

Ա. ԱԲՐԱՀԱՄԵՍՆ - ԵՌԻ. ԲԱՐ
Դ. ՄԱՐԿԱՐՈՎ - Կ. ՄԱՐԿՈ

**ԻՆՖՈՐՄԱՏԻԿԱՅԻ
ՀԻՄՈՒՆՔՆԵՐ**



ՀՐԱՏԱՐԱԿԻՉՆԵՐՈՒ ԿՈՂՄԻ

«Տեղեկադիտութեան» լեզուն՝ միայն քոաներորդ դարու զարգացումին խորհրդանշանը չէ. անիկա կրնայ նկատուիլ նաեւ իբր խաղաղութեան լեզու։ Աշխարհի բոլոր երիտասարդները, որոնք մինչեւ երեկ կը զանազանուէին տարրեր լեզուներով, կրօնքներով, սովորութիւններով ու բարքերով, համակարգիչի լեզուին ճամբով եւ զիրենք իրարու զօղող սիրով՝ կրնան հաղորդակցութեան կերպ մը զանել, որ կը զերազանցէ աշխարհազրական ամէն սահման։

«Տեղեկադիտութեան» լեզուին միջոցով՝ համախմբուած իրիտասարդներուն կը վերաբերի զերազանցել հին արդելքներ եւ հակադրութիւններ, յաղթահարելով հազարամեայ առելութիւնը՝ որ կը կաշկանդէ զանազան ժողովուրդներ։

Մեր մաղթանքն է որ այս ուղեգիծ-զիրքը կարենայ նպաստել ո՛չ միայն Հայաստանի յառաջդիմութեան ու վերելքին այլ նաեւ ժողովուրդներու խաղաղութեան։

ՊԵՏՐՈՍ ՔՈԽՉՈՒՔԵԱՆ
Խտախոյ Հայ Գաղուք
Միլանոյ

Նկատի առնելով Հայ ժողովուրդի միասնութեան ներկայ օրերու հրամայականը՝ յարմար նկատեցինք այս հրատարակութիւնը հանրութեան յանձնել Մերոպիան դասական ուղղագրութեամբ, իբր խորհրդանիչ՝ Հայ ժողովուրդի միասնութեան իղձերուն իբր գործնական արտայայտութիւն։

ՄԽԻԹԱՐԵԱՆ ՀՐԱՏԱՐԱԿՉՈՒՏՈՒՆ
ՎԵԼԵԽԵԼԻ - Ս. Ղազար

ԲՈՎԱՆԴԱԿՈՒԹԻՒՆ

1. ՆԵՐԱԾՈՒԹԻՒՆ

7

2. ՔՈՄՓԻՒԹԵՐՆԵՐԸ Ա.ՐԴԻ ՀԱՍՏԱՏԱԿԱՐԳՈՒՄ

2. 1. Հասարակարգի ինֆորմատացումը եւ նոր ինֆորմացիոն տեխնոլոգիան	9
2. 2. Հաշուողական տեխնիկայի պատմութիւնից	11
2. 3. Անհատական կիրառական ծրագրերի յարմարաւ և տուրիներ	18
2. 4. Տեխստերի վերամշակման համակարգեր	22
2. 5. Աղիւսակների վերամշակման համակարգեր	26
2. 6. Ինֆորմացիոն համակարգեր	30
2. 7. Գործավարման ամբողջացուած համակարգեր	36
2. 8. Պատկերների վերամշակման համակարգեր	40

3. ՔՈՄՓԻՒԹԵՐՆԵՐԻ ԱԾԽԱՏԱՆՔԻ ՄԿՋԲՈՒՆՈՒՅՆԵՐԸ

3. 1. Քոմփիւթերի ընդհանուր կառուցուածքը	51
3. 2. Կենտրոնական պրոցեսոր	55
3. 3. Յիշող համակարգերը	59
3. 4. Ինֆորմացիայի ներածման եւ արտածման սարթեր	66

4. ՍԿԱՆԱՌԱԿԱՑԻՒՆ ՕՊԵՐԱՅԻՌՈՒՆ ՀԱՄԱԿԱՐԳ

4. 1. Օպերացիոն համակարգ կառուցուածքը	75
4. 2. Սշխատամբ սկաւառակների հետ	78
4. 3. Մրագրերի թողարկում	84

5. ԱԼԳՈՐԻԹՄՆԵՐ

5. 1. Ալգորիթմի հասկացութիւնը - Ալգորիթմի ներկայացման եղանակները	87
5. 2. Ալգորիթմի յատկութիւնները	93
5. 3. Ալգորիթմների տիպերը	95
5. 4. Ցիկլային ալգորիթմներ	102
5. 5. Զանգուածների մշակման ալգորիթմներ	116

5

6. ՔՈՄՓԻՒԹԵՐՆԵՐԻ ՆՐԱԳՐԱԿՈՐՈՒՄԸ

6. 1. Բարձր մակարդակի ծրագրաւորման լեզուներ	131
6. 2. Ծրագրաւորման թեյսիկ լեզուն	132
6. 3. Թեյսիկ լեզուի կառուցուածքը	134
6. 4. Թեյսիկի այրութենք	135
6. 5. Հաստատութենք եւ վովտխականներ	136
6. 6. Զանգուածներ	138
6. 7. Ստանդարտ ֆունկցիաներ	140
6. 8. Արտայայտութիւններ	144
6. 9. Յարաբերութիւններ	146
6. 10. Տրամարանական գործողութիւններ	146
6. 11. Թեյսիկի հիմնական հրահանգները	152
6. 12. Արժէնների վերագրման հրահանգներ	152
6. 13. Խնդրմացիայի արտածման հրահանգը	157
6. 14. Մելքանաբութիւնները Թեյսիկ ծրագրում	162
6. 15. STOP եւ END հրահանգները	163
6. 16. Ղեկավարման հրահանգներ	163
6. 17. Ցիլինդրի կազմակերպման հրահանգներ	171
6. 18. Զանգուածների մշակման ծրագրեր	178
6. 19. Օգտուողի կողմից սահմանուած ֆունկցիաներ	185
6. 20. Ներածրագրեր	185
6. 21. Դրաֆիկական հրահանգներ	191
6. 22. Աշխատանք ֆայլերի հետ	199

1. ՆԵՐԱԾՈՒԹԻՒՆ

Ինֆորմատիկայի հնտեհնուի զարդացումը եւ տարածումը րերեցին քոմպիւթեյին թեմատիկայի նկատմամբ հետաքրքրութեան կտրուկ մեծացման։ Ընթերցողին առաջին հերթին կը հետաքրքրէր «Ի՞նչ են քոմփիւթերները եւ ի՞նչ կը տայ դրանց օգտագործումը մարդկային գործունէութեան այս կամ այն բնագաւառում» հարցը։

Քոմփիւթերի մասին պատկերացման խորացմանը զուզընթաց առաջանում է «Ի՞նչպէ՞ս կարելի է քոմփիւթերի հիման վրայ կառուցել մի համակարգ, որը կ'օգնի մարդուն կատարել այս կամ այն աշխատանքը կամ կը փոխարինի նրան ընդհանուրապէս» հարցը։ Հենց այդպիսի ուզուածութիւն ունի Ձեր ուշագրութեանը ներկայացուող զերքը։

Գիրքը իրենից ներկայացնում է քոմփիւթերային տեխնիկայի ներածութիւն։ Նրանում պարունակւում են դրանց ուսումնասիրման համար բոլոր տուեալները։ Դիսարկում են քոմփիւթերների կոնկրետ տիպերը եւ դրանց կիրառման բնագաւառուները։ Հաւատարապէս ուշադրութիւն է զարձում ինչպէս քոմփիւթերների բաղկացուցիչ մասերի նկարադրման, այնպէս էլ նրանց օգտագործման եւ ծրագրաւորման հարցերի շարադրման վրայ։

Գիրքը պայմանականուն կարելի է բաժանել երկու մասի։ Առաջնում (Գլ. 2, 3, 4) զարդարում են քոմփիւթերների եւ նրանց հիման վրայ ստեղծուած ինֆորմացիոն համակարգերի մասին ընդհանուր պատկերացումները։ Երկրորդը (Գլ. 5, 6) նույրուած է ինդիքների ալղորիթմացման հարցերին եւ ծրագրաւորման լեզուներին։

Չնայած այս զերքը նախատեսուած է միջնակարգ ուսումնական հաստատութիւնների սովորողների եւ զամաւանդողների համար (զրգում մանրամասնորին զիտարկ կրւում են այն հարցերը, որոնք ըստ մեղ լաւ չեն լուսաբանուած ինֆորմատիկայի զարդարական դասագրքերում), այն օգտակար կարող է լինել քոմփիւթերային տեխնիկայի բնագաւառում չմասնագիտացող ուսանողների, զիտարտադրական տարբեր բնագաւառների աշխատողների համար։

Քոմփիւթերների հիման վրայ ստեղծուած ինֆորմացիոն համակարգերը կարող են կիրառել քոմփիւթերը իրենց խնդիրների լուծման համար օգտագործել ցանկացող ինժեներները, օրինակ՝ բեռնափոխադրման զեկավարման, արտադրութեան աւտոմատացման, կապի համակարգերի զեկավարման ինդիքներում։ Ֆիզիկուները, քիմիկուները եւ կենսաբանները այն կարող են կիրառել փորձերի յատկանիշների խիստ հսկողութեան կազմակերպման, տուեալների աւտոմատ հաւաքման եւ վերլուծման զորձնախցում, նկարիչներն ու ճարտարապետները քոմփիւթերի օգնութեամբ սահմանում են որոշակի զինապահի միջավայր օգտագործելու հնարաւորութիւն, հոգեբանները քոմփիւթերները կարող են օգտագործել մարդկային մտածողութեան օրինաչափութիւնների ուսումնասիրման բնագաւառում։

Մինչեւ գրքի լոյս ընծայելը հեղինակային խումբը 5-6 տարի զբաղուել է միջնակարգ ուսումնական հաստատութիւններում «Ինֆորմատիկայի և հաշուողական տեխնիկայի հիմունքներ» առարկան դասաւանդող ուսուցիչների որակաւորման բարձրացման եւ վերապատրաստման հարցերով։

Այդ ընթացքում մենք ունեցել ենք բազմաթիւ հանդիպումներ այս գրքի հիմնական պահանջն ունեցողների հետ անմիջապէս աշխատող մանկավարժների հետ։ Գրքի ստեղծման գործում մեծ նշանակութիւն ունեցան նրանց հետ աշխատելու փորձը եւ նրանց ցանկութիւններոն ու դիտողութիւնները։ Մենք մեծ օգնութիւն ենք ստացել ԵՊՀ-ի դոցենտ Վ.Ս. Եղիազարեանից, որը մանրամասն ուսումնասիրել է զիրքը եւ նրա կառուցուածքին վերաբերող հարցերում օգտակար դիտողութիւններ կատարել։ Այդ դիտողութիւնները անշուշտ օժանդակել են նիւթի շարադրման բարելաւմանը եւ մենք բարձր ենք զնահատում նրա ջանքերը։ Մեր պարտքն ենք համարում խորին շնորհակալութիւն յայտնել «Կապոյտ Խաչ» կազմակերպութեանը եւ պարոն Անրի Փափազեանին, որոնք իրենց վրայ են վերցնել գրքի հրատարակման հոգսերը։

2. ՔՈՄՓԻՒԹԵՐՆԵՐԸ ԱՐԴԻ ՀԱՍԱՐԱԿԱՐԳՈՒՄ

2. 1. Հասարակարգի ինֆորմատացումը եւ նոր ինֆորմացիոն տեխնոլոգիան

Հաշուողական տեխնիկայի սրընթաց զարգացումը պայմանաւորուած է որոշակի դիտատեխնիկական մակարդակի հասած մարդկային հաստիարգի որբենական պահանջներով։ Զարգացած երկրներում քոմմիթերների ներթափանցումը կեանք գրեթէ բոլոր ոլորտները այնպիսին է, որ չափազանց մեծ աղդեցութիւն ունի հասարակուում տեղի ունեցող ինֆորմացիոն, տեխնոլոգիական, տնտեսական եւ այլ գործընթացների վրայ։ Քոմմիթերների օգնութեամբ դիտական, տեխնիկական եւ գործնական խնդիրների լուծման հետ կապուած բոլոր երեսոյթների բազմութիւնը ընդունած է կոչել հասարակարգի ինֆորմատացում։ Անյայտ է, որ ինտենսիվ ինֆորմատացումը պրոցեսի երեսոյթ է, քանի որ վերջինիս զարգացման աստիճանից է կախուած այս կամ այս պետութեամ տնտեսական պոտենցիալը։

Ինֆորմացիան արդէն պաշար է զարգածել եւ աւանդական պաշարների (ածով, չներզիա, աշխատութ եւ այլն) պէս կարող է արգիւնահանուել, վերամշակուել, օգտագործուել եւ տարածուել։ Ինֆորմացիայի հիմնական տարրերութիւնը աւանդական պաշարներից կայանում է նրանում, որ նրա ծառալր չի նուազում օգտագործման հետեւանքով։ Իրականում տուեալների վերաբերման ընթացքում ստացուած արդիւնքները կարող են դիտարկուել իրրե նոր տուեալներ։ Աւելի լուրջ տարրերութիւն է ի յայտ գալիս գրաելիքների վերամշակման ժամանակ, երբ գործնականում կատարում է նոր գիտելիքների դուրս բերում, ինչպէս մարդիմատիկայում գոյութիւն ունեցող թէորեմների հիման վրայ զուրու է բերում նոր թէորեմ։ Արդէն յայտնի են քոմմիթերների օգնութեամբ կատարուած յայտնագործութիւններ։

80ական թուականների սկզբին ներկայացուեց մի նոխագիծ, որն ստացուած նոր ինֆորմացիոն տեխնոլոգիա անուանումը։ Նոր տեխնոլոգիայի անհրաժեշտութիւնը պայմանաւորուած էր նրանով, որ աւանդական եղանակով մեծածաւալ աւտոմատացուած համակարգերի նախագծման տեսլութիւնը սկսում էր գերազանցել նրանց բարեկարգութիւն մասնակի բարոյական մաշման ժամկետները։ Իրրե ստեղծուած վեճակից զուրու զարու միակ միջոց, առաջարկուում էր համակարգերի ստեղծմանը ի սկզբէ նախագծի՝ համակարգի շահագործողները օգտագործելու յատուկ մասնագիտացուած եւ պարզեցուած լեզուներ, ստեղծուում են ապագայ համակարգի նախամիաց։ Այդ նախամիացու միականացում են ապագայ համակարգի լուրոր ֆունկցիաները, բայց առփորար չի բաւարարում արդիւնականութեամ որոշ պայմանները (արագագործութիւն, օգտագործուուղ յիշողութեան ծաւալ եւ այլն)։ Մասնագէտ ծրագրաւորողները այդ գէպքում զրագուում են հիմնականում նախամիացի արդիւնականութեամ ապագութեամով եւ այդ եղանակով ճեւաւորում ապագայ համակարգի

ծրագրային ապահովումը։ Ընդ որում բաւական յաճախ ծրագրելի արդիւնաւէտութեան բարձրացում չի պահանջւում։ Օրինակ, ասենք ոչ ծրագրաւորողի կողմից գրուած ծրագիրը մի քանի հազար նիշով աւել յիշողութիւն է դրազեցնում եւ համապատասխանաբար խնդրի լուծման համար մի քանի հազար գործողութիւն աւել է կատարում, քան մասնագիտ ծրագրաւորողի կողմից գրուած ծրագիրը։ Տուեալ դէպ-քում արդեօք միտք ունի՞ աշխատել ծրագրի արդիւնաւէտութեան բարձրացման ռւդղութեամբ, եթէ քոմիկիթերի յիշողութեան ծաւալը կազմում է ասենք 1 միլիոն նիշ, իսկ արագազործութիւնը 1 միլիոն գործողութիւն վայրկեանում։ Գրեթէ ակնյայտ է, որ՝ ոչ։ Սա է նոր ինֆորմացիոն տեխնոլոգիայի բուն էութիւնը։

Առաջին քայլը դէպի նոր ինֆորմացիոն տեխնոլոգիան արուեց աւելի չուստ, երբ քոմիկիթերները, որոնք նախկինում օգտագործում էին միայն դիտական խնդրների լուծման ընթացքում հիմնականում հաշուային գործողութիւններ կատարելու համար, սկսեցին օգտագործուել դորժնական խնդրների լուծման համար, որպիսիք են՝ հիմնարկութիւնների գործունելութեան գեկավարումը, պլանաւորումը եւ այլն։ Դրանով քոմիկիթերների օգտագործման ոլորտները զգալիօրէն ընդլայնուեցին։ Նշենք, որ այս փուլում քոմիկիթերները զեւուեւ հեռու են գտնուում լայն խաւերից, իսկ որպէս միջնորդ հանգու էր զալիս մասնագիտ ծրագրաւորողը։

Յաջորդ քայլը դէպի նոր ինֆորմացիոն տեխնոլոգիան արուեց 70ական թուականների վերջում, երբ երեւան եկան առաջին անհատական քոմիկիթերները, որոնք նախատեսուած էին հիմնարկունում ծրագրաւորման ասպարէզում մասնագիտական դիտելիքներ չունեցող մարդկանց համար։ Այժմ, տասից քիչ աւել տարի կարելի է յստակ ասել, որ դա ինֆորմացիոն տեխնոլոգիայի «դիտութիւն - արտադրութիւն - կենցաղ» շղթայի վերջին կապակցու օղակին էր։ Ներկայումն զարգացած երկրների գրեթէ ամբողջ բնակչութիւնը ներկրաւուած է ինֆորմացիայի վերամշակման ոլորտում։ Զարդարակալի չէ, որ այլ երկրներում սրոշումներ են ընդունուած զպրոցներում նոր առարկայի՝ ինֆորմատիկայի վերաբերալ։

Դէպի նոր ինֆորմացիոն տեխնոլոգիան կատարուած է երկու կարեւոր քայլ, առաջին անհատական հաշուարկների գարգացումը գետ չի աւարտած։ Ապագայում դեռ կը լինեն նոր նուաճումներ, սկզբում, ի հարկ է, քանակական։ Կամի միջնորդներին միացուած անհատական քոմիկիթերը պէսք է դառնայ սովորական կենցաղային մի սարք, որպիսին հետախոսն է։ Այնուհետև քանակական փոփոխութիւնները անխաւասփելիքրէն կը բերեն որակական փոփոխութեան։ Եթէ նախկինում մենք գործունիքին ինֆորմացիայի վերամշակման հետ, ապա յիշտագայում գրազուած կը լինենք դիտելիքների վերամշակուով։ Այսօր մենք զեր այլ ճանապարհի նախաչմին ենք, բայց արդէն ի յայտ են զալիս ապազայ տեխնոլոգիայի ուրուազծերը։ Անցումը հիմնարկում գիտելիքների վերամշակմանը երրորդ, ամենակարեւոր քայլն է դէպի նոր ինֆորմացիոն տեխնոլոգիան։

Աստիճանաբար անցումը դէպի նոր ինֆորմացիոն տեխնոլոգիան ունի նաեւ իր հասարակական հետեւանքները, որոնց մի մասը ակնյալա է արդէն այսօր։ Գոյութիւն ունի մարդկանց մի մեծ ենթախումբ, որոնց արտադրական դործունէութիւնը զրեթէ լիովին կատարած է ինֆորմացիայի վերամշակման հետ։ Դրանց թուին են պատառ նախապահութեան անհանդացումը, որուն մասնակիցները աշխատանքը, անկամ կողեկիւթ աշխատանքը, կարող է կաղամակերպուել միացեալ ցանցի մէջ ընդգրկուած անհատական

քոմիկիթերների միջոցով։ Որպէս հետեւանք կ'անհետանայ մասնագէտների աւան դական աշխատատեղերում գտնուելու անհրաժեշտութիւն։ Ընդհանուր տուեալների հիմնապահէնսաի առկայութիւնը հնարաւորութիւն է տալիս ապահովել արդիւնաւէ։ Կողեկտիւ աշխատանքը, որի ընթացքում հնարաւոր է դառնուում առանձին անհատի աշխատանքի ընթացքի վիճակի հսկումը։ Բացի զրանից՝ անհատական քոմիկիթեր ների միջոցով կաղմակերպուած աշխատանքը տնային պայմաններում հիմնականուու շահաւէտ է հիմնարկութիւնների համար քանի որ նուազեցնում է պահանջուող արտադրական տարածութիւնները։ Սակայն գործը ցոյց է տալիս, որ աշխատանքի նման կազմակերպումը հիմնարկանում չի բաւարարուած աշխատանքներին։ Իրենց բացասական կան վերաբերունքունք այդպիսի աշխատանքնին նրանք բացատրում են առաջացու մարդկային հաղորդակցութեան պակասութեամբ, որից առանց այդ էլ տառապու։ Է արդի հասարակարդը։ Աշխատանքի տնային կաղմակերպման հանդէպ նման վերաբերունքի հիմնարկանութիւնը գժուար է զնահատելով այս սեփական օրինակի վրայ։

Յաջորդ կարեւոր հասարակական փոփոխութիւնը կազուած է արտադրական գործութիւններ մէջ ստեղծադադարական բովանդակութեան կտրուկ ածի հետ։ Իրականում ինչ է վերապահաւում արդի բիւրուկրաստին, եթէ ըոլոր փաստաթվերը եւ տեղի կոնքները սկսեն ձեւակերպուել ըոլոր օրէնքներին և հրահանդամներին համաձայն և անդամ հաստատուեն աւատամատ կերպով։ Այս պարագայում զգալիօրէն կը բաժնանայ ակտիւ, ստեղծադադար մարդկանց հասարակական կշիռը։ Անհատական քոմիկիթերների միջոցով ստեղծուուղ զիւրայարմարութիւնը անխուսափելորէն կը բիրը կեանքի տեւողութեան եւ, որպէս հետեւանք, բնակչութեան միջին տարիքի աճի։

2. Հաշուողական տեխնիկայի պատմութիւնից

Հաշուողական տեխնիկայի պատմութիւնը սկսում է 17րդ դարի կէսերից, երաժեղուեց առաջին մեխանիկական թուարանական մեքենան, որ կատարում է միայն գումարման եւ հանման գործողութիւններ։ Այնուհետև ստեղծուեցին մի շարու մեխանիկական մեքենաներ, որոնք արդէն կատարում էին չորս պարզադու։ Թուարանական գործողութիւններ, ընդ որում բաղմապատկման գործողութիւնը կատարում էր բաղմակի գումարման եղանակով, իսկ բաժանման գործողութիւնը բաղմակի հանման եղանակով։ 19րդ դարի վերջին ԱՄՆ-ուում մարդահամար անց կացնելու նպատակով ստեղծուեց ինֆորմացիայի վերամշակումը աւտոմատացնուառաջին մեքենան, որում ինֆորմացիան ներկայացնում էր յասուկ քարտերի չորս պակումների ձեւու։ Ընթացիկ գործիք 40-50ական թուականներին, երբ զգալի արդիւնք ունեցած է գաղափարական աշխատանքունք։ Ընթացիկ գործիք 40-50ական թուականներին կատարուած է ինֆորմացիայի վերամշակման հետ։ Դրանց թուին են պատառ նախապահութեան անհանդացումը, որուն մասնակիցները աշխատանքը բարեկարգութեան անհանդացումը։ Վերջին չորս տասնամեանների, ընթացքում ստեղծուեց բովանդակութեան գործիքը չորս սերունդ եւ սկսուեցին հինգերուց նախապահութեան աշխատանքները։

Հաշուողական տեխնիկան չափագանց արագ է զարգանում։ Այդ սրբնիաց զարգումը բարգարաւուում է նրանով, որ մարդու գործունէութեան մի շարք ուրարտուած միաժամանակ ինֆորմացիան գործունէութիւնների կորու։

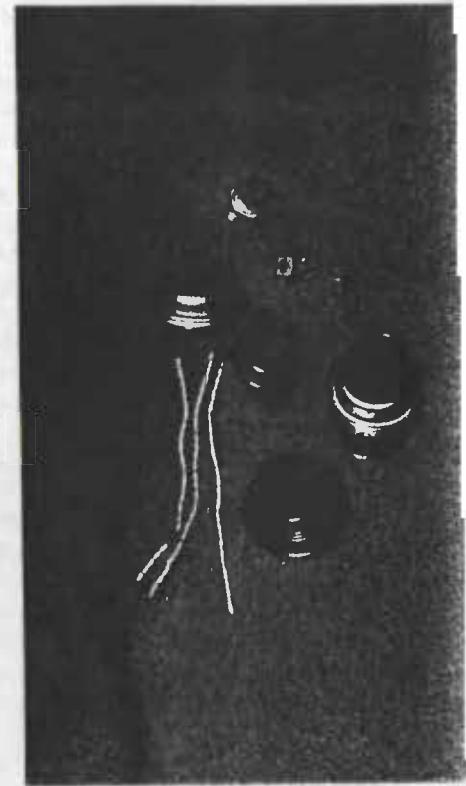
արագացման պահանջ առաջացաւ: Անցած աւելի քան 40 տարիների ընթացքում քոմ-փիւթերների տարրերի չափսերը նուազել են 10 միլիոն անդամ, սպառուող հօրութիւնը՝ 1 միլիոն, զինը՝ 200 հազար, արագագործութիւնը աճել է 100 հազար, իսկ յուսալիութիւնը՝ 40 հազար անդամ: Համեմատութեան համար ասենք, որ եթէ նոյն թափով զարգանար աւտոմոբիլաշխնութիւնը, ապա ժամանակակից աւտոմեքենան կ'արժենար ոչ աւել, քան մի տուփ լուցկին, կը զարգացնէր մինչեւ 1 միլիոն Կմ/ժամ արագութիւն, 100 Կմ զարգի դէպեռում ծախսելով 1 միլիգրամ զառելանիւթ, կունենար գնդասեղի զլսիկի չափ շարժիչ եւ գործնականօրէն շարքից դուրս չէր զայ մարդու կեանքի միջին տեսողութեան ընթացքում:

Առաջին սերնդի քոմփիւթերների արդիւնաբերական արտադրութիւնն սկսուեց 40ական թուականների վերջին: Այս մեքենաներում, որպէս տարրային հիմք, հիմնականում օգտագործուում էին էլեկտրոնային լամպեր (նկ. 1): Առաջին սերնդին են պատկանում հայրենական «Մտրելա», «Մինսկ - 1», «Ուրալ - 1», «Մ - 20», ամերիկան «MARK - 1», «ENIAC» (նկ. 2) եւ այլ քոմփիւթերները, որոնք հիմնականում օգտագործուում էին գիտատիենիկական խնդիրների լուծման համար: Լամպային քոմփիւթերները սպառուում էին մեծ հօրութիւն, ունիէն մեծ չափսեր, յիշողութեան փոքր ծաւալ եւ յուսալիութիւն՝ յատկապէս էլեկտրոնային լամպերի յաճախակի շարքից դուրս գալու պատճառով: Օրինակ՝ «ENIAC» քոմփիւթերը պարունակում էր 18 հազար էլեկտրոնային լամպ, սպառուում էր 100 կիլովատից աւել էլեկտրաէներգիա, կշռում էր 30 տոննա եւ զրագեցնուում էր աւելի քան 300 քառակուսի մետր տարածութիւն: Այս սերնդի քոմփիւթերների համար բնորոշ է այսպէս կոչուած «քաց» օգտագործման երանալլ, որի դէպեռում ծրագրաւորողը մեքենայի դէկավարման փահանակից ներածում, յարդարում եւ կատարում էր ծրագիրը, ընդ որում հաշուարկներ կատարելու վրայ ծախսուում էր ընդհանուր ժամանակի ոչ աւել, քան 10 տոկոսը:

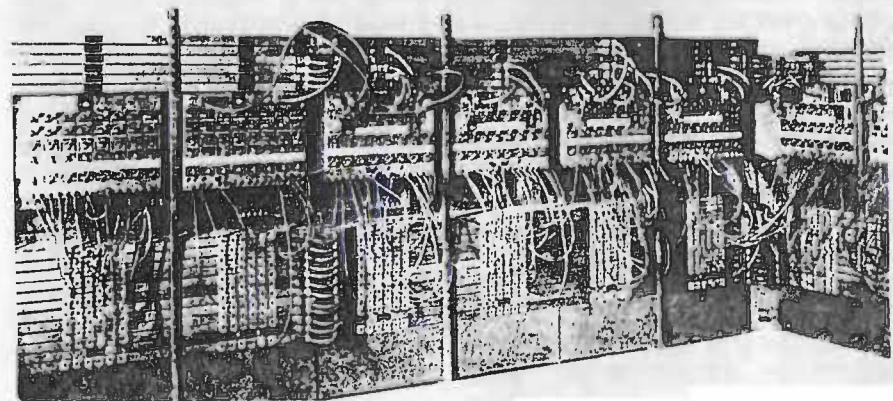
Երկրորդ սերնդի քոմփիւթերների արդիւնաբերական արտադրութիւնը սկսուեց 50ական թուականների վերջին: Այս մեքենաներում էլեկտրոնային լամպերը փոխարինուեցին կիսահաղորդչային տրանզիստորներով եւ դիոդներով (նկ. 3): Այդպիսի փոխարինումը նպաստեց յուսալիութեան բարձրացմանը, նրանց չափսերի եւ սպառուող հօրութեան նուազմանը: Գիտական հաշուարկների համար նախատեսուած մեծ քոմփիւթերներին («Մինսկ - 22», «Ուրալ - 14», «Հրազդան - 3» եւ այլն) զուգընթաց ստեղծուեցին փոքրերը («Նայիրի», «Դնեպր», «Միր» եւ այլն), պլանատնտեսական եւ արտադրութեան կառավարման խնդիրների լուծման համար: Այս սերնդի քոմփիւթերների համար բնորոշ է այսպէս կոչուած «փակ» օգտագործման եղանակը, որի դէպեռում ծրագրաւորողը անմիջական չփում չունէր քոմփիւթերի հետ: Նա կազմում էր ծրագիրը յանձնուում յատուկ սպասարկման խմբին, որը մի քանի ծրագրերից կազմում էր «փաթեթ» եւ թողարկում քոմփիւթերի վրայ միասին: Դա հնարաւորութիւն էր տալիս շատ աւելի արդիւնաւէս օգտագործել քոմփիւթերի հնարաւորութիւնները, բայց զգայիօրէն մեծացնուում էր առանձին խնդրի լուծման ժամանակը, քանի որ ծրագրի յարդարումը, այսինքն՝ սպասարկման լուծումը պէտք է կատարուեր մի քանի թողարկման ընթացքում: Տարբեր ծրագրերի միաժամանակ թողարկումը սպահովելու եւ քոմփիւթերի առանձին սարքերը խնդիրների միջեւ վերարաշխելու համար ստեղծուեցին յատուկ դէկավարող ծրագրեր, որոնք քոմփիւթերի անբաժան մասն էին կազմում:



Նկ. 1



Նկ. 2



Նկ. 3

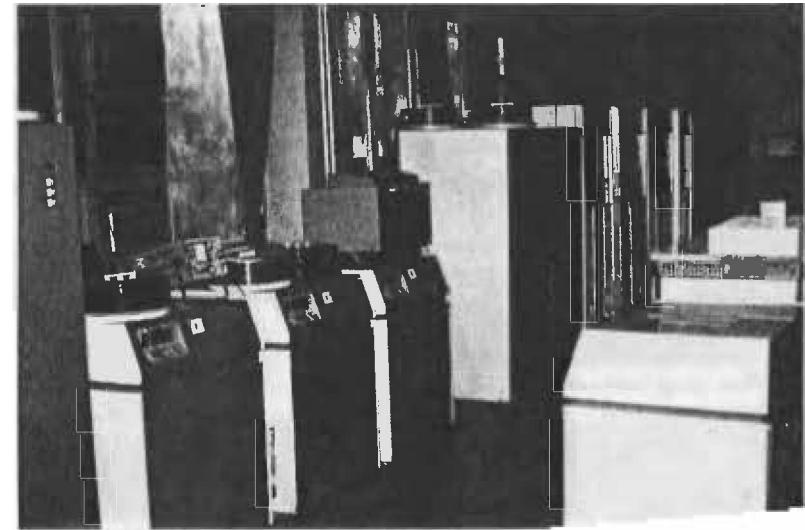
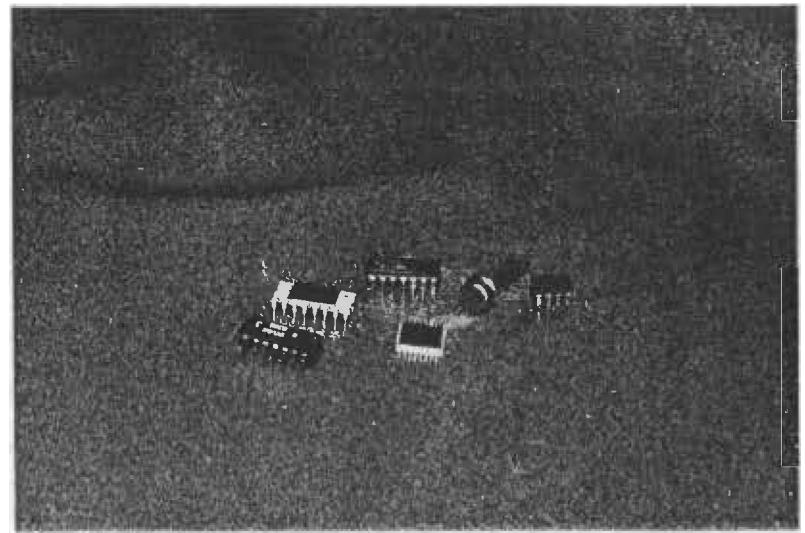
Երրորդ սերնդի առաջին քոմֆիւթերները ստեղծուեցին 70ական թուականների կողմանը: Այստեղ, որպէս տարրային հիմք, հիմնականում օգտագործում էին կիսանդրութային ինդեքրալ միկրոսիմեաները (նկ. 4), որոնց իրանում մէկ բիւրեղեայ թիթեղի վրայ սաեղծուում էին սասնեակ փոքրածաւալ տրանզիստորներ եւ դիոդներ: Երրորդ սերնդին է պատկանում «Նայիրի - 4» քոմֆիւթերը: ԱՄՆ-ու այս սերնդի յրաժողութեան ստեղծուեց լայն ճանաչում ստացած IBM-360 քոմֆիւթերների ընտանիքը, որը բաղկացած էր մի շարք տարրեր հզօրութիւն ունեցող եւ ծրագրային ահասկէտից համատեղելի քոմֆիւթերներից:

70ական թուականների սկզբին անտեսական փոխօգնութեան խորհրդի մասնակից երկրները, իրագործելով սոցիալիստական ինսեգրացման ծրագիրը, սկսեցին երրորդ սերնդի քոմֆիւթերների երկու ընտանիքների նախադումն ու արտազրութիւնը: Դրանք էին քոմֆիւթերների միացեալ համակարգը (EC - նկ. 5) եւ փոքր քոմֆիւթերների համակարգը (CM - նկ. 6): Դրա հետ մէկտեղ ստեղծուեցին այնպիսի քոմֆիւթերներ, որոնք իրենց ընութազրերով միջանկեալ զիրք էին զրաւում մէծ եւ փոքր քոմֆիւթերների միջև: Երլորդ սերնդի երեւան զալը նպաստեց լուծուող գիտատեխնիկական, պահանանտեսական եւ ղեկավարման ինդիրների շրջանակների բնակայնամանը: Քոմֆիւթերները սկսեցին կիրառուել ուսուցման, գունագան գործընթացների մոդելաւորման եւ այլ խնդիրների լուծման համար:

Քոմֆիւթերների հետ աշխատանքը դարձաւ չսա աւելի արդի նա էտ, երբ երրորդ սերնդի մերենաներում ամէն մի օգտագործող ստացա իր առանձին աշխատատեղը, որը կոչում էր տերմինալ: Քոմֆիւթերն միացւում էին մի շարք տերմինոլոգներ, եւ յասուկ ղեկավարող ծրագրերի համակարգը (օպերատորն համակարգը) կողմանիրավում էր մերենայի աշխատանքը այնպէս, որ առանձին օգտագործողին թւում էր, թէ քոմֆիւթերի ամրող միջոցները զանուում են իր արամազրութեան տակ: Քոմֆիւթերի հետ աշխատելու ամենաբրդիւնական է բնական եղանակը դառնում է սկսիւ երկխօսութիւնը:

Քոմֆիւթերների չորրորդ սերնդի առաջին ներկայացրուցիչները, որմանց տարրային հիմքը հիմնականում մէկ բիւրեղի վրայ հաղարաւոր տրանզիստորներ պարունակով մէծ եւ զերմէծ ինսեգրալ սինեմաներն են (նկ. 7), ոտեղծուեցին 70ական թուականների վերջին: Չորրորդ սերնդի մի շարք քոմֆիւթերներ իրենց կառուցուածքով կրկնում էին երրորդ սերնդի լուսագոյն քոմֆիւթերները, տակայն, մէծ եւ զերմէծ ինսեղարակ սինեմաների օգտագործման շնորհիւ ունիւն աւելի բարձր տեխնիկական ընութազրեր: Այս սերնդի մէքենաների շարքում յատակ տեղ են զրադեշնուում միկրո քոմֆիւթերները, սպոնտ թէ եւ հիմնական լուսագրելով զիջում են փոքր քոմֆիւթերներին, սակայն անհամեմատ էժան են եւ ունեն աւելի փոքր չափսեր, այնպէս որ կարող են տեղաւորուել տպորակիսն դրասեղունի վրայ: Միկրո քոմֆիւթերների փոքր արժէքը, չափսերը եւ չահագործման պարզութիւնը դրանց օգտագործումը նախկինում չնպաստեսուած բազմաթիւ բնագաւառներում: Միկրո քոմֆիւթերները լայնօրէն օգտագործում են հասացների եւ տեխնոլոգիական սարքաւորումների, նոյնիսկ էնցուարային սարքերի կառավարման համար: Ընդուում, դրանք յաճախ ղեկավարուող սարքերի բազադրիչ մասն են կազմում:

Ներկայումս արտադրուող քոմֆիւթերները հիմնականում ըստ արտագործութեան պայմանականորէն բաժանում են մի քանի դասի: Այսպէս՝ զերքոմֆիւթերները կապուղացներ էնդիուդ մէկ միջիացրդ զործողութիւն:



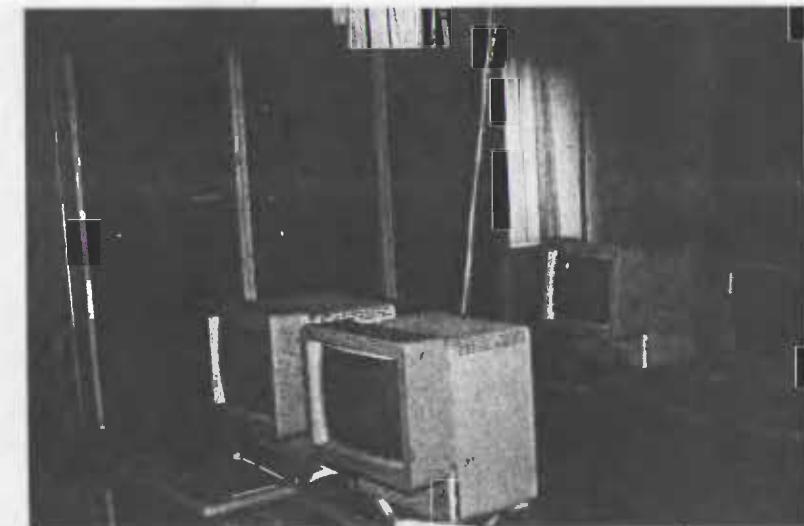
մեծ քոմփիւթերները՝ 100 միլիոն, միջին քոմփիւթերները՝ 10 միլիոն, իսկ փոքր եւ միկրո քոմփիւթերները՝ աւելի քան 1 միլիոն գործողութիւն վայրկեանում: Դրա հետ կապուած տարրեր դասի քոմփիւթերների պահանջւում է մի քանի գերքոմփիւթեր, իսկ միկրո քոմփիւթերների պահանջարկը կարող է հասնել միլիոնների:

Վերջին տասնամեւակում միկրո քոմփիւթերների ընտանիքում առանձնացաւ մեքենաների մի մեծ խումբ, որոնց շահագործումն այնքան պարզ է, որ հաշուողական տեխնիկայի բնագաւառի յատուկ զիտելիքներ չի պահանջում: Այս մեքենաները օգտագործելու լիովին բաւական է ունենալ զիտելիքներ «Խնֆորմատիկայի եւ հաշուողական տեխնիկայի Հիմունքներ» առարկայից եւ եթէ ծրագաւորում է պէտք պահանջուքի ծանօթ լինել համապատասխան ծրագրաւորման լիցուին, սովորաբար՝ բէյսիկ լեզուին: Այդպիսի մեքենաները կուլում են անհատական քոմփիւթերներ: Անհատական բառը տուեալ դէպքում նշանակում է ոչ թէ այն, որ մեքենան պատկանում է առանձին անհատի, այլ այն, որ քոմփիւթերը հնարաւորութիւն է տալիս լիովին անկախ իրականացնել անհատ օգտագործողի լուծումը մեքենայի բոլոր հնարաւորութիւնների օգտագործման պայմաններում: Անհատական քոմփիւթերների առանձնութեան հաշուողական տեխնիկայում կարելի է համեմատել էլեկտրոնիկայում տրանզիստորների, թեթեւ արդիւնաբերութեան մէջ սինթետիկ նիւթերի, առօրեայում զնդիկաւոր գրիչների առաջացման հետ:

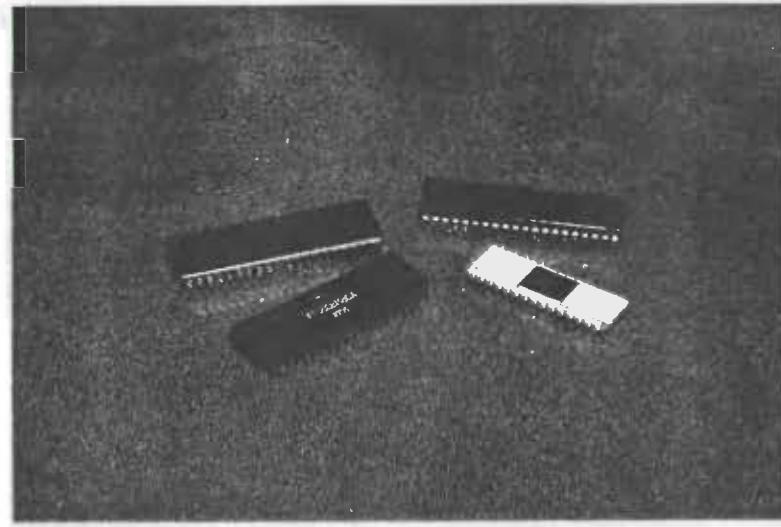
Անհատական քոմփիւթերները հնարաւորութիւն առուեցին կրճատել մեքենայի զինը, բարձրացնել շահագործման յարմարաւէտութիւնը եւ ստեղծել հաշուողական միջոցների մի շարք նոր, նախկինում անմատչելի օգտագործման եղանակներ: Նրանց կիրառուման պոտենցեալ ոլորտները ներառում են ինֆորմացիայի ընկալման, մշակման եւ փոխանցման հետ կապուած մարդու գործունէութեան տեսակները: Կոնկրետ ձեռնորդկութեան համար անհատական քոմփիւթերների որոշակի մոդելի ընտրութիւնը կատարում է այնպիսի գործօնների վերլուծութեան հիման վրայ ինչպիսիք են զինը, նոր մեքենայի համատեղելիութիւնը ձեռնարկութիւնում արդէն դոյութիւն ունեցող մեքենաների հետ, ձեռք բերուող համակարգի սպասարկման եւ զարգացման հնարաւորութիւնները եւ այլն:

Անհատական քոմփիւթերները նոյնական ընդունուած է դասակարգել ըստ սերունդների: Ի տարրերութիւն քոմփիւթերների դասական սերունդների՝ այստեղ որոշվէ օգտագործուող հիմնական հաշուողական գործողութիւնները կատարող սարքի, այսպէս կոչուած միկրոպրոցեսորի կարգայնութիւնն է, որն իր հերթին որոշում է քոմփիթերի արագագործութիւնն եւ գործողութիւնների կատարման ճշգրտութիւնը:

Անհատական քոմփիւթերների առաջին սերունդը հիմնուած է 8 կարգանի միկրոպրոցեսուների վրայ եւ հիմնականում կիրառում է կենցաղում քոմփիւթերային խաղերի կամ փոքրածաւալ անհատական ինֆորմացիոն համակարգերի ստեղծման համար: Այս տիպի անհատական քոմփիւթերներում որպէս արտաքին սարքեր հիմնականում օգտագործում են կենցաղային հեռուստացոյցները եւ ձայնագրիչները: Արտասահմանում այդ սերնդի ամենասարածուած մեքենաները արտադրում են Commodore, Atari, Sinclair, Sharp եւ այլ ֆիրմաների կողմից: Դրանք լայն տարածում են զտել աշխարհում: Տուեալ սերնդին են պատկանում հայրենական «Միկրոշա», «Էլեկտրոնիկա-բկ 0010» եւ այլ քոմփիւթերները: Ներկայումս աշխարհում տարեկան արտադրում է շուրջ 6 միլիոն առաջին սերնդի անհատական քոմփիւթերներ:



Նկ. 6



Նկ. 7

Երկրորդ սերնդի անհատական քոմփիւթերները կառուցուած են 18 կարգանի միկրոպլոցեսորների հիման վրայ եւ նախատեսուած են հիմնականում մասնագիտական կիրառուամների համար: Նախորդ սերնդի համեմատութեամբ այս մեքենան ունեն հաշուարկների կատարման աւելի բարձր ճշգրտութիւն, 5 - 10 անգամ աւելի մեծ ծաւալի յիշողութիւն եւ չուրջ 10 անգամ աւելի մեծ արագագործութիւն: Այս առաւելութիւնները հարաւարութիւն են տալիս մեծացնել մասնադիտացուած արտաքին սարքերի տեսականին: Արտասահմանում մեծ տարածում են զտել այնպիսի ֆիրմաների արտադրած անհատական քոմփիւթերները, ինչպիսիք են IBM, Apple եւ այլն: Հայրենական արտադրութեան քոմփիւթերներից կարելի է նշել ԴՎԿ, «Էլեկտրոնիկա-85», «Խոկրա-1030», «Նեյրոն» եւ այլն մեքենաներ: Աշխարհում տարեկան արտադրում է այս սերնդի շուրջ 5 միլիոն անհատական քոմփիւթերներ:

Երրորդ սերնդի անհատական քոմփիւթերներում օգտագործուած են 32 կարգանի միկրոպլոցեսորներ, որը հարաւարութիւն է տալիս ապահովել հաշուարկների չատակի մեծ ճշգրտութիւն եւ արագագործութիւն: Այս յատկութիւնները նպատակայարմար են դարձնում երրորդ սերնդի անհատական քոմփիւթերների կիրառում գիտական խնդիրների լուծման, արագընթաց գործընթացների կառավարման համար, նախագծման աւտոմատացուած համակարգերում եւ այլն: Այս սերնդի անհատական քոմփիւթերների մասսայական արտադրութիւնը նոր է սկսում եւ չունի այն լայնածաւալութիւնը, որը բնորոշ է առաջին երկու սերնդների համար:

Ներկայումս ըստ իրենց բնութագրերի, տարածման աստիճանի եւ կիրառման ոլորտների տառաելագոյն հետաքրքրութիւն են ներկայացնում երրորդ սերնդի անհատական քոմփիւթերները, որոնք յաճախ կոչւում են մասնագիտական անհատական քոմփիւթերներ: Այդ զատի քոմփիւթերներից ամենասարածուածը այսօր IBM ֆիրմայի անհատական քոմփիւթերներն են, որոնք, ըստ իրենց կառուցւածքի եւ ծրագրային տպահովման, զրեթէ բոլոր երկրներում, այդ թւում եւ մեր երկրում, ընդունուած են որպէս նախատիպ երկրորդ սերնդի անհատական քոմփիւթերների նախագծման եւ արտադրութեան համար: Վերջին տարիների արտադրուած մասնագիտական անհատական քոմփիւթերների ճնշող մեծամասնութիւնը կազմում են IBM ֆիրմայի անհատական քոմփիւթերների հետ լիովին համատեղելի մեքենաները, այդ թւում եւ հայրենական «ЕС-1840/41» (նկ. 8), «Խոկրա-1030» (նկ. 9), «Նեյրոն» անհատական քոմփիւթերները: Հաշուի առնելով այս հանգամանքը եւ այն որ կրթութեան բնագաւառում մօտ ապագայում կ'օգտագործուեն հիմնականում այս տիպի անհատական քոմփիւթերներ, յետազայ չարաղրանքը կը ներկայացնի բացառապէս նրանց տեխնիկական ծրագրային միջցների օրինակի վրայ:

2. 3. Անհատական կիրառական ծրագրերի յարմարաւետութիւնը

Անհատական քոմփիւթերների հաշուողական հզօրութիւնը անմիջապէս օգտագործման հարաւարութիւնը ընձեռեց բազմազան մասնագիտական առարկէղներում: Անհատական քոմփիւթերների օգնութեամբ սկսեցին լուծել հաշուապահական խընդիրներ, վերամշակել տեքստեր, ստեղծել անհատական տուեալների հիմնապահուներ եւ այլն: Կողմնորոշումը դէպի տարբեր մասնագիտութեան տէր մարդկանց



Նկ. 8



Նկ. 9

լայն խաւերը սկզբունքօրէն փոխեց նման տիպի խնդիրների լուծման համար նախառեսուող ծրագրային համակարգերի ստեղծման սկզբունքները: Ոչ շատ վաղուց վերը նշուած տիպի աշխատանքներ կատարելու համար գրեթէ իւրաքանչիւր օգտագործող ստիլուած էր ստեղծել իր կիրառական ծրագրերը օգտուելով բարձր մակարդակի որեւէցէ ծրագաւորման լեզուից: Այդ հանդամանքը դժուարացնում էր կիրառական խնդիրների զգալի մասի լուծումը, քանի որ տուեալ բնագաւառի մասնագիտները ստիլուած էին լինում դիմել ծրագրաւորողների օգնութեանը կամ ինքնուրոյն տիրապետի որեւէ ծրագաւորման լեզուի՝ փաստորէն ձեռք բերելով ծրագրաւորողի երկրորդ մասնագիտութիւն: Ակնյայտ է, որ ըլոր այն մասնագիտները, որոնց գործունութեան մէջ քոմիտերների օգտագործման անհրաժեշտութիւն է առաջացել, չին կարող ծրագաւորող դառնալ:

Անհատական քոմիտերների գարզացումը երեւան բերեց համապիտանի ծրագրային համակարգեր, որոնք բաւական լայն դասի խնդիրների լուծման հնարաւորութիւն են ընձեռնում: Այդպիսիք են այսօր տեքստերի վերամշակման, աղյւսակային հաշուարկների, ինֆորմացիոն համակարգերի ստեղծման եւ այլ գործողութիւնների համար նախատեսուած ծրագրային միջոցները: Համապիտանի ծրագրային համակարգերը ներկայումս ամենասուարածուած ծրագրային միջոցներն են: Այդ դասի առանձին ծրագրային համակարգերը ունեն միլիոնաւոր տպաքանուններ:

Կիրառական ծրագրային համակարգը նախատեսուած է ինֆորմացիոն օրիենտաների որոշակի դասի վերամշակման համար՝ տեքստ, աղյւսակ եւ այլն: Քոմիտերների վրայ աշխատելիս օգտագործողը դործ ունի ոչ թէ ոչալ օրկիելատի, այլ նրա մէքենայական մողելի հետ: Իւրաքանչիւր ծրագրային համակարգի համապիտանութեան աստիճանը պայմանաւորուած է նրանով, թէ որքանով են համարժէք միքենայական մողելը եւ ոչալ օրիենտը: Մեքենայական մողելի յաջող ընարութիւնը, ինֆորմացիոն օրիենտի յատկութիւնների եւ նաև նկատմամբ կատարուող գործողութիւնների արտապատկերման ընականութիւնը եւ ակնառութիւնն են որոշում ծրագրային համակարգի հետ աշխատելով յարմարաւէտութիւնը:

Օրինակ՝ տեքստերի վերամշակման համակարգը պէտք է ապահովի տարրեր տիպի փաստաթիթերի հետ աշխատանքը բովանդակային մակարդակի վրայ եւ հնարաւորին չափամբովին հաշուի առնի ոչալ տեքստային փաստաթիթերի կառուցուածքները, այսինքն՝ կարողանայ ընկալել տեքստի բաղկացուցիչ մասերը (բառ, պարբերութիւն եւ այլն) եւ ձեւաւրոյի էջերից, պարագրաֆներից եւ զլուխներից բաղկացած փաստաթիթերին տեսքը: Լաւ համակարգը վերամշակման ընթացքում տեքստը տեսատիպի վրայ պէտք է ներկայացնի հնարաւորին չափ ընական, «թղթային» տեսքով: Քոմիտերուում ներկայացուած ինֆորմացիոն օրիենտի օգտագործողի համար ընականութիւնը եւ նրա հետ գործողութիւնների սովորականութեան աստիճան են որոշում ծրագրային համակարգի յարմարաւէտութիւնը:

70ական թուականների կիրառական ծրագրային համակարգերը դժուար է կոչել յարմարաւէտ: Այդ համակարգերի համար բնորոշ է օգտագործողի հետ հրամայական երկխօսութիւնը: Ընդ որում ինֆորմացիոն օրիենտ սովորաբար ներկայացնում էր վերացական ձեւով եւ օգտագործողը նրա վրայ ազդում էր ստեղնաշարից ներածուող հրամանների միջոցով: Համակարգը, ընկալելով այդ հրամանները, սովորաբար տեքստային հաղորդագրութիւնների ձեւով յայտնում էր օգտագործողին նրանց

կատարման արդիւնքների մասին: Այսինքն՝ օգտագործողը օրիենտի վերջնական տեսքի փոխարին տեսնում էր նրա վերածական մեքենայական զնահատականութիւնը:

Ինֆորմացիոն օրիենտաների հետ աշխատանքի նոր եղանակների որոնումները բերեցին օրիենտային երկխօսութեան զարափարին, որը իրենից քոմիտերի ղեկավարման եւ բազմաբովանդակ ինֆորմացիոն օրիենտաների հետ աշխատանքի բաւականին յարմար միջոց է ներկայացնում: Ի տարբերութիւն նախկին հրամայական երկխօսութեան, օրիենտային երկխօսութեան մամանակ արտապատկերուում է օրիենտի մեջնայական մողելի ընթացիկ վիճակը: Օգտագործողի տեսակի այց նա ոչ թէ որեւէց վերացական լինուուկ երկխօսութիւն է տանում, այլ պարզունակ ստեղնակը սեղմուում է սեղուում եւ նախարարի մասնակի արտապատկերուում: Պոմպիւթիւնը հետո այց առաջարկուում է սեղուում եւ ներկայական է սեղուում բոլոր այցերի մասնակի արտապատկերուում:

Մրագրային համակարգերի յարմարաւէտութեան կարեւոր գործներից է օգտագործողին աշխատանիքի բնմացքքում համակարգի մասին անհրաժեշտ ինֆորմացիայով ապահովիլը: Համակարգի համարաւորութիւններին եւ նրա հետ աշխատելու սկզբունքներին համապատասխան ժամանակի ներկայացնուուղ ինֆորմացիան, եւ լրացուցիչ զրականութիւնը սովորաբար չի պահանջնուում: Բառական է հետեւել տեսատիպի վրայ տեղի ունեցող իրադարձութիւններին եւ կատարել քոմիտերի կողմից տուածրակուուղ դործողութիւնները: Տարբեր բարդութեան տատիճանի ծրագրային համակարգերի օժանուած են լինում տարբեր մանրամասնութեան տատիճան ունեցող բացատրական ինֆորմացիայի ներկայացնուած կամ, ինչպէս յաճախ կոչուում են, յուշման ենթահամակարգերով: Բարձր յարմարաւէտութիւն ունեցող ծրագրային համակարգի ընդհանուր ծաւալի զերակշուուղ մասն է կազմում սավորաբար յուշման ենթահամակարգը:

Յաջորդ կարեւոր գործնր, որ ապահովուում է ծրագրային համակարգի յարմարաւէտութիւնը՝ ուսուցման ենթահամակարգն է: Պարզուացն զէպրում ուսուցման ենթահամակարգը բարձրացած է մի շաբաթ բնորոշ օրինակներից, որոնք արտում են օգտագործողին ծրագրային միջոցների հետ մէկսեղ: Վերլուծելով այդ օրինակներից այլ օրինակների բնորոշուուղ իր գործողութիւնները բացատրագրի մէջ բերուածի հետ՝ օգտագործողը ըմբռնուում է աշխատանիքի հիմնական նրբութիւնները: Աւելի բարդ ծրագրային համակարգերը պարտանակուում են յատուկ ուսուցանող ծրագրեր, որոնք յատուկ կերպով կազմակերպուած երկխօսութեան միջոցով մանրամասնորէն համակարգի հետ աշխատելու ունակութիւններ են հաղորդուում օգտագործողին:

Վերջապէս ծրագրային համակարգի յարմարաւէտութեան ոչ պակաս կարեւոր գոծոնն է կազմուում մանրամասնորէն եւ մատչելի ձեւով գրուած բացատրագիրը, որը պէտք է ընդդրկի համակարգի բոլոր հնարաւորութիւնները եւ նրա հետ աշխատանիքի նրբութիւնները: Հարգ է նշել, որ անկախ բուռն ծրագրային, յուշով եւ ուսուցանող ենթահամական համակարգի համար նախաթիւնից համակարգի հետ օգտագործողի աշխատելու արդիւնականութիւնը: Աւելի հարգ ծրագրային համակարգերը կատարուած են յատուկ ուսուցանող ծրագրեր, որոնք յատուկ կերպով կազմակերպուած երկխօսութեան միջոցով մանրամասնորէն համակարգի հետ աշխատելու ունակութիւններ են հաղորդուում օգտագործողին:

Ա. 4. Տեխնոլոգիական համակարգերի համար ստեղծուած առաջին ծրագրային միջոցների

Անհամական քոմիքթերների համար ստեղծուած առաջին ծրագրային միջոցների թույլ էին տեքստերի գերամշակման համակարգերը։ Այս համակարգերի յարժարագութեանը մատչելի դարձեց տեքստերի մեքենայական գերամշակումը տարրեր մասնագիտութեան տէր մարդկանց լայն խաւերի համար։ Ներկայիս անհամական քոմիքթերների զգալի մասն օգտագործում է հիմնականում տեքստերի գերամշակում համար։

Տիպային տեքստերի գերամշակման համակարգը զարդացնում է աւանդական մեքենագրման հնարաւորութիւնները, եւ քոմիքթերը դրցնում է տեքստերի նախապատճառաման, խմբագրման եւ տպագրման հզօր գործիք։ Տեքստի նախապատճառամանը կամ ներածումը ոչնչով չի տարբերում սովորական մեքենագրումից։ Ներածուող տեքստը տող առ տող արտապատկերում է տեսասիպի էկրանի վրայ, ինչպէս մեքենագրման ժամանակ թղթի վրայ։ Ընդ որում տեքստի ներածման ժամանակ պարտապիլը չէ հետեւել տեսատիպին, քանի որ լնիմացիկ տողի աւարտից յետոյ ներածուող բառը համակարգի կողմից ամրողովին տեղափոխում է յաջորդ տողի վրայ։ Համակարգը հնարաւորութիւն է տալիս ներածուած տեքստը ոչ միայն պարզուին ինմրազըել, այլև կատարել բաւական բարդ գերամշակման զործողութիւններ։ Ստացուած տեքստը կարող է զրանցուել քոմիքթերի արտաքին յիշողութիւնում, մասնաւորապէս մազնիսկան միաւուասիկի վրայ, տուեալ քոմիքթերի օպերացիոն համակարգում ընդունուած Փայլի, այսինքն՝ որոշակի անուանուած գրանցումի ձեւով։ Այս դէպքում քոմիքթերի տեսատիպի էկրանի իրենից ներկայացնում է մի պատուհան, որի միջով երեւում է տեքստի որոշակի մասը։ Քոմիքթերի սակեցնաշարի համապատասխան սաեղների սեղմելու միջնորդ այդ պատուհանը կարելի է տեղաշարժել երեւակայական տեքստի վրայով եւ տեսանելի զարձնել խմբազուուղ տեքստի անհրաժեշտ մասը։ Տեսատիպի վրայ արտածուած տեքստի սահմաններում յասուկ ցուցիչը հնարաւոր է տեղաշարժել յաջորդ ներածուող նիշի տեղը ցոյց տալու նպատակով։ Համակարգը ներածուած տեքստերի խմբագրման լայն հնարաւորութիւններ է ընձեռնում։ Անհրաժեշտութեան դէպքում հնարաւոր է հետացել աւելորդ տառը, բառը, տողը կամ անդամ տեքստի մի ամբողջ հատուածը։ Համապատասխան տեկումից հերթին հնարաւոր է աւելացնել բաց թողնուած տառը կամ նոյնիոկ տեքստի մի ամբողջ հատուածը։ Վերջինը հնարաւորութիւն է ստեղծում նախապէս պատրաստած հատուածներից հատուել տեքստը՝ կրկնելով համապատասխան հատուածները անհրաժեշտ անդերում։

Երկիոսութիւնը քոմիքթերի հետ կադմակերպում է այնպէս որ օգտագործողը միշտ անուում է տեսատիպի վրայ տեքստի եւ համակարգի լնիմացիկ վիճակը։ Համակարգի վիճակը եւ հիմնականում անհրաժեշտ հրամանների ցանկը մինիուրի ձեւով արտապատկերում է սովորաբար տեսատիպի էկրանի վերին մասում, իսկ էկրանի միւս մասը ամբողջովին օգտագործում է խմբագրուող տեքստի համապատասխան մասի պատապատկերման համար։ Օգտագործողը, սաեղնաշարի որոշակի սաեղներ ուղարկուի, համակարգին հաղորդում է պահանջնուող հրամանը, որի կատարման հետեւանուով փոփոխում է համակարգի եւ տեքստի լնիմացիկ վիճակը կամ կատարուում է անցուկ մէկ ուրիշ մենիուի, այսինքն՝ ուրիշ հրամանների ցանկը։ Ոգտագործողի բայոր թոյլաւուած զործողութիւնները համակարգի կողմից պար-

զարանուում են տեսատիպի վրայ համապատասխան յիշումների եւ մենիւների արտածման միջոցով։ Համակարգում օգտագործողին մատչելի են աւելի քան 120 հրամաններ, որոնք ըստ նշանակութեան խմբաւորուած են 7 մենիւների մէջ։ Զնայած դրան համակարգի հետ արդիւնաւէս աշխատանքը սկսելու համար բաւական է ձանօթ լինել մի քանի հիմնական հրամանների եւ սատիճանաբար ուսումնասիրել անհրաժեշտնոր հնարաւորութիւնները եւ նրանց համապատասխանող հրամանները։ Էկրանի վրայ շատ տեղ չզրադեցնելու նկատառութիւն է կատարուում էկրանի մէջ հրամանները ըստ անակտում են շատ կարճ ձեւով։ Հրամանների մասին աւելի հանդամանալից ինֆորմացիա ստանալու նպատակով կարելի է օգտուել համապատասխան յուշումից։ Դիտարկենք համակարգի մենիւներից երկու կարեւորագոյնները։

Համակարգի թողարկումից յետոյ օգտագործողն առաջարկուում է մուտքային մենիւյից (նկ. 10) ընտրել կատարման համար անհրաժեշտ զործողութիւնը։ Ընտրութիւնը կատարուում է մենիւյի մէջ պահանջնուող զործողութեանը համապատասխանող տարի ներածման եղանակով։ Տուեալ դէպքում մատչելի են հետեւեալ դործողութիւնները։

Ն - հետագայ տեխստերի գրանցման կամ լնիմացման համար նոր ծառայողական սկավառակի ընտրութիւն։

Ց - ծառայողական սկավառակի վրայ գրանցուած ֆայլերի ցուցակի արտածման անհրաժեշտ հշուարձեալ սկզբունքը մէջ համապատասխանաբար նշուում է ՄԱՅթրած թէ ԱՆՁատած է, ցուցակի արտածումը։ Սուածին դէպքում ֆայլերի ցուցակը արտածում է մուտքային մենիւյի տակ։

Ր - անհրաժեշտ բացատրման կամ յուշման մակարդակի ընտրութիւն։ Նշուում է, թէ աշխատանի ընթացքում համակարգի ը՞ր հրամանները եւ մենիւնները պահանջման է պարզաբանուեն։

Փ, Ծ - վերամշակող տեխստի տիպի (փաստաբույր կամ ծրագիր) ընտրութիւն։

Տ - Նշուած տեխստի արտածում տպոյ սարքի վրայ։ Վ, Պ, Հ - սկավառակի վրայ գրանցուած տեխստային ֆայլի վերանուանում, պատթեւաւորում, հեռացում։

Ա - համակարգի հետ աշխատանի աւարտում։

Թ - համակարգի հնարաւորութիւնները ընդլայնող լրացուցիչ ծրագրի բողարկում։

ՄՈՒՏՔԱՐԿԻՆ ՄԵԽՆԻ		
- Նախնական հրամաններ -	- Տեխստային հրամաններ -	- Ելբային հրամաններ -
Նոր սկավառակ	Տ Տպում	Տ Աւարտ
Ց Ցուցակ	Ց ՄԱՅթ (ԱՆՁ)	Ց Թողարկում
Ր Բացատրման մուկարդակ	Ր Փաստաբույր ստուգում	Ր Թողարկում
Պ Խմբագրման մուկարդակ	Պ Պատթեւաւորում	
Ծ Խմբագրման մուկարդակ	Ծ Հեռացում	
Ֆ Փաստաբույր		
Ծ Ծրագիր		

Համապատասխան ընտրութիւնների աւարտից եւ պահանջուող տեքստային ֆայլի անուանման ներածումից յետոյ համակարգը՝ կ'անցնի զիսաւոր մենիւյի ուժիմ եւ պատրաստ կը լինի նշուած տեքստի ներածմանը կամ խմբադրմանը։ Այս ուժիմում տեսատիպի էկրանի վերին մասում արտածում է համակարգի ընթացիկ վիճակը եւ զիսաւոր մենիւն (նկ. 11)։ Տեսատիպի վերին տողում, որը ցոյց է տալիս համակարգի վիճակը, ներկայացուած են վերամշակուող տեքստի անուանումը, ցուցիչի դիրքը խմբագրուող տեքստի մէջ (էջ, տող, սինակ) եւ ինֆորմացիայի ներածման ձեւը՝ վերադիր կամ ներդիր։ Առաջին դէպում ներածուած նիշը զրագեցնում է ցուցիչի դիրքում մինչ այդ գտնուող նիշի տեղը չնշելով այն, իսկ երկրորդ դէպում ցուցիչի դիրքից աջ գտնուող տողի մասը ներածուող նիշի համար տեղ ազատելով՝ տեղաշարժում է կ'էպի աջ։ Գլաւոր մենիւյի լույրը հրամանները ներկայացուած են համապատասխան տառերով, որոնց նախորդում է յատուկ «յայտանիշը»։ Հրամանի ներածման համար անհրաժեշտ է քոմիտեների ստեղնաշարի վրայ սղմած պահել յատուկ դեկարման ստեղնը եւ սեղմել համապատասխան տառի ստեղնը։

A: TEXT

կջ 2

ՏՈՂ 13 ՍԻՒՆՍԿ 26

ԳԼԽԱԼԻՌ ՄԵԽՆԻ

ՆԵՐԴԻՐ

— Ցուցիչի տեղաշարժ —	— Հեռացում —	— Լրացուցիչ —	— Այլ մենիւ —
Նիշ՝ »Ձ ձախ« »Ձ աջ Բառ՝ »Թ ձախ« »Ծ աջ Տող՝ »Ձ վերև« »Ձ ներքեւ — Պատուիանի տեղաշարժ —	»Ց նիշի »Զ քառի »Լ տողի	»Բ Քայլ »Ն ներդիր »Փ Փնտումը կրկնել »Ը Ընդհատում »Զ Զեւաչափի	»Բ Բացարման »Խ Խմբային »Է Էկրագ »Ա Արագ »Տ Տպման

Նկ. 11

Համակարգի միւս մենիւնները մի շարք աւելի բարդ տեքստի վերամշակման գործողութիւններ կատարելու հնարաւորութիւն են տալիս։ Դրանց թույն են պատկանում խմբային տեղաշարժերի, բառականակցութիւնների փնտուման եւ փոխարինման, տեքստի ձեւափառուման, տպող սարքի դեկարման եւ այլ գործողութիւններ։

Խմբային տեղաշարժի գործողութիւններ կատարելու համար պէտք է »Ձ հրամանի միջոցով դիսաւոր մենիւյից անցնել խմբային մենիւյի։ Այս մենիւյին հնարաւորութիւն է տալիս տեքստի մի մասը նշել որպէս մի այլ տեղ, դրանցել սկաւառակի վրայ որեւէ Քայլի մէջ եւ առանձին ըլուկ ընթերցել սկաւառակի վրայ զրանցուած ֆայլից խմբադրուող տեքստի մէջ տեղադրելու նպատակով։ Խմբային մենիւյից հնարաւորութիւն են ընձեռնում գոյութիւն ունեցող տեքստերի առանձին մասերից բաւական արագ հաւաքել մի նոր տեքստ։

Մեծածալ տեքստերի վերամշակման ժամանակ յաճախ որիւէ տեքստի համար արագ գործողութիւններ կատարելու միջոցներ է տրամադրում։ Օրինակ՝

տուեալ տեքստում անհրաժետ է «մեքենայ» բառը փոխարինել «քոմիտեներ» բառով։ Տուեալ դէպում պէտք է արագ մենիւյից ընտրել փոխարինման հրամանը, որից յետոյ պատասխանելով համակարգի հարցերին որպէս հատուած, որը պէտք է գտնուի նշել «մեքենայ» բառը, իսկ որպէս հատուած, որով պէտք է փոխարինուի գտնուածը՝ «քոմիտեներ» բառը։ Տեքստի մէջ «մեքենայ» բառը գտնելուց յետոյ համակարգը տեսատիպի վրայ կ'արտածի հետեւեալ հարցը՝

ՓՈԽԱՐԻՆՄՅՆ (Ա/Ռ)

Ի պատասխան՝ ներածուած Ա տառը հաստատում է փոխարինումը, իսկ Ո տառը՝ արգելում այն, որից յետոյ համակարգը գտնում է տեքստի մէջ յաջորդ «մեքենայ» բառը եւ կրկնում հարցը։ Անհրաժեշտապես նշել այս գործողութեան կատարման այնպիսի ուեժիմ, որի դէպում փոխարինումը կատարում է առանց հարցման եւ համակարգը ամբողջ տեքստում հանդիպող «մեքենայ» բառերը աւտոմատ կերպով կը փոխարինի «քոմիտեներ» բառով։

Տեքստի ձեւաչափաւորման գործողութիւնները կատարուում են էկրանային մենիւյի հի հրամանների միջոցով, որոնց կարելի է անցնել զիսաւոր մենիւյից »Ձ հրամանի միջոցով։ Այս հրամանները հնարաւորութիւն են տալիս սապարաշինել վերնազերել, ընարել կը չափականներ եւ պարբերութիւններ ձեւաորի։ Ձեւաչափաւորմը վերականում է մարդու համար բնակուն տեսրով այնպիսի տարրեր, ինչպիսիք են՝ բառը, նախադասութիւնները եւ պարբերութիւններ։ Էկրանային մենիւյից մէջ նախատեսուած են ձեւաչափաւորման յատկանիշների նշոնակման հրամաններ։ Այդ յատկանիշների թույն են պատկանում տեքստի ձախ եւ աջ սահմանները, միջտողային սարածութիւնը, եղբերի հաւասարեցումը, բառերի փոխանցումը յաջորդ տողի վրայ եւ այլն։ Տեքստի ձախ եւ աջ եղբերի հաւասարեցումը հնարաւորութիւն է տալիս բառերի միջեւ լրացուցիչ բացատրելու աւելացնելու եղանակով՝ ակրսութ բերել աւելի դիրքին տեքստի։ Տեքստի աջ սահմանը անցնող ներածուուղ բառերի աւտոմատ տեղափոխուումը յաջորդ տողի վրայ հնարաւորութիւն է ստեղծում արագ ներածում կատարել առանց հետեւելու տեքստին տեսատիպի վրայ։ Ներածման ժամանակ համակարգը բաժանում է տեքստը չէկրել՝ նշելով մի էլեկտ միւս սանդան անցնուած աւելացնելու համար։

Գլաւոր մենիւյից »Ձ հրամանի միջոցով կատարուում է անցում ապման մենիւյին, որում ընդպրկուած են տպող սարքի դեկավարման հրամանները։ Այդ հրամանները նախատեսուած են տպատառերի տեսքը, չափսերը, խտութիւնը, միջտողային հեռաւորութիւնը եւ տպող սարքի մի շարք ուրիշ բնութագրերը փոփոխելու համար։

Տեքստերի վերամշակման համակարգը աշխատանքի բնփացքում (ցանկացած իրաւիճակներ) օգտագործողին մանրամասն ինֆորմացիան է ներկայացնուում տուեալ պահին բոլոր հնարաւոր գործողութիւնների մասին։ Անպատ կարող է տպատրութիւն ստեղծուել, թէ քոմիտեները «գլաւակցօրէն» յուշում, քանի ող նա յարմարուում է օգտագործողի աշխատանքի տեմպուն։ Փոքր չունեցող օգտագործողը դանդաղ է ներածում հրամանները եւ համակարգը ամէն հրամանից յետոյ աեսատիպի վրայ արտածուում է անհրաժետ բնիւններ եւ համապատասխան մենիւն։ Երբ փորձառու օգտագործողը բառականին արագ ներածում է հրամանների յաջորդականութիւնը, համակարգը չի հասցնուած յուշումներ արտադրել եւ անմիջապէս անցնուում է նրանց կատարմանը։

Համսկարզի հնարաւորութիւնների, մենիւների եւ հրամանների մասին մանրամասն ինֆորմացիա կարելի ստանալ բացաւումն մենիւնի հրամանների միջով, որոնց կարելի է անցնել զիմաւոր մենիւնից »Տ հրամանի միջոցով: Մուռքային մենիւնի Բ հրամանի միջոցով օգտագործողը կարող է նշանակել իր փորձառութեանը համապատասխան յուշման ասուիճան: Սկսնակ օգտագործողի համար արսուածուում են լուլոր մենիւները եւ բացասարութիւնները: Փորձառու օգտագործողը կարող է արդեւ մենիւների եւ բացասարութիւնների սրտածումը, լրացուցիչ տեղ ազատելով անսասափի էկրանի վրայ եւ արագացնելով համակարգի աշխատանք:

2. 5. Աղիւսակների վերամշակման համակարգեր

Անհատական քոմիւթերների լայն սարածումը անհրաժեշտ դարձեց նոր տիսլի ծրագրային համսկարգերի ստեղծումը, որոնք հնարաւորութիւն են տալիս հաշուացին լրացր զործողութիւններ կատարել տանոց բարձր մակարդակի լեզուների օգտագործման եւ աւանդական իմաստով ծրագրաւորելու: Միայն այս զէպում կարճ ժամկատում քոմիւթերների օգտագործողները կր դառնան «Հաշուացին» աշխատադիրի՝ սննեսագչտների, հաշուապահների, ինժեներների եւ հիմնարկների ծառայողների:

Աւանդական «Քղթային» ինֆորմացիոն տեխնոլոգիայում տուելուների եւ արդիւնքների ներկայացման ստենսկարեւոր եւ յաճախ օգտագործուած եղանակը աղիւսակի տեսքով թուային ինֆորմացիայի ներկայացումն է: Աղիւսակի անեսքով կազմուամ են լրազմապիսի ցուցակներ, ամփոփագրեր, հաշիւներ, ինժեներներն էականական էամար անհրաժեշտ ելակէտային տուելուներ եւ այլն: Աղիւսակների մերժով կատարուող վերամշակման ժամանակ կուղմուում է լրանկ, որի առանձին վանդակները ծառայում են կամ որպէս ելակէտային տուելուներ եւ նախապէս լրացուում են թուերով, կամ որպէս աղիւսակում արդէն զոյութիւն ունեցող տուելուների հանդէս հաշուացին զործողութիւնների արդեւնքները: Այս եղանակով են հաշուարկուում եւ լրացուում աշխատավարձի ցուցակները մշակւում զիտափարձերի արդիւնքները և այլն: Ինֆորմացիայի աղիւսակուին ներկայացման եղանակը այնքանով լրամար և զիրքին եղաւ որ մի շորբ բարձր մակարդակի ծրագրաւորման լիզուներում հաշուացին խնդիրների ծրագրաւորումը եւ արդեւնքները ներկայացուում պարզինակով աւելացուեցին ինֆորմացիան աղիւսակային տեսքով արտածող բահանդիներ:

Անկան թուականների սկզբին ստեղծուեցին ինֆորմացիան աղիւսակի տեսքով գրգուցնազ և վերամշակող յասուուկ ծրագրային համակարգեր՝ այսպէս կոչուած էրկարուային ողիւսակներ: Էլեկտրոնային աղիւսակները սովորական աղիւսակների մերժաւական մոդելներն են, որոնք կազմուած են քոմիւթերի էլեկտրոնային յիշացուինաւում մէջ պահուող եւ վերամշակուող տողերի եւ սիւնակների լրազմութիւնից: Անվարուապի աղիւսակի տողերը նշանակուում են թուերով, իսկ սիւնակները՝ տակրուի: Տողի եւ սիւնակի համամա վայրը իրենից աղիւսակի վանդակ կամ ըջիջ և ներկայացնուում: Աղիւսակի տոանձին վանդակի նշիչ (հասցէ) է հանդիս զովիս տուուկին համապատասխանող տառերը եւ թիւը: Օրինակ՝ դ10ը դոյլը և տուիս զովիսը սիւնակի (դ) եւ տասներորդ հատման վայրում գտնուուու

վանդակը: Էլեկտրոնային աղիւսակի չափսերը սահմանափակւում են միայն քոմիւթերի յիշողութեան ծաւալով եւ ժամանակակից քոմիւթերում կարող է հասնել մինչեւ հարիւրաւոր տողերի եւ սիւնակների: Աղիւսակի վանդակների հասցէաւուրումը պարզաբանուում է նկ. 12ով:

Այս աղիւգի ծրագրային համակարգը հնարաւորութիւն է տալիս անմիջապէս քոմիւթերի տեսատիպի վրայ լրացնել մի էլեկտրոնային աղիւսակ, որի վանդակներում կարող են տեղադրուել կամայական թուային տուեալներ, սեքստեր բանաձեւեր: Համակարգը հնարաւորութիւն է րնձեւուում տուեալների ներսուում մից կամ փոխութիւնից յետոյ բանաձեւերի վերահաշուարկ կատարել եւ ստացուած արդիւնքը յիշել կամ արածել թղթի վրայ աղիւսակի ձեւով: Համակարգութեամբ նախատեսւում են նաեւ աղիւսակի աւատումատ վերամշակման յատուկ ծրագրերի ստեղծուան եւ կատարման միջուցներ: Օգտագործողի աղիւսակնը համակարգի հետ կատարուում է բուռն կատարուում է երկանուութեան կանակնուու: Ինդ սրում իւրաքանչիւր դորձողութիւնից յետոյ օգտագործողը տեսնուում է աղիւսակի վերջնական տեսքը: Համար նախաւակուրի համար աշխատասխան յուշող համակարգ: Ամէն կոնկրետ իւրաքանչիւր մէջ սեղմելով հարցական նշանի ստեղնը (՞՝ կարելի է տեսնել նրա համապատասխան հատուածը: Աղիւսակնը կամ աղիւսակը բնիորմացիան տուեալ պահուի վրայ առանուալ սպառիչ բնիորմացիան տուեալ պահուի հերարերեալ: Մրազը թղարկութիւնից յետոյ տեսատիպի վրայ արտապատկերուում է էլեկտրոնային աղիւսակը (նկ. 12): Աղիւսակի սիւնակները նշանակուում են աւանձին սառերուկ ա, բ, գ, ..., փ կամ տառերի գոյգերով՝ աա, ար, ատ, ..., աֆ, բա, բբ, ..., ֆի, իսկ տողերը՝ թուերով: Եթէ աղիւսակը աղիւսակը չի անդառանաւուում է կարելի է կատարուում, որ աղեաչարժելով աղիւսակի վրայով կարելի է տեսնել նրա համապատասխան հատուածը: Աղիւսակնը կամ բնիորմացիան աղիւսակների խմբերու լրուկները: Բնիկը աղիւսակի մի ուղղանկան հատուած է, որը հասցէաւուում է երկու կտոսվ բաժանուած է աղիւսակների միջոցավու: Եթէ ցոյցիչը աղեաչարժելում է աղիւսակի էլեկտրոնայից զուրս, ապա կարծես իրենից մի պատուհան ներկայացնուու է կրանակները աղիւսակի վրայով այնպէս, որ աղիւսակի վանդակը միշտ երեւայ տեսատիպի վրայ: Որոշ սիւնակներ կամ տողեր են փիքսուել տեսատիպի վրայ, որպէսով պատուհան մերժաւական մոդելը նշանակուում է յարմար: Օրինակ՝ յարմար է փիքսը աղիւսակի վիստամասը միւս բնուր տողերը դիտարկելիս: Անհրաժեշտութեամբ զէպում տեսատիպի էլեկտրոնը կարելի բաժանել երկու մասի՝ հորիզոնական կամ ուղղահայեց ուղղութեամբ, որպէսով աղիւսակի բահանդին դառնուու:

Վանդակներից մէկը, որը կոչուում է ակսի վանդակ, նշուում է յատուկ աղիւսակային ցոյցիչի (< >) միջոցով: Ինֆորմացիայի ներկայացնուումը հնարաւորութիւն է նաև աղիւսակներ: Վանդակների կամ տողերի կամ փիքսների սպառունակող ցոյցիչի տեղադրուումը սակագնու աղիւսակի միջոցով: Եթէ ցոյցիչը աղիւսակի վանդակը աղիւսակի միջոցով աղիւսակի վանդակը միշտ երեւայ տեսատիպի վրայ: Աղիւսակի վանդակը միշտ երեւայ տեսատիպի վրայ, որպէսով պատուհան մերժաւական մոդելը նշանակուում է յարմար: Օրինակ՝ յարմար է փիքսը աղիւսակի վիստամասը միւս բնուր տողերը դիտարկելիս: Անհրաժեշտութեամբ զէպում տեսատիպի էլեկտրոնը կարելի բաժանել երկու մասի՝ հորիզոնական կամ ուղղահայեց ուղղութեամբ, որպէսով աղիւսակի բահանդին դառնուու:

Համակարգի հասպարկուումը անմիջապէս յետոյ վանդակները միայն տեսականօրէն գոյութիւն ունեն աղիւսակներ: Եթէ ցոյցիչը իշխողութեան մէջ աղել չեն զրացենուում: Վանդակի բամփիւթերի յիշողութեան մէջ աղել է զրացենուում միայն նրա

	ա	բ	գ	դ	ե	զ	է	օ	ռ	ժ	ի
1	ա1										ր1
2	ա2										
3		բ3									
4											
5											
6											
7											
8											
9											
10				դ10							
11											
12											
13	ա13										ր13

Նկ. 12 Էլեկտրոնային աղիւսակի վանդակների հասցեալորումը

առևելայ ներածելուց յետոյց: Հարկ է նշել, որ վանդակի պարունուկութիւնը եւ նրա արժեքը միշտ չեն համընկում: Եթէ վանդակը բանաւում է պարունակում, ապա նրա արժեքը հաւասար է բանածեւով հաշուարկուած արդիւնքին: Օրինակ՝ եթէ Ե4 վանդակը պարունակում է (Ե1+Ե2)/Ե3 եւ հաշուարկը կատարելու պահին Ե1, Ե2 եւ Ե3 վանդակներում գտնուում էին համապատասխանաբար 5, 13 եւ 3 ժուերը, ապա Ե4 վանդակի արժեքը կը լինի Յ թիւը: Տուեալ դէպում Ե4 վանդակը պարունակում է (Ե1+Ե2)/Ե3 բանածեւը, իսկ նրա արժեքը հաւասար է Յի: Աղիւսակի հետ կարելի է աշխատի երկար ժամանակ՝ անհրաժեշտութեան դէպում աղիւսակի պարունակութիւնը կարելի է զրոնցել արտաքին յիշողութեան մէջ, օրինակ՝ մազնիսական սկառակի վրայ՝ յետագայում օգտագործելու նպաստակով:

Տեսադրի ստորին մասում միշտ տեղաւորուած հն երեք ծառայողական տողեր, որոնցում ներկայացուած են աղիւսակի վիճակը, յուշումները եւ ներածուող ինֆորմացիան (տես նկ. 13): Առաջին տողը ներկայացնուում է ակտիւ վանդակի մասին ինֆորմացիան եւ նրա պարունակութիւնը: Երկրորդ տողը ներկայացնուում է աղիւսակի ընթացիկ վիճակը՝ վանդակի լայնութիւնը, քոմիեւթերի աղստ մնացած յիշողութեան ծաւալը, աղիւսակի զրադեցուած վերջին վանդակի հասցէն եւ որոշակի յուղում: Վերջապէս երրորդը տուեալների եւ Հրամանների ներածուածն տողն է:

Ստեղնաշարից ներածուած ինֆորմացիան արտապատկերուում է ներածման (երրորդ) տողի վրայ: Այդ տողը պարունակում է նաև մի ցուցիչ, որը ցոյց է տալիս թէ տողի որ զիրքում է ներածուած յաջորդ նիշը: Առաջին զիրքում ներածուած նիշը որոշում է աշխատանքի յետագայ ուղղութիւնը՝ տուեալների ներածուած կամ Հրամանների ներածուած: Հրամանների ներածուածը կատարուում է, եթէ առաջին զիրքում

	ա	բ	գ	դ	ե	զ	է	օ	ռ	ժ	ի
1	ԱԱՌՈՒՆ										
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
>ա1											
ԼԱՅՆՈՒԹԻՒՆ	6										
ՑԻՇՈՂՈՒԹԻՒՆ	356										
A>—											

Նկ. 13

ներածուած էր հեետեւեալ նիշերից մէկը՝ =, !, ՞, /: Այդ նիշերը ունեն հետեւեալ նշանակութիւնները՝

= դէպի նոր ակտիւ վանդակի անցում իրագործելու հրաման: Տուեալ նիշից յետոյ նշում է պահանջուող վանդակի հասցէն, իսկ Անրածման աւարտից յետոյ աղիւսակային ցուցիչը տեղափոխում է նշուած վանդակը.

! արդէն ներածուած բանաձեւերով վերահաշուարկ կատարելու հրաման: Վերահաշուարկի անհրաժեշտութիւն է առաջանում բանաձեւերում օգտագործուած որեւէ վանդակի պարունակութիւնը փոխելուց յետոյ.

; տեսադրի էկրանը երկու մասի բաժանելու դէպում մի մասից միասին անցնելու հրաման.

/ համակարգի հիմնական հրամաններից մէկի ներածման յայտանիշ: Հիմնական հրամանների ցուցակը կարի բացարութիւններով հետեւեալն է՝

- | | |
|-------------|---|
| Ա ւարտ | → համակարգի հետ աշխատանքի աւարտ. |
| Գ րանցում | → աղիւսակի գրանցում արտաքին յիշողութեան մէջ. |
| Ե կրան | → էկրանի բաժանում երկու մասի. |
| Ը ներբեցում | → աղիւսակի ընթերցում արտաքին յիշողութեան մէջից. |
| Լ բացում | → աղիւսակում նոր տողի կամ սինակի աւելացում. |
| Խ մրագրում | → վանդակի պարունակութեան խմբագրում. |
| Կ րինում | → նշուած վանդակների բլոկի պատճենաւորում. |
| Հ եռացում | → տողի, սինակի կամ բլոկի հեռացում. |

- | | |
|-------------|--|
| Մ աքրում | → աղիւսակի ամբողջովին հեռացում յիշողութիւնից. |
| Շ եղում | → տուի կամ սիւնակի տեղափոխում դեպի աղիւսակի մեկ ուրիշ մաս. |
| Զ պաշտպանել | → վանդակի պարունակութեան պաշտպանութեան հանում. |
| Պ պաշտպանել | → վանդակի պարունակութիւնը փոփոխութիւններից պաշտպանում. |
| Զ նջում | → նշուած վանդակների պարունակութեան ջնջում. |
| Ռ եժիմ | → աղիւսակի հետ աշխատելու ռեժիմի ընտրութիւն. |
| Տ պում | → աղիւսակի տարածում տպող սարֆի կամ տեսատիպի վրայ. |
| Ց իբում | → տուիրի կամ սիւնակների փիբում տեսատիպի վրայ: |

Միւս բոլոր նիշերը նշանակում են տուեալների ներածում, ընդ որում ներածուած առաջին նիշը ցոյց է տալիս տուեալների տեսակը: Տիքստի ներածումը սկսում է "կամ" նիշերով, իսկ ցանկացած ուրիշ նիշ, բացի վերը նշուածներից, նշանակում է բանաձեկի ներածում, ընդ որում թիւը զիտարկում է որպէս բանաձեկի մասնաւոր դէպք:

Համակարգի հիմնական հրամանները / նիշի ներածումից յիշոյ առաջարկում են կառարման հետեւեալ յուշումով՝

ՆԵՐԱՆՑՈՒՄ

Ա, Գ, Է, Ղ, Լ, Խ, Կ, Հ, Մ, Շ, Զ, Ջ, Պ, Ջ, Ռ, Տ, Ց, Ֆ, Ռ, Ց, Ց, Ց, Ց

Ցուշման մէջ թուարկած են հիմնական հրամանների առաջին տառելրը: Հրամանների մեծ մասը ունեն մի քանի մակարդակի ենթահրամաններ եւ հրամանի առաջին տառի ներածումից յիշոյ առաջարկում է յաջորդ մակարդակի յուշման տողը, որից պէտք է ընտրել համապատասխան հնժահրամանը եւ այզպէս չարունակ մինչեւ հրամանի կատարումը: Նման երկիխօսութեան ցանկացած պահն լրացուցիչ աեղեկութիւններ ստանալու համար պէտք է ներածել հարցական նշանը (Պ):

Նմանատիպ էլեկտրոնային աղիւսակները հնարաւորութիւն են տալիս արագ կառարելու րարդ թուարանական հաշուարկներ առանց որեւէ պատկերացում ունենալու բոմբիթերի ներքին կառուցուածքի եւ ծրագաւարման մասին:

2. Ինֆորմացիոն համակարգեր

Ժամանակակից անհատական քոմիֆիթերները բաւական հզօր անհատական ինֆորմացիոն համակարգեր ստեղծելու հնարաւորութիւն են ընծեռում, իսկ անհատական քոմիֆիթերները բաշխուած ցանցի մէջ ընդունումը տեսականորէն անսահման հզօրութեան ինֆորմացիոն համակարգ ստանալու հնարաւորութիւն է ստեղծում: Պարզույն ինֆորմացիոն համակարգի մոդելը կարելի է ներկայացնել քոմիֆիթերի յիշուածքեան մէջ պահուող ինֆորմացիոն գրառումներ բազմութեան տեսքով: Գրառումը իրբեւ մի ամբողջ դիտուող կապակցուած տուեալների տարբերի բազմութիւն և ինֆորմացիան համակարգում իրենից ներկայացնում է ըստ պարունակութեան

միասնու զրառումների մի խումը, որը յիշուում է քոմիֆիթերի արտաքին յիշողութիւնում, օրինակ մազնիխական սկաւառակների վրայ: Ինֆորմացիոն համակարգի պարզագոյն օրինակ կարող է հանդէս զալ հեռավուագիւթելը, որտեղ զրառումը կարող է ընդուրկել ազգանունը, անունը եւ հեռախօսահամարը: Ինֆորմացիայի ներածումը, ճշտումը, հեռացումը եւ փնտումը համակարգում է համապատասխանարար գրառումների աւելացման, ճշտման, հեռացման եւ դրանցից անհրաժեշտը գտնելու միջոցով: Դրառումը տուեալների գամախումբը է, իսկ դաշտը իր հերթին զրառման տարրն է: Դաշտը նկարագրում է յիշուում է տուեալների առանձին տարրերը: Հեռախօսազգրքի օրինակում իրեւել զաշտեր հանդէս զալիս առանձին ազգանունը, անունը, հեռախօսահամարը: Ամէն առանձին դաշտ համակարգում նկարագրում է համապատասխան յատկանիշերով, որոնք պարզաբանում են դաշտը (թուային կամ նշանային) եւ չափաը (թուային կամ նշանային դաշտի երկարութիւնը):

Ինֆորմացիոն համակարգում փնտուման գործողութիւններ կազմակերպելու համար որոշ զաշտեր պէտք է յայտարարուեն որպէս յայտանիշային, զրանք սովորաբար ամենասրնորոշ զաշտերն են: Հեռախօսազգրքի օրինակում դա կարող է լինել ազգանունը: Պահանջուող ինֆորմացիան գտնելու համար անհրաժեշտ է ներածել համապատասխան յայտանիշի արժէքը եւ համակարգը կ'արտածի այն գրառումները, որոնց համապատասխան զաշտը համընկնում է ներածուած յայտանիշի հետ:

Անհատական քոմիֆիթերների երեւան զալը անհրաժեշտ զարձրեց այնպիսի ինֆորմացիոն համակարգերի ստեղծումը, որոնք մատչելի լինեն ծրագրաւորման լեզուներ չիմացող օգտագործողների լայն իաւերին: Ինչպէս վերը դիտուած միւս համակարգերը, ինֆորմացիոն համակարգերը անհատական քոմիֆիթերների վրայ կառուցուած են օրիկապային երկխօսութեան զաղափարի հիման վրայ: Ինֆորմացիոն համակարգի ծեւաւորման համար անհրաժեշտ դաշտերի եւ գրառումների նկարագրութիւնը, հայցերի ձեւակերպումը եւ միւս պահանջուող դործողութիւնները կատարուելու հետ երկիխօսութեան հղանակով:

Տիպային ինֆորմացիոն համակարգն ունի աշխատանիքի երկու հիմնական սեժիմ՝ ինֆորմացիոն համակարգի ծեւաւորման եւ տուեալների հետ աշխատանիքի սեժիմ: Ինֆորմացիոն զրառման մոդելը համակարգում օգտագործողին ներկայացնում է լայն եւ երկար ձեւի կամ տիպարի տեսքով: Տիպարի հետ աշխատանիքը կազմակերպուում է յարմարաւաւ կը լրացնային երկխօսութեան ձեւով: Ինչպէս հետ վերը դիտուած տեսքերի վերամշակման համակարգը էլեկտրոնային մենիւնների համապատասխան մենիւն, իսկ կը լրացն միւս ինֆորմացիոն պատուհանն է, որը կարելի է տեղաշարժել տիպարի վրայով՝ նրա համապատասխան մասը տեսաների զարձնելու նպատակով: Տուեալ համակարգի մենիւնները չափ նման են տեսքերի վերամշակման համակարգը մենիւններին:

Համակարգի ծեւաւորման, այսինքն տիպարի նախագծման, ընթացքու տեսատիպի վերին մասում արտապատկերուում է համակարգի վիճակի տողը եւ ձեւաւորման մենիւն (նկ. 15): Վիճակի առող ցոյց է տալիս համակարգի ընթացքի վիճակը, աւելի ճիշտ՝ ցուցիչի ընթացքի վիճացքը: Այլ տողի վրայ արտածուում է ցուցիչը պարունակութիւնը եւ ցուցիչի վիճակը համարները, ինչպէս նաև տուեալ դաշտի երկարութիւնը եւ ցուցիչի վիճակը դաշտուած: Ցուցիչի տեղաշարժի դէպքում վիճակի վրայ արտածուած տականիքը կարուելու վերաբերութիւնը եւ ցուցիչի վիճակը դաշտուած:

Ա	Բ	Գ	Դ	Ե	Զ	Է	Ը	Թ	Ւ
1									
2			Բ2						
3				Դ3			Ը3		
4									
5			Դ5						
6									
7									
8									
9						Է9			
10			Բ10						
11									
12									
13									

Նկ. 14

ՏՈՂ 11 ՍԻՒՆԱԿ 20 ՀԱՄԱՐ 1 ԵՐԿԱՐՈՒԹԻՒՆ 21 ԴԻՐՔ 20
ՉԵԽԱԽՈՐՄԱՆ ՄԵՆԻ

- Ցուցիչի տեղաշարժ -	- Հեռացում -	- Աւելացում -	- Լրացուցիչ -
Դաշտ »մ ձախ »ձ աջ	»Ձ դաշտի »Ն նիշի	»Ք դաշտի »Խ նիշի	»Բ Բացատրում »Ճ Տպում
Սիւն »Թ ձախ »Մ աջ	»Ձ սիւնի »Լ տողի	»Շ սիւնի »Ռ տողի	»Ա Աւարտ »Ց Յայտանիշ »Դ Դաշտ
Տող »Ձ վերև »Ժ ներքեւ			

Նկ. 15

Տիպարը սովորաբար բաղկացած է համապատասխան կապակցող, յաճախ՝ յուշող տեղափոխ եւ դաշտերից: Տեղաշարժելով ցուցիչը էկրանի վրայով՝ կարելի է այն տեղաշրջել ցանկացած տեղում եւ այնտեղ ցանկացած տեքստ ներածել կամ ընդգծման գծիկների միջոցով դաշտ նշել: Գոյութիւն ունեցող տիպարը կարելի է տեղափոխ համանմանորէն խմբագրել՝ աւելացնել կամ հեռացնել որեւէ նիշ, տող կամ սիւնակ, աւելացնել, ընդլայնել, կրճատել կամ հեռացնել որեւէ դաշտ: Տիպարի ստեղծման կարեւորագոյն փուլը դաշտերի յատկանիշների նշանակումն է: Դրա համար ցուցիչը պէտք է տեղաշրջել համապատասխան դաշտում եւ ներածել »Հ կամ »Դ հրամանները: Առաջին հրամանը (»Հ) սուեալ դաշտը նշանակում է որպէս յայտանիշը (*): Եթէ մի քանի դաշտեր նշանակուուծ են որպէս յայտանիշ, ապա դրանք միաւորում են մէկ յայտանիշի մէջ, որով համակարգում կարող է փնտում կատարուել: Տիպարի ստեղծման ժամանակ առնուազը մէկ դաշտ պէտք է նշել որպէս յայտանիշ: Երկրորդ հրամանով (»Դ) համակարգը տալիս է աւելի քանի քսան հարց ընթացիկ դաշտի պարունակութեան վերաբերեալ: Հարցելը հիմնականում միաբժիք պատասխան են պահանջում (այս կամ ոչ) և նկարագրում են զաշտիքի սնցման հերթականութիւնը, դաշտի տիպը, նրա լրացման, սուուզման կամ հաշուարկման եղանակը, դաշտերի լրացման եւ տուեալների արտածման ձեւաչափերը: Տիպարի բոլոր դաշտերին նշանակում են իրենց համարները եւ տուեալներին ներածման կամ խմբագրման ընթացքում ցուցիչը սնցնում է դաշտերով՝ բայց նշանակուած համարկալման: Յայտանիշային դաշտերին նոյնպէս համարներ են նշանակուում եւ համաձայն այդ համարների հերթականութեանը՝ նրանք միաւորում են մէկ միացեալ յայտանիշում:

Դաշտի պարունակութիւնը կարող է պատճէնաւորուել մէկ ուրիշ զրառումից կամ անզամ ուրիշ ինֆորմացիոն համակարգից: Թուային դաշտի պարունակութիւնը կարող է հաշուարկուել համապատասխան բանաձեռով, որի մէջ կարող են օգտագործուել ուրիշ դաշտերի պարունակութիւնները: Ինֆորմացիայի ներածման ժամանակ սիւներից խուսափելու համար կարող են տրուել դաշտերի լրացման յատուկ ձեւաչափեր, որոնցում նշանային դաշտի համար թուարկուում են բոլոր, տուեալ դիրքում թոյյատրուող նիշերը, իսկ թուային դաշտի համար՝ արժեքի բնդզերումը: Ներածուող ինֆորմացիայի համապատասխանութիւնը նկարագրութեանը հսկում է համակարգի կողմից անմիջապէս դաշտի պարունակութեան ներածումից յետոյ: Հնարաւորութիւն կայ նաեւ ներածումը կատարել առանց սուուզման, այնուհետեւ սուուզել ներածուած ինֆորմացիան ամբողջովին: Անհրաժեշտութեան դէպում «Բ հրամանի միջոցով կարելի է համակարգի ցանկացած հնարաւորութեան եւ հարցի վերաբերեալ սուանալ մանրամասն բացատրութիւն: Յատկանիշների նշանակման ժամանակ՝ »Ձ հրամանի միջոցով կարելի է վերադառնալ նախորդ հարցին: Որոշ հարցերի կարելի է չպատասխանել, այդ դէպում համակարգն ընտրում է ամենայարժ պատասխանը:

Նախադենք հեռախոսագրի օրինակի համար համապատասխան տիպար: Տուեալ դէպում գրառումը պարունակում է ազգանուն, անուն եւ հեռախոսահամար, որոնք էլ համակարգի դաշտերն են հանդիսանում: Քանի որ հեռախոսագրում գրառումները յարմար է վերադառնարել ըստ ազգանունների ապա որպէս յայտանիշ նշանակենք ազգանունի դաշտը: Միւս դաշտերը պարունակում են կամայական նշանա-

յին ինքորմացիա եւ նրանց յատկանիների նշանակումը կարող ենք յանձնարարել համակարգին, այսինքն՝ »Դ հրամանը կարող ենք չկատարել: Տիպարում զաշտերը տեղաւորենք առանձին տողերի վրայ եւ առնջն դաշտից առաջ գրենք նրա անուանումը (նկ. 16): Տիպարը յիշելու և տուեաների հետ աշխատանքի ռեժիմին անցնելու համար անհրաժեշտ է ներածել »Ա. հրամանը: Եթէ տիպարի ստեղծման ժամանակ ուսաներ են տեղի ունեցել, ապա համակարգը նրանց պատճառների մասին հաղորդագրութիւն է արտածում տեսատիպի վրայ եւ առաջարկում է օգտագործողին վերադառնալ եւ ուղղել սիսաները:

Տուեալների հետ աշխատանքի ռեժիմում տեսատիպի վրայ արտածում է զործողութիւնների ընտրման մենին (նկ. 17): Տուեալների հետ աշխատանքի ցանկացած զործողութեան մասին մանրմանն ինքորմացիա կարելի է սահանալ »Բ հրամանի միջոցով: Ակցում, երբ համակարգում դեռ գրառումներ չեն պարունակւում, մատչելի է միայն նոր գրառումների ներածման զործողութիւնը: Այդ գործողութեան կատարման ժամանակ մատչելի են տիպարի դաշտերով ցուցիչի տեղաշարժման, զաշտի պարունակութեան խմբագրման, տպող սարքի վրայ տիպարի պարունակութեան արտածման եւ տուեալների ներածման տարրափ հրամանները: Համակարգի մէջ որոշ քանակութեամբ գրառումների ներածումից յետոյ կարելի է անցնել նրանց հետ ուրիշ զործողութիւններ կատարելու: »Ե հրամանի միջոցով զործողութիւնների բնարկման մենին միջակութիւնների:

Ներածուած գրառումները կարելի է զիտարկել ներածման հերթականութեամբ »Ճ հրամանի միջոցով կամ յայտանիշին համաձայն՝ »Հ հրամանի միջոցով: Վերջին զկուում հեռախոսազբում գրառումները ներկայացւում են ազգանունների այրենական կարգով: Տուեալների զիտարկման ժամանակ արտածում է համապատասխան մենիւն (նկ. 18), որը հնարաւորութիւն է տալիս կոնչել տեսատիպի վրայ յանրդ կամ նախորդ գրառումը համապատասխանաբար՝ »Ն կամ »Ե հրամանների միջոցով: Տուեալների դիտարկման ընթացքում հնարաւոր է առանձին դաշտերի պարունակութիւնը ճշտել եւ փոփոխութիւնները հաղորդել համակարգին: Այդ զէպում հին գրառումները ոչնչացուածի նշանակում են ստանում, որից յետոյ ճշտուած ինքորմացիան պարունակութիւնը նոր գրառումը տեխլացւում է համակարգի մէջ: Ժամանակ առ ժամանակ անհրաժեշտութիւն է առաջանում համակարգը մարրել ոչնչացուած գրառումներից եւ յայտանիշին համաձայն՝ զերազասաւորելով միւս բոլոր գրառումները: Այդ նպատակին է ծառայում զործողութիւնների ընտրման մենիւն: »Ս հրամանը:

Եթէ անհրաժեշտ է արագ գտնել կոնկրետ արոնենի հեռախոսահամարը, կարելի է օգտակա զործողութիւնների ընտրման մենիւնը՝ »Գ հրամանով: Համակարգը կը քարցնի յայտանիշի արժեքը, այսինքն՝ արոնենի ազգանունը, որից յետոյ կ'ընտրի անհրաժեշտ գրառումը: Յայտանիշը չգտնելու զէպում համակարգը համապատասխան հաղորդագրութիւն է արտածում տեսատիպի վրայ: Գործնական աշխատանքում զատկոր է որոշակի դիմակով ինքորմացիայի փնտում կատարել: Օրինակ՝ եթէ յայտնի է միայն արոնենի անունը, ապա »Դ հրամանի միջոցով անհրաժեշտ է ներածել կամ խմբագրել դիտարկման դիմակը, որում միայն անունի դաշտում պէտք է նշուի կոնկրետ անունը, իսկ միւս բոլոր դաշտերում՝ աստղանիշեր: Այդ զէպում դիտորդիման ընթացքում կ'արտածուեն միայն այն գրառումները, որոնցում համապատասխան դաշտում գրանցուած է նշուած անունը:

Ազգանուն	—
Անուն	—
Հետախոսահամար	—

Նկ. 16

ԳՈՐԾՈՂՈՒԹԻՒՆՆ ԸՆՏՐՄԱՆ ՄԵՆԻՒՆ

»Ն նոր գրառումների ներածում
»Բ նայլային փարերի ընտրութիւն
»Գ Դրաման յայտանիշով փնտուում
»Ը Տուեալների փայլի դիտարկում
»Դ Դիմակի խմբագրում

»Բ Բացարութիւն
»Փ Փարերի դիտարկում
»Յ Յայտանիշով դիտարկում
»Ս Սպասարկում
»Ե Ելք տիպարից

Նկ. 17

ՏՈՒԵԱԼՆԵՐԻ ԴԻՏԱՐԿՄԱՆ ՄԵՆԻՒՆ

— Ցուցիչի տեղաշարժ —
Դաշտ »Ա ճախ» »Ճ աջ»
Սիւն »Բ ճախ» »Շ աջ»
Դաշտ »Զ առաջին» »Ժ վերջին»

— Խմբագրում —
»Հ հեռացում»
»Ս Աւելացում»
»Վ Վերականգնում»
»Ղ Յաջորդ գրառում»
»Է նախորդ գրառում»

— Տպում —
»Ճ Տուեալների»
»Զ Ձեւի»
— Աւարտ —
»Ն ներածման»
»Ռ Ռեժիմի»

Նկ. 18

2. 7. Գործակարման ամրողացուած համակարգեր

Անհատական քոմիտերների վրայ ծրագրային համակարգերի շահագործման փորձը ցոյց տուեց, որ օգտագործողների մեծ մասը մշտական օգտառում են մի քանի համակարգերից եւ մէկից միւսին ժամանակաւոր անցումը զգալի դժուարութիւնների հետ է կապուած։ Օգտագործողների պահանջները հնարաւորին չափ լիովին բաւարեկուու նպատակով ստեղծւում են այսպէս կոչուած ամբողջացուած ծրագրային համակարգեր, որոնք պարունակում են մի շարք տարբեր գործողութիւններ կատարող եւ միմեանց հետ սերտօրէն կապակցուած ծրագրեր կամ ենթածրագրեր։ Այդպիսի համակարգերի հետաքրքիր օրինակ են Հանդիսանում այսպէս կոչուած էլեկտրոնային քարտուղարները, որոնք լայն ճանաչում են ստացել գործակարման աշխատանքներ կատարող տարբեր մասնագիտութեան տէր օգտագործողների մօտ։ Նման համակարգի կորիզը յաճախ իրենից ներկայացնում է մի ուեղիզենտ ծրագիր, որ թողարկումից յետոյ մշտական մնում են քոմիտերի յիշովութեան մէջ՝ չխանգարելով միւս ծրագրերի կատարումը։ Դա հանրաւորութիւնն է ասլիս ցանկացած պահին արագործէն կանչել համակարգը նրա հետ որեւէ գործողութիւն կատարելու նպատակով անկախ նրանից գտնուում է արդեօք տուեալ պահին քոմիտերի յիշովութեան մէջ մէկ ուրիշ ծրագիր, թէ՝ ոչ։ Այս հանդամանքը չափազանց յարմար է, փաստորէն էլեկտրոնային քարտուղարը աննկատ միշտ գտնուում է քոմիտերում, րայց ակտիւացուում է միայն անհարժեշտ ժամանակ։ Էլեկտրոնային քարտուղարը ակտիւացուած է երկու որոշակի ստեղների, օրինակ՝ ղեկավարման եւ լրացման սահեղների միաժամանակ սեղմման միջոցով։ Այդ դէպքում տեսատիպի էկրանի վրայ արտածւում է համակարգի զիմանիք մենիւն (նկ. 19)։ Ընդ որում՝ մենիւն վերադրում է մինչ այդ էկրանին եղած պատկերի վրայ՝ չխախտելով նրա մենիւյի տարածքից դուրս գտնուող մասը։ Գիշաւոր մենիւյից սեղմելով ստեղնաշարի համապատասխան ֆունկցիոնալ ստեղնը՝ (F1 - F5) կարելի է յուշման կարգով համակարգի մասին մանրամասն ինֆորմացիա ստանալ (F1) եւ ակտիւացնել տեքստերի վերամշակման (F2), հեռախոսազրքի (F3), օրացոյցի (F4) կամ հաջուկի (F5) ծրագրերը։ Եթէ համակարգը ակտիւացուել էր մէկ ուրիշ ծրագրի կատարման ժամանակ, ապա վերջինս ընդհանուում է՝ փիքսելով տեսատիպի վրայ արտածուող ինֆորմացիան։ Ընդհատուած ծրագրի կատարումը վերսկւում է համակարգից բանալու ստեղնը սեղմելու միջոցով դուրս գտնուց։

ԳԼԽԱԼԻՐ ՄԵՆԻ
F1 Յուշում
F2 Տեխն
F4 Հեռախոս
F5 Հաշուկի
ԲՈՒՆԱԼԻ ԵԼՔ

նկ. 19

Տեքստերի վերամշակման ծրագիրը իր ֆունկցիոնալ հնարաւորութիւններով հիմնականում համարժէք է զիտարկուած տեքստերի վերամշակման համակարգին։ Սկզբունքային տարբերութիւնը կայանում է նրանում, որ առևիալ եւ համակարգի կողմից ընդհատուած ծրագրերի միջեւ ինֆորմացիայի փոխանցում կատարելու հնարաւորութիւնն է ընձեռուում։ Տուեալ ծրագրից ինֆորմացիան ընդհատուած ծրագրին փոխանցելու համար անհրաժեշտ է պահանջուող ինֆորմացիան նշել որպէս բրոկ եւ յատուկ հրամանի միջոցով վերագրել այն որեւէ ստեղնի։ Այսուհետեւ ընդհատուած ծրագրի աշխատանքը վերականգնելուց յետոյ համապատասխան պահին պէտք է սեղմել այդ ստեղնի ինֆորմացիան հաղորդելու համար։ Ինֆորմացիան մնում է ստեղնին վերագրուած՝ մինչեւ յատուկ հրամանով վերագրման վիմացումը։ Հակառակ ուղղութեամբ փոխանցում կատարելու համար անհրաժեշտ է համապատասխան հրամանի միջոցով վերականգնել տեսատիպի վրայ ընդհատուած ծրագրի կողմից արտածուած պատկերը, որի վրայ նշել պահանջուող ինֆորմացիան որպէս բրոկ եւ պատճենաւորման հրամանի միջոցով տեղադրել նշուած բլոկով իմբազրուուղ ափստի համապատասխան մասում։ Այս հնարաւորութիւնը բաւականից յարմարեցնում է արագացնում է խմբագրուուղ տեքստի մէջ որեւէ ծրագրի աշխատանքը բլոկուուղ տապահուող ինֆորմացիայի ընդդրկումը։ Բացի դրանից՝ ծրագրիը հնարաւորութիւնն է տալիս տեքստի վերանցել մէկ հրամանի միջոցով։

Հեռախոսագիրը իրենից ներկայացնում է մի պարզ մասնագիտացուած ինֆորմացիոն համակարգ։ Այս հնարաւորութիւնն է տալիս ոչ միայն զանիկ պահանջուող արունենակի հեռախոսագիրը, այլև նրա հետ հեռախոսակապ հաստատել, եթէ, ի հարկէ, քոմիտիթերը համապատասխան սարքաւորման միջոցով միացուած է հեռախոսագիրն։ Ինֆորմացիան յիշուում է մագնիսական սկաւառակների վրայ սովորական ֆայլերի տեսքով։ Ընդ որում այդ ֆայլերի ստեղծումը եւ խմբազրուումը կատարուում է տեքստերի վերամշակման ծրագրի միջոցով։ Համակարգի միջոցով հեռախոսակապ ապահովելու համար անհրաժեշտ է որպէս զիմանիք ֆայլի առանձին տողի վրայ տեղաւորուած լինի միայն մէկ հեռախոսակամար եւ այն պարունակի գոնչ մի փակագիծ կամ գծիկ (-)։ Բացի դրանից՝ որոնման զործողութիւնը ճիշտ կադմակիրպելու համար ցանկալի է, որ տեղի սկզբից զրուած լինի արոնենակի պահանունը կամ աւելի ճիշտ համապատասխան յայտանիչը։

Հեռախոսագրքի ընտրութիւնից յետոյ տեսատիպի վրայ համապատասխան պատճենանում արտածուում է նշուած ֆայլի մի մասը (նկ. 20)։ Ֆայլի միւս մասերը դիտարկելու համար անհրաժեշտ է ցուցիչի տեղաշարժի ստեղնիով չարժեկ պատու հանը ֆայլի վրայով։ Որեւէ ուրիշ ֆայլը ընտրելու համար պէտք է սեղմել F3 ստեղնը եւ այնուհետեւ ներածել նոր ֆայլի անուանումը։ Համակարգը հնարաւորութիւնն է տալիս ինֆորմացիայի որոնում կազմակերպել՝ նշելով յայտանիչը կամ նրա սկզբանը F5 ստեղնի միջոցով։ Ցանկացած դէպքում որոնման զործողութիւնը սկսուում է ինֆորմացիայի համապատասխան հատուածի ներածումից յետոյ եւ աւարտուում է այդ հատուածը պարունակող տողը տոնը գտնելուց յետոյ։ Որոնումը այն կամ այն ուղղութեամբ շարունակելու համար կարելի է օգտուել ցուցիչը համապատասխան ուղղութեամբ տեղաշարժող ստեղնից։ Պահանջուող հեռախոսակապ սլատուհանի միջին տողի վրայ տեղաւորելուց յետոյ հեռախոսակապ համար պարզապէս պէտք է սեղմել ներածման ստեղնը։

ՀԵՌԱԽՈՍԱԳԻՐՔ

Տիգրանեան Դայանէ Արխտակէսեան Մուշեղ	Խմբագիր տնօրէն	21 - 36 - 49 54 - 77 - 77
Պետրոսեան Վահան	Փոխտնօրէն	54 - 74 - 47
Բագրատեան Կիրակոս Համբագապեան Անահիտ	տնտեսագիտ բաժնի վարիչ	73 - 21 - 11 66 - 09 - 56

Նկ. 20

Օրացոյցի ծրագիրը արտածում է աեսատիպի վրայ բնթացիկ ամսուայ օրացոյցի աղիւսակ (նկ. 21), ըստ որում ցուցիչի Հորիզոնական տեղաշարժի սաեզներով կարիի և փոփոխել արտածուող տմիսը, խակ ուղղահայեցով՝ սարին։ Օրացոյցի հետ կապուած է յատուկ գործարական օրագրի ենթահամակարգը, որին անցներս համար անհրաժեշտ է ներածել դահնինջառ սամափիւր, եւ կ'արտածւի համուպասասախոն օրագիրը (նկ. 22)։ Օրագիրը կարելի է թերթել, այսինքն՝ փոփոխել ամսումիւր ցուցիչի հորիզոնական տեղաշարժի սաեզներով։ Ուղղահայեց տեղաշարժի սակաները հնարաւորութիւն են տափս օրագրի սուղերով շարժուել համապատասխան գրանցումներ կատարելու նպաստիպով։ Օրագրիրը դրանցում են մաղնիսական սկառակաների վրայ յետաղայ օգտագործման համար։ Համակարգը սրոշակի նշուած ժամանակահատուածի համար օրագրիրը սպոզ սարքի վրայ արտածելու հնարաւորութիւնն է տալիս։

Հաշուիչի ծրագիրը նախատեսուած այն օգտագործողների համար, որոնք յաճախ ող չառ րարդ մաթեմատիկական հաշուարկներ են կոստարում եւ սովոր են պարզապայն հաշուիչով օգտաւել։ Այս ծրագրի ընարութիւնից յետոյ աեսատիպի վրայ արտածում է հաշուիչի (նկ. 23) պատկերը, որի վրայ նշուած են կիրաւուու սաեզները։ Հայուիչը հնարաւորութիւն է տայնու հաշուարկներ կատարել ոչ միայն տասական (10⁶), այլ նաև երկուական (Bis) եւ տասնեցսական (Hex) համակարդերում։ Վերջին դիպում F5 - F10 սուղները օգտագործուում են համապատասխարար տասնտեսական A, B, C, D, E, եւ F թուանշաները ներածելու համար։ Չորս օպարգոյն թուառանակոն գործողութիւններ կատարելուց բացի՝ կարելի է նաև կատարել երեք արտարարական գործողութիւն՝ ԵԽ (And), ԿԱՄ (Or) եւ ՈՉ-ԿԱՄ (Xor)։ Մրագրի կարեւորակոյն տասնձնայատկութիւնը կայանում է նրանում, որ Հաշուարկների արդիւնքը յատուկ համանի միջոցով կարելի է վերագրել սուեզնաշարի որեւէ տուղին եւ այսուհետեւ այն օգտագործել մէկ ուրիշ գործողութիւն կատարելու իսկ արքի ծրագրաւմ։ Ինչպէս եւ տեքստերի վերամշակման ծրագրի դէպում վերադրումը պահպանուած է մինչեւ յատուկ համանով վերացնելը։

Այսպիսուած՝ զիտարկուած դործագարան համակարգը մի շարք կարեւոր եւ հետաքրիւ հնարաւորութիւններ է բնդպրիում եւ կարող է լայնօրէն օգտագործուել

Յունուար				22	1990	
Կիր	Երկ	Երր	Զոր	Հիւն	Ուր	Զար
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Նկ. 21

Յունուար			22	1990
ՖԱՅԼ A: DIARY. PRI			22	1990
Յունուար	22	1990		
3				
	Անուանում			
	08:00			
	08:30	Խորհրդակցութիւն		
	09:00			
	09:30	Դասախոսութիւն ՀՏՀ		
2	10:00			
	10:30			
	11:00	Դասախոսութիւն Մրագրաւոր.		
	11:30			
	12:00			
	12:30			
	13:00	Բանավէն Ա-Բ հեռամիարները		
	13:30			
	14:00			

Նկ. 22

4726. 0283			
Dec			
S. Վեց	Ցիշող.	Թիւ/նիշ	
	R C	= 7 8 9 -	
A B	M + -		
F5 F6	* /	/ 4 5 6	
C D	-Համակարգ		
F7 F8	Dec	* 1 2 3 +	
E F	Bin		
F9 F10	Hex	O	
Տրամադրանական		ՉՆՉՈՒՄ	
And Or Xor		C CE ←	

Նկ. 23

քոմիտեներով կահաւորուած տարրեր մասնագէտների աշխատատեղերում՝ զործակարման աշխատանքները արագացնելու և աւելի արդիւնաւէտ դարձնելու նպաստակով:

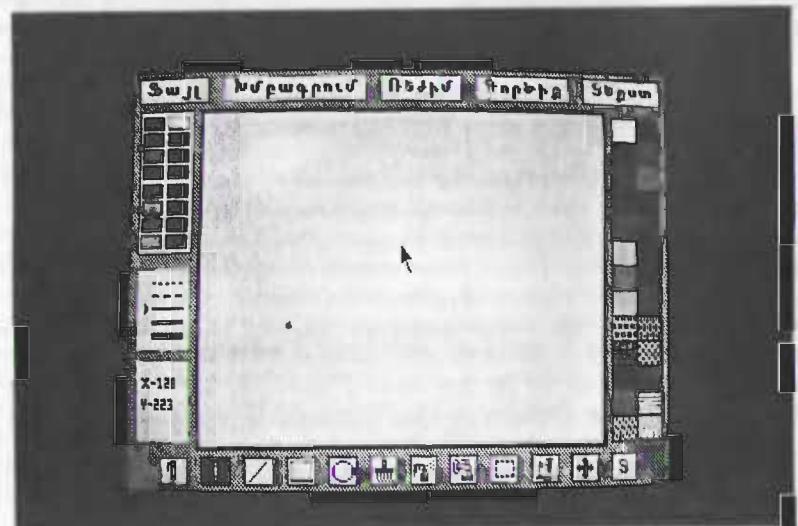
2. 8. Պատկերների վերամշակման համակարգեր

Դիտարկուած բոլոր ծրագրային համակարգերը իրենցից ներկայացնում էին նշանային ինֆորմացիայի վերամշակման միջոցներ: Նրանց յարմարաւէտութիւնը ապահովելու համար պահանջուում էր տեսատիպի միայն մէկ կարեւոր հնարաւորութիւն՝ էլեկտրոնի վրայով ցուցիչի տեղաշարժիքի ծրագրային դեկավարման հնարաւորութիւնը, այսինքն կամացական յաջորդականութեամբ տեսատիպի վրայ ինֆորմացիայի արտածման և նրա առանձին մասերի փոփոխման հնարաւորութիւնը: Ժամանակակից անհատական քոմիտեները օժտուած են ինֆորմացիայի արտապատկերման և վերամշակման աւելի լայն հնարաւորութիւններով: Դրանցից կարեւորագրյանը գունաւոր զրաֆիկական պատկերների արտապատկերման և վերամշակման հնարաւորութիւնն է, որը լայնօրէն կիրառուում է գործարարական ինֆորմացիայի վերամշակման մէջ զրաֆիկներ և գծագրեր կառուցելու համար, ուսուցանող համակարգերում որպէս բարդ գործնիմեցների և երեւոյթների ներկայացման միջոց, աւտոմատ նախագծման համարգերում համապատասխան պահանջներին բաւարարող գծագրեր ստունալու և նախագծողի առարկայի պատկերաւոր ներկայացումը ապահովելու համար եւ այլն:

Պատկերների մեքենայական վերամշակումը ակտիվօրէն մուտք է զործում ոչ միայն զործարար կեանքի մէջ, այլեւ այն ասպարէզները, որոնք ընդունուած էին

պատկերացնել որպէս բացառապէս ստեղծագործական: Արդէն վաղուց քոմիտերի օգնութեան են դիմում մուլտիպլիկացիոն եւ ֆանտաստիկ ֆիլմերի ստեղծման ժամանակ: Վերջին տարիներին ըուսն բանվէճերի առարկայ դարձաւ այսպէս հոչուած քոմիտերային գեղանկարչութիւնը: Ոմանք դա գրեթէ սրբապղծութիւն են համարում, ասելով, որ այդ երեւոյթը գեղանկարչութիւնը իբրեւ բացառիկ ստեղծագործական ոլորտ ներկայացնելու աւանդական գաղափարն է վերանայում, իսկ մէկ ուրիշները այն հոչակում են քոմիտերային դարաշրջանի միակ արուեստ, որը լիովին արտայայտում է մեր ժամանակակիցի պատկերացումները եւ որտամադրութիւնները:

Պատկերների վերամշակման համակարգի թողարկումից յիսոյ տիսաափակի վրայ արտապատկերում է համակարգի աշխատանքային ռեժիմը ներկայացնող նկարը, որի օրինակը ներկայացուած է նկ. 24ում: Նկարի կենտրոնում պարփ ձեռով ներկայացուած է գրաֆիկական ցուցիչը, որը կարող է տեղաշարժուել ցուցիչի տեղաշարժի ստեղների կամ յատուկ սարքի (օրինակ մէնինի) միջոցով: Նկարի ձախ ստորին անկիւնի ուղղանկեան մէջ արտածում են ցուցիչի ընթացիկ գիրքի կորոդինատները՝ համապատասխանաբար հորիզոնական եւ ուղղահայիս ուղղութիւններով: Նկարի վերին ստորի վրայ թուարկուած են համակարգի հինգ հրմանական մէնիւնները, որոնցից ամէն մէկը պարունակում է մի շարք զործողութիւններ կամ ռեժիմներ (նկ. 25): Որեւէ մէնիւյի զիմենու համար անհրաժեշտ է ցուցիչը տեղապարել համապատասխան անուանման վրայ եւ սեղմի թողարկման ստեղնը՝ օրինակ + ստեղնը, որից յիսոյ մէնիւյի անուանման տակ կ'արտածուի նրա պարունակութիւնը: Մէնիւյի հնարաւորութիւններից մէկը ընտրելու համար անհրաժեշտ է ցուցիչը տեղապարել նրա վրայ եւ սեղմել թողարկման ստեղնը, իսկ մէնիւյից զուրս զարու համար պէտք է սեղմել զալարեցման ստեղնը՝ օրինակ - ստեղնը:



Նկ. 24

Ֆայլ	Խմբագրում	Ռեժիմ	Գործիք	Տեքստ
Ընթացում	Շրջում	Տեղաշարժ	Վրձինի ձեւը	Ցուցակ
Դրամցում	Հորիզ. հայելի	Պատմէն	Փոխանշատում	Տպատառ
Հեռացում	Ուղղ. հայելի	Թափանցիկ	Ստուեր	
Արտածում	Պոյոյտ	Ցանց	Գրաֆիկ	
Աւարտ	Մասշտար	Խաչ	Գրադարան	
	Վերականգնում			

Նկ. 25

ԶԵՐԴ ՆԿԱՐԻ ԿԻՆՈՒՐՈՆԱԿԱՆ ՄԱՏՈՐ ՀԱՄԱԼԱՐԳԻ ԱՊԱՏՈՒՀԱՆՆ է, որի վրայ արտածում է վերամշակուող պատակերի մի մասը, իսկ ստորին մասի փոքրիկ վանդակիներում ներկայացուած են պատակերի վերամշակման Համար օգտագործուող տասներկու հնարաւոր միջոցները։ Դրանցից մէկը ընարելու Համար պէտք է ցուցիչը տեղադրի Համալարատախան վանդակի մէջ եւ սեղմել թողարկման ստեղնը։ Այնուհեաւ բնարուած միջոցը զարծածելու Համար պէտք է տեղափոխել ցուցիչը պատուհանի տարածքի պահանջուող մասը եւ սեղմել թողարկման ստեղնը, իսկ զործողութիւնը գաղարեցնելու Համար՝ գաղարեցնել ստեղնը։ Դիտարկենք այդ միջոցները հերթականորէն՝ ճամփորդ:

Առաջին վանդակի ընարութիւնը հնարաւորութիւնն է տալիս կամայական գծեր անցկացնել՝ ինչպէս մատախով նկարելիս։ Ընդ որում՝ թողարկման ստեղնը մէկ անդամ սեղմելուց յետոյ ցուցիչը ցանկացած տեղաշարժ իր հետքն է թուղում պատուհանում, այսինքն՝ Համալարատախան զիժ է նկարում։ Բնական է, որ այդ զիտրում ցուցիչը հնարաւոր չէ հանել պատուհանի տարածքից զորա։ Երկրորդ անգամ թողարկման ստեղնը սեղմելուց յետոյ նկարելը դադարեցնում է, այսինքն՝ ցուցիչը տեղաշարժի ընթացքում այլեւս հետք չի թողում։ Վերջին զործողութեան մամանակ նկարած գծի մի մասը ջնջելու Համար պէտք է սեղմել դադարեցման ստեղնը և անցկացնել ցուցիչը գծի Համալարատախան մասի վրայով, իսկ ջնջումը դադարեցնելու Համար պէտք է մէկ անդամ եւս սեղմել դադարեցման ստեղնը։ Եթէ դիմը անցկացուած էր ուրիշ գծերի վրայով, ապա տուեալ զէպքում կը վերականգնի նախորդ պատակերը, քանի որ կը չնշուի միայն անցկացուած զիժը։

Երկրորդ վանդակի ընարութիւնը հնարաւորութիւնն է տալիս ռեախինի նման ջնջել պատակերի մի մասը։ Թողարկման ստեղնի առաջին սեղմումից յետոյ սկսւում, իսկ երկրորդ սեղմումից յետոյ զադարեցում է ջնջում։ Դադարեցման ստեղնը հնարաւորութիւնն է տալիս վերականգնել նոր ջնջուած պատկերի պահանջուող մասը։

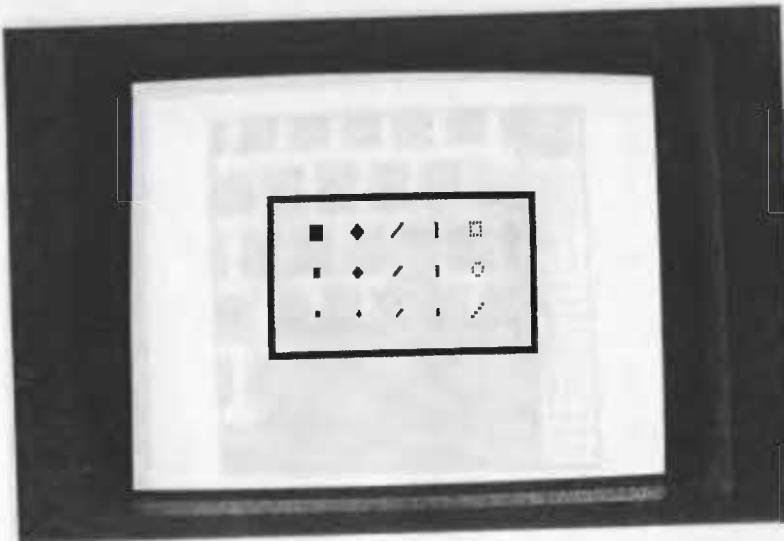
Երրորդ, չորրորդ եւ հինգերորդ վանդակները նախատեսնուած են Համապատասխանաբար ուղղող գիծ, ուղղանկիւն եւ շրջանագիծ կամ էլիպս նկարելու Համար։ Բոլոր զէպքերում կառուցում կատարում է երկու կէտով, որոնցից առաջինը փիրուում է թողարկման, իսկ երկրորդը զադարեցման ստեղների միջոցով։ Ուղիղ զիժ զէպքում այդ կէտերը ցոյց են տալիս հաստածի սկիզբը եւ վերջը, ուղղանկիւն զէպքում՝ նրա հակագիր անկիւնները, շրջանագիծ կամ էլիպսի զէպքում առաջին

կէտը ցոյց է տալիս նրանց գիրգիկական կենորունը իսկ երկրորդ կէտը գտննում է Համապատասխան զիժի վրայ։ Նկարելիս զծի տեսքը կամ հաստութիւնը կարելի է հարարել նկարի ձախ կենարունական մասում զտնուող ուղղանկեան միջից, որի հարարութիւնը մարդ ցուցիչը պէտք է տեղուղբեկ Համապատասխան զիժի վրայ և սեղմել թողարկման ստեղնը։ Ընտրուած զծի զիմաց կէտ է դրուում։

Վեցերորդ վանդակը, ըստ իր օգտագործման սկզբունքի, համընկնում է առաջին վանդակի հետ։ Տարրերութիւնը կայանում է նրանում, որ տուեալ զէպքում մատիսի փոխարէն զործ ունենք վրձինի հետ։ Վերջինիս ձեւը և չափոր բնորուում է 26րդ նկարութիւնը սեղկայացուած ապրերկաներից, որոնք արածւում են զործիքների մինիյի համապատասխան հնարաւորութիւնը բնարելուց յետոյ։

Եօթներորդ վանդակը նոյնպէս, ըստ իր օգտագործման սկզբունքի, համընկնում է առաջին վանդակի հետ։ Տարրերութիւնը կայանում է նրանում, որ տուեալ զէպքում մատիսի փոխարէն զործ ունենք վուշի հետ։ Վերջինիս որում փոշեցրման խոռութիւնը տուեալ զէպքում կախուած է ցուցիչի տեղաշարժման արագութիւնից։

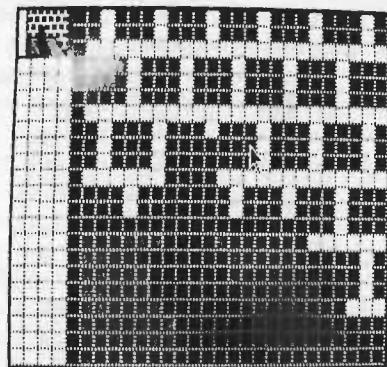
Ութերորդ վանդակը հնարաւորութիւնն է տալիս նկարի սպարփակուած ալիքովները լցոնել որեւէ զոյնով կամ նախով։ Տիրոյթը լցուում է սահմանադից սկսած մինչեւ կենարուն և այն ոչ լրիւ լցոնելու համար անհրաժեշտ է հսմալարատախան սրահին կենարուն և այս լրիւ լցոնելու համար անհրաժեշտ է սեղմել վազալիքման ստեղնը։ Վեցերորդ, եօթներորդ եւ ութերորդ վանդակների միջին զիժուում է նկարի տջ կազմից բնարուուղ զոյնը միջոցով նկարելու ժամանակ օգտագործւում է նկարի տջ կազմից բնարուուղ զոյնը մատիսի փոխարէն ժամանակ կամ նախուցը։ Ընտրման համար անհրաժեշտ է տեղափոխել զադարեցման ստեղնը և անդամ նախուցը։ Ընտրման համար անհրաժեշտ է սեղմել դադարեցման ստեղնը։ Վերջինի վանդակի վրայ և սեղմել թողարկման ստեղնը։ Ընդ որում՝ զոյների եւ նախչերի վանդակի վրայ և սեղմել թողարկման ստեղնը։ Վերջինի վանդակի վրայ և սեղմել թողարկման ստեղնը։ Վերջինի վանդակի վրայ և սեղմել թողարկման ստեղնը։



Պատկերի որոշակի ուղղանկիւն մասի յատկացման համար օգտագործում է ին-ներորդ վանդակը։ Ընդ որում յատկացումը կատարում է ուղղանկեան երկու հա-կաղիր անկիւնները համապատասխանաբար թողարկման և գաղարեցման ստեղների միջոցով։ Ամրագրման մենիւյի համապատասխան հնարաւորութիւններից օգտուելով՝ յատկացուած մասը կարելի է շրջել, այսինքն՝ բոլոր զոյները փոխարինել նրանց հակաղիր գոյներով։ Հորիզոնական կամ ուղղահայեաց հայելային տրտապատկերում կատարել, պատել կենտրոնի շուրջը կամ մասշտաբաւորել, այսինքն՝ մեծացնել կամ փորացնել հորիզոնական և ուղղահայեաց ուղղով թիւններով։ Բացի դրանից, ռե-ժիմների մենիւյի համապատասխան հնարաւորութիւններից օգտուելով, յատկաց-ուած մասը կարելի է տեղաշարժել կամ պատճէնաւորել պատկերի մէկ այլ մասում։ Նոյն մենիւյի թափանցիկ ռեժիմը հնարաւորութիւն է տալիս տեղաշարժը կամ պատ-ճէնաւորումը կատարել պատկերի ֆոնը թափանցիկ ներկայացնելով։ Այդ դէպքում տեղաշարժից կամ պատճէնաւորումից յետոյ նոր պատկերը վերադրում է հնի վրայ։

Տասներորդ վանդակի ընտրութիւնը հնարաւորութիւն է տալիս պատկերի որոշակի մասը խոչրացնել եւ ճշտել այն կէտ առ կէտ։ Այդ դէպքում ամբողջ պատուհանը զբաղեցնում է խոչրացուած պատկերի հասուածը։ միայն ձափ վերին անկիւնում ներկայացնում է նրա տեսքը բնական չափով (նկ. 27)։ Վերջինիս վրայ անմիջապէս արտապատկերում են խոչրացուած պատկերի վրայ կատարուած փոփոխութիւնները։

Վերամշակուող պատկերը կարող է համակարգի պատուհանից մէծ լինել։ Այդ դէպքում, օգտուելով տասնմէկերորդ վանդակից կարելի է պատուհանը տեղաշար-ժել պատկերի վրայով՝ նրա պահանջուոր մասը տեսանելի դարձնելու նպատակով։ Տուեալ վանդակը ընտրելուց յետոյ անհրաժեշտ է ցուցիչը տեղադրել պատուհանի տարածքում, սեղմել թողարկման ստեղնը եւ տեղաշարժել ցուցիչը համատապասխան ուղղութիւնուր։ Երբ պատուհանում կ'արտապատկերուի պատկերի պահանջուոր մասը, պէտք է ֆիքսել այն՝ դաշտարեցման ստեղնը սեղմելու միջոցով։



Վերջապէս, տասներկուերորդ վանդակը հնարաւորութիւն է տալիս պատկերի վրայ տարբեր տեսքի եւ չափսի տպատառերով դրառումներ կատարել։ Համապա-տասխան փոփոխութիւնները կատարուում են տեքստիկի մենիւյի հնարաւորութիւն-ների միջոցով։ Տեքստի ներածման ընթացքում կարելի խմբագրել այն՝ աեքստերի վերամշակման համակարգի պարզագոյն հնարաւորութիւնների նման միջոցներով։ Առաջին, երրորդ, չորրորդ, հինգերորդ եւ տասներկուերորդ վանդակների միջո-ցով նկարիլու ժամանակ օգտագործում է նկարի ձափ վերին մասի ուղղանկիւնից ընտրուու գոյնը։ Ընտրման համար անհրաժեշտ է տեղադրել ցուցիչը համապատաս-խան վանդակի վրայ եւ սեղմել թողարկման ստեղնը։ Ընդ որում՝ զոյնիրի քանա-կութիւնը կախուած է օգտագործուող տեսատիպի տեսակից։ Ընտրուած զոյնի վան-դակի ասհմանը պատկերում է կէտագծով։

Համակարգում պատկերները յիշուու են մագնիսական ոկաւառակների վրայ ստանդարտ ֆայլերի ձեւով։ Ֆայլերի մենիւն հնարաւորութիւն է տալիս պատկեր-ների ֆայլերը ընթերցել գրանցել եւ հեռացնել մագնիսական ոկաւառակի վրայից։ Բացի դրանից՝ այս մենիւն պարունակում է վերամշակուող պատկերը տպող սարքի վրայ արտածելու հնարաւորութիւն։

Խմբագրման մենիւն, բացի վերը նշուած հնարաւորութիւններից, պարունակում է վերականգնման գործողութիւնն, որի միջոցով կարելի է հրաժարուել վերջին կա-տարուած գործողութեան արգիւնից, այսինքն՝ վերականգնել պատկերի նախկին տեսքը։ Վերականգնման գործողութիւնն տարածուում է վերջին ընտրուած վանդակի միջոցով՝ կատարուած բոլոր փոփոխութիւնների վրայ։

Ռեժիմների մենիւն պարունակում է երկու ծառայուղական ոեժմի, որոնք կոչուած են աւելի դիւրին զարձնել պատկերների վերամշակուում։ Դրանցից առաջնը համա-կարգի պատուհանը ծածկում է հորիզոնական և ուղղահայեաց բարակ զերերի ցան-ցով, որը օգնուած է համաշափ, ճիշտ հրկաչափական կառուցումներ կատարել։ Այս միջոցը համարժէք է աւանդական գծագրութեան մէջ միջմնարամիջիլի օգտագործ-մանը։ Զարկ է նշել, որ այդ ցանը անհրաժեշտութեան դէպքում է՝ առանց պատկերի վրայ որեւէ հետք թողներու։ Երկրորդ ոեժմը փոխուում է գրաֆի-կական ցուցիչի տեսքը գորբիկ պաքի փոխարէն արտածելով տմբուց կըրանի վրայ տարածուու իսչուածքով, որը հեշտացնուում է պատկերի տարբեր մասերի դիրքերի համեմատումը։

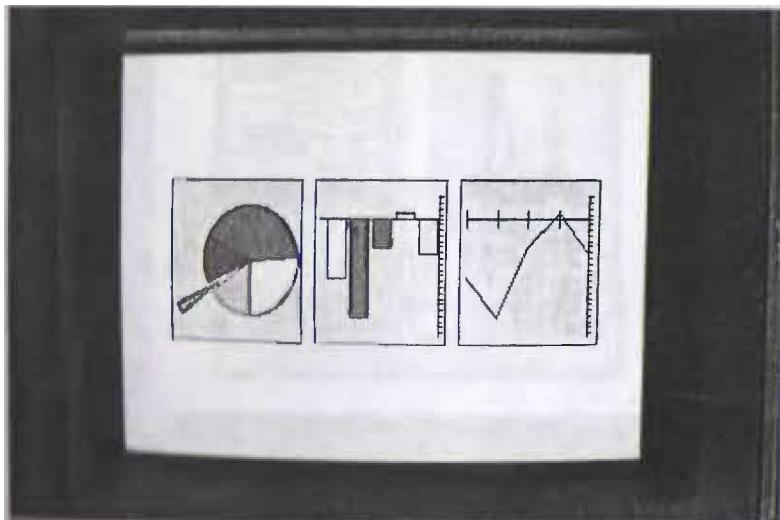
Համակարգը միաժամանակ քոմփիւթերի յիշողութեան մէջ կարող է պահել երկու վերամշակուող պատկեր, որոնցից ցանկացած պահին միայն մէկը կարող է ակտիւ լինել եւ արտածուել տեսատիպի վրայ, իսկ միւսը այդ պարագայուում պահեստային դերում է։ Գործիքների մենիւյի մէջ պատկերների փոխանչատման միջոց է պա-րունակուում։ Փոխանչատումը հնարաւորութիւն է բնձենուում յատկացյած մասերը տեղաշարժի կամ պատճէնաւորման ընթացքում փոխանցել մէկ պատկերից մրւսին։ Այս մենիւյի յաջորդ միջոցը հնարաւորութիւն է տալիս կառուցուող երկրաչափական պատկերները աւտոմատ կերպով օժտել համապատասխան ստուերներով։

Պատկերների վերամշակման համակարգի կարեւորագոյն հնարաւորութիւններից մէկը գրաֆիկների կառուցումն է տուեաների հիման վրայ, որը նախատեսուած է ոեժմների մենիւյի մէջ։ Այս ոեժմի ընտրման դէպքում տեսատիպի վրայ արտած-ուում է համապատասխան նկարը (նկ. 28), որի վրայ ձափ աղիւսակում ներածւում են տուեալները, իսկ ստորին աղ աղիւսակում՝ գրաֆիկի յատկանիշները։ Այնուհե-

Տվյալներ	
1	1
2	-3.45
3	13
4	43
5	26
6	
7	2.5 քայլ
8	-18 ստորին սահման
9	50 վերին սահման
10	

Հատկանիցներ

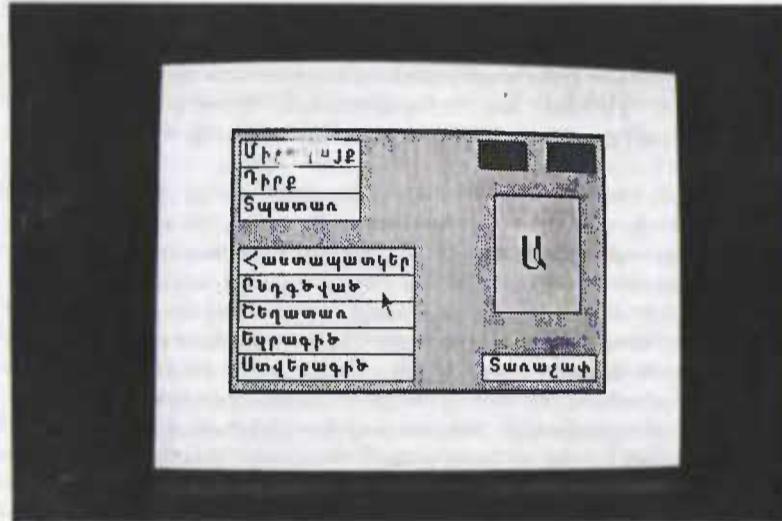
և նկարի միջին մասում պէտք է բնարութի զրաֆիկի տիպը՝ ցուցիչը աեղաշարժը ռելյու վանդակի և սեղմել թափարկման սահմանը։ Օրինակի համար նկ. 29ում ներկայացրուած են նախարդ նկարի տառապներով կառաւցուած բոլոր տիպի զրաֆիկները։



Գործիքների միջին մերջին միջոցը, որի բնարում արտահուում է նկ. 30ում ներկայացրուած նկարը, հնարաս որութիւն է տալիս սկաւառակների վրայ դրանցուած պատկերների զրագրանից ընտրել եւ օգտագործուել անհրաժեշտ ստունդարտ դրաֆիկական պատկերները։ Դրա համար զիտարկման միջոցով պէտք է գտնել համապատասխան պատկերը, ցուցիչը տեղաշարժել «Այս» վանդակը եւ սեղմել թուղարկման սահմանը։

Տեքստերի մենիսյի համապատասխան հնարաւորութեան բնարութիւնից յետոյ տեսամիպի վրայ արտածուում է զրագոււներ կատարելիս օգտագործուող տպաստոփ աեսքը և չափով ցոյց տուուլ յատկանիշների ցուցակը պարունակող նկարը (նկ. 1)։ Այս նկարի աջ ստորին վանդակի միջոցով կարելի է ընտրել տպաստոփի չափով իսկ, ձափի ստորին մասի ուղղանկելիւնից՝ տպաստոփի տեսքը, ՅԶրդ նկարում ներկայացուած են վերջինիս հնարաւոր ստորբերակները։ Նկարից հրեւում է, որ միաժամանակ կարող են ընտրել մի քանի յատկանիշներ։ Ընտրաթիւնը կատարելու համար անհրաժեշտ է ցուցիչը տեղադրել համապատասխան յատկանիշի վրայ եւ սեղմել թուղարկման ստեղնը։ Բայց զրանից՝ հնարաւորութիւն է ընճենուում ձափի վերին մասի ուղղանկեան առաջին առղի միջոցով ընտրել ներածուող տառերի եւ տողերի միջև հեռաւորութիւնը, այսինքն՝ միջակայք։ Նոյն ուղղանկեան երկրորդ տողի ընտրման միջոցով կարելի է փափոխել ներածուող տեքստի զիրքը տեսամիպի վրայ։ Երրորդ տողը հնարաւորութիւնն է տալիս մազնիսսկան սկաւուակի վրայից ընթերցել նախասէս զրանցուած մէկ այլ տպաստոփի ֆայլ։ Ընլ որում նրան ընդհանուր տեսքի մասին ինֆորմացիա առլու նախառուկով տպաստոփ առաջին տասը արտահուում է էկրանի աջ միջին մասում։ Համապատասխան ընարաթիւններ կատարելուց յետոյ տեքստի ներածման անցնելու համար անհրաժեշտ է աեղափակի ցուցիչը «Այս» վանդակի վրայ եւ սեղմել թուղարկման ստեղնը։ Տեքստի ներածման

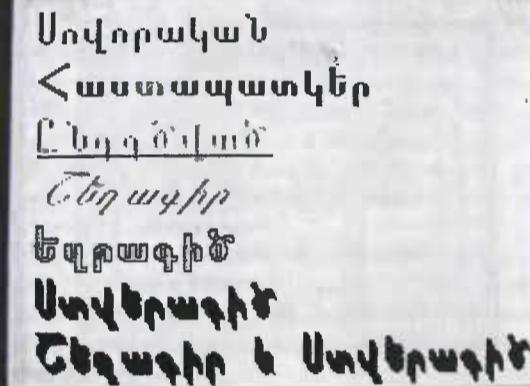
Հարագարանի Փոփոխում	
Պատկերների գրտարկում	
1	Տնակ
2	Ագտութենա
3	Ինքնաթիռ
4	Գնացք



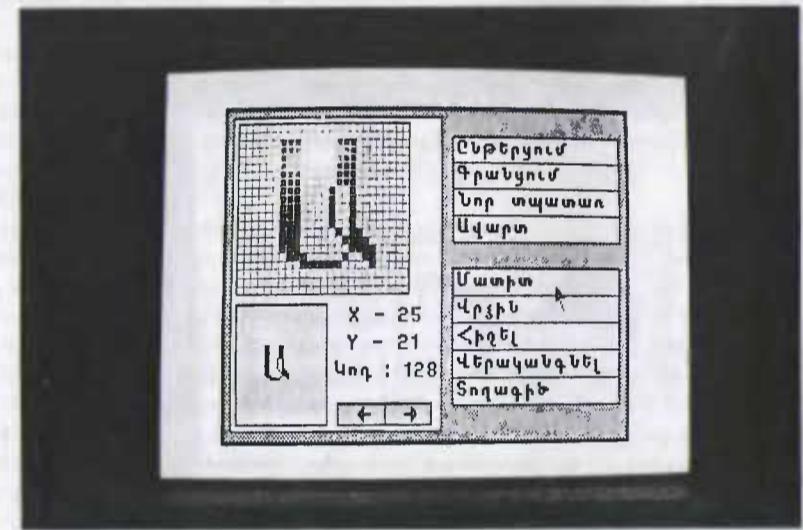
Նկ. 31

ընթացքում յառկանիշների ցուցակին անցնելու անհրաժեշտութեան դէպքում՝ պէտք է սեղմիլ գաղարեցման ստեղնը:

Տեքստերի մենյուի երկրորդ հնարաւորութիւնը կազուած է տպատափի խմբագրման, այսինքն՝ նրա ձեւափոխման եւ ճշաման հետ Տպատափի խմբագրման ժամանակ տեսատիպի վրայ արտածուում է համապատասխան նկարը (նկ. 33), որի ձախ մասում ներկայացնուում են խմբագրուող տառի ընական եւ խոչորացուած տեսքը, նրա առաւելագոյն չափսերը՝ հորիզոնական եւ ուղղահայեաց ուղղութիւններով եւ զիրքը կողային աղիւսակուում, այսինքն՝ կողը: Համապատասխան կողի ներածման կամ նկարի վրայ պատկերուած սլաքների օգտագործման միջոցով կարելի է ընտրել խմբագրուող տառը: Նկարի աջ վերին մասից համապատասխան ընտրութիւն կատարելով, կարելի է տպատառը ստանդարտ ֆայլի տեսքով ընթերցել եւ գրանցել մագնիսական սկանառակի վրայ կամ նոր տպատառ սկսել: Վերջին դէպքում ցուցիչի տեղաշարժի ստեղների միջոցով կարելի է փոփոխել տառի առաւելագոյն չափսը, որը ընդհանուր է տպատափի բոլոր տառերի համար: Նկարի աջ ստորին մասում ներկայացնուած են տպատառի խմբագրման հնարաւոր միջոցները: Խմբագրուող տառի տեսքը փոփոխելու համար, կարելի է օգտուել մատիոնի կամ վրձինի հնարաւորութիւններից, ի տարբերութիւն վերը դիտարկուած պատկերի խմբագրման զործընթացի, տուեալ դէպքում ջնջելու նպատակով օգտագործուում է սպիտակ գոյնով նկարելու հնարաւորութիւնը: Անհրաժեշտութեան դէպքում կարելի է ընթացիկ տառի տեսքը յիշել, վերականգնել եւ տեղաշարժել այն համապատասխան ժամանակ: Տողագծի նշումը հնարաւորութիւն է տալիս պահպանել տառերի մակարդակը անկախ նրանից, թէ նրանք տողից ցածր գտնուող մաս ունեն, թէ՝ ոչ:



Նկ. 32



Նկ. 33

Ամփոփելով՝ նշենք, որ պատկերների վերամշակման նման համակարգերը, որոնք սովորաբար կոչում են քոմիլիֆերային զրաֆիլայի համակարգեր, չեն պարունակում ոչ մի ներքին իմաստայնութիւն: Դրանց միջոցով, օրինակ, հնարաւոր չէ էկրանի վրայ ունենալով աւտոմիքնայի պատկերը, պատել այն՝ հակառակ կողմը տեսնելու նպատակով: Նման համակարգի միջոցով մենք կարող ենք վերամշակել միայն այն պատկերը, որը տուեալ պահին արտածուած է տեսուաժարի վրայ: Բաւականին չաս կիրառումներում անհրաժեշտ է ունենալ նաև վերամշակման օրինական երկրաշափական մոդելները: Օրինակ՝ չենք նախադեմին ճարտարապետի աշխատանքի արդիւնքը գնահատելու համար անհրաժեշտ է տեսնել չենքի տեսքը տարրեր կողմերից: Նման հնարաւորութիւնները են պահանջում առողջապահական, մեքենաշրնչական եւ մի շարք այլ խնդիրների լուծման ընթացքում: Երկրաշափական մոդելների կառուցման եւ վերամշակման գործողութիւններ իրազործող համակարգերը քոմիլիֆերային երկրաշափութեան համակարգեր են կոչում: Բնական է, որ քոմիլիֆերային երկրաշափութեան համակարգերը շատ տեղի բարդ են եւ պահանջում են քոմիլիֆերի աւելի բարձր բնաւթագրեր, քան քոմիլիֆերային զրաֆիլայի համակարգեր:

3. ՔՈՄՓԻԼԻԹԵՐՆԵՐԻ ԱՇԽԱՏԱՆՔԻ ՍԿԶԲՈՒԽՔՆԵՐԸ

3. 1. Քոմիլիթերի ընդիհանուր կառուցուածքը

Նախորդ զվարում դիտարկուած եւ նրանց նման ծրագրային համակարգերի յարմարաւետութեան աստիճանը, ինչպէս համոզուեցինք, համապարութիւնն է տալիս բաւական արգիւնաւէլու օգտագործել անհամական քոմիլիֆերը՝ զրեթէ առանց որեւէ պատկերացում ունենալու նրա կառուցուածքի եւ աշխատանքի սկզբունքների մասին: Այսուամենայնիւ, քոմիլիֆերների աւելի լայն եւ արդիւնաւէլու կիրառման, նրանց հնարաւորութիւնների լիովին օգտագործման համար անհրաժեշտ է բնդհանուր տեղեկութիւններ ունենալ քոմիլիֆերի բաղկացուցիչ մասերի եւ նրանց համատեղ աշխատանքի սկզբունքների մասին:

Չնայած այն հանդամանքին, որ զոյսութիւն ունեցող քոմիլիֆերները իրենց արտաքին տեսքով, չափերով եւ գործողութիւններով զգալիօրէն տարրերուում են միմեանցից, զրանց աշխատանքի սկզբունքները եւ կառուցուածքը զրեթէ նոյնն են: Քոմիլիֆերի կառուցուածքը ամրողջապէս պատկերացներու եւ նրա աշխատանքի սկզբունքները աւելի լաւ պատկերացներու համար կիրայիշենք, թէ սովորական պարզագոյն հաշուիչի օգնութեամբ լայնորդականութեամբ են կատրուում հաշուարկները: Որպէս օրինակ զիտարկենք հետեւեալ լինդիրը. գանել $A=128$, $B=46$ եւ $C=231$ երեք թուերի միջն թուարանականը: Թղթի վրայ նախապէս զրւում են սկզբնական տուեալները (նկ. 34, առաջին տող) եւ երեք թուերի միջն թուարանականի հաշուման բանաձեւը (երկրորդ տող): Այսուհետեւ կարդացուած են Δ եւ B թուերը եւ հաշուիչի միջոցով գումարուում: Հաշուիչի վրայ ստացուած միջանկեալ արդիւնքը դրւում է թղթի վրայ (երրորդ տող): Ստացուած արդիւնքը գումարուում C թուերն եւ ստացուած է երկրորդ միջանկեալ արդիւնքը, որ նոյնպէս գրւում է թղթի վրայ (չորրորդ տող): Վերջապէս, երկրորդ միջանկեալ արդիւնքը հաշուիչի միջոցով բաժանուում է 3ի եւ ստացուած վերջական արդիւնքը գրւում է թղթի վրայ (չինդիրորդ տող):

Դիտարկենք այս խնդրի լուծման ընթացքում ֆունկցիաների բաշխումը:

1. Հաշուիչը կատարուում է արտած թուերի հետ մարդու կողմից առաջարուած գործողութիւններու:
2. Թղթի վրայ նը գրանցուում խնդրի լուծման բանաձեւը, ելակէտային տուեալները, միջանկեալ եւ վիրջնական արդիւնքները: Այս եղանակով և յիշուում խնդրի լուծման բանաձեւը օգտագործուու ինֆորմացիան:
3. Մարդը դրդի վրայ գրում է ելակէտային տուեալները եւ բանաձեւը, օգտագործելու տուեալ բանաձեւը, ղեկավարում է կատարուուղ հաշուարկները,

այդ քում ինֆորմացիան ստեղծաշարի միջոցով փոխանցում է հաշուիչին
եւ հաշուիչից՝ բղբին:

Որպէսզի ստեղծել այնպիսի սարք, այլ խօսքով՝ քոմֆիթեր, որը առանց մարդու անմիջական մասնակցութեան կարող է լուծել տուեալ ինդիրը, անհրաժեշտ է հաշուիչին աւելացնել մի շարք հանդույցներ։ Փորձենք ներկայացնել այդ հանդույցներ։

1. Քոմֆիթերը պլտք է ունենայ այնպիսի հանգոյց, որը մեր օրինակում կը փոխարիթի բղբին։ Դա ինֆորմացիայի գրանցման, պահպանման եւ ընթերցման համար նախատեսուած յիշող սարքն է, կամ յիշողութիւնը, որում պահեում են խնդրի ելակետային տուեալները, միջանկեալ եւ վերջնական արդիւթեները։ Բացի դրանց՝ յիշողութեան մէջ պահեում է խնդրի լուծման յաջորդականութեան մասին ինֆորմացիան։ Մեր օրինակում դա միջին քուարանկանի բանաձեւից քինող գործողութիւնների կատարման յաջորդականութիւնն է։ Այսպիսով, յիշողութեան մէջ գրանցում է ոչ թէ խնդրի լուծման բանաձեւը, այլ յաջորդականութիւնը։ Յանախ քոմֆիթերի կազմի մէջ մտնում են մի բանի տարրեր տիպի յիշող սարքեր, որոնք համախմբում են մէկ յիշող համակարգում։

2. Խնդրի լուծման ընթացքին մասնակցում է քոմֆիթերի կազմի մէջ մտնող դեկավորող սարքը, որը յիշողութեան մէջ պահուած գործողութիւնների յաջորդականութեան հիման վրայ կազմակերպում է ոչ միայն քոմֆիթերի միւս բոլոր սարքերի աշխատանքը, այլև ինֆորմացիայի փոխանցումը նրանց միջեւ։

3. Մինչեւ խնդրի լուծելը, անհրաժեշտ է յիշողութեան մէջ գրանցել ելակետային տուեալները եւ գործողութիւնների կատարման յաջորդականութիւնը։ Այդ նպատակին է ծառայում ինֆորմացիայի ներածման սարք։

4. Խնդրի լուծումն աւարտելուց յետոյ անհրաժեշտ է յիշողութիւնից արտածել վերջնական արդիւնքը։ Այդ նպատակին է ծառայում ինֆորմացիայի արտածման սարք։

5. Քոմֆիթերում հաշուիչի գործողութիւնները կատարող սարքը, սովորաբար, կոչում է քուարանկան գործողութիւնների կատարման համար։ Յանախ այս սարքը կատարում է նաև այսպէս կոչուած տրամարանական, օրինակ՝ համեմատման գործողութիւնները, որոնք կարող են անհրաժեշտ լինել մէկ այլ խնդրի լուծման համար։

Այսպիսով, մեր խնդրի աւտոմատ լուծման համար սկալազոյն քոմֆիթերը պէտք է քաղկացած լինի հետեւեալ հինգ սարքերից՝ յիշողութեան, զեկավարման, ներածման, արտածման եւ թուարանական։ Այս հինգ սարքերը բաւական են կամայական խնդրի լուծման համար։ Նշուած սարքերի միջեւ անհրաժեշտ կապերը ցոյց են արուած քոմֆիթերի կառուցուածքային սիսեմայում (նկ. 35)։ Նկարում լայն սլաքներով նշուած են ինֆորմացիայի, իսկ նեղերով՝ զեկավարման ասղանչանների փոխանցման ուղղութիւնները։ Ինչպէս երեւում է նկարից՝ ինֆորմացիան սարքերի միջեւ հրմանակում փոխանցուում է թուարանական սարքի միջոցով, իսկ զեկավարման ազդանշանները բոլոր սարքերին արուած են զեկավարող սարքից։ Այս երկու սարքերը, որոնք սերտօրէն կապուած են միմեանց հետ եւ իրազորուած են ինֆորմացիայի վերամշակման գիմնական աշխատանքը, յաճախ միաւորուում են պրոցեսոր կամ աւելի ճիշտ կենտրոնական պրոցեսոր կոչուող մէկ սարքի մէջ։

$$A=128, \quad B=46, \quad C=231$$

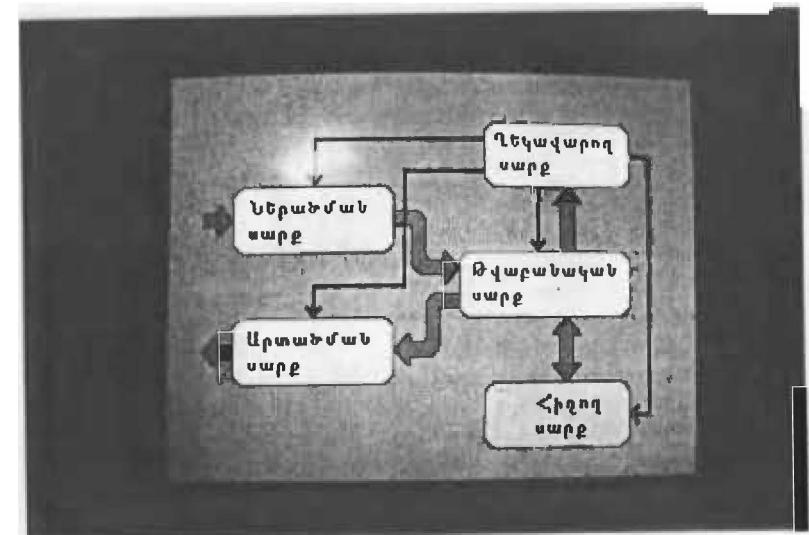
$$(A+B+C)/3$$

$$A+B=174$$

$$A+B+C=405$$

$$(A+B+C)/3=135$$

Նկ. 34



Նկ. 35

Դիտարկենք, թէ ինչպէս է լուծում վերը նշուած խնդիրը քոմիլթերի վրայ։ Նախապէս մարզը կազմում է բանաձեռների համապատասխան խնդրի լուծման համար անհրաժեշտ զործողութիւնների յաջորդականութիւնը։ Այնուհետեւ ներածման սարքի միջոցով նա կատարում է այդ գործողութիւնների յաջորդականութեան եւ հելակտային տուեալների ներածումը յիշողութեան մէջ։ Իսկ երբ արդէն քոմիլթերը պատրաստ է խնդրի լուծման համար, զեկավարման սարքը յիշողութիւնից ընթերոցում է առաջին զործողութիւնը և զեկավարում է նրա կատարումը։ Այսինքն, ընթերոցում է յիշողութիւնից Ա և Յ թիւերը, որոնց հետ պէտք է կատարուի զործողութիւնը, փոխանցում դրանք համապատասխան գործողութեան (մէր օրինակի դէպքում գումարման) կատարման ցուցմունքի հետ թուարանական սարքին։ Գործողութեան աւարտից յետոյ զեկավարման սարքը զրանցում է արդինքը յիշողութեան մէջ։ Դրանից յետոյ զեկավարող սարքը յաջորդականորէն ընթերոցում է յաջորդ զործողութիւնները եւ զեկավարում նրանց կատարումը։ Վերջնական արդինքի սարցումից յետոյ արտածման սարքի միջոցով այն յիշողութիւնից փոխանցուում է մարդուն։

Քոմիլթերի անրաժան բաղկացուցիչ մասն է կազմում նաեւ նրա ծրադրույին տպահովումը։ Առանց համապատասխան ծրագրերի զործնականում հնարաւոր չէ սովորել մեթոնային որեւէ օգտակար զարժ կատարել։ Ներկայումս տարբեր տիպի անհատական քոմիլթերների համար ստեղծուած են տասնեակ հաղորդաւոր ծրագրեր և ծրագրային համակարգեր, որոնք կարելի է բաժանել հետեւեալ երեք հիմնական զասերի՝

1. Օպերացիոն համակարգեր.
2. Կիրառական համակարգեր.
3. Մրագրաւորման եւ գործիքային համակարգեր.

Օպերացիոն համակարգը ծրադրույին ապահովման կարեւորագոյն մասն է կաղմամատ եւ քոմիլթերի բոլոր աերինիկական միջոցների ընդհանուր զեկավարումն է կաղմանիկապում։ Վերջինս հնարաւորութիւն է տալիս միւս զասերի ծրագրերը աղաւույ աերինիկական միջոցների հետ անմիջական շփումից եւ նրանց առանձնայատկութիւնները։ Հաշուի առնելու անհրաժեշտութիւնից։ Երկրորդ սերնդի անհատական քոմիլթերներում հիմնականում օգտագործուում է սկաւառակային օպերացիոն համակարգը, որը մազնիսական սկաւառակների վրայ զրուած ինֆորմացիայի վերամրական լայն հնարաւորութիւններ է ընձեւուում։ Մրագրաւորման եւ գործիքային ամերկարդիրը անհրաժեշտ են ծրագրաւորողներին ուրիշ ծրագրային միջոցներ սփոյծերու կամ օգտագործողներին որոշ մասնագիտական զործողութիւններ, օրինակ մոպնիսական սկաւառակների հետ աշխատանքը պարզեցնելու համար։ Բոլոր նորինում, եթէ օգտուենք քաղտնաեսութեան տերմինաբանութիւնից, զրանք ինտենսիվայի արագորութեան միջոցներն են, իսկ կիրառական համակարգերը զաման միջոցներն են։ Կիրառական համակարգերը, որոնցից մի քանիսի հետ և ներածանուցանք երկրորդ զմանում, կազմում են սովորաբար ծրագրեր չզգող, այլ չի ոչի իր ինդիբների լուծման համար զրանք օգտագործող մարդուն ուղղուած ծրագրի ապահովման մտկարգակը։

8. 2. Կենտրոնական պրոցեսոր

Անհատական քոմիլթերի կենտրոնական պրոցեսորը իրենից ներկայացնուում է տասնեակ պարունակող մեծ ինտեղրալ սիսեմա, որը կոչւում է միկրոպրոցեսոր։ Նա պարունակում է բաւական բարդ տրամաբանական սիսեմա, որը ներառուում է թուականական սարքը, զեկավարման սարքը եւ վերջիններիս աշխատանքի համար անհամար հրաժեշտ փոքրիկ յիշող սարք։

Միկրոպրոցեսորում, ինչպէս եւ բոլոր տեսակի քոմիլթերների պրոցեսորներում, ինֆորմացիան ներկայացւում է երկուական հաշւման համակարգում։ Այդ համակարգը տարրերուում է աւանդական տասական հաշւման համակարգուց՝ միայն թուակ ներկայացման համար օգտագործուող թուանշանների քանակով։ Տասական համակարգում հաշւմակարգում մինչեւ ինը՝ յաջորդ թիւը զրանցում ենք կով։ Տասական համակարգում հաշւմակարգում մինչեւ ինը՝ յաջորդ թիւը զրանցում ենք երկու կարգի միջոցով, քանի որ բոլոր թուանշանները արդէն օգտագործուած էին։ Երկուական համակարգում անցումը երկարագանի թուին կատարուում է արդէն հաշւման երկրորդ քայլում։ Այսպիսով՝ քանի որ երկուական համակարգում օգտագործուում են միայն երկու թուանշաններ (0 և 1), ապա «երկու» թիւը այդ համարգործուում են միայն երկու թուանշաններ 10 տեսքը։ Որպէս օրինակ նկ։ 36ում ներկայացւուած են տասակարգում կան թուանշաններին համապատասխանող երկուական թուերը։

Տասական	Երկուական
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Թուարանական հաշուարկները երկուական համակարգում շատ աւելի պարզ են կատարւում, քան տասականում: Օրինակ՝ նկ. 37ում համեմատութեան համար ներկայացուած են տասական եւ երկուական համակարգերում գումարման կանոնները ներկայացնող աղիւակները: Աղիւակների տողերը եւ սիւնակները ներկայացնում են գումարելները, իսկ նրանց հատման վանդակում գրանցուած է համապատասխան գումարը: Ընդ որում փակագծերի մեծ ներկայացուած է յաջորդ կարգին անցնող միաւորը: Հանման գործողութիւնը երկուական համակարգում նոյնպէս շատ աւելի

Տասական համակարգ

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	(1) 0
2	2	3	4	5	6	7	8	9	(1) 0	(1) 1
3	3	4	5	6	7	8	9	(1) 0	(1) 1	(1) 2
4	4	5	6	7	8	9	(1) 0	(1) 1	(1) 2	(1) 3
5	5	6	7	8	9	(1) 0	(1) 1	(1) 2	(1) 3	(1) 4
6	6	7	8	9	(1) 0	(1) 1	(1) 2	(1) 3	(1) 4	(1) 5
7	7	8	9	(1) 0	(1) 1	(1) 2	(1) 3	(1) 4	(1) 5	(1) 6
8	8	9	(1) 0	(1) 1	(1) 2	(1) 3	(1) 4	(1) 5	(1) 6	(1) 7
9	9	(1) 0	(1) 1	(1) 2	(1) 3	(1) 4	(1) 5	(1) 6	(1) 7	(1) 8

Երկուական համակարգ

	0	1
0	0	1
1	1	(1) 0

պարզ է, քան տասականում: Զնայած դրան՝ թուարանական սարքի կառուցուածքը պարզեցնելու նպատակով հիմնականում հանման գործողութեան փոխարէն կատարւում է բացասական թուի հետ գումարման գործողութիւն: Ինչպէս յայտնի է՝ դրանք լիովին համարժեք գործողութիւններ են: Դիտարկենք քոմիկիւթերներում բացասակայացնող աղիւակները: Աղիւակների տողերը եւ սիւնակները ներկայացնում են գումարելները, իսկ նրանց հատման վանդակում գրանցուած է համապատասխան գումարը: Ընդ որում փակագծերի մեծ ներկայացուած է յաջորդ կարգին անցնող միաւորը: Հանման գործողութիւնը երկուական համակարգում նոյնպէս շատ աւելի

Բաղմապատկման եւ բաժանման գործողութիւնները պրացեսորում հաշուի առնելով երկուական համակարգի կատարում հինգ կարգանի երկուական թուերի հետ, որոնցից գոյց է տալիս թուի նշանը, ապա տասական 3 թիւր կ'ունինայ 00011 տեսքը, իսկ բացասական -3ը 11101 տեսքը: Եթէ փորձենք զումարելի 7 եւ -3 տասական թուերը, այսինքն՝ զումարենք պրոցեսորում 00111 եւ 11101 թուերը, կը ստանանք 00100 հինգ կարգանի թիւր (տասական 4) եւ մէկ միաւոր փոխանցուուղ դէպի փերբորդ կարգը, որը կը կորչի վերջինիս բացակայութեան պատճառով:

Տրամաբանական գումարույթ

	0	1
0	0	1
1	1	1

Տրամաբանական բազմապատկույթ

	0	1
0	0	0
1	0	1

Պրոցեսորի կարեւորագոյն ընութաղիրը նրա կարգայնութիւնն է, որը ցոյց է տալիս նրա մէջ զիրամշակուող երկուական թուերի կարգիրի քանակը: Ինչպէս արդէն նշուել է, անհատական քոմֆիւթերների միկրոպրոցեսորների կարգայնութիւնը սովորաբար հաւասար է 8ի, 16ի կամ 32ի: Միկրոպրոցեսորի կազմի մէջ մտնող յիշողութիւնը ըստկացած է մի քանի ուեզիսաորներից, որոնց կարգայնութիւնը համապատասխանում է միկրոպրոցեսորի կարգայնութեանը: Ուեզիսաորները հիմնականում օգտագործում են միկրոպրոցեսորում՝ գործողութիւններ կատարելու ընթացքում միջանկետ տուեայնիւրա միեւնու համաս:

Կողեւոր նշանակութիւն ունի ոչ միայն միկրոպրոցեսորի, այլև նրա ստւելա-ների հաղորդաթիթեղի կարգայնաւիճնը, այսինքն՝ այն հազորպարաբերի քանակը, սրսոց միջոցով միկրոպրոցեսորները ինֆորմացիա է փոխանակում քոմփիթերի կազմի մեջ մանող մերս սալֆերի հետ: Տուեալների հաղորդաթիթեղի կարգայնութիւնը կարող է ասլրիերուել միկրոպրոցեսորի կարգայնութիւնից: Օրինակ՝ անհատական քոմփիթերներում լայն կիրառութիւն ստացած 8088 եւ 8086 միկրոպրոցեսորները ֆունկցիոնալ տեսակէտից լիովին համանման են, միայն 8088ը տնի 8 կարգանի տուեալներ հաղորդաթիթեղի իսկ 8086՝ 16 կարգանի: Տուեալների հաղորդաթիթեղի մեծ կարգայնութեան դէպքում ինֆորմացիայի փոխանակումը միկրոպրոցեսորի եւ միւս սալֆերի միջն կատարում է աւելի արագ, սակայն դրա հետմէկուեցած անում է նաև պահանջուող սպասարկուող էլեկտրոնային սիստեմների քանակը եւ համապատասխանաբար քոմփիթերի դինը: Այդ պատճառուով երկրորդ սերնդի պարզագոյն անհատական քոմփիթերում նախընտրում են օղտագործել 8088 միկրոպրոցեսորը:

Միկրոպլազմա սրբ միայն կարեւուրագոյն բնութափիրը ասակտացին յաճախտականութիւնն է, որը լնորոշ է միկրոպլազմա պարզադոյն գործողութիւնների կասասրման արագագութիւնը կամ այլ կերպ տասձ՝ միկրոպլազմա արագագործութիւնը։ Տակտացին յաճախտականութիւնը տրւում յատուկ տակտացին գեներատորի կորմից։ Ժամանակակից 8 կարգանի միկրոպլազմա ատակտացին յաճախտականութիւնը կազմում է 1ից 6 մեղահերց, 16 կարգանիներինը՝ 4ից 16 մեղահերց, իսկ 32 կարգանիներինը՝ 20ից 35 մեղահերց։ Հարկ է նշել, որ մէկ պարզագոյն զսրծողութիւն (արինակ՝ միկրոպլազմա ու ռեզյստորներում զրանցուած ելիքու ամրազջ թիւ գումարել եւ արդիւնքը զրանցել ու եղիսարում) կատարելու համար պահանջում է 4 տակտ։ Այսինքն՝ 16 կարգանի միկրոպլազմա 10 մեղահերց տակտացին յաճախտականութեան դէպքում այդպիսի զործողութիւնների կատարման արակտալութիւնը կը լինի 2,5 միլլուն զործողութիւն վայրիկեանում։ Եթէ թուերը ընթիւրցւում են յիշչողութիւնից, եւ արդիւնքը պէտք է դրանցուի այնաեղ, ապա զործողութեան կատարման համար պահանջում է 20-25 ատակտ, այսինքն՝ արագագործութիւնը կը լինի 400-500 հազար զործողութիւն վայրիկեանում։ Երական բազմապատկման զործողութիւնների դէպքում միկրոպլազմա արագագործութիւնը նուազում է եւս 20ից 40 անգամ եւ կոտզում է մեր օրինակում ընդամէնը 10-25 հազար զործողութիւն վայրիկեանում։ Սակայն այս դէպքում օգնութեան են գալիս այսպէս կոչուած պրոցեսորակիցները, այսինքն՝ զիտաւոր միկրոպլազմա կամ կոշտացուած միկրոպլազմա արագագործութիւնը, որն յատուկ նախատեսուած է միայն արագագործութիւնների արագ կատարման համար։ Թուարանական պրոցեսորակիցները չեն կարող ինքնուրան աշխատել, նրանք պէտք է միշտ կցուած լինեն գիտաւոր միկրոպլազմա արագագործութիւնների արագ կատարման համար։

ցեսօրին: Օրինակ 8088 և 8086 միկրոպլատֆորմների հետ օգտագործվում է 8087 պլատֆորմակիցը, որը հնարաւորութիւն է տալիս իրական թուերի հետ կատարուղ գործողութիւնները արագացնել 5ից 10 անգամ: Պրացեսորակիցը հնարաւորութիւն է տալիս բարձրացնել նաեւ թուաբանական գործողութիւնների ճշգրաւութիւնը, քանի որ սովորաբար այն ունեն մի քանի անգամ աւելի մեծ կարգայնութիւն:

Ֆունկցիոնալ անսակէափց միկրոպարոցեսորը բնութագրւում է իր հրամանների համակարգով, որը որոշում է ըոլոր գործողութիւնների ծրագրւորման կարգը, օպերանդների արմուն եղանակները, հատցէաւորման ձեւերը եւ այլն։ Համակարգում պարունակացիքի առկայութեան զէքֆում սրոշ հրամաններ աւտոմատ կերպայի փոխանցուում են նրան կասարման համար, այնպէս որ Հիմնական ծրագիլը կարծեն թէ չի նկատում, թէ որտեղ են իրականում կառարւում նրա հրամանները։ Դա հնարաւորութիւն է առլիս ծրագրաւորիրիս հիմնականում ուշազրութիւն չդարձնել պրոցեսորակցի առկայութեան վրայ եւ կազմել ծրագիրը՝ առանց նրան հաշուի առնելու։ Եթէ այն քոմիքիթերում, սրի վրայ օգտագործում է կաղմուած ծրագիրը պրոցեսորակցի և աեղազդուած, ապա ծրագիրը կ'աշխատի բաւականին արագ, իսկ հակառակ դէքրում՝ հսմիկաստարար զանդադա։

❖ Պոմփիթերի հետ շխտելիս յաճախ յարմար է օգասուել ոչ թէ մարդու համար ընական տասուկան, այլ հաշումն տասնուեցական համակարգից: Այդ համակարգի տարբերակինը առասկանից կայտնում է միայն նրանում, որ ասսի փոխարքէն կիրառում են սահմանական իրանշաններ: Բայց առանձինական տասից, զրանք սովորաբար լրաբնական այրութենի առաջին վեց մեծասառերն են՝ A, B, C, D, E և F: Համեմատութեան համար նկ: Յուրաքանչյուր տասնուեցական մուանշանները և նրանց համապատասխանող առասկան ու երկուական թուերը: Տառնուեցական համակարգը շատ յարմար է երկուական թուերը կամբ ձևավարութեան համար, քանի որ, ինչպէս տեսնում ենք նկ, ՅՈՒՅ չորս կարգանի երկուական եւ մի կարգանի առանուեցական թուերի միջեւ փոխառութ միարժէք համապատասխանութիւն կայ: Այսինքն՝ ամէն հնարաւոր չորս կարգանի երկուական թուերն համապատասխանում է որոշակի տասնուեցական նիշ եւ՝ հակառակի: Նոյն նկարից երեւում է, որ նման համարժէքութիւն երկուական եւ տասկան համակարգերի միջեւ չկայ:

Յ. Յ. Յիշող համակարգեր

Գոմիկւեերի յիշող համակարգի մէջ կարող են մանել չորս հիմնական դասի յիշող սարքեր, որոնք տարրերում են նշանակութեամբ, օլտագործման եղանակներով, տեխնիկական բնութագրերով և այլն: Դրանք են՝ հաստատուն, օպերատիք, փոխարինող արագարին և չփոխարինուող արագաքին յիշող սարքը: Առաջին երկուսը իրենցից ներկայացնում են քամֆիների այսպէս կոչուած ներքին յիշողութիւնը, իսկ վերջինները, բնականարար, արագաքին յիշողութիւնը:

Ինֆորմացիոն յիշուղ սարքերում ընդունում, պահպանում եւ Հաղորդում էր երկուական կարգերի կամ բիտերի (1 կամ 0) բազմութեան ձեւով: ԸսդՀանրապէս կարելի է նշել, որ ոչ միայն սլուցեսորը եւ յիշող սարքերը, այլէւ քոմիգիթերը կազմի մէջ մասնապ զործնականում բոլոր սարքերը աշխատում են երկուական Հա-

Տասնվեցական	Տասական	Երկուական
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

նկ. 39

մակարդում: Թէ՛ նշանային, թէ՛ դրաֆիկական ինֆորմացիան յատուկ կերպով կոռաւորում են երկուական կողերով: Չնայած նրան, որ աշխատանքներ են տարւում զրաֆիկական ինֆորմացիայի կոդաւորման հիմնօրինակների ստեղծման ուղղութեամբ, ներկայումս առանձին պատկերների վերամշակման համակարգեր օգատգործ են զբաֆիկական ինֆորմացիայի կոդաւորման իրենց ուրոյն եղանակներ, որոնք կարող են սկզբունքորէն տարբերուել մէկ այլ համակարգում ընդունուածներից: Առանձին նշանի կոդաւորման համար սովորաբար օգտագործում է 8 երկուական կարգ, որոնք միասին կոչւում են բայտ: Դժուար չէ հաշուարկել, որ գոյութիւն

ունի ընդամէնը 256 տարբեր 8 կարգանի երկուական կոդային յաջորդականութիւն: Սովորաբար զրանցից առաջին 32ը օգտագործուում են քոմիտեների սարքերում որպէս գեկավարող ինֆորմացիա, իսկ միւսները կարող են օգտագործուել նշանային ինֆորմացիայի կողաւորման համար: Այսինքն՝ քոմիտեները միաժամանակ կարող է զործ ունենալ ոչ աւել քան 224 տարբեր նշանների հետ: Նշանային ինֆորմացիայի կողաւորման համար գոյութիւն ունեն համապատասխան հիմնօրինակներ, մասնաւորապէս նկ. 40ում ներկայացրուած է հայբէն այբուբենը եւ համապատասխան կետաղական նշանները պարունակող հիմնօրինակը, որը համաստուած է Հայկական ԽՍՀ Գիտութիւնների ակադեմիայի տիրմինարանական հոմիարէի կողմից: Նկարում ներկայացրուած են կողի համարը տասական եւ տասնուեցական համակարգերում, ինչպէս նաև նրան համապատասխանուղ նշանի տեսքը:

Ինֆորմացիայի ծաւալը յաճախ չափում են նրա մէջ պարունակուող նշանների քանակով, այսինքն՝ բայտերով: Համապատասխանաբար ինֆորմացիայի այն առաւելագոյն ծաւալը, որը հնարաւոր է գրանցել առևեալ յիշտ սարքի մէջ, այսինքն վերջինս ինֆորմացիոն ծաւալը նոյնպէս պէտք է չափուի բայտերով: Առօրեայում աւելի յարմար է օգտուել աւելի մէծ միաւորներից, ինչպիսիք են՝ կիլորայտը (1024 բայտ) եւ մեծարայտը (1024 կիլորայտ կամ 1048576 բայտ):

Հաստատուն յիշող սարքը կառուցրում է յատուկ կիսահաղորդչային մեծ ինտեղրականի մասին իրայ, որոնց մէջ ինֆորմացիան համապատասխան անվիրադարձ կասուցուածքային ձեւափոփութիւնների միջոցով զրանցուում է քոմիտեների արտադրման ժամանակ եւ նրա շահագործման ընթացքում չի փոփոխուած: Այսինքն այս սարքից քոմիտեների աշխատանքի ընթացքում կարելի է կասարել միայն ինֆորմացիայի ընթերցման գործողութիւն եւ հնարաւոր չի զրանցում կատարել: Անհամական քոմիտեների հաստատուն յիշող սարքի մէջ զրանցում է մի շարք յատուկ ծրագրերից կազմուած այսպէս կոչուած յենակէտային ներածման համակարգի հիմնական մասը: Այդ համակարգը նախատեսուած է քոմիտեների հիմնական հանդույնների ստուգման եւ նախնական թողարկման համար, ինչպէս նաև տուեալների ներածման եւ արտածման գործողութիւնների կատարումը ապահովելու համար: Բացի զրանից՝ այդ սարքում տեղ են գտնում մի շարք օժանդակ ծրագրեր եւ որոշ դէպքելում որեւէ ծրագրաւորման լեզուով, օրինակ՝ բջջուիկով ծրագրաւորման պարզագոյն միջոցներ: Հաստատուն յիշող սարքում գտնուող ինֆորմացիան նախահատուած է բայտերութիւն կատարուած է համապատասխան կիսահաղորդչային մեծ ինտեղրական սիսեմաների հիման վրայ, որոնցում պահուող ինֆորմացիան ննջուում է քոմիտեների սնուցման անհատութիւց յետոյ: Ցաւօք այս թերութիւնը դեռևս բնորոշ է բոլոր կիսահաղորդչային օգերաաի յիշող միկրոսիսեմների համար:

Անհատական քոմիտեների ներքին յիշողութեան ծաւալը սահմանափակուած է առաջին հերթին օգտագործուող միկրոպրոցեսորի հասցէի հաղորդաթիթեղի կար-

10-ական	16-ական	Նշան
032	20	!
033	21	"
034	22	#
035	23	\$
036	24	%
037	25	&
038	26	,
039	27	(
040	28)
041	29	*
042	2A	+
043	2B	.
044	2C	-
045	2D	.
046	2E	.
047	2F	/
048	30	0
049	31	1
050	32	2
051	33	3
052	34	4
053	35	5
054	36	6
055	37	7
056	38	8
057	39	9
058	3A	:
059	3B	:
060	3C	<
061	3D	=
062	3E	>
063	3F	?
064	40	@
065	41	A
066	42	B
067	43	C
068	44	D
069	45	E
070	46	F
071	47	G
072	48	H
073	49	I
074	4A	J
075	4B	K
076	4C	L
077	4D	M
078	4E	N
079	4F	O
080	50	P
081	51	Q
082	52	R
083	53	S
084	54	T
085	55	U
086	56	V
087	57	W

10-ական	16-ական	Նշան
088	58	X
089	59	Y
090	5A	Z
091	5B	[
092	5C	\
093	5D]
094	5E	^
095	5F	-
096	60	a
097	61	b
098	62	c
099	63	d
100	64	e
101	65	f
102	66	g
103	67	h
104	68	i
105	69	j
106	6A	k
107	6B	l
108	6C	m
109	6D	n
110	6E	o
111	6F	p
112	70	q
113	71	r
114	72	s
115	73	t
116	74	u
117	75	v
118	76	w
119	77	x
120	78	y
121	79	z
122	7A	{
123	7B	}
124	7C	‘
125	7D	’
126	7E	—
127	7F	—
128	80	Ա
129	81	Բ
130	82	Գ
131	83	Դ
132	84	Ե
133	85	Վ
134	86	Շ
135	87	Ջ
136	88	Ռ
137	89	Ջ
138	8A	Շ
139	8B	Ջ
140	8C	Շ
141	8D	Ջ
142	8E	Շ
143	8F	Ջ

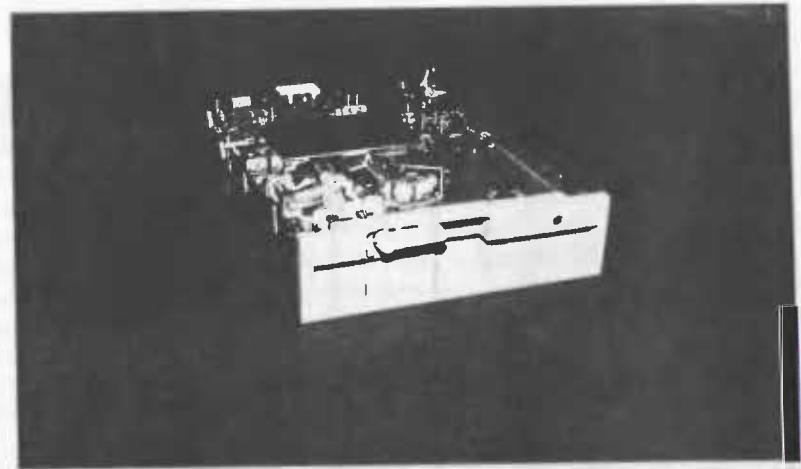
10-ական	16-ական	Նշան
144	90	Ճ
145	91	Շ
146	92	Ծ
147	93	Ը
148	94	Ծ
149	95	Ը
150	96	Ը
151	97	Ը
152	98	Ը
153	99	Ը
154	9A	Ը
155	9B	Ը
156	9C	Ը
157	9D	Ը
158	9E	Ը
159	9F	Ը
160	A0	Ը
161	A1	Ը
162	A2	Ը
163	A3	Ը
164	A4	Ը
165	A5	Ը
166	A6	Ը
167	A7	Ը
168	A8	Ը
169	A9	Ը
170	AA	Ը
171	AB	Ը
172	AC	Ը
173	AD	Ը
174	AE	Ը
175	AF	Ը
176	BO	Ը
177	B1	Ը
178	B2	Ը
179	B3	Ը
180	B4	Ը
181	B5	Ը
182	B6	Ը
183	B7	Ը
184	B8	Ը
185	B9	Ը
186	BA	Ը
187	BB	Ը
188	BC	Ը
189	BD	Ը
190	BE	Ը
191	BF	Ը
192	CO	Ը
193	C1	Ը
194	C2	Ը
195	C3	Ը
196	C4	Ը
197	C5	Ը
198	C6	Ը
199	C7	Ը

10-ական	16-ական	Նշան
200	C8	Ը
201	C9	Ը
202	CA	Ը
203	CB	Ը
204	CC	Ը
205	CD	Ը
206	CE	Ը
207	CF	Ը
208	DO	Ը
209	D1	Ը
210	D2	Ը
211	D3	Ը
212	D4	Ը
213	D5	Ը
214	D6	Ը
215	D7	Ը
216	D8	Ը
217	D9	Ը
218	DA	Ը
219	DB	Ը
220	DC	Ը
221	DD	Ը
222	DE	Ը
223	DF	Ը
224	EO	Ը
225	E1	Ը
226	E2	Ը
227	E3	Ը
228	E4	Ը
229	E5	Ը
230	E6	Ը
231	E7	Ը
232	E8	Ը
233	E9	Ը
234	EA	Ը
235	EB	Ը
236	EC	Ը
237	ED	Ը
238	EE	Ը
239	EF	Ը
240	FO	Ը
241	F1	Ը
242	F2	Ը
243	F3	Ը
244	F4	Ը
245	F5	Ը
246	F6	Ը
247	F7	Ը
248	F8	Ը
249	F9	Ը
250	FA	Ը
251	FB	Ը
252	FC	Ը
253	FD	Ը
254	FE	Ը
255	FF	Ը

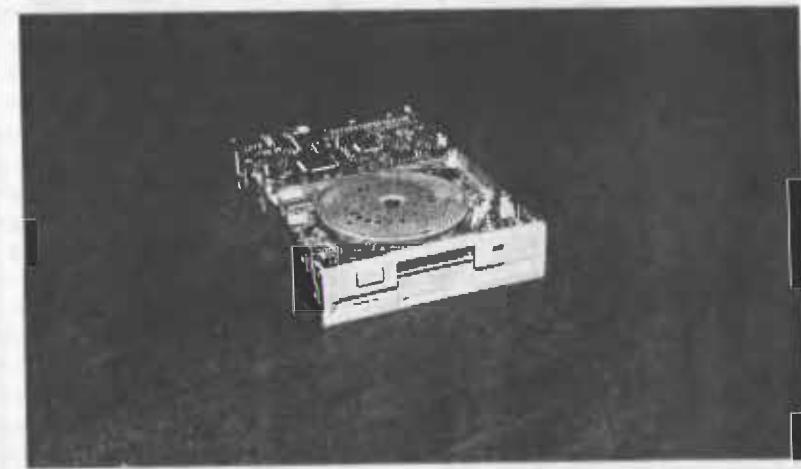
գայնութիւնով, այսինքն՝ այն հաղորդակարերի քանակով, որոնց միջոցով միկրոսկոպացնուող ինֆորմացիայի հասցէն փոխանցում է քոմիտերի կազմի մէջ մտնող միւս սարքերին: Հիմնականում 8 կարգանի միկրոպրոցեսորի յիշողութեան հասցէաւորման հնարաւորութիւնները սահմանափակում են 64 կիլոայտով, իսկ 16 կարգանիներինը՝ կից մինչեւ 16 մեղարայտով: Որոշ դէպքերում ներքին յիշողութեան ծաւալը սահմանափակող հանդէս գալ եւ զինը, քանի որ ներքին յիշողութեան տեսակարար (1 բայտի) արժէքը չաս աւելի մեծ է, քան արտաքինինը: Բացի զրանից հասցէների մի մասը մատչելի չէ ծրագրերի համար, քանի որ այն զբաղեցնուում է հաստատուն յիշող սարքի կողմից եւ պահեստաւորւում է համակարգի ծառայողական նպատակների համար՝ հիմնականում անհատական քոմիտերի սարքերի աշխատանքը ապահովելու նպատակով: Օրինակ՝ 8088 միկրոպրոցեսորի հասցէների տիրոյթը կազմում է 1 մեղարայտ, որից միայն 640 կիլորայտը կարող է զբաղեցնել օպերատիւ յիշող սարքի հասցէներով, իսկ մուացած 384 կիլորայտը զբաղեցնուում է հաստատուն յիշող սարքի հասցէների համար եւ պահեստաւորւում է:

Անհատական քոմիտերներում լայնօրէն կիրառուող մէկ այլ միկրոպրոցեսոր՝ 80286ը հնարաւորութիւն է տալիս հասցէաւորել մինչեւ 4 մեղարայտ յիշողութիւն, որից նոյն 384 կիլորայտը զբաղեցնուում է հաստատուն յիշող սարքի հասցէների համար եւ պահեստաւորւում է: Այսինքն՝ օպերատիւ յիշողութեան ծաւալը կարող է կազմիւ աւելի քան 3,5 մեղարայտ: Սակայն այս անգամ առաջ են զալիս օպերացիոն համակարգի կողմից զրուող սահմանափակումները: Վերը նշուած երկու միկրոպրոցեսորների հիման վրայ կառուցուած անհատական քոմիտերների ծրագրային համատեղելիութիւնը ապահովելու նպատակով ամենատարածուած սկաւառակային օպերացիոն համակարգը սահմանափակում է ծրագրերի համար օգտագործուող օպերատիւ յիշողութեան ծաւալը 640 կիլորայտով: Միւս 3 մեղարայտ օպերատիւ յիշողութիւնը հնարաւոր է օգտագործել այսպէս կոչուած էլեկտրոնային սկաւառակ սեղծելու նպատակով: Վերջինս իրենից ներկայացնում է օպերատիւ յիշողութեան մի տիրոյթ, որին դիմումը կատարուում է մագնիսական սկաւառակների համար բնորոշ եղանակներով:

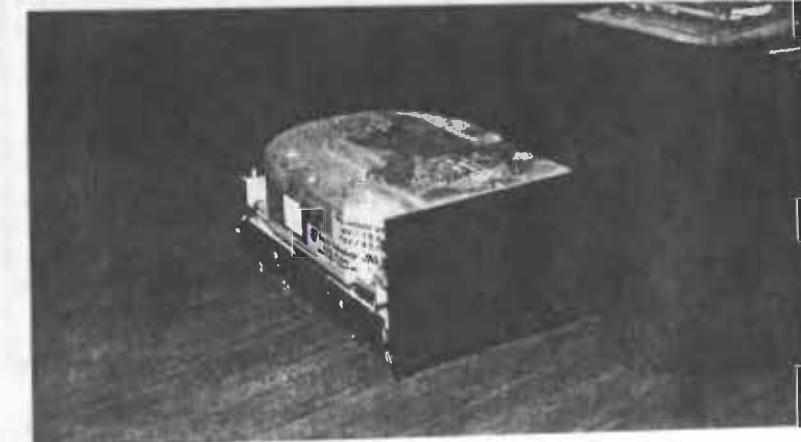
Ինֆորմացիայի երկարատեւ պահպանման համար քոմիտերներում հիմնականում օգտագործուում են արտաքին յիշող սարքերը: Ժամանակակից քոմիտերներում լայն տարածուում են գոտել ճկուն մատնիսական սկաւառակների վրայ հիմնուած սարքերը, որոնցում ինֆորմացիան դրանցուում մագնիսական շերտով ծածկուած եւ համապատասխան պատեանի մէջ տեղադրուած փոքր, ճկուն սկաւառակների վրայ: Ներկայուած ամենալայն տարածուումն են ստացել յատկապէս անհատական քոմիտերներում 133 միլիմետր տրամագիծ ունեցող սկաւառակների վրայ յիշող սարքերը (նկ. 41), որոնք ունեն բաւականին փոքր չափսեր (150×250×43 միլիմետր) եւ հնարաւորութիւն են տալիս (կախուած կառուցուածքից) մէկ սկաւառակի վրայ դրանցել 360 կիլորայտ կամ 1, 2 մեղարայտ ինֆորմացիա: Դա հիմնականում բաւարարում է անհատական քոմիտերների ճնշող մեծամասնութեան պահանջները: Վերջերս սկսեցին կիրառուել 89 միլիմետր տրամագիծ ունեցող սկաւառակների վրայ յիշող սարքերը (նկ. 42), որոնք ունեն աւելի փոքր չափսեր (100×130×40 միլիմետր) եւ զրահ հետ մէկտեղ հնարաւորութիւն են տալիս մէկ սկաւառակի վրայ դրանցել անգամ աւելի շատ ինֆորմացիա քան 133 միլիմետրանոց սկաւառակի վրայ՝



Նկ. 41



Նկ. 42



Նկ. 43

Հիլորայտ կամ 1,44 մեգաբայտ։ Դա պայմանաւորուած է այդ սկաւառակների յատուկ կառուցուածքով եւ համեմատարար աւելի կարծր պատեանով, որոնք սկաւառակի մակերեսին մագնիսական զլիիկների աւելի ճշգրիտ մօտեցում են ապահովում։ Այդ սարքերի լայն կիրառումը հիմնականում սահմանափակում է նրանով, որ անհաւական քոմիտիթերների ծրագրային համակարգերի մեծ մասը տարածում է 133 միլիմետրանոց սկաւառակների վրայ։ Նշուած սարքերը պատկանում են փոխուող արտաքին յիշող սարքերի զասին, քանի որ նրանց մէջ ինֆորմացիայի կրիչը՝ այսինքն՝ ձկուն սկաւառակը, բաւականին հեշտ տեղադրուում եւ հանուում է։ Դա հնարաւորութիւն է տալիս սարքի հետ համապատասխան քանսկութեամբ սկաւառակներ օգտագործելով պահանջուող ծաւալի ինֆորմացիա յիշել։

Գոփիուուղ արտաքին յիշող սարքերի թուին են պատկանում կարծր մագնիսական սկաւառակների վրայ յիշող սարքերը, որոնցում ինֆորմացիան զրանցուում է մի քանի մագնիսական շերտով ծածկուած մետաղեայ սկաւառակների վրայ։ Քանի որ սկաւառակները գոնուում են հերմետիկ պատեանի մէջ, ապա մագնիսական զլիիկները շատ մօտ են տեղաւորուում սկաւառակների մակերեսներին եւ վիրջիններս պտտուում են շատ աւելի արագ, քան ձկուն սկաւառակները համապատասխան սարքերում։ Դա ապահովում է շատ աւելի մեծ ինֆորմացիայի զրանցման խոտութիւն և փոխանցման արագութիւն։ Այս սարքերը, որոնք յաճախ կոչւում են «վինչեստեր» ափի սարքեր, սովորաբար 133 միլիմետրանոց ճկուն սկաւառակների վրայ յիշող սարքերին հաւասար չափսեր ունեն (նկ. 43), սակայն նրանց ինֆորմացիոն ծաւալը տասնեակ, նոյնիսկ հարիւրաւոր անգամ աւելին է։ Բնական է, որ դրանց զինը նոյնպէս աւելի բարձր է (միջինուում 5 անգամ)։ Տուեալ սարքերի հիմնական թերութիւնը կայանում է նրանում, որ նրանց ինֆորմացիոն կրիչը հնարաւոր չէ փոփոխ։ Վերջիւն փորձնական յիշող սարքեր ստեղծուեցին, որոնցում օգտագործուում են 99 միլիմետր տրամադով փոխուող կարծր մաղնիսական սկաւառակներ։ Այդ սարքերը արտաքին բնութաղրերով նման են «վինչեստեր» ափի սարքերին, սակայն նրանց ինֆորմացիայի կրիչը, յատուկ հերմետիկ պատեանի մէջ պարունակուող սկաւառակները, կարող են հեշտութեամբ հանուել եւ տեղադրուել։

Օպերացիոն համակարգի տեսահիմունքը սկաւառակների վրայ յիշող սարքերը ինֆորմացիային անմիջական դիմում են ապահովում։ Այսինքն համապատասխան ակինիկական եւ ծրագրային միջոցները հնարաւորութիւն են ընձեռնուում ինֆորմացիան զրանցել կամ ընթեցել սկաւառակի ցանկացած մասից սկաւառ։ Ընդ որում փոխուում է միայն մինչեւ սկաւառակի պահանջուուղ հատուածը մագնիսական զլիիկների տեղաշարժի ժամանակը։ Ի տարբերութիւն դրան՝ անհաւական քոմիտիթերի կազմի մէջ մտնող միւս սարքերի մեծ մասը յաջորդական դիմումով սարքեր են, որոնց հետ աշխատանքը կատարուում է յաջորդական քրանցման կամ ընթեցման սկզբունքով։

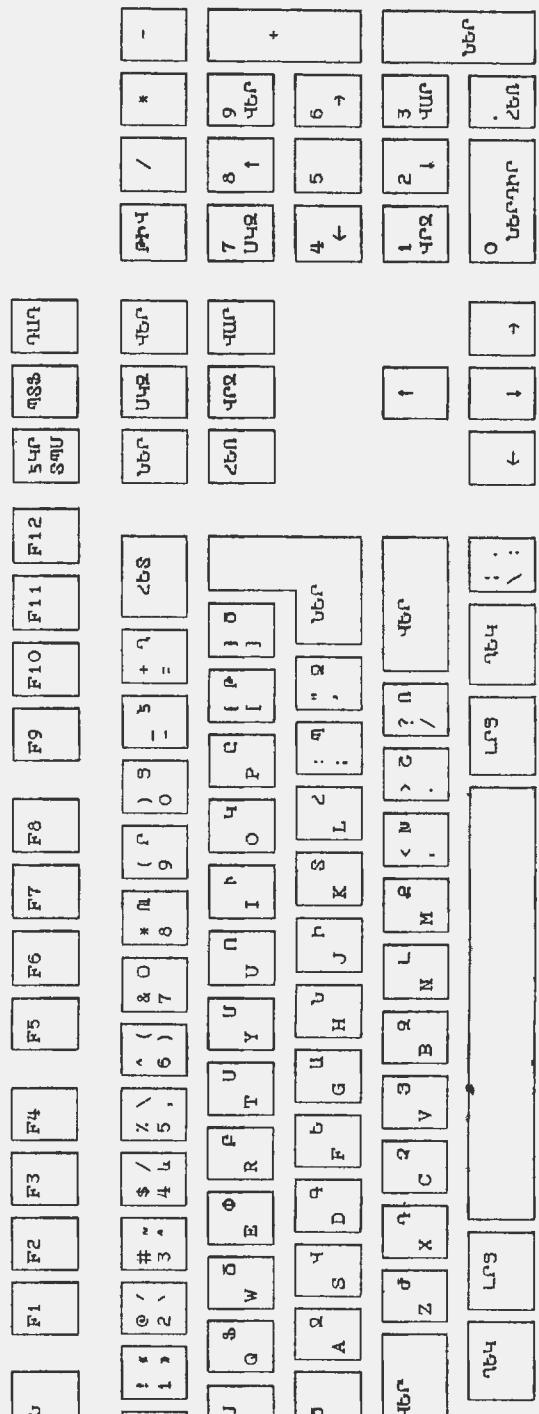
Ինֆորմացիայի ներածման եւ արտածման սարքերը

Անհաւական քոմիտիթերներում ինֆորմացիայի ներածման հիմնական սարքը ստեղնաշարժ է (նկ. 44), որը շատ նման է աւանդական գրամեքենայի ստեղնաշարժի։ Մտեկնաշարժի հիմնական մասը կազմում են տառերի, թուանշանների ստեղ-

ները, որոնք հիմնականում տեղաւորուած են համապատասխան գրամեքենաներում ընդունուած ձեւով։ Հիմնական մասի շուրջը տեղաւորուած են յատուկ ստեղնիրը, որոնց մի մասը նոյնպէս կային գրամեքենայի ստեղնաշարժի վրայ։ Դրանք են՝ հետաղարձ քայլի (ՀԵՁ), ներածման (ՆԵՌ), վերին ռեզիստրի (ՎԵՌ), մեծասառերի ֆիքսման (ՄԵՄ) եւ աղիւսակատիպիչի (ԱՂՄ) ստեղները։ Նոյն մասերում են տեղապուած զիկավարման (ՂԵԿ) եւ լրացման (ԼՐՑ) յատուկ ստեղնիրը։ Ստեղնաշարի աջ մասում գտնուում են թուանշանների եւ թուարանշանների գործողութիւնների նշանների ստեղնիրը։ Այդ սահեղների մի մասը թուանշանների վիքսման (ԹԻՒ) ստեղնի միջոցով փոխուում են՝ ցուցիչի տեղաշարժի եւ խմբագրման գործողութիւնների կատարմամբ։ Յուցիչի տեղաշարժի ուղղութիւնը նշուում է ստեղների վրայ պաֆներով, բացի զրանից ցուցիչը կարելի է տեղափոխել տեքստի սկիզբը (ԱՂԶ), վերջը (ՎՐՁ), մէկ էկրան կամ էլ զէպի վեր (ՎԵԲ) կամ վար (ՎԱԲ)։ Այս մասում տեղաւորուած ներդրման (ՆԵՐԴԻԲ) ստեղնը փոխանչատում է ներդրման եւ վերադրման ռեժիմները, իսկ հետացման (ՀԵՌ) ստեղնը հետացմում է ընթացիկ նիշը։ Յարմարութիւնն համար այդ բոլոր ստեղները առանձին կրկնուում են ստեղնաշարի վերին մասում տեղաւորուած են բանալու (ԲԱՆ), էկրանի տպման (ԷԿԻ ՏՊՄ), պտուման ֆիքսման (ՊՖՖ), զաղարի (ԴԱԴ) եւ 12 ֆունկցիոնալ (F1-F12) ստեղները։ Յատուկ եւ փունկցիոնալ ստեղների նշանակութիւնը հիմնականում ծրագրաւորուում է, եւ տարբեր ծրագրերուում կարող է սարգեր լինել։

Ժամանակակից անհաւառական քոմիտիթերների ստեղնաշարերն ունեն մի կարեւոր առանձնայատկութիւն։ Ի տարբերութիւն միւս քոմիտիթերների ստեղնաշարերի՝ անհաւառական քոմիտիթերի ստեղնաշարը միելուպուցիւրը ուղարկուում է ոչ թէ նշանի կողը, այս սեղմուած ստեղնի յաջորդական համարը։ Սեղմուած ստեղնի իմաստի լմանման հետ կապուած բոլոր աշխատանքները կատարուում են ծրագրային հետանակուու։ Տուեալ զէպֆուած ստեղների կողաւորուումից անկախ է զանուում, որը կարուել պարզեցնում է ստեղնաշարի հետ աշխատանքը եւ միաժամանակ աւելի ճկուն է զարգնուում այն։ Կիրտուկան համակարգերի նախագծողները առաւելացնուույն յարմարաւչութիւն ասպահովերու նպատակով աշխատուում են կրճատել անհաւառական քոմիտիթերի վրայ իւրաքանչիւր գործ կատարելու համար անհրաժեշտ ստեղների սեղմուածների քանակը։ Դրան հասնելու պարզագոյն միջոցը ծրագրային եղանակուու առանձին ստեղների նշանակութիւնների փոփոխուուն է, որն էլ ապահովուում է վերը նշուած առանձնայատկութիւններան հաշուին։

Ծրագրերի աշխատանքի ընթացքում ինֆորմացիայի արտածուումը հիմնականում կատարուում է անհաւառական քոմիտիթերի տեսաստիպիչի վրայ։ Տեսափերը կարող են զգալիօրէն տարբերուել միմեանցից, եւ նրանց տեսնիկական բնութագրելուն են հիմնականում որոշուում անհաւառական քոմիտիթերների եւ նրանց վրայ օպտագործուուղ ծրագրային համակարգերի հնարաւորութիւնները։ Առաջին հերթին միայն թուառային ինֆորմացիայի արտածման համար նախատեսնուուած տեսատիպերը պէտք է տարբերել զրափիկական տեսաստիպերից, որոնց էկրանին կարող է արտածուել ինչպէս զրափիկական, այնպէս էլ տասաթուային ինֆորմացիա։ Տեսափերը միւս կարեւոր յատկանիշը կապուած է զունաւոր կամ միայն մէներանգ (սեւ-սպիտակ) պատկերների արտածման հետ։ Հական նշանակութիւններու ունեն տեքստուային ճեւաչափը եւ պատկերի ճշգրտաւթիւնը։ Տեսասային ճեւաչափը բնութագրուում է մէկ տողի վրայ տեղաւորուուղ նշանների ստեղների սահակական վրայ առաջարկուույն քանակով եւ էկրանի վրայ տեղաւորուուղ



Նկ. 44

տողերի քանակով։ Գրաֆիկական ռեժիմում ճշգրտութիւնը տրւում է էլեկտրանի վրայ հորիզոնական եւ ուղղահայեց ուղղութիւններով տեղաւորուող կէտերի առաւելադոյն քանակով։ Մեներանդ ռեժիմում պայծառութեան մակարդակների առաւելադոյն քանակը կամ գունաւոր ռեժիմում գոյների առաւելադոյն քանակը նոյնակչ խստ կարեւոր բնութագրեր են։ Վերջապէս էլեկտրանի վրայ ամբողջ պատկերը կամ նրա առանձին մասերը, որոյն թուամ եւ նշաններ յստակ տարրերելու տեսակէտից մեծ նշանակութիւն ունի էլեկտրանի չափութիւնը։

Նշուած բնութագրերը կախուած են տեսատիպի կառուցուածքից եւ նրան քոմիտերի հետ կապուող զեկավարող սարքից, որը յաճախ կոչւում է աղապտեր։ Ներկայումս տեսատիպի վրայ տեքստերի կամ պատկերների ձեւաւորումը հիմնականում կատարում է յատուկ էլեկտրանի յիշողութեան հիման վրայ։ Պատկերի ամէն մի տարր, տեսատիպի վրայ առանձին կէտ, ձեւաւորում է այդ յիշողութեան 1, 2, 8 բիտերից կազմուած հաստուածի հիման վրայ։ Նշուած բիտերում պարունակուող ինֆորմացիան որոշում է համապատասխան կէտի պայծառութիւնը կամ գոյնը եւ մի շարք ուրիշ յատկանիշներ։ Էլեկտրանի յիշողութեան ծաւալը ուղղակիրէն կապուած է տեսատիպի պատկերի արտածման ճշգրտութեան հետ։ Այսպէս օրինակ, 640×200 կէտերի եւ մեներանդ տեսատիպի պայծառութեան երկու մակարդակի դէպքում պահանջուող էլեկտրանի յիշողութեան նուազադոյն ծաւալը կազմում է 16 կիլորայտ, նոյն ճշգրտութիւնը 16 գոյների դէպքում պահանջում է 64 կիլորայտ յիշողութիւն եւ այլն։ Պատկերների արտածման այս եղանակի դէպքում նշանները արտածուում են յատուկ կէտերունային սինեմայի միջոցով, որի ճշգրտութեանը համապատասխան՝ կէտային մատրիցայում (8×8, 8×12 եւ այլն) ձեւաւորում է ատակի եւ միւս նշանների տեսքը։

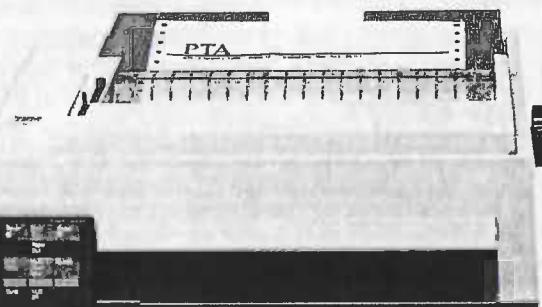
Երկրորդ սերնդի անհասական քոմիտերների մեծ մասը ունեն մեներանգ կամ գունաւոր էլեկտրոնաճառագայթային խողովակի հիման վրայ կառուցուած տեսատիպեր։ Էլեկտրոնաճառագայթային խողովակի զեկավարումը իրադրութեամբ է հիմնականում հինգ ստանդարտ ազդենչաններով՝ կարմիր, կանաչ, կապոյտ գոյների եւ հորիզոնական ու ուղղահայեց սինքրոնացման ազդենչաններով։ Տեսատիպը միացուում է համակարգին զեկավարող սարքի միջոցով, որը հիմնականում իրենից առանձին տպատախտակ է ներկայացնում։ Այդ սարքի կազմի մէջ մտնում են էլեկտրանի յիշողութիւնը եւ համապատասխան զեկավարող սինեմաները։ Բացի զրանից նախասեսուում է արտածուող տառերի եւ նշանների տեսքը, այսինքն՝ տեսատիպի տպատառը պահպանելու համար յատուկ յիշողութիւն։ Հիմնականում զա հաստատուն յիշող սարք է եւ տեսատիպի տպատառը փոփոխելու համար պէտք է զեկավարուղ սարքում եղածի փոխարէն տեղադրել նոր հաստատուն յիշող սարք։ Որոշ դէպքերում այդ յիշողութիւնը կառուցուում է օպերատիր յիշող տափքի ինտերակտ սինեմաների հիման վրայ, եւ տեսատիպի տպատառը փոփոխուում է ծրագրային եղանակով։ Անհասական քոմիտերներում համապատասխան զեկավարող սարքի միջոցով կարելի է որպէս տեսատիպ օգտագործել սովորական հեռուստացոյցը, սակայն նա բառականին ցածր ճշգրտութիւն է պահանջում՝ սովորաբար 256×256 կէտ։ Դիրակիր անհասական քոմիտերներում (նկ. 45) կիրառուում են հեռուկ բիւրեղների վրայ հիմնած տեսատիպերը, որոնք որոշ դէպքերում պահանջում են էլեկտրոնաճառագայթային խողովակներին բնորոշ ճշգրտութիւն։ Այդ տեսատիպերը պահանջում են չափործման յատուկ պայմաններ, լաւ լուսաւորուածութիւն եւ



Նկ. 45

համապատասխան դիտարկման անկիւն, որի պատճառով նրանք դեռևս լայն կիրառություն չեն գտել:

Անհատական քոմֆիւթերներում ինֆորմացիան թղթի վրայ արտածելու համար հիմնականում օգտագործում են տպող սարքերը, որոնց զիմանը ընութաղերը տպման որակը եւ արագությունն են: Այդ սարքերը կարող են ունենալ տարրեր աշխատանքի սկզբունքներ եւ դրա հետ կապուած տարրեր բնութագրեր: Մատրիցային տպող սարքերը (Նկ. 46) ներկայումս ամենատաղածուածն են: Նրանցում առանձին նշանների պատկերը կառուցւում է, ինչպէս եւ տեսափակերում, համապատասխան կէտային մատրիցայում (9×9 , 9×12 , 12×24 եւ այլն), որի կէտերը ձեւաւորւում են յատուկ ներկող ժապաւէնի վրայով բարակ ձողիկների (0 , $2-0$, 3 միլիմետր տրամագծով): Հարուածների միջոցով՝ Զողիկների քանակը կարող է լինել 9 կամ 24 նրանց հարուածներից թղթի վրայ ստացուող կէտերը զործականում անընդհատ գծեր են կազմում: Տպող հանգոյցը, այսինքն՝ ձողիկները պարունակող զվիկը շարժում է հորիզոնական ուղղութեամբ եւ սովորաբար կարող է տպել ինչպէս դէպի ձախ, այնպէս էլ դէպի աջ շարժուելիս: Դա ապահովում է տպման բաւականին բարձր արագութիւն՝ մինչեւ 200 սիշ վայրկեանում: Այս ափսի սարքերը ունեն ոչ այնքան բարձր տպման որակ, սակայն նրանք տպում են սովորական թղթի վրայ եւ շատ ցածր գին ունեն: Բացի դրանից նրանց կորիւոր առակելութիւնը այն է, որ զվիկի հորիզոնական եւ թղթի ուղղահայեաց ուղղութիւններով շարժումը զեկավարուում է ծրագրային եղանակով: Դա հնարաւորութիւնն է տպիս հեշտութեամբ ստեղծել եւ օգտագործել տարրեր ափսի ապատառեր եւ, բացի զրանից, արտածել կամայական գրաֆիկական պատկերներ: Այս գասի որոշ տպող սարքերը պարունակում են հաստատուն յիշող սարքի մէջ զրանցուած մի քանի տպատառեր, իսկ ուրիշներում նախատեսում է յատուկ օպերատի յիշող սարք, որի մէջ ծրագրային եղանակով զրանցում է նախապէս պատրաստուած տպատառը:



Նկ. 46

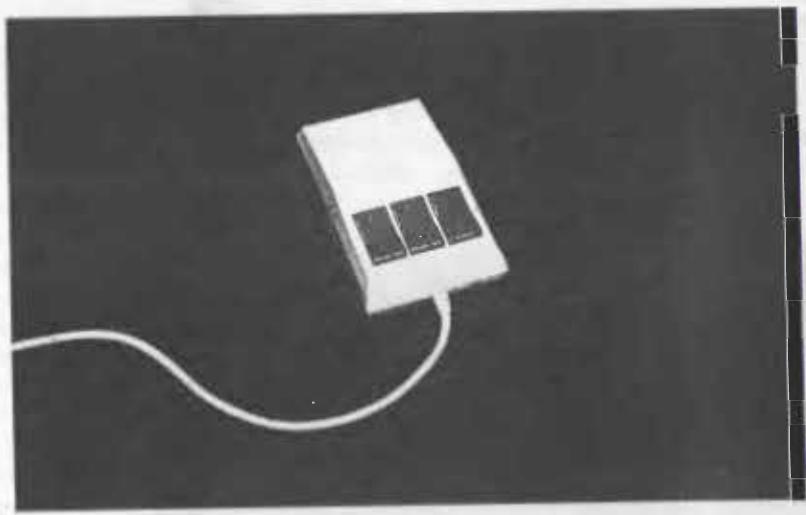
Մէկ այլ տհամակի տպող սարքերում, որոնք յաճախ թերթիկային են կոչւում, նշանները ձեւաւորում են յատուկ սկավառակի վրայ: Այս սարքերը ապահովում են շատ բարձր տպման որակ բայց ունեն ցածր տպման արագութիւն՝ մինչեւ 60 նիշ վայրկեանում: Բացի դրանից, որպէս թերթիւն, հարկ է նշել այն, որ նրանք դրաֆիկական պատկերներ տպելու հնարաւորութիւն չեն տալիս եւ կարող են օգտագործել միայն ստանդարտ տպատառեր: Զերծազրաֆիկական տպող սարքերը տարրերում են մատրիցայիններից նրանով, որ տպում են միայն յատուկ թղթի վրայ: Նրանցից մի քանիսում ջերմութիւնը ազդում է անմիջապէս թղթի վրայ, իսկ մէկ սպիշներում ներկող նիւթը հալեցում է եւ այնուհետեւ տեղափոխուում թղթի վրայ: Այս դասի տպող սարքերը յաճախ օգտագործում են դիւճակիր անհատական քոմիտերներում, քանի որ ունեն փոքր չափսեր: Վերջին ժամանակներս ուշաղրութիւն են գրաւում շիթային տպող սարքերը, որոնք անազմուկ են եւ ապահովում են տպման բաւականին բարձր որակ ու կէտերի տարրեր պայծառութիւն: Նրանց աշխատանքի սկզբունքը կայանում է նրանում, որ հորիզոնական ուղղութեամբ շարժուող ծայրափողակից թղթի վրայ են արձակուում թանաքի փոքրիկ կաթիլներ, որոնք էլ ձեւաւորում են արտածուող պատկերը: Այս դասի տպող սարքերը լայն տարածում չին ստացել, քանի որ յատուկ ծայրափողակը չաղբակալող թանաք են պահանջում եւ ունեն թղթի որակից զգալիօրէն կախուած տպման որակ:

Տպող սարքերի ասպարեգում վերջին նորոյթն են կազմում լազերային տպող սարքերը: Նրանցում լազերային ճառագայթը ծրագրի զեկավարման տակ ձեւաւորում է արտածուող պատկերը յատուկ թմրուկի վրայ, որից այնուհետեւ թղթի վրայ է տեղափոխուում ներկող նիւթը: Այս դասի սարքերը ապահովում են տպման շատ բարձր որակ եւ արագութիւն (մինչեւ 10 էջ րոպէում): Գրաֆիկական ուժիմուում նրանց տպման ճշգրտութիւնը մէկ միջիմետրում հասնում է մինչեւ 20 կէտի: Ինչպէս եւ մատրիցային սարքերում տուեալ գէպֆում հնարաւոր է հաստատուն կամ օպերատիւ յիշող սարքերի մէջ զգանցուած արբեր տպատառերի օգտագործումը: Լազերային տպող սարքերը ներկայում ունեն բաւականին մեծ չափսեր եւ զին: Բացի դրանցից բաւականին թանկ է գեւուելու փոխուող տպող հանգույցը, որը պարունակում է վերը նշուած թմբուկը եւ ներկող նիւթը: Լազերի կառուցուածքի եւ տպող հանգույցի օպտիկական համակարգի յետագայ զարգացումը կապահովի այս տիպի սարքերի չափսերի եւ զնի նուազումը, ինչպէս նաև նրանց յուսալիկութեան բարձրացումը:

Գոյոյնիւն ունեն դիտարկուած բոլոր սկզբունքներով աշխատող տպող սարքեր, որոնք ինֆորմացիան տարբեր գոյներով արտածելու հնարաւորութիւն են տալիս: Բոլոր տիպի մեներանդ եւ գունաւոր տպող սարքերը միացուում են քոմիտիթերին նոյն զեկավարող սարքի միջոցով եւ այդ պատճառով ի տարբերութիւն տեսատիպի անհատական քոմիտիթերի տպող սարքը շատ հեշտ կարելի է փոխարինել մէկ ուրիշով:

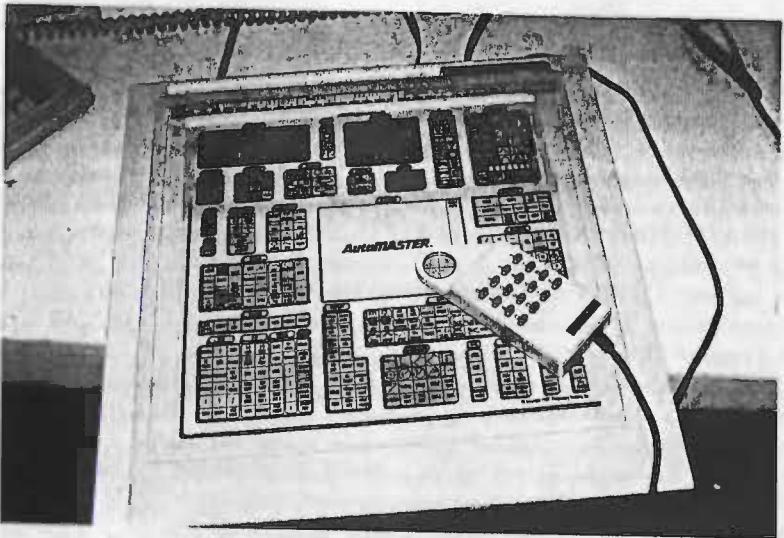
Բացի վերը նշուած ինֆորմացիայի ներածման եւ արտածման հիմնական սարքերից, անհատական քոմիտիթերներում որոշ դէպքերում, մասնաւորապէս զրաֆիկական պատկերների վերամշակման ընթացքում օգտագործուում են մի շարք յատուկ ներածման եւ արտածման սարքեր:

Յատուկ ներածման սարքերից առաջին հերթին հարկ է նշել այսպէս կոչուած մկնիկը (նկ. 47), որը իր ենից ներկայացնուում է երկու կամ երեք ստեղն ունեցող եւ հաղորդաբարերով քոմիտիթերի հետ միացուած մի փոքրիկ սալքը: Նրա մէջ պարունակուող փոքրիկ գնդիկը եւ յատուկ էլեկտրոնային սխեման հարթութեան վրայ,

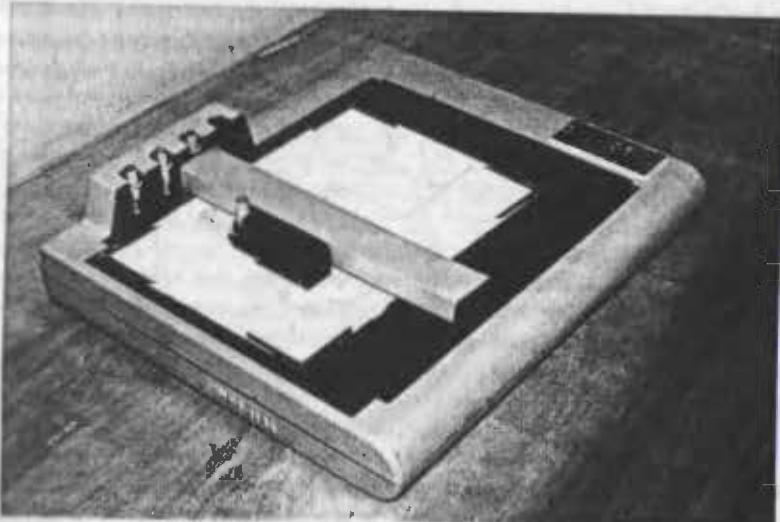


Նկ. 47

օրինակ սեղանի վրայ մկնիկի տեղաշարժը ձեւափոխում են էլեկտրոնական ազդանշանների եւ փոխանցում քոմիտիթերին: Յատուկ ծրագրի միջոցով այդ աղդանշանները մկնիկին համապատասխան տեսատիպի էլեկտրանի վրայ տեղաշարժուում են ցուցիչը: Մկնիկի վրայ աեղազրուած ասեղները օգտագործուում են համապատասխան ծրագրերուում գործողութիւնների թողարկման համար: Բնական է, եթէ մկնիկը տեղաշարժուուի այսպէս, որ գնդիկը չէպսիկ հարթութեան հետ, սպացուցիչը տեսատիպի վրայ չի շարժուի: Մկնիկը յարմար է օգտագործել զրաֆիկական ինֆորմացիայի վերամշակման եւ մի շարք այլ համակարգերուում, սովորյան նա չի կարող օգտագործուել պատկերների ներածման համար, քանի որ չի ապահովում համապատասխան ծշլլուսութիւնն: Այդ նպատակին են ծառայում յատուկ զրաֆիկական պլանշտիպութեան մէջ (նկ. 48): Նրա վրայ տեղազրուում է ներածուուղ պատկերը պարունակուութիւնը (նկ. 48): Նրա վրայ տեղազրուում է ներածուուղ պատկերը պարունակուութիւնը եւ այնուհետեւ օգտագործուուղ պատկերի վրայով տեղաշարժուում է յատուկ ցուցիչը եւ սեղմուում նրա վրայ գտնուուղ սուեղնը ուղղի գծերի հատման կէտերուում: Ընդ որում հատուածների ծայրերի կորողինաստները փոխանցուում են քոմիտիթերին՝ համապատասխանաբար կոդաւորելով ներածուուղ պատկերը: Գրաֆիկական պատկերները կարելի է աւելի արագ եւ ճշգրիս աւտոմատ կերպով ներածել տեսալիքի կամ յատուկ տեսախցիկի կամ յատուկ տեսածրուուղ սարքերի միջոցով: Գրաֆիկական ինֆորմացիան թղթի վրայ արտածելու համար համապատասխան տպող սարքերից բացի կարող են օգտագործուել կորակառուցիչ կոչուուղ սարքերը (նկ. 49): Այդ սարքերուում որպէս գծող հանգույց կիրառուում են յատուկ գրիչներ, որոնք գծերի շատ փոքր լայնութեան (0,025-0,25 միլիմետր) են ապահովուում: Ի հայլ գրան՝ կորակառուցիչ միջոցով ստացուած պատկերը սովորաբար աւելի ճշգրիտ է, քան տպող սարքի միջոցով ստացուածը, եւ այն հիմնականուում օգտագործուում է աւտոմատ նախագծման համակարգերուում գծագրական աշխատանքներ կատարելու նպատակով:



Նկ. 48



Նկ. 49

4. ՍԿԱՆԱՌԱԿԱՑԻՆ ՕՊԵՐԱՑԻՈՆ ՀԱՄԱԿԱՐԳ

4. 1. Օպերացիոն համակարգի կառուցուածքը

Օպերացիոն համակարգը երկու կարեւորագոյն հնարաւորութիւններ է սպառվում՝

- բոլոր տիպի ծրագրերի կատարման օժանդակում և սարքաւորման հետ մրանց համապրծակցութեան ապահովում.
- օգտագործողին հոմիլիթերի ընդհանուր դեկավարման հնարաւորութեան ընձեռնում:

Առաջին հնարաւորութիւնը ապահովում է ծրագրերի համադրծակցութիւնը արտաքին սարքերի և ուրիշ ծրագրերի հետ, օպերատոր յիշողութեան վերաբաշխումը, աշխատանքի ընթացքում առաջացող տարրեր իրավարձութիւնների յայտնաբերումը ու նրանց համապատասխան դործողութիւնների իրագործում եւ այլն: Երկրորդ հնարաւորութիւնը իրականացնում է հրամանների լեզուի հիման վրայ, որի միջոցով օգտագործողը կատարում է բոլոր անհրաժեշտ գործողութիւնները:

Երկրորդ սերնդի անհատական քոմփիթերներում ամենաառածուած MS-DOS (MicroSoft Disk Operating System - MicroSoft ֆիրմայի սկաւառակային օպերացիոն համակարգ) օպերացիոն համակարգը բաղկացած է չորս հիմնական մասերից՝

- յենակետային ներածման արտածման համակարգ.
- յենակետային ներածման արտածման համակարգի բնդլայնիչը.
- ընդհատումների վերամշակման համակարգ.
- հրամանների պրոցեսոր:

Ենակետային ներածման արտածման համակարգը պարունակում է արտաքին սարքերի զեկավարման և քոմփիթերի բոլոր հանդույնների սգուզման մի շարք յատուկ ծրագրեր: Այս համակարգը գտնուում է հասաւուում յիշող սարքի մէջ և ընդհանուր է առանց անհատական քոմփիթերի հետ կիրառուող բոլոր օպերացիոն համակարգերի համար: Նոր արտաքին սարքի զեկավարող ծրագիր աւելացնելու համար է նախառառուած յենակետային ներածման արտածման համակարգի ընդլայնիչը, որը գրանցուում է օպերացիոն համակարգը պարունակող (համակարգային) սկաւառակի վրայ: Համակարգային կարող է լինել որեւէ ճկուն կարծր մագնիսական սկաւառակ: Հանդույնների ստուգման աւարտից յետոյ քոմփիթերը աւտոմատ կերպով դիմում է ճկուն սկաւառակին կամ նրա բացակայութեան դէսքում կարծր սկաւառակին՝ օպերացիոն համակարգի միւս մասերը օպերատոր յիշողութեան մէջ բեռնաւորելու նպատակով: Ընդհատումների վերամշակման համակարգը բաղկացած է

մի չորք ծրագրերից, որոնք հնարաւորութիւն են տալիս կիրառական ծրագրերում օգտուել օպերացիոն համակարգի հնարաւորութիւններից, մասնաւորապէս զիմել ոկաւառակներին եւ միւս արտաքին սարքերին: Վերջապէս՝ հրամանների պրոցեսորը ապահովում է օգտագործողի երկխօսութիւնը օպերացիոն համակարգի հետ եւ կիրառական ծրագրերի թողարկումը: Այն նոյնպէս ապահովում է կիրառական ծրագրերի աշխատանքի ընթացքում հանդիպող սխալների վերլուծութիւնը եւ նրանց աւարտից յիսոյ վերադարձ օպերացիոն համակարգին:

Քոմիտիթերի սարքերը նշելու համար օպերացիոն համակարգում օգտագործում են յատուկ անուններ, որոնք աւարտում են երկու կէտով (): Դրանք են՝

A:	B:	C:	Եւ այլն	մագնիսական սկաւառակների վրայ յիշող սարքեր.
PRN կամ LPT1			-	հիմնական տպող սարք.
LPT2, LPT3			-	լրացուցիչ տպող սարքեր.
AUX կամ COM1			-	ինֆորմացիայի փոխանակման հիմնական սարք.
COM2			-	ինֆորմացիայի փոխանակման լրացուցիչ սարք.
CON			-	բարձակ, այսինքն արտածման ժամանակ տեսատիպ, իսկ ներածման՝ ստեղմաշար.
NUL			-	գոյութիւն չունեցող վերացական սարք.

Սովորաբար A: եւ B: անուններով նշանակուում են ճկուն սկաւառակների վրայ յիշող սարքերը, իսկ C:, D: եւ այլ անուններով՝ կարծր սկաւառակների վրայ յիշող սարքերը: Քոմիտիթերի կազմի մէջ աւելացուած լրացուցիչ սարքերին իրենց ուրոյն անուններն են նշանակուում համապատասխան ղեկավարող ծրագրերում:

Ֆայլերը, որոնք նախատեսուած են սկաւառակների վրայ տարբեր բնոյթի ինֆորմացիա գրանցելու համար, նոյնպէս նշուում են յատուկ անուններով: Ֆայլի անունը կարող է բաղկացած լինել ոչ աւել, քան 11 նիշից, որոնց թւում կարող են լինել յատինական տառերը, թուանշանները եւ. մի շարք այլ նշաններ (, , (,),), —, ”, ‘): Ֆայլի անունը կէտի (.) միջոցով բաժանուում է երկու մասի, որոնցից մինչեւ 3 նիշ պարունակող աջ մասը սովորաբար ցոյց է տալիս ֆայլի տիպը: Ֆայլերի անունների օրինակներ են՝ COMMAND, COM, REPORT, TEXT, A36, եւ այլն: Ընդհանուր առմամբ ֆայլի անունը պէտք է պարունակի առնուալը մէկ նիշ եւ նրա տիպը կարող է չնշուել: Զնայած նրան, որ հմտորէն ընտրուած ֆայլերի անունները օգտագործուին կարող են օգնել պահանջուուղ ինֆորմացիան, սկաւառակի վրայ գրանցուած ֆայլերի մեծ քանակութեան դէպքում որոշակի դժուարութիւն է առաջանուում: Սկաւառակի վրայ ինֆորմացիայի տեղաբաշխման մէջ որոշակի կարգ մտցնելու հնարաւորութիւն են տալիս ֆայլերի գրացուցակները: Գրացուցակը մէկ կրիչի վրայ գրանցուած տրամաբանօրէն շաղկապուած ֆայլերի մի խումբ է: Այն կարող է դիտարկուել որպէս արտաքին յիշողութեան որոշակի մաս, որի հետ կարելի է աշխատել բաւականաչափ անկախ: Գրացուցակը որոշակի ինֆորմացիա է պարունակում եւ այն որպէս ֆայլ կարելի է գրանցել մէկ այլ գրացուցակի մէջ: Այսպիսով՝ ստացուում են ներդրուած գրացուցակներ: Սկաւառակի վրայ կարող է ստեղծուել մէկ հիմնական կամ արմատային գրացուցակ, որի մէջ կարող են իրենց հերթին ստեղծուել ենթակայ գրացուցակներ: Վերջիններիս մէջ կարող են ստեղծուել նրանց ենթակայ գրացուցակներ եւ այլն: Համակարգի համար իւրաքան-

չիւր արտաքին յիշող նմաքի գրացուցակներից մէկը ընթացիկ է եւ այն կարող է համապատասխան հրամանի միջոցով փոխուել օգտագործողի կողմից:

Ներդրուած գրացուցակներ պարունակող սկաւառակների օգտագործման դէպքում Փայլը միարժէք որոշելու համար բաւական չէ նշել նրա անունը, պէտք է նշու նաեւ նրա գտնուելու տեղը՝ յաջորդականօրէն ենթակայ գրացուցակների շղթայի ձեւով: Այդ շղթան, որը ուղի է կոչուում, կազմուում է թեր գծով () բաժանուած յաջորդական գրացուցակների անուններից: Ուղիների օրինակներն են՝ \TEXT\BOOK, \DOS, \SERVIS\FONT\ARM եւ այլն: Այդ առումով Փայլը ամրողովին տրուում է որոշակի անուանազրուք, որը կարող է պարունակել համապատասխան յիշող սարքի անունը, օրինակ՝ C:\DOS\3_3\FORMAT.COM, A:\TEXT\HELP, 1 եւ այլն: Անուանագրում կարող է բացակայել յիշող սարքի անունը կամ ուղին, ընդ որում առաջին դէպքում ենթացրում է ընթացիկ սարքը, իսկ երկրորդում՝ ընթացիկ գրացուցակը: Որոշ հրամաններում միաժամանակ մի քանի Փայլեր նշելու նպատակով օգտագործուում են ? եւ * նշաններ պարունակող տիպարներ: Առաջին նշանը (?) տիպարում նշում է տուեալ զիրքում կամայական նշանի (նոյն թւում եւ բացատի) առկայութիւնը, իսկ երկրորդը (*) կամայական քանակութեամբ նշաններ: Օրինակ AR*, * տիպարը նշում է AR տառերով սկսուող բոլոր Փայլերը, *:EXE* բոլոր EXE տիպի Փայլերը, ???.TXT* բոլոր միատառ կամ երկտառ անուններով TXT տիպի Փայլերը եւ այլն:

Եթե օպերացիոն համակարգը պատրաստ է օգտագործողի կողմից տրուած հրամանները կատարելու, տեսատիպի վրայ արտածուում է համապատասխան հրաւերքը, որը պարզագոյն դէպքում նշում է ընթացիկ սարքի անունը: Օրինակ C> հրաւերքը նշում է C: սարքը, A> A: սարքը եւ այլն: Համապատասխան հրաւերքի բացակայութիւնը նշանակում է, որ կատարուում է նախորդ հրամանը կամ որեւէ կիրառական ծրագիր: Այդ դէպքում հնարաւոր յիշող սարքին համար հաջողութեան մէջ կարող է ստեղծաչարի միջոցով ներածել պահանջուուղ հրամանը: Օպերացիոն համակարգի հրամանների մի մասը կատարուում է անմիջապէս հրամանների պրոցեսուրի կողմից, որը գտնուում է քորմիկերի օպերատիւ յիշողութեան մէջ: Դրանք կոչուում են ներքին հրամաններ եւ զիմում են սկաւառակներին միայն ինֆորմացիայի վրանցման կամ ընթերցման անհրաժեշտութեան դէպքում: Արտաքին հրամանների իւրաքանչւելը կատարուում է օպերացիոն համակարգը պարունակող սկաւառակի վրայ գրանցուած յատուկ ծրագրի միջոցով: Այսինքն՝ այլ հրամանների կատարման համար, որը ոչնչով չի տարրերուում միւս բոլոր ծրագրերի կատարուակի անդամագիր, անհրաժեշտ է, որ օպերացիոն համապատասխան սարքի մէջ:

Համակարգի կողմից արտածուող հրաւերքը կարելի է փոխել ներքին PROMPT հրամանի միջոցով, օրինակ՝

PROMPT Հրաման ներածենք

հրամանի կատարումից յետոյ կ'արտածուի

Հրաման ներածենք

հրաւերքը: Ընդհանուր առմամբ հրաւերքի երկարութիւնը կարող է հասնել 100 նշանի: PROMPT հրամանը առանց որեւէ լրացուցիչ տեքստի վերականգնում է

Համակարգի սովորական պարզագոյն հրաւերքը: Այս հրամանը հնարաւորութիւն է սամիս համակարգի հրաւերքի մէջ բնդգրկել որոշ օգտակար ինֆորմացիայ ֆ նշանից յետոյ զրելով որոշակի յատուկ նշաններ, որոնք համապատասխան ինֆորմացիա են տեղադրում հրաւերքի մէջ: Անդ տառերի նշանակութիւնը հետեւեալն է՝

B	-	Աշան
D	-	ընթացիկ ամսաթիւ
E	-	դեկավորող ESC նշան
G	-	> Աշան
H	-	քայլ հետ' վերջին Աշանը ջնջելով
L	-	< Աշան
N	-	ընթացիկ յիշող սարքի անուն
P	-	ընթացիկ գրացուցակի անուն
Q	-	= Աշան
T	-	ընթացիկ ժամանակ
V	-	օպերացիոն համակարգի տարբերակի անուն
-	-	անցում յաջորդ տողին

Օրինակ՝ եթէ 1990 թուականի Յունուարի 22ի ժամը 10:30ին ընթացիկ դարձնելով
C: սարքի DOS գրացուցակը ներածել

PROMPT Ամսաթիւ \$D , ժամ \$Q\$T\$ _ \$P

Հրամանը, ապա համակարգի կ'արտածի

Ամսաթիւ = երկ 22-01-1990 , ժամ = 10:30:12.43

C:\DOS

Հրաւերքը: Քանի որ ընթացիկ գրացուցակի անունը պարունակում է նաև ընթացիկ սարքի անունը տուեալ հրամանում՝՝ Պատուի հետ N տառի օգտագործումը աւելորդ է:

Անհատական քոմփիթերի կողմի մէջ նախատեսուած է ժամացոյց, որը սովորաբար սննում է յատուկ կուտակիչից եւ այդ պատճառով շարունակում է իր աշխատանքը անգամ մեքենան անշատելուց յետոյ: Համակարգը այդ ժամացոյցից է րնկերցում ընթացիկ ամսաթիւը եւ ժամը: Որոշ գէպքերում անհրաժեշտ է լինում հրաւերքի ժամացոյցում զրանցուած ամսաթիւը կամ ժամը, ինչի համար նախատեսուած են համապատասխանաբար DATE եւ TIME ներքին հրամանները:

4. Աշխատանք սկաւառակների հետ

Անհատական քոմփիթերի աշխատանքի ընթացքում իւրաքանչիւր պահին արտադրի յիշող սարքերից մէջը ընկալուում է օպերացիոն համակարգի կողմից որպէս ընթացիկ սարք: Ընդ որում՝ եթէ սկաւառակնի զիմող որեւէ հրամանում չի նշում սարքի անունը, ապա ենթադրում է ընթացիկ սարքը: Այդ պատճառով յաճախ

յարմար կամ անզամ անհրաժեշտ է լինում փոխել ընթացիկ սարքի անունը, որի համար բաւական է համակարգի հրաւերքը ստանալուց յետոյ ներածել պահանջուող սարքի անունը: Օրինակ՝

C>A:

A>

տուեալ դէպքում համակարգի պարզագոյն հրաւերքը ցոյց էր աալիս, որ ընթացիկ է C: սարքի եւ ներածելով A: սարքի անունը, մենք այն զարձեցինք լինթացիկ սարք: Մեծ ինֆորմացիոն ծաւալ ունեցող կարգը մազնիսական սկաւառակները յատուկ ծրագրերի միջոցով կարելի բաժանել մի քանի մասերի, որոնցից իւրաքանչիւրը օպերացիոն համակարգի կողմից ընկալում է որպէս առանձին սարք եւ նրան համապատասխան յաջորդական անուն է տրւում:

Նոր ծկուն սկաւառակը կամ կարծր սկաւառակի մասը օգտագործելուց առաջ պէտք է յատուկ կերպով ձեւաչափի արտաքին FORMAT հրամանի միջոցով: Այդ հրամանը բաժանում է սկաւառակի մակերեսին սեկտոր կոչուող առանձին հատուածների եւ լրացնում զրանք բացատի կողերով: Սսվորաբար մէկ սեկտորի ծաւալը կազմում է 512 բայտ եւ, կախուած յիշող սարքի անսակից, օպերացիոն համակարգը մէկ գործողութեան լինթացքում զրանցում կամ լինթեցքում է 1, 2 կամ 4 սեկտոր միաժամանակ: Բացի զրանից այս հրամանը ստուգում է սկաւառակի մակերեսը, զտնում շարքից զորու եկած սեկտորները եւ յատուկ ձեւով նշելով՝ արդիում յետազայ զիմումը նրանց: Գործողութեան աւարտից յետոյ համակարգը յայտնում է օգտագործողին սկաւառակի ընդհանուր ծաւալը եւ օգտագործողին մատչելի ծաւալը: FORMAT հրամանում, որպէս պարամետր կարող է նշուել այն սարքի անունը, որի վրայ պէտք է կատարուի սկաւառակի ձեւաչափումը: Զակառակ դէպքում՝ ձեւաչափումը կատարուում է ընթացիկ սարքի վրայ: Որոշ լրացուցիչ պարամետրեր՝ հնարաւորութիւն են տալիս յատուկ ձեւով կատարել ձեւաչափումը: Օրինակ \S պարամետրը հնարաւորութիւն է տալիս նոր սկաւառակի վրայ ձեւաչափման հետ մէկտեղ զրանցել օպերացիոն համակարգի հրմանական մասը, որպէսզի յետազայում այդ սկաւառակը հնարաւոր յինի օգտագործել համակարգի թողարկման համար, /1 պարամետրը՝ ձեւաչափի միայն սկաւառակի մէկ երեսը եւ այն: Ձեւաչափման ընթացքում սկաւառակի վրայ ստեղծուում է հիմնական լրացուցակը եւ այն կարելի է օգտագործել Փայլերի զրանցման կամ հնթակայ զրացուցակներ ստեղծելու համար:

Սպանիսական սկաւառակը միւսներից տարրերիւ նպաստակով արտաքին LABEL հրամանի միջոցով նրան կարելի է վելազլել մինչեւ 11 նշանից բաղկացած վելնացիքը: Օրինակ՝

LABEL A: GAMES

A: սարքում տեղադրուած սկաւառակին վիրագրում է GAMES վելնացիքը: Սկաւառակների բազմութեանից պահանջուող զտնելու նպաստակով նրանց վերնազրերը կարելի է ընթերցել ներքին VOL հրամանի միջոցով՝ ի հարեւէ յաջորդաբար տեղադրելով սկաւառակները համապատասխան սարքի մէջ:

Սկաւառակի վրայ համապատասխան զրացուցակի մէջ գտնուած Փայլերի եւ ենթակայ զրացուցակների անունների ցոյցուկը տարածելու է ներքին DIR հրամանի միջոցով: Օրինակ՝

DIR C:\SERVIS\FONT

Հրամանի միջոցով կարող է արտածւել նկ. 50ում ներկայացուած ինֆորմացիան: Տուեալ դէպում գրացուցակում գրանցուած են 6 սովորական ֆայլեր եւ 2 ենթակայ զրացուցակ, ընդ որում վերջիններս յատուկ <DIR> նշանակումն ունեն: Բացի դրանից կայ նաեւ երկու առանձնայատուկ տող՝

.	<DIR>	11-01-89	14:20
..	<DIR>	11-01-89	14:20

Այդ տողերը համապատասխանաբար խորհրդանշում են տուեալ գրացուցակը ու նրանց մէկ աստիճան բարձր գտնուող գրացուցակը եւ ցոյց են տալիս, որ արտածուածը ենթակայ զրացուցակ է: Գրացուցակում բացի ֆայլերի լիով անուններից, տրամածուում է նրանց ծաւալը բայտերով եւ գրանցման ամսաթիւն ու ժամը: Վերջին տողում արտածուում է տուեալ զրացուցակում զրանցուած ֆայլերի եւ զրացուցակների քանակը եւ ընդհանրապէս սարքի վրայ ազատ մնացած ինֆորմացիոն ծաւալը: Անհամար համապատասխանաբար նեթայի մէկ ընդգրիելով ֆայլի անունի որոշակի տիպար՝ կարելի է ստանալ միայն նրան համապատասխանող ֆայլերի ցուցակը: Օրինակ՝

DIR C:\SERVIS\FONT A*. *

Հրամանը կ'արտածի միայն A տառով սկսուող ֆայլերը, ինչպէս ներկայացուած է նկ. 51ում: Եթէ DIR հրամանում բացակայում են սարքի անունը կամ ուղին, ապա համապատասխանաբար ենթադրում է ընթացիկ սարքը կամ ընթացիկ զրացուցակը:

Ընթացիկ զրացուցակում ներքին MKDIR (կամ կրճատ ձեւով MD) հրամանի միջոցով կարելի է ենթակայ զրացուցակ ստեղծել: Հրամանում պէտք է նշուի նոր զրացուցակի անունը, որի կառուցուածքը համարժէք է ֆայլի անունի կառուցուածքին: Օրինակ՝

MD LIBRARY

Հրամանը ստեղծում է LIBRARY անունով մի զրացուցակ, որում գեռեւս ֆայլեր դրանցուած չեն: Նոր զրացուցակը սկաւառակի վրայ սովորաբար 4 կիլոբայտ անունում, որտեղ յետագյում գրանցուում է նրա մէջ տեղադրուող ֆայլերի մասին համապատասխան ինֆորմացիան:

Ֆայլեր չպարունակող զրացուցակը կարելի է հեռացնել RMDIR (կամ RD) հրամանի միջոցով: Օրինակ՝

RD LIBRARY

Հեռացնում է LIBRARY անունով զրացուցակը: Հարկ է նշել, որ հնարաւոր չէ հեռացնել հիմնական, ընթացիկ եւ ֆայլեր պարունակող զրացուցակները:

Սկաւառակի վրայ գտնուող զրացուցակներից որեւէ մէկը կարելի է ընթացիկ դարձնել CHDIR (կամ CD) հրամանի միջոցով: Օրինակ՝

CD \SERVIS\FONT\ARM

Ինթացիկ է զարձնում ARM զրացուցակը: Նոյն գործողութիւնը կարելի է կատարել նաեւ հետեւեալ երեք հրամանների միջոցով՝

Սարք - C: Սկաւառակի վերնագիր - 1 HARD DISK
Դրացուցակ C:\SERVIS\FONT

	<DIR>	11-01-89	14:20	
..	<DIR>	11-01-89	14:20	
FONTEdit	EXE	70996	14-01-89	14:27
ARM	BAT	112	23-07-89	10:55
HELP	TXT	1953	22-07-89	13:23
EDITOR	COM	58075	3-04-88	17:16
PAINTER	EXE	125451	20-06-87	16:42
TEST		26	22-07-89	9:15
ARM	<DIR>		23-07-89	10:42
FONTS	<DIR>		15-01-89	18:56
8 Ֆայլ	Ազատ է	62036992	բայտ	

Նկ. 50

Սարք - C: Սկաւառակի վերնագիր - 1 HARD DISK
Դրացուցակ C:\SERVIS\FONT

ARM	BAT	112	23-07-89	10:55
ARM	<DIR>		23-07-89	10:42
2 Ֆայլ	Ազատ	62036992	բայտ	

Նկ. 51

CD SERVIS

CD FONT

CD ARM

Մէկ աստիճան վեր, տուեալ դէպքում FONT գրացուցակին անցնելու համար կարելի է օգտուել:

CD ..

Հրամանից, իսկ

CD \

Հրամանը ընթացիկ է դարձնում հիմանական գրացուցակը:

Օպերացիոն համակարգի հիմնական ներքին հրամաններից է COPY հրամանը, որը հնարաւորութիւն է տալիս ֆայլերը փոխանցել տարրերի սարքերի, մասնաւորապէս մագնիսական սկաւառակների վրայ յիշող սարքերի միջեւ, այսինքն՝ պատճէնաւորում կատարել: Պատճէնաւորման համար անհրաժեշտ է նշել ինֆորմացիայի աղբյուրը, այսինքն՝ պատճէնաւորուող ֆայլի անուանազրումը, և ընդունիչը, այսինքն՝ այն ֆայլի անուանազրումը, որի մէջ պէտք է տեղադրուի աղբյուրի պարունակութիւնը: Բնական է, որ այդ երկու անուանազրերը չեն կարող համընկնել և կարող են ախաղար պարունակել: Անուանազրի մէջ սարքի անուան կամ ուղղութափայութեան դէպքում ենթադրում է ընթացիկ սարքը կամ գրացուցակը: Ընդունիչը անուանազրի մէջ ֆայլի անուան բացակայութիւնը նշանակում է, որ ֆայլի պատճէնը կ'ունենայ բնագրի անունը, տիպարի օգտագործումը՝ որ պատճէնի անունը կը պարունակի բանագրի անուան համապատասխան մասերը: Օրինակ՝

COPY A: B;

Հրամանը A: սարքի բոլոր ֆայլերի նոյն անուններով կրկնում է B: սարքի վրայ,

COPY FONTEDIT. EXE TYPE*. *

Հրամանը ընթացիկ գրացուցակի FONTEDIT. EXE ֆայլը կրկնում է նոյն սարքի վրայ TYPEEDIT. EXE անունով,

COPY A*. * C:\TEXT\B*. *

Հրամանի ընթացիկ գրացուցակի A տառով սկսուող բոլոր ֆայլերը կրկնում է C: սարքի TEXT գրացուցակի մէջ՝ սկսելով ֆայլերի անունները B տառով:

COPY Հրամանի միջոցով հնարաւոր է ֆայլերի միաւորում կատարել, օրինակ՝

COPRE TEXT. 1+TEXT. 2 TEXT

Հրամանը ընթացիկ գրացուցակի TEXT. 1 և TEXT. 2 ֆայլերը միաւորում է եւ դրանցում TEXT ֆայլի մէջ,

COPY T*. * TEXT

Հրամանը ընթացիկ գրացուցակի T տառով սկսուող բոլոր ֆայլերը միաւորում է եւ դրանցում FILEXT ֆայլի մէջ, ընդ որում ֆայլերը միաւորում են գրացուցակի հերթականութեամբ:

Որպէս աղբյուր կամ ընդունիչ, COPY Հրամանի մէջ ֆայլի անուանազրի փոխարժեալ կարող են օգտագործուել քոմիքիթերի տարրեր սարքերի անուններ: Ընդ որում

աղբյուր կարող են լինել ինֆորմացիայի ներածման, իսկ ընդունիչ՝ արտածման սարքերը: Օրինակ՝

COPY TEXT. 1 PRN

Հրամանը արտածում է ընթացիկ գրացուցակի TEXT. 1 ֆայլը տպող սարքի վրայ,

COPY CON TEXT. 3

Հրամանը ընթացիկ գրացուցակի TEXT. 3 ֆայլի մէջ գրանցում է ստեղնաշարի միջոցով ներածման ինֆորմացիան: Վերջին դէպքում ներածուող ինքառանչիւր տողը պէտք է աւարտուի ներածման ստեղնի միջոցով, իսկ ներածման աւարտը նշուում է դեկավարող ստեղնի եւ: Զ ստեղնի միաժամանակ սեղմումով: Այնուհետեւ ներածման ստեղնի սեղմելուց յետոյ ներածուած ամբողջ ինֆորմացիան գրանցում է սկաւառակի վրայ:

Սկաւառակի վրայ գրանցուած ֆայլերը կարող են վերանուանուել REN ներքին Հրամանի միջոցով, որին ունի COPY Հրամանին համարժէք կառուցուածք: Օրինակ՝

REN FONTEDIT. EXE TYPE*. *

Հրամանը ընթացիկ գրացուցակի FONTEDIT. EXE ֆայլին վերագրում է TYPEEDIT. EXE անունը:

Գործածութիւնից դուրս եկած ֆայլերի հեռացման համար նախատեսուած է DEL ներքին Հրամանը, որի մէջ նոյնպէս կարող է նշուել տիպար պարունակող ֆայլի անուանազրիը: Օրինակ՝

DEL A*. *

Հրամանը ընթացիկ գրացուցակից հեռացնում է բոլոր A տառով սկսուող ֆայլերը:

Ճկուն մաղնիսական սկաւառակի վրայ գրուած ինֆորմացիան կարող է ամբողջվին պատճէնաւորուել DISKCOPY արտաքին Հրամանի միջոցով: Ընդ որում պատճէնաւորուում է նաև սկաւառակի ձեւաչափումը, այնպէս որ այս Հրամանի միջոցով նոր սկաւառակի վրայ պատճէնաւորում կատարելիս անհրաժեշտ չէ այն նախազէն ձեւաչափի: Հարք է նշել, որ DISKCOPY Հրամանի միջոցով հնարաւոր չէ պատճէնաւորել կարծը մաղնիսական սկաւառակները: Այս Հրամանում սովորաբար նշուում են ինֆորմացիայի աղբյուրը եւ ընդունիչը, օրինակ՝

DISKCOPY A: B:

Հրամանը A: սարքում աեղազբուած սկաւառակի վրայ պարունակուուղ ինֆորմացիան ամբողջովին կրկնում է B: սարքում աեղազբուած սկաւառակի վրայ: Պատճէնաւորումը կարող է կատարուել նաև որպէս աղբյուր եւ ընդունիչ օգտագործելով նոյն սարքը, օրինակ՝

DISKCOPY A: A:

Տուեալ դէպքում քոմիքիթերի օգիրատիւ. յիշող սարքի ծաւալից կախուած համարժը օգտագործութիւն հոմեզպատասխան յուշումների ձեւով մի քանի անգամ առաջիկում է A: սարքի վրայ աեղազբուի բնագրերը կամ պատճէնը պարունակող սկաւառակը: Նոյն քոմիքութիւնը ընթացիկ սարքի վրայ իրագործում է սարքերի անուններ չպարունակուից, օրինակ՝

DISKCOPY

Հրամանիր:

4. 3. Ծրագրերի թողարկում

Ֆայլերի կարեւորագոյն դաս են կազմում մեքենայական կոդով, այսինքն՝ միկրոպրոցեսորի հրամանների յաջորդականութեան ձեւով ծրագրեր պարունակող ֆայլերը: Ի տարբերութիւն բարձր մակարդակի ծրագրաւորման լեզուով, օրինակ րէյսիկով, սկաւառակի վրայ գրանցուած ծրագրերի, մեքենայական կոդով գրանցուած ծրագիրը կարող է անմիջականօրէն բեռնաւորուել քոմֆիւթերի յիշողութեան մէջ եւ թողարկել: Այդ պատճառով յաճախ մեքենայական կոդով գրանցուած ծրագրերը կոչում են բեռնաւորուող կամ թողարկուող ծրագրեր: Բարձր մակարդակի ծրագրաւորման լեզուով գրանցուած ծրագրերից յատուկ թարգմանող ծրագրերի միջոցով ստեղծուում եւ սկաւառականների վրայ մեքենայական կոդով գրանցուում են թողարկող ծրագրերը: Այդպիսի ծրագրեր պարունակող ֆայլերը ունեն .COM կամ .EXE տիպերը, որոնք իրարից տարբերուում են միայն արտաքին յիշող սարքերից ուկերատիւ յիշողութեան մէջ բեռնաւորելու եւ վերջինիս հասցէների վերաբաշման սկզբունքներով: Օգոտագործողի համար այդ տարբերութիւնը չօշափելի չէ, եւ երկու սիմպի ծրագրերն էլ նոյն եղանակով են թողարկուում: Ծրագրի թողարկման համար անհրաժեշտ է համակարգի հրաւերքի առկայութեան դէպքում ներածել նրա անունը տուանց սիմպի: Ընդ որում այդ անունով ֆայլը պէտք է դտնուի ընթացիկ գրացուցակի մէջ: Օրինակ՝ եթէ ընթացիկը FONT գրացուցակն է, ապա տեքստերի խմբագրման ծրագիրը կարելի է թողարկել ներածելով

EDITOR

անունը, իսկ պատկերների վերամշակման ծրագիրը՝ ներածելով

PAINTER

անունը: Որոշ դէպքերում ծրագրերի աշխատանքի համար պահանջւում է ֆայլի անունից յետոյ ներածել լրացուցիչ յատկանիչներ, որոնք կարող են բաղկացած լինել կամայական նշաններից: Յաճախ որպէս յատկանիչ օգտագործուում են նոյն կամ մէկ այլ գրացուցակում գտնուող ֆայլերի անունները: Օրինակ՝ եթէ պէտք է խմբագրուի HELP.TXT ֆայլի մէջ պարունակուող տեքստը, ապա ծրագիրը կարող է թողարկուել

EDITOR HELP.TXT

Հրամանի միջոցով: Այսպիսով՝ կիրառական ծրագրերի թողարկումը ոչնչով չի տարբերուում օպերացիոն համակարգի հրամանների կատարումից:

Ուրիշ (ոչ ընթացիկ) գրացուցակի մէջ ընդգրկուած ծրագիրը կարելի է թողարկել՝ նշելով նրա ամբողջ անուանագիրը: Օրինակ, անկախ նրանից, թէ որ գրացուցակն է ընթացիկ, պատկերների վերամշակման ծրագիրը կարելի է թողարկել

C:\SERVIS\FONT\PAINTER

Հրամանի միջոցով:

Ծրագրերի թողարկման համար պարտադիր չէ նշել նրա ամբողջ անուանումը կամ ընթացիկ դարձնել այն պարունակող գրացուցակը: Եթէ պահանջուող ֆայլը ընթացիկ գրացուցակի մէջ բացակայում է, ապա այն յաջորդաբար որոնուում է PATH հրամանի մէջ թուարկուած ուղիներով: Այդ հրամանի մէջ կարող են գրուել մի

շարք միմեանցից կէտ ստորակէտով (;) բաժանուած սարքի անուններ կամ ուղիներ, օրինակ՝

PATH C:\;C:\DOS;C:\SERVIS;C:\SERVIS\FONT

Հրամանի կատարումից յետոյ համակարգը յիշում է նրա մէջ ընդգրկուած ուղիների յաջորդականութիւնը եւ օգտագործում այն ամէն անգամ, երբ օգտագործողը օպերացիոն համակարգի արտաքին հրաման է կատարում կամ ծրագիր է թողարկում: Բերուած հրամանը նշանակում է, որ եթէ նշուած արտաքին հրամանին կամ ծրագրին համապատասխանող ֆայլը ընթացիկ գրացուցակում չի գտնուում ապա այն որոնուում է սկզբում C: սարքի հիմնական գրացուցակում, այնուեւեալ DOS անունով գրացուցակում եւ այլն: Տուեալ դէպում անկախ նրանից, թէ որ գրացուցակն է ընթացիկ, պատկերների վերամշակման ծրագիրը կարելի է թողարկել ներածելով միայն ծրագիր անունը՝

PAINTER

նշանակուած որոնման ուղիները վերացնել կարելի է

PATH;

Հրամանի միջոցով:

Թողարկուող ֆայլերի առանձնայատուկ դաս են կազմում այսպէս կոչուած փաթեթային ֆայլերը, որոնք միշտ .BAT տիպն ունեն: Նկ. 50ում ներկայացուած ֆայլերից միայն մէկն է (ARM, BAT) այդ դասին պատկանում: Փաթեթային ֆայլերը, ի տարրերութիւն մեքենայական կոդով գրանցուած ծրագրերի, սովորական տեքստ են պարունակում, որում ամէն տողի վրայ գրուած է օպերացիոն համակարգի հրաման կամ ծրագրի անուն եւ թողարկուում նոյն եղանակով, ինչպէս միւս ծրագրերը: Այդ առումով փաթեթային ֆայլերը կարող են ստեղծուել, խմբագրուել եւ գրանցուել սկաւառակի վրայ տեքստերի վերամշակման ծրագրերի միջոցով: Օպերացիոն համակարգը սկսում է իր աշխատանքը սկաւառակի վրայ յատուկ AUTOEXEC.BAT ֆայլի որոնումից եւ եթէ գտնում է այն, ապա անցնուում է նրա մէջ թուարկուած հրամանների կատարմանը: Դա բաւական յարմար է, քանի որ հնարաւորութիւն է տալիս քոմփիւթերի նախնական թողարկման ժամանակ առանց օգտագործողի մասնակցութեան ոչ միայն համապատասխան սարքերը նախապատրաստել, այլև թողարկել նշուած ծրագիրը: Օրինակ, եթէ օպերացիոն համակարգը պարունակող սկաւառակի վրայ գրանցուած է նաեւ տեքստերի վերամշակման EDITOR ծրագիրը եւ

EDITOR

տեքստը պարունակող AUTOEXEC.BAT ֆայլը, ապա քոմփիւթերի թողարկումից յետոյ համակարգը անմիջապէս կ'աշխատանքնի տեքստերի վերամշակման ծրագիրը:

Օպերացիոն համակարգի մէջ մտնում են մի չափ ներփին հրամաններ, որոնք հիմնականում օգտագործուում են փաթեթային ֆայլերում՝ ECHO, PAUSE, REM եւ այլն: Սովորաբար փաթեթային ֆայլի կատարման ժամանակ տեսատիպի վրայ յաջորդաբար արտածեում են նրա մէջ մտնող բոլոր հրամանները: Եթէ ֆայլի մէջ ընդգրկուի

ECHO OFF

Հրամանը, ապա նրան յաջորդող հրամանները չեն արտածուի, իսկ

ECHO ON

Հրամանից յետոյ արտածումը կը վերականգուի: Ֆայլի մէջ աւելցրած

PAUSE

Հրամանից յետոյ յաջորդ հրամանը կը կատարուի միայն օգտագործողի կողմից որեւէ ստեղն սեղմելուց յետոյ: Այս հրամանում կարող է ընդունել լրացուցիչ Հաջորդագրութիւն, որը նրա կատարման ժամանակ կ'արտածուի տեսատիպի վրայ: Օրինակ՝

PAUSE Այխապատրաստէք տպող սարքը

Հաջորդագրութիւնը չ'արտածուիր, եթէ ֆայլի մէջ տուեալ հրամանին նախորդել է ECHO OFF հրամանը: REM հրամանը նախատեսուած է փաթեթային ֆայլի կատարման ընթացքում տարբեր տիպի հաջորդագրութիւններ տեսափակի վրայ արտածելու համար եւ ի տարբերութիւն PAUSE հրամանի՝ չի ընդհատում ֆայլի կատարումը: Օրինակ՝

REM Ենրածում է տպատառը

Բացի վերը բերուած օրինակներից փաթեթային ֆայլերը կարող են պարունակել ուրիշ յատուկ հրամաններ, որոնք հնարաւորութիւն են տալիս կատարման պայմանական, կրկնուող եւ այլ գործողութիւններ:

5. ԱԼԳՈՐԻԹՄՆԵՐ

5. 1. Ալգորիթմի հասկացութիւնը – Ալգորիթմի ներկայացման եղանակներ

Դուք գիտէ՞ք ինչպէս են պատրաստում խորովածը: Փորձենք նկարագրել այդ հաճելի արարողութիւնը:

1. Վերցմել միս, սխ, բացախ, պղպեղ եւ աղ:
2. Խորատել միսը եւ սոխը:
3. Խառնել մէկ ամամի մէջ միսը եւ սոխը:
4. Աւելացնել աղը, պղպեղը եւ բացախը:
5. Թողմել ստացուածը 3-4 ժամով:
6. Շարել միսը շամփուրների վրայ:
7. Վառել կրակը:
8. Սպասել մինչեւ կրակը նստի:
9. Դնել շարուած միսը կրակի վրայ:
10. Հետեւել կրակին:
11. Եթէ միսը եփուել է, կատարել 12րդ հրահանգը՝ հակառակ դէպքում՝ 10րդը:
12. Մատուցել խորովածը:

Մենք ներկայացրել ենք ցուցում, որը բաղկացած է առանձին յատակ հրամաններից: Այդ հրամանների յաջորդական կատարումը, յոյսով ենք, կը բերի նպատակին՝ խորովածի պատրաստմանը: Նկատենք որ՝

1. Բոլոր հրամաններում առաջին հերթին նշուած է կատարուող գործողութիւնը (վերցնել, կտրատել, քողնել եւ այլն) եւ յետոյ այն առարկաները, որոնց նկատմամբ կատարում է գործողութիւնը (միսը, սոխը, կրակը . . .).

2. Բացակայում են հրամանների բացատրութիւնները՝ օրինակ, ինչի՞ համար է պէտք աւելացնել պղպեղը եւ բացախը:

Համարենք որ այս գործողութիւններն իրագործող մարդու համար ամենեւին էլ պարտադիր չէ իմանալ իւրաքանչիւր հրամանի իմաստը: Նա պէտք է միայն հետեւի կարգադրադրին, չիսիսուելով՝ հրամանների յաջորդականութիւնը:

Ճշգրիտ կարգապահքիրը՝ կատարելու գործողութիւնների այնպիսի յաջորդականութիւններ, որն աւզուած է ցանկալի արդիւնքի ստացմանը կամ առաջադրուած խնդրի լուծմանը անուանում է ալգորիթմ:

Ալգորիթմը բառի պատամաթիւնը կապուած է IXրդ գարի ուզբէկ մաթեմատիկոս Ալ-Խորեզմի անուան հետ, որը ճեւակերպել է տասական համակարգում թուարանական գործողութիւնների կատարման կանոնները: Ալգորիթմի հասկացութիւնը

Դայն կիրառում է գտել մաթեմատիկայի, ֆիզիկայի, տնտեսագիտութեան եւ այլ բնագաւառներում:

Ալորեայ կեանքում մեր բոլոր գործողութիւնները նկարագրում ենք բառերով: Այդ նոյն խօսակցական լեզուն մենք կարող ենք օգտագործել ալգորիթմների նկարագրման ժամանակ, եթէ համարակալենք ներկայացուած գործողութիւնները: Աղորիթմների ներկայացման այդ եղանակը կարելի է անուանել բանաւոր շարադրութիւն քայլերի միջոցով:

Բերենք այդպիսի ալգորիթմի եւս մի օրինակ (դուք հաւանաբար կուահեցիք, որ խորոշած պատրաստելու ալգորիթմը կազմուած էր հենց այդ եղանակով): Ներկայացնենք գործողութիւնների յաջորդականութիւնը, որը պէտք է կատարել ճանապարհին անցնելու համար (նկ. 52):

Ինչպէս տեսնում ենք ալգորիթմ կազմում են ոչ միայն հաշուարկային խնդիրներ լուծելու համար: Իսկ ի՞նչ եղանակներ կան հաշուարկային խնդիրների ալգորիթմների ներկայացման համար:

Դիտարկենք օրինակներ:

Խնդիր.- Հերոնի բանաձեւով հաշուել ա, բ, ս կողմեր ունեցող եռանկեան մակերեսը:

1. ՆԵՐԱՄԵԼ ա, բ, ս ;
2. ՀԱՇՈՒԻԾԼ $p = (a+b+c)/2$;
3. ՀԱՇՈՒԻԾԼ $y = p(p-a)(p-b)(p-c)$;
4. ՀԱՇՈՒԻԾԼ $S = \sqrt{y}$
5. ԱՐՏԱՄԵԼ S ;
6. ՎԵՐՋ :

Ալգորիթմների ներկայացման այս եղանակը անուանենք բանաձեւերի յաջորդականութիւն:

Ալգորիթմների միասնական եւ ճշգրիտ գործուման համար ստեղծուած է նշանակումների եւ կանոնների ձեւայնացուած համակարգ, որը կոչում է ալգորիթմի լեզու:

Նկարագրենք հետեւալ խնդրի լուծման գործողութիւնների յաջորդականութիւնը ալգորիթմական լեզուով:

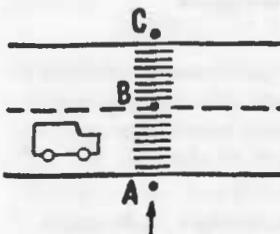
Խնդիր.- Կառուցել $1/(X-1) + 1/(X-3)$ արտայատութեան արժէքի հաշւման ալգորիթմը:

ԱԼԳ ԱՐԺԵՔ (ԻՐԿ x , y , ՏԵՔՍՏ p)

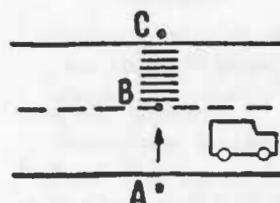
Ա.ՐԴ x

Ա.ՐԴ y, P

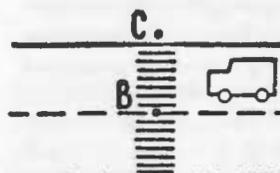
ՃԱՆԱՊԱՐՀՆ ԱՆՑՆԵԼՈՒ ԱԼԳՈՐԻԹՄ



- 1 ՔԱՅԼ. ԿԱՆԳՆԵԼ **A** ԿԵՏՈՒՄ ;
 2 ՔԱՅԼ. ՆԱՅԵԼ ԶԱԽ ;
 3 ՔԱՅԼ. ԵԹԵ ՄԵՔԵՆԱՑ Է ՄՈՏԵՆՈՒՄ,
ԿԱՏԱՐԵԼ 2 ՔԱՅԼԸ, ՀԱԿԱՌԱԿ
ԴԵՊԲՈՒՄ ԿԱՏԱՐԵԼ 4 ՔԱՅԼԸ ;

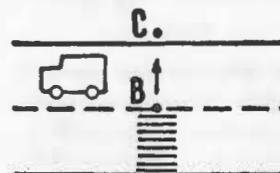


- 4 ՔԱՅԼ. ԳՆԱԼ ԴԵՊԻ **B** ԿԵՏԸ ;



- 5 ՔԱՅԼ. ԿԱՆԳՆԵԼ **B** ԿԵՏՈՒՄ ;

- 6 ՔԱՅԼ. ՆԱՅԵԼ ԱԶ ;



- 8 ՔԱՅԼ. ԳՆԱԼ ԴԵՊԻ **C** ԿԵՏԸ ;

?

- 9 ՔԱՅԼ. ՎԵՐՋ :

Եթէ $x=1$ ԿԱՄ $x=3$

ԱՊԱ R := "y-ի ԱՐԺԵՔԸ ՈՐԾՈՒԱԾ ԶԷ"

ԱՅԼԱՊԷՍ R := "y-ի ԱՐԺԵՔԸ ՈՐԾՈՒԱԾ Է"

$$y := 1/(x-1) + 1/(x-3)$$

ԱԻԱՐՏ

ԳԵՐՁ

Ինչպէս տեսնում են ալգորիթմական լիցուի բառակազմն ընդգրկում է բացի սովորական տեքստից նաև մաթեմատիկական պայմանանշաններ՝ հաստատուններ, փոփոխականներ, գործողութիւնների նշաններ, փակագծել եւ այլն:

Ալգորիթմաների նկարագրման յարմար եւ տեսանկի եղանակներից մէկն է գրաֆիկական ներկայացումը բուկ-սիմետրի միջոցով:

Առանձին բլոկներում, որոնք իրենցից ներկայացնում են երկրաչափական պատկերներ, նկարագրում են կատարուաց գործողութիւնները, իսկ կատարման հերթականութիւնը ցոյց է տրում պարագաների միջոցով (նկ. 53, 54):

Ցոյց տրուած օրինակի 13րդ բլոկում առկայ է որեւէ պայման՝ դիցուք $a > b$: Եթէ պայմանը բաւարարում է, այսինքն a -ն խորագիս մեծ է b -ից, ապա ալգորիթմը շարունակում է "այո" ճիշով իւ կատարում է 14րդ բլոկում նշուած գործողութիւնը, իսկ եթէ a ոչ՝ ապա 15րդը: Ինչպէս տեսնում ենք պայման ստուգող բուկը պէտք է ունենայ երկու իւ միայն երկու ելք՝ "այո" և "ոչ": Ի տարրերութիւն, օրինակ, հաշուղական գործողութիւնն պատկերող բլոկին, որն ունի միայն մէկ ելք:

Դիտարկուած հղանակներով կազմուած ալգորիթմանը նախատեսուած են իրագործուելու մարդու կողմից: Էլեկտրոնային հաշուիչ մեքենայի միջոցով ալգորիթմը իրացրուելու համար, այն պէտք է ներկայացնի որեւէ ծրագրաւորման լիցուով:

Մրացրաւորման լիցուն իրենից ներկայացնում է մեւական նշանային համակարգ, որը օգտագործում է մարդու իւ հաշուիչ մեքենայի միջեւ կապ ստեղծելու համար եւ նախատեսուած է տուեալների իւ ալգորիթմների նկարագրման ու մշակման համար:

Բերնք մի ծրագրի օրինակ, որը դրուած է բլուիկ լիցուով:

Կազմել $Y=1/(x-1) + 1/(x-3)$ արտայայտութիւն արժեքի հաշւման ծրագիրը բէյսիկ լիցուով:

10 INPUT X

20 IF X=1 THEN 70

30 IF X=3 THEN 70

40 LET Y=1/(x-1) + 1/(x-3)

50 PRINT "Y-ի արժեքը որոշուած է՝ Y=" ; Y

60 STOP

70 PRINT "Y-ի արժեքը որոշուած չէ"

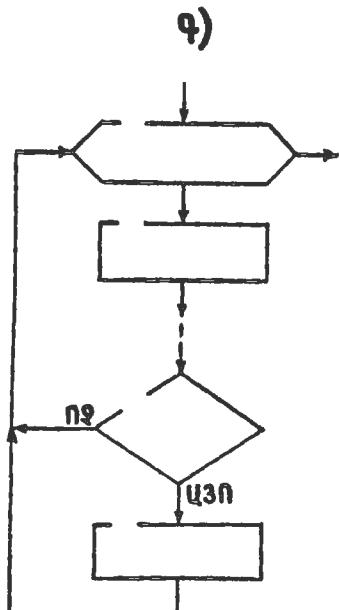
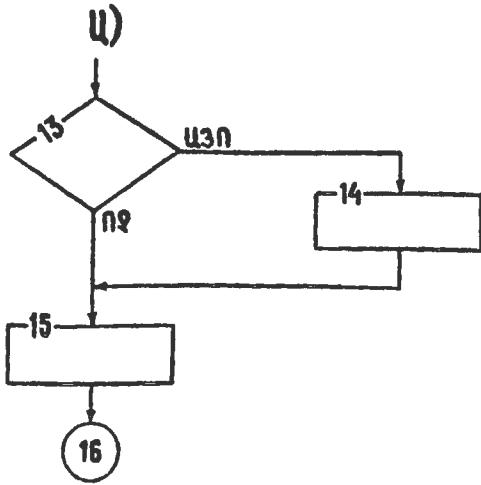
80 STOP

90 END

ԲԼՈԿ - ՍԽԵՄԱՅՈՒՄ ՕԳՏԱԳՈՐԾՈՒՈՂ ՀԻՄՆԱԿԱՆ ՊԱՏԿԵՐՆԵՐ

ԱՆՈՒԱՆՈՒՄ	ՆՇԱՆԱԿՈՒՄ	ԲԱՑԱՏՐՈՒԹԻՒՆ
1		ՀԱՅՈՒԱՐԿԱՆ ԳՈՐԾՈՂՈՒԹԻՒՆ
2		ՊԱՅՄԱՆԻ ՍՏՈՒԳՈՒՄ
3		ՑԻԿԼԻ ՍԿԻԶԲԵ
4. ԿԱՆԽՈՐՈՇՈՒՄՆԵՐ ՊՐՈՑԵՍ		ՀԱՅՈՒԱՐԿԱՆ ԵՆԹԱԾՐԱԳՐՈՎ
5. ՓԱՍՏԱԹՈՒՂԹ		ԱՐԴԻԽՆՔԻ ԱՐՏԱՆՅՈՒՄ
6. ՊԵՐՖՈՔՍՐ		ՏՈՒԵԱԼԵՐԻ ՆԵՐԱՆՅՈՒՄ ՊԵՐՖՈՔՍՐԻ ՄԻՋՈՑՈՎ
7. ԱՆՑՈՒՄ		ԳՆԵՐԻ ԸՆԴՀԱՏՈՒՄ
8. ԳՈՐԾԱՐԿՈՒՄ ԴԱԴԱՐԵՑՈՒՄ		ՍԿԻԶԲ, ՎԵՐՁ
9. ՄԵԿՆԱԲԱՆՈՒԹԻՒՆ		ԲԱՑԱՏՐՈՒԹԻՒՆ

ԲԼՈԿԵՐԻ ՄԻԱՅՄԱՆ ՁԵՒԵՐ



նկ. 54

Եւ այսպէս մենք նկարագրեցինք ալգորիթմի ներկայացման 5 եղանակ (նկ. 55): Յաջորդ պարագափաներում մենք կը դիտարկենք բազմաթիւ խնդիրների լուծման ալգորիթմները, որոնք կ'օգննեն մեզ հասկանալ ալգորիթմների կազման հիմնական մեթոդները եւ սկզբունքները: Ալգորիթմների ներկայացման համար մենք ընտրել ենք զրաֆիկական եղանակը (բլոկ-սխեմաներ), քանի որ զանում ենք, որ այն դարձնում է ալգորիթմը պատճեռաւոր եւ հչշտ ուրացուող:

5. 2. Ալգորիթմի յատկուրիւնները

Մենք արդէն դիտենք, թէ ինչ է իրենից ներկայացնում ալգորիթմը: Այժմ տեսնենք, թէ ինչ յատկութիւններ պէտք է այն ունենայ:

1. Դիսկրետուրիւն:

Ինչպէս դիտենք ալգորիթմի կատարումը բաժանում է առանձին գործողութիւնների-քայլերի: Յաջորդ գործողութեանը անցնելու համար իրազրծողը պարտաւոր է լրիւ աւարտել նախորդը:

2. Ճշգրտութիւն:

Ալգորիթմը պէտք է կազմուած լինի այնպէս, որ իրազրծողը միարժէքօրէն իմանայ, թէ որ քայլն է կատարելու յաջորդը:

3. Որոշակիութիւն:

Ալգորիթմը չպէտք է պարունակի այնպիսի գործողութիւններ, որոնք հնարաւոր չեն իրադրել: Այլ կերպ ասած ալգորիթմը պէտք է հասկանալի լինի:

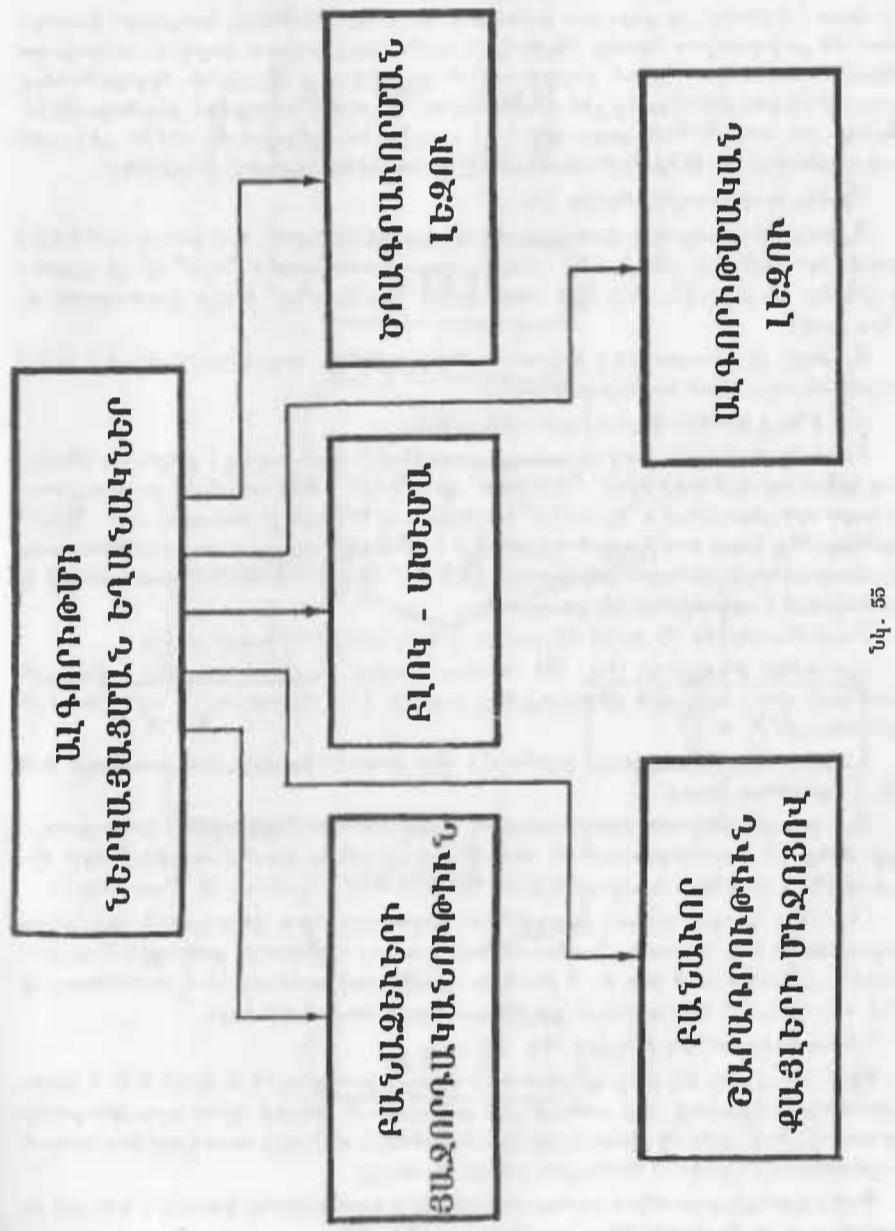
Եթէ անդրադառնանք մեր դիտարկած օրինակներին, ապա կը տեսնենք, որ խորոված չի նշուած օգտագործուող մարդ, սովոր եւ այլ մթերքների քանակը: Այդ թերութիւնը ալգորիթմը իրազրծողի համար դարձնում է անհասկանալի: Առաջարկում ենք լրացնել խորոված պատրաստելու ալգորիթմը՝ հաշուի առնելով նշուած թերութիւնները:

4. Արդիւնաւետութիւն:

Ալգորիթմի գործողութիւնների կատարումը սկզբնական տուեալների նկատմամբ պէտք է բերի որոշակի արդիւնքի: Նշենք, որ "խնդիրը լուծում չունի" պատասխանը համարւում է լիարժէք արդիւնք:

5. Մասսայականութիւն:

Այս յատկութիւնը ենթադրում է, որ նոյն ալգորիթմի միջոցով հնարաւոր է լուծել բազմաթիւ միատիպ խնդիրներ՝ նախնական տուեալների զանազան տարբերակների դէպքում:



Նկ. 55

6. Վերջաւորութիւն:

Ցանկացած ալգորիթմ պէտք է վերջաւոր քանակի գործողութիւնների կատարումից յիսոյ հասնի արդիւնքի:

Հարցեր եւ վարժութիւններ.

1. Զեւակերպեմք ալգորիթմի սահմանումը:
2. Բերէք ձեզ յայտնի ալգորիթմների օրինակներ:
3. Ի՞նչ եղանակով են ներկայացնուում ալգորիթմները:
4. Ներկայացրէք միջբաղամայն աւտոմատ հեռախոսով օգտուելու ալգորիթմը բանաւոր շարադրութեան եղանակով:
5. Կազմեմք սուրճ պատրաստելու ալգորիթմ:
6. Զեւակերպէք եւ զրի առէք ալգորիթմ, որը մախատեսուած է կարկինի եւ բանոնի օգնութեամբ.
ա) կիսել տրուած նաևուածը.
բ) բաժանել տրուած նաևուածը երեք հաւասար մասերի:
7. Նշենք ալգորիթմների յատկութիւնները:
8. Ի՞նչ է նշանակում ալգորիթմի որոշակիութիւնը:
9. Ճշգրիտ է արդեօֆ նախապարհին անցնելու ալգորիթմը:
Հիմնաւորէք ձեր պատրախամը:
10. Քանի՞ մուտք կարող է ունենալ.
ա) ուղղանկին բլոկը.
բ) շեղանկին բլոկը:

5. 3. Ալգորիթմների տիպերը

Ալգորիթմները լինում են 2 տիպի՝ դժային եւ ճիւղաւորուած:

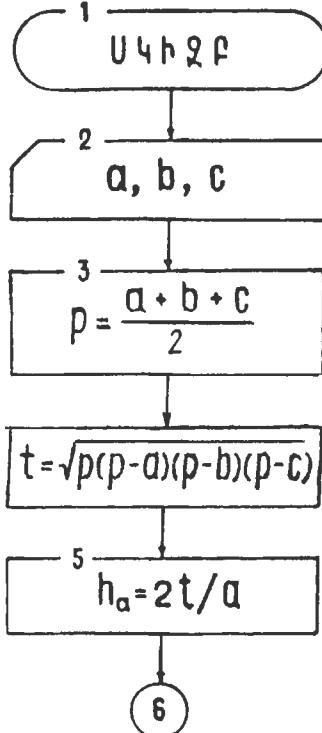
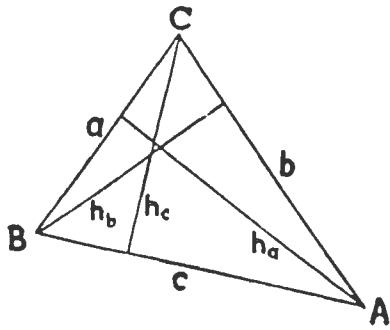
Ճիւղաւորուած ալգորիթմների զասում առանձին հետաքրքրութիւն են առաջնում ցիկլային ալգորիթմները, որոնք պարունակում են կրկնուղ գործողութիւններ:

Դիստրկենք մի օրինակ (նկ. 56):

Ինչպէս երեւում է ներկայացնուած բլոկ-սխեմայից Զրդ բլոկը ներածում է նախնական տուեաները՝ հսանկեան կոզմեմը ա, բ, ս: Ցրդ բլոկում կատարում է կիսապարագծի հաշուարկը: Նշենք, որ հաշուարկները կրծատելու նպատակով Հերոնի բանաձեւից սատցուած անդամը, որը նոյնն է բլոկը երեք բարձրութիւնների հաշուարկի բանաձեւերում, հաշուած ենք միայն մէկ անդամ եւ նշանակում է (4րդ

ԳԾԱՅԻՆ ԱԼԳՈՐԻԹՄ

ՀԱՇՈՒԵԼ a, b, c ԿՈՂՄԵՐ ՈՒՆԵՑՈՂ ԵՌԱՆԿԱՆ ԲԱՐՁՐՈՒԹԻՒՆՆԵՐԸ՝

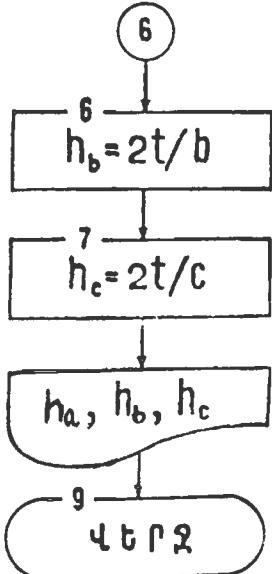


$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)}$$

$$h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)},$$

ՈՐՏԵՇ $p = \frac{a+b+c}{2}$



բլոկ): Տրդ բլոկի միջոցով տպում ենք բարձրութիւնների մեծութիւնները, որոնց հաշուարկը կատարուել է 5, 6, 7 քայլերում:

Հեշտ է նկատել, որ բերուած օրինակում զործողութիւնները (քայլերը) կատարուած են յաջորդաբար նրանց հերթական համարների աճման կարգով, առանց ետք վերադասնակու կամ որեւէ գործողութիւն չըջնցելու: Այդպիսի ալգորիթմները կոչում են գծային: Նրանք չեն պարունակում պայմանի ստուգման գործողութիւն: Ասենք, որ առաել հետաքրքրութիւն են առաջացնուած ծիւղաւորուած եւ ցիկլային ալգորիթմները եւ մենք հիմնականում կը դիտարկենք այդպիսի խնդիրներ:

Ակնենք պարզագոյն խնդիր (նկ. 57):

Այստեղ ծիւղաւորուած կատարուած է Յրդ բլոկի միջոցով, երբ ստուգում է $x \geq 0$ պայմանը: Եթէ այն տեղի ունի ($x \geq 0$), ապա կատարուած է "այո" ճիւղի գործողութիւնը (Յրդ բլոկ), իսկ եթէ տեղի չունի ($x \leq 0$)՝ "ոչ" ճիւղի գործողութիւնը (4րդ բլոկ):

Այսինքն ծիւղաւորուած է կոչուած այն ալգորիթմը, որը պարունակում է որեւէ պայմանի ստուգման գործողութիւն:

Իսկ ի՞նչ է իրենից ներկայացնուած պայմանը:

Պայմանը՝ տրամաբանական արտայայտութիւն է, որը կարող է ընդունել հետեւեալ երկու արժեքներից մէկը՝ "ճշմարիտ" կամ "կեղծ": Եթէ պայմանը բաւարարուած է, ապա այն ընդունուած է "ճշմարիտ" արժեքը, իսկ եթէ այն չի բաւարարուած՝ "կեղծ" արժեքը: Այլ կերպ ասած պայմանը կարելի է դիտարկել որպէս հարց, որի հնարաւոր պատասխաններն են "այո" (ճշմարիտ) կամ "ոչ" (կեղծ): Ըստ այդ պատասխանի էլ կատարուած է ալգորիթմի ծիւղաւորուածը:

Ուսումնասիրենք մի քանի ծիւղաւորուած ալգորիթմների օրինակներ:

Այս տիպի խնդիրներ (նկ. 58) լուծելու համար նպատակայարմաք է թուային առանցքի վրայ նշել այն միջակայքերը, որտեղ Y -ը հաջուքուած է այս կամ այն բանաձեւով:

$] -\infty, -2]$ միջակայքուած գործում է 2րդ բանաձեւը, $[-2, 2]^*$ առաջինը, իսկ $[2,]^*$ դարձեալ 2րդը:

Այս փոքրիկ վերլուծութիւնը զգալիօրէն պարզեցնուած է ալգորիթմի կառուցուածը, որը հանդում է համեմատասխան միջակայքի որոշմանը համեմատութիւնների միջոցով (3, 4 բլոկներ):

(նկ. 59) ներկայացնուած ալգորիթմում օգտագործելով վերադրման հրահանքը կարողանուած ենք կրճատել համեմատական գործողութիւնների քանակը: Ա փոփոխականով նշանակուած ենք A, B թուերից փոքրը, այնուհետեւ Ս-ն համեմատելով C-ի հետ որոշուած ենք ամենափոքր թիւը, որը եւ արտածուած ենք:

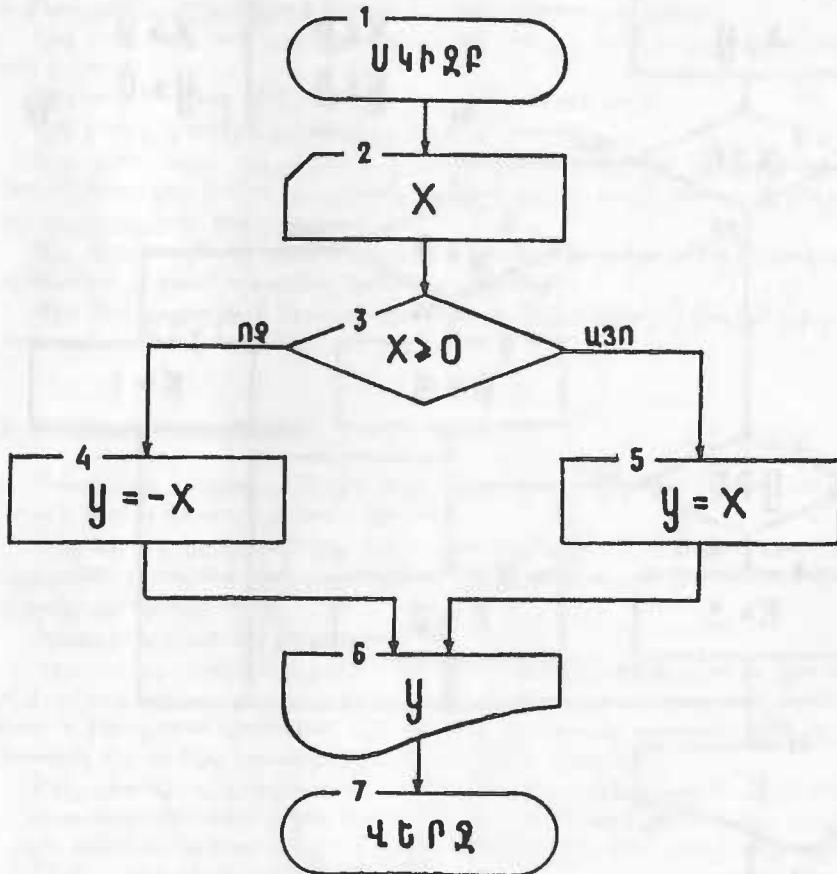
Դիտարկենք յաջորդ խնդիրը (նկ. 60):

Ակքում յիշենք թէ ի՞նչ պայմանների պէտք է բաւարարեն A կէտի X եւ Ս կորդի ի ինատները, որպէսզի այս գտնուի այն քառորդում: Նկարի վրայ այդ յիշեցուած արուած է: Իսկ այժմ մի քանի համեմատութիւնների միջոցով ստուգում ենք նշուած պայմանների եւ գտնուած համապատասխան քառորդը:

Յաջորդ խնդիրը լուծելու համար մենք պէտք է կարողանանք կառուցել գծային եւ քառակուսային ֆունկցիաների գրաֆիկները (նկ. 61):

ԲԱՑԱՐՁԱԿ ՄԵԾՈՒԹԵԱՆ ՀԱԾՈՒՅՄԱՆ ՕՐԻՆԱԿ՝ $y = |x|$

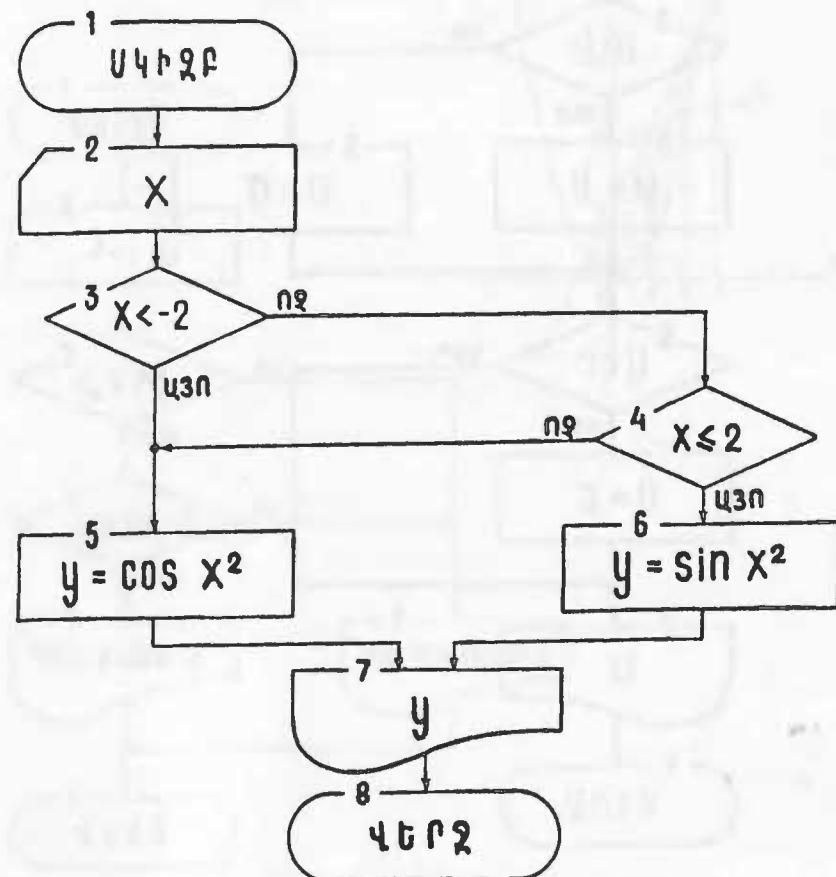
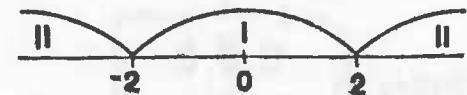
$$y = \begin{cases} x, & \text{եթե } x \geq 0, \\ -x, & \text{եթե } x < 0, \end{cases}$$



Խկ. 57

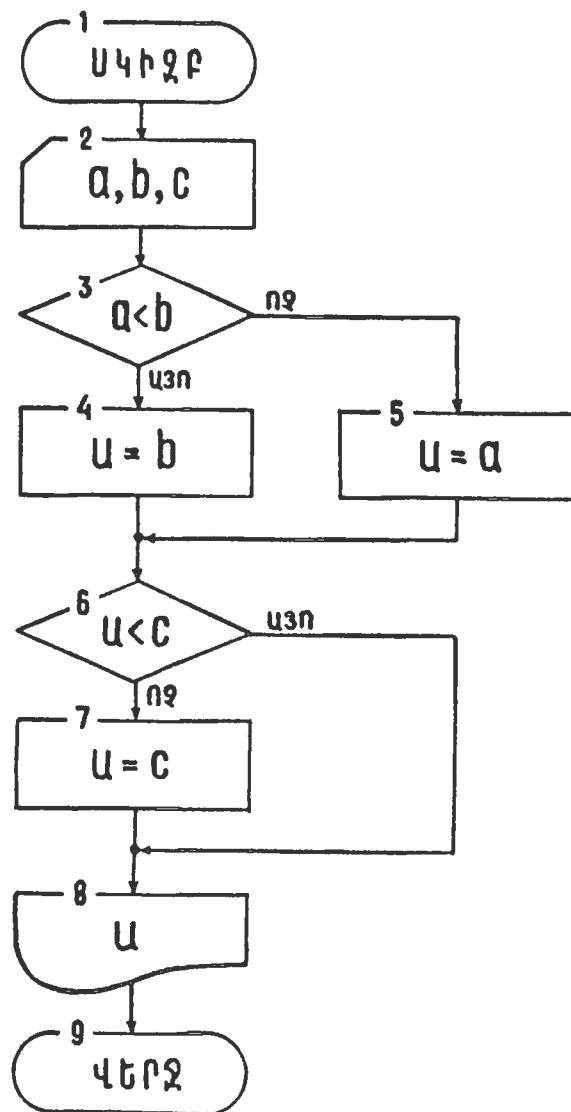
ՏՐՈՒԱԾ X-Ի ՀԱՄԱՐ ՀԱԾՈՒԵԼ ՍՊՈՒՆԿՑԻԱՅԻ ԱՐԺԷՔԸ

$$y = \begin{cases} \sin x^2, & \text{եթե } |x| \leq 2 \\ \cos x^2, & \text{ՀԱԿԱՊԱԿ ԴԵՊՔՈՒՄ} \end{cases}$$

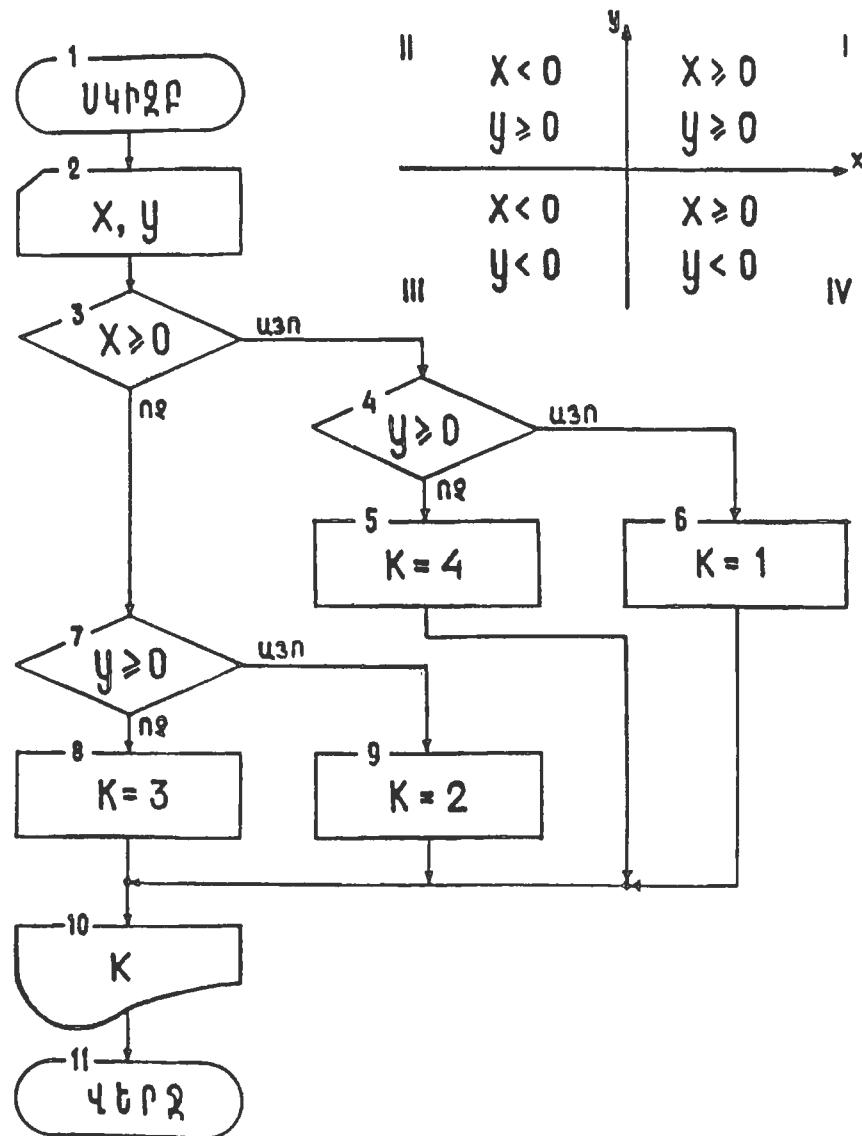


Խկ. 58

ԳՏՆԵԼ a, b, c ԹՈՒԵՐԻՑ ԱՄԵՆԱՓՈքՐԸ



ՈՐՈՇԵԼ ԱՅՆ ՔԱՌՈՐԴԻ ՀԱՄԱՐԸ, ՈՐՏԵղ ԳՏՆԻՌՄ Է $A=(x, y)$ ԿԵՏԸ:



Կառուցելով $y=x^2$ եւ $y=2x+2$ ֆունկցիաների զրաֆիկները առանձնացնում ենք տիրոյթը: Որպէսզի $A(x_0, y_0)$ կէտը գտնուի տիրոյթի ներսում, այն պէտք է լինի $y=2x+2$ ուղղից աջ եւ $y=x^2$ կորից վերեւ, այսինքն միաժամանակ պէտք է բաւարարուեն հետեւեալ երկու պայմանները՝

$$y \leqslant 2x+2 \quad \text{եւ} \quad y_0 \geqslant x_0^2$$

Սուսակելով նշուած պայմանները ալգորիթմը որոշում է պատկանո՞ւմ է արդեօք $A(x_0, y_0)$ կէտը տիրոյթին թէ ոչ:

Վերլուծենք եւս մի ալգորիթմ (*նկ. 62*):

Այս խնդրում ալգորիթմի արդիւնաւէտութիւնը եւ մասսայականութիւնը ապահովելու համար անհրաժեշտ է դիտարկել բոլոր հնարաւոր դէպքերը՝

Եթէ $a=0, b=0, c=0$, ապա ցանկացած թիւ հանդիսանում է արուած հաւասարման լուծում:

Եթէ $a=0, b=0, c \neq 0$, ապա հաւասարումը լուծում չունի:

Եթէ $a=0, b \neq 0$, ապա միակ լուծումն է՝ $x=-c/b$:

Եթէ $a \neq 0$, ապա հաշըրում է քառակուսի հաւասարման զիսկրիմինանտը՝ $D=b^2-4ac$, եթէ $D < 0$ հաւասարումը իրական արմատ չունի, հակառակ դէպրում հաւասարումը ունի երկու արմատ:

Այս վերլուծութիւնը ցոյց է տալիս, որ a, b, c փոփոխականների ցանկացած արժեքների դէպքում ալգորիթմը բերում է արդիւնքի:

Մեծ հետաքրքրութիւն են առաջացնում նաև երկարաչափական խնդրիների լուծման ալգորիթմները (*նկ. 63*):

5. 4. Ցիկլային ալգորիթմներ

Շատ յաճախ հանդիպում են այնպիսի խնդիրներ, որոնց լուծման համար անհրաժշտ է կրկնել միեւնոյն դործողութիւնները:

Ցիկլ ասելով հասկանում ենք հրամանների այնպիսի յաջորդականութիւնը, որը արդորիթմի կատարման ընթացքում կրկնում է մի քանի անգամ որեւէ փոփոխականի տարրեր արժեքների համար:

Դիտարկենք հետեւեալ խնդրիը (*նկ. 64*):

Բերուած ալգորիթմում բազմից կրկնում են 4, 5, 6 բլոկերի զործողութիւնները X փոփոխականի տարրեր արժեքների դէպքում: Յատուկ ուշադրութիւն դարձնենք $x=x+h$ վերագրման հրամանին, որի միջոցով է հնարաւոր դառնում ամէն անգամ փոփոխել x -ի արժեքը (տուեալ դէպքում աւելացնել ի քայլով):

Մենք գիտենք, որ ցանկացած ալգորիթմ պէտք է լինի վերջաւոր: Դա նշանակում է, որ գործողութիւնների կրկնութիւնը պէտք է սահմանափակ լինի: Տուեալ դէպքում արգիս սահմանափակում հանդէս է զալիս $x \leqslant b$ պայմանը (τ րդ բլոկ): Այդ արժեքը (b -ն) կոչում է ցիկլի սահման:

Երկնուող գործողութիւնների համախմբութիւնը կազմում է ցիկլի մարմինը: Քիրուած օրինակում դա 5 եւ 6 բլոկերն են:

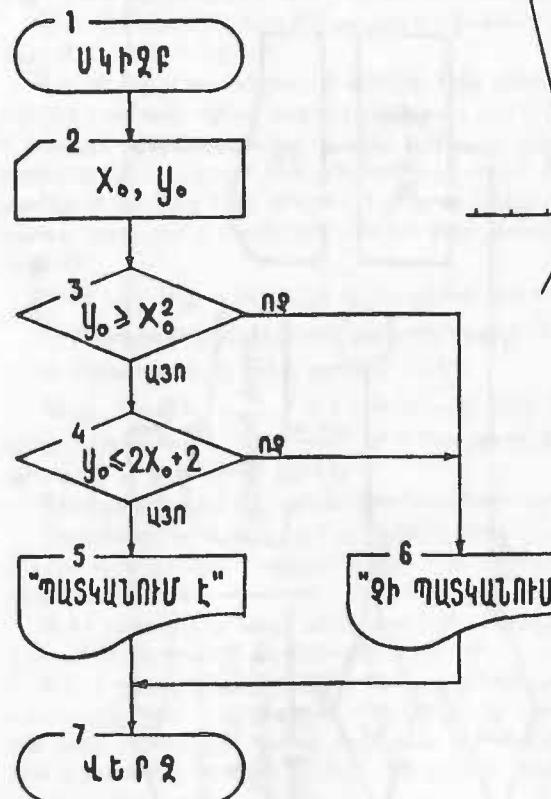
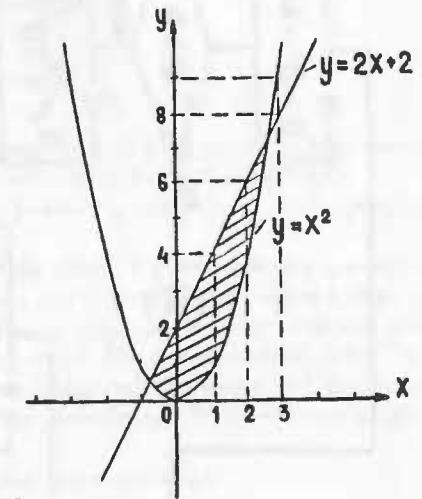
Փորձենք մատչելի լեզուով նկարագրել բերուած ցիկլով կատարուող գործողութիւնները:

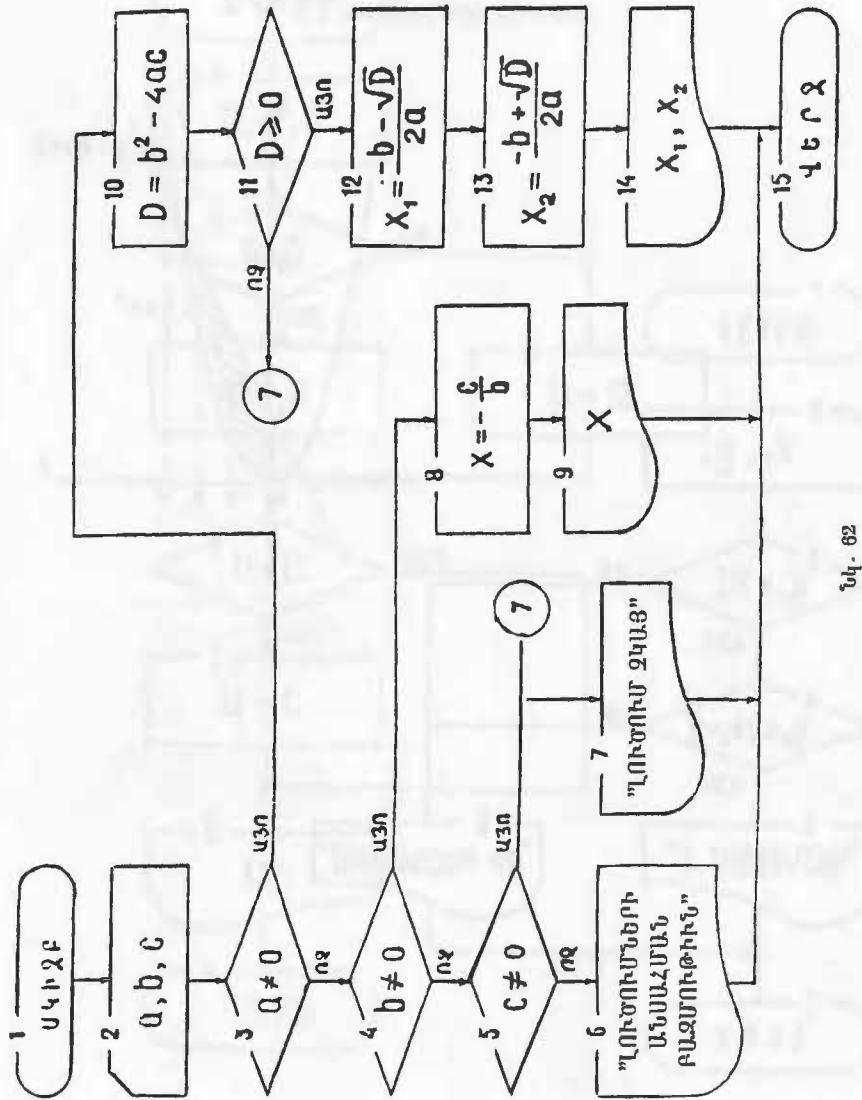
ՏԻՐՈՅԹԸ ՏՐՈՒԱԾ Է ՀԵՏԵՒԵԱԼ ՀԱՒԱՍՍՐՈՒՄՆԵՐՈՎ

$$y=x^2 \quad \text{եւ} \quad y=2x+2$$

ՈՐՈՇԵԼ, ՊԱՏԿԱՆՈՒՄ Է, ԱՐԴԵՕՔ, $A(x_0, y_0)$ ԿԵՏԸ

ՏՈՒԵԱԼ ՏԻՐՈՅԹԻՆ :





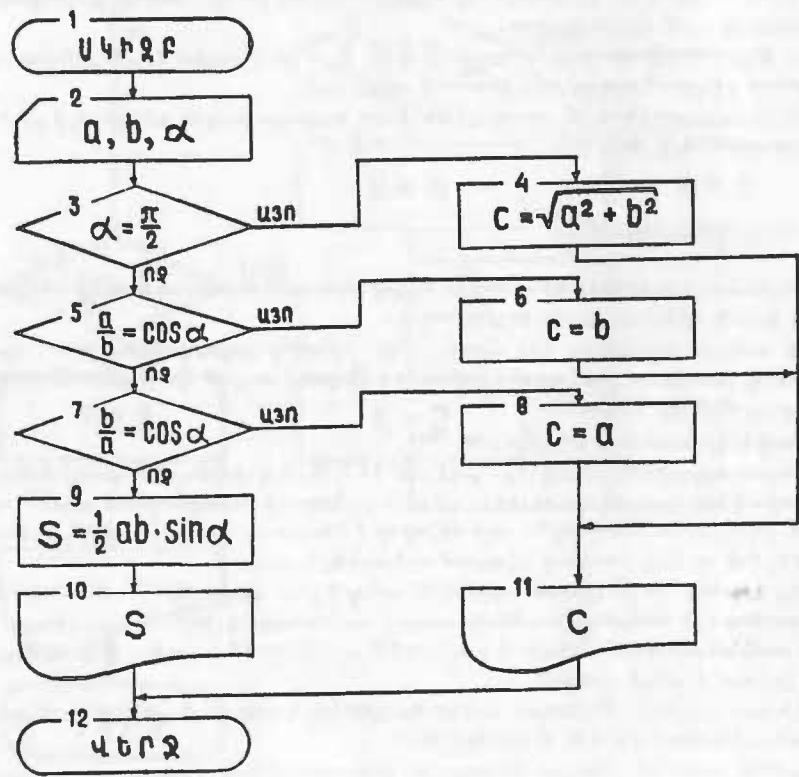
Նկ. 62

ՏՐՈՒԱԾ ԵՆ ԵՐԵՔ ԹԻՒ՝ a , b և c , ՈՐՏԵղ, $a \neq 0$ և $b \neq 0$ - ԵՌԱՆԿԵԱՆ
ԿՈՂՄԵՐՆ ԵՆ, ԻՍԿ α - ԻՐԵՆՑ ՄԻՋԵՒ ԵՂԱԾ ԱՆԿԻՒՆԸ:
ՀԱՅՈՒԵԼ՝ ԵԹԷ ԵՌԱՆԿԻՒՆԻՆ ՈՒՂՂՄԱՆԿԻՒՆ Է - ՆՐԱ ՆԵՐՔԵՍՁԻԳԸ
ԵԹԷ ՈՉ - ՄԱԿԵՐԵՍԸ :

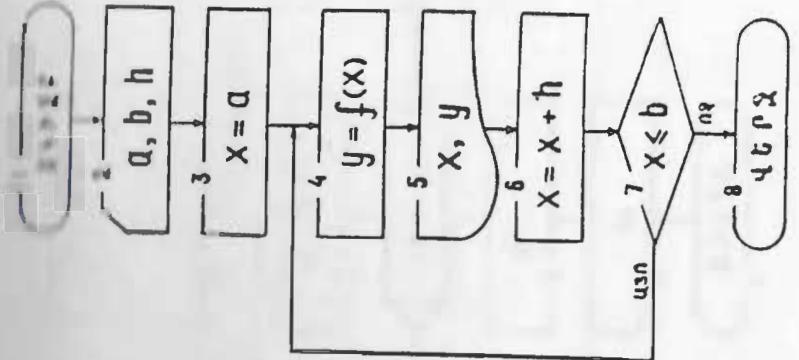
ԼՈՒՇՈՒՄ : ՀԱՅՐԱԿՈՐ Է ԵՐԵՔ ԴԵՊՔ

1. $\alpha = \frac{\pi}{2}$
2. $\frac{a}{b} = \cos \alpha$
3. $\frac{b}{a} = \cos \alpha$

$$S = \frac{1}{2} ab \cdot \sin \alpha$$



Նկ. 63



Հ ղեկավարող փոփոխականի իւրաքանչիւր արժեքի դէպքում (x -ը փոխում է ձևականին արժեքից մինչեւ Ե վերջնական արժեքը ի քայլով) կատարել հետեւեալ գործողութիւնները՝

1. Հաշուել $f(x)$ փունկցիայի արժեքը Ե վերագրել Ս փոփոխականին.

2. Արտածել x Ե Ս փոփոխականների արժեքները:

Իսկ ինչպէ՞ս հաշուել քանի անգամ է կատարում ցիկլի մարմինը։ Դրա համար կարելի է օդտագործել հետեւեալ պարզ բանաձիւր՝

$$n = [(b-a)/h] + 1$$

որտեղ $[y]$ -ով նշանակուած է Ս-ի ամբողջ մասը։

Լուծենք յաջորդ խնդիրը (նկ. 65):

Առաջարկում ենք ինքնուրոյն առանձնացնել այն գործողութիւնները, որոնք կազմում են ցիկլի մարմինը և հաշուել քանի անդամ են նրանք կրկնուում։

Հիմա տեսնենք թէ ինչպիսի ցիկլը են կազմում գումարներ եւ արտադրեալներ հաշուելու համար (նկ. 66):

Այս խնդրում պահանջուում է հաշուել 1-ից մինչեւ Ա ընտական թուերի քառակուսիների գումարը։ Ցիկլը կաղմակերպելուց առաջ Տ փոփոխականը պէտք է ընդունի 0 արժեքը, որպէսզի հնարքուր լինի կրկնուուղ դործուութիւնների միջացով կուտակի (y = y + 1²) նշուած մեծութիւնների գումարը։ Ցիկլի փոփոխականը 1-ն է, որի արժեքը փոփուում է 1-ից մինչեւ Ա 1-քայլով։ Ցիկլի մարմինը կազմում է 5 բղդ բլոկ, որտեղ հաշուելում է ճերթական բնական թուի քառակուսին եւ գումարուում ընդհանուրին։

Մենք ցոյց ենք տուել ցիկլի նկարագրման երկու եղանակ՝

1. Օգտագործելով պայմանի ստուգման բլոկը ($1 < n$).

2. Օգտագործելով ցիկլի բլոկը ($1 = I, n$):

Ցիկլի մարմինը կատարուում է Ո անգամ։ Երբ 1-ն ընդունում է $n+1$ արժեքը ցիկլը աւարտուում է Ե եւ կատարուում է Յրդ լրոկի գործողութիւնը (սպառում է արդիւնքը՝ Ս փոփոխականի արժեքը)։

Նման ցիկլ կարելի է կաղմակերպել նաև արտադրեալ հաշուելու համար (նկ. 67)։

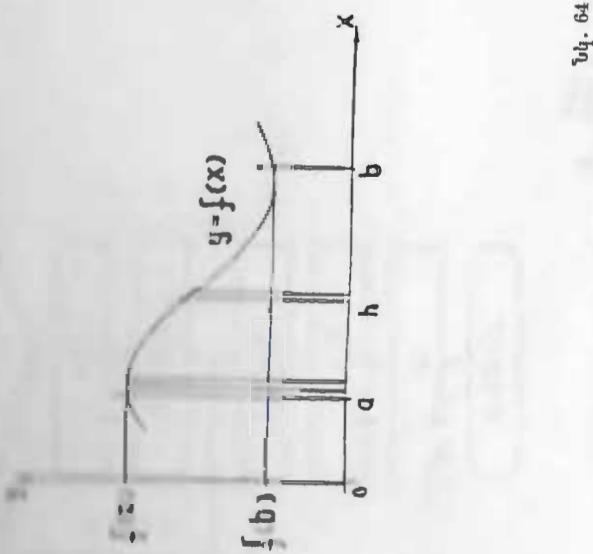
Ցիկլեցնենք, որ Ո Փակտուրեալը իւրենից ներկայացնուում է 3-ից մինչեւ Ա ընտական թուերի արտադրեալը։ Ի տարրերութիւննախորդ ալգորիթմի այսաեղ Ս փոփոխականի պղբնական արժեքը հաւասար է 1-ի, իսկ ցիկլի կազմակերպումը նոյնն է։

Այժմ դիտարկենք աւելի բարդ ալգորիթմ, որտեղ ցիկլի մարմինը պարունակուում է պայմանի ստուգման գործողութիւնը (նկ. 68)։

Տ եւ Կ փոփոխականներին վիելագրուում է 0 սկզբնական արժեքը Ե եւ կաղմակերպուում է ցիկլ ըստ Խ ղեկավարսղ փոփոխականի, որը փոխուում է ա-ից Ե ի քայլով։ Խ-ի ամէն մի արժեքի համար հաշուելում է $y = \sin \cos x$ Ե համեմատուում 0-ի հետ։ Եթէ Ս-ի արժեքը դրական է, ապա Տ-ը Ե Կ-ն ստանուում են համապատասխան աճ։

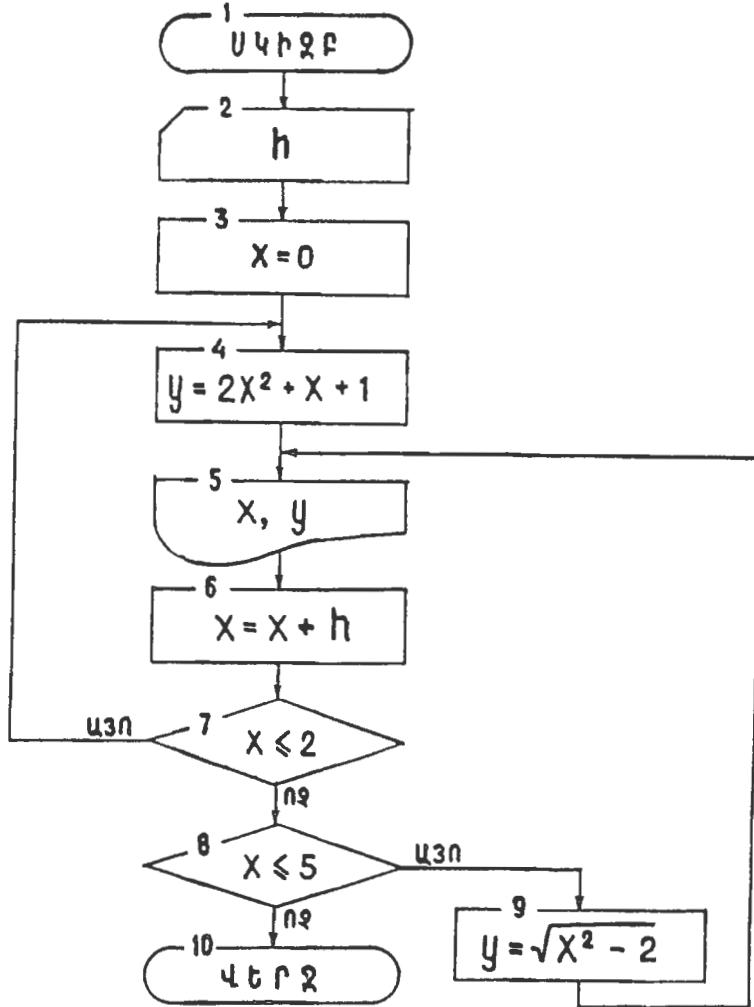
Դիտարկենք յաջորդ խնդիրը (նկ. 69)։

Նկատենք որ արդարիթմուում չի օգտագործուած աստիճան բարձրացման գործողութիւնը, որը փոփորինուած է ցիկլի մէջ արտադրեալով (Յրդ բլոկ)։



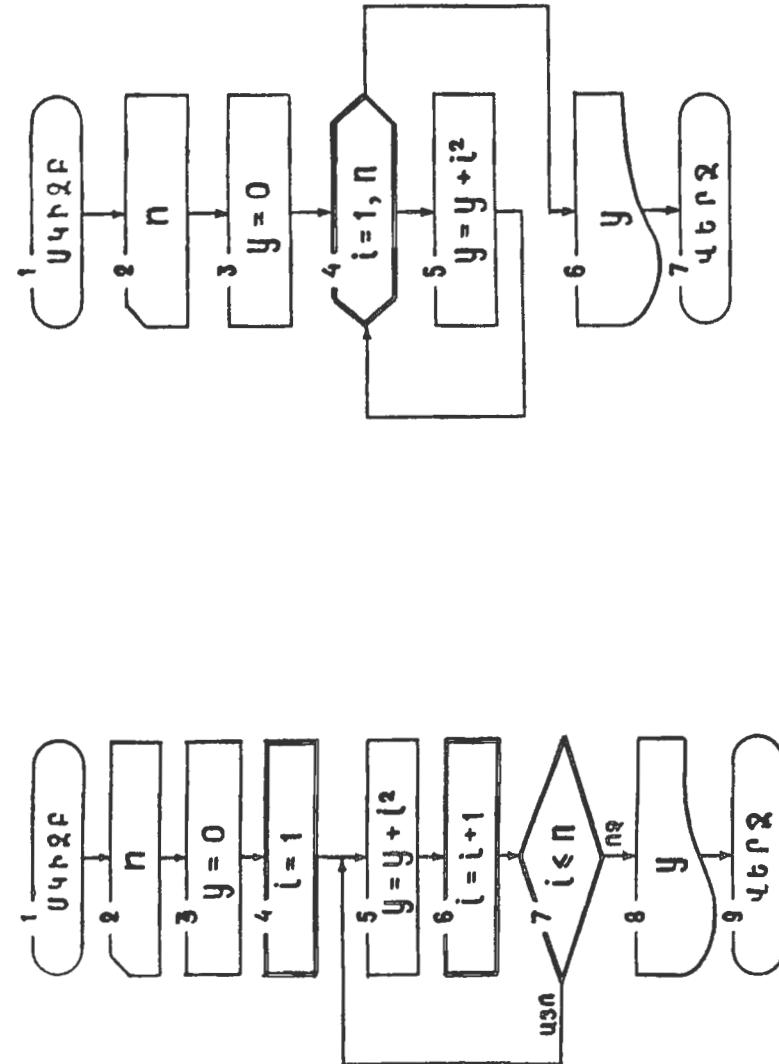
ՀԱՅՈՒՆԵԼ [0 ; 5] ՀԱՏՈՒԱԾԻ ՎՐԱՑ $\Delta x = h$ ՔԱՅԼՈՎ ՀԵՏԵՒԵԱԼ
ՖՈՒՆԿԻԱՑԻ ԱՐԺԵՔՆԵՐԸ :

$$y = \begin{cases} 2x^2 + x + 1, & \text{եթե } 0 \leq x \leq 2, \\ \sqrt{x^2 - 2}, & \text{եթե } 2 < x \leq 5 : \end{cases}$$



Նկ. 65

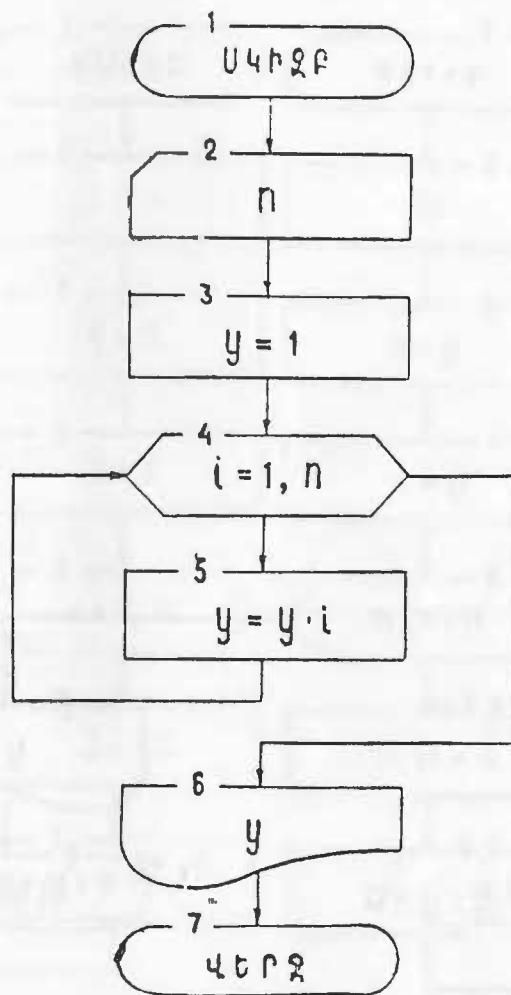
$$\text{ՀԱՅՈՒՆԵԼ ԳՈՒՄԱՐԸ } y = \sum_{i=1}^n i^2 = 0 + 1^2 + 2^2 + \dots + n^2$$



Նկ. 66

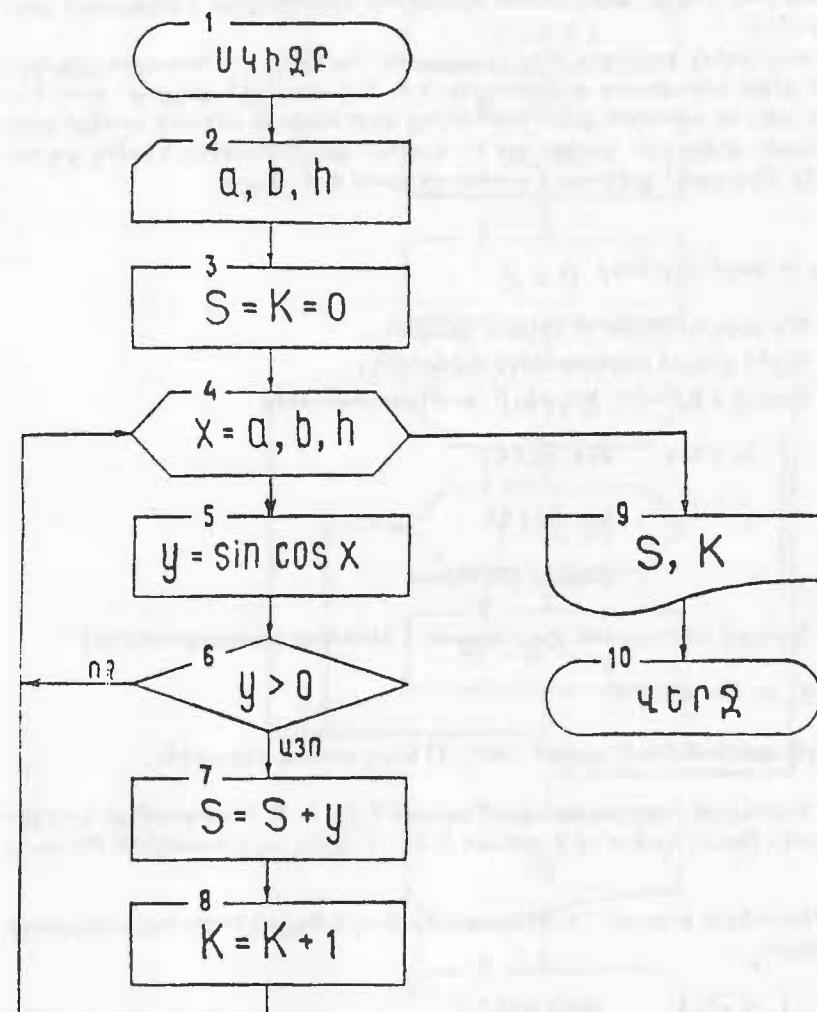
ՀԱՇՈՒԵԼ $y = n!$

$$y = 1 \cdot 2 \cdot \dots \cdot n = \prod_{i=1}^n i$$



Խկ. 67

ՓՈՓՈԽԵԼՈՎ, ԽԱՐԳՈՒՄԵՆՏԻ ԱՐԺԵՔԸ Ի ՔԱՅԼՈՎ [a ; b] ՀԱՏՈՒԱԾԻ
ՎՐԱՅ, ՀԱՇՈՒԵԼ $y = \sin \cos x$ ՖՈՒՆԿՑԻԱՅԻ ԴՐԱԿԱՆ ԱՐԺԵՔՆԵՐԻ
ԳՈՒՄԱՐԸ (s) ԵՒ ԱՅԴ ԿԵՏԵՐԻ ՔԱՆԱԿԸ (k) :



Խկ. 68

(Նկ. 70) բերուած ալգորիթմում հետաքրքրութիւն է առաջացնում $(-1)^k$ հաշուարկը, որը կատարում է նոր փոփոխականի (z) ներածման միջոցով: և զեկավարող փոփոխականի կէնտ արժէքների դէպքում $z = -1$, իսկ ցոյդ արժէքների դէպքում $z = 1$:

Ցիկլերը կարող են ներդրուած լինել մէկը միւսի մէջ:

Դիտարկենք խնդիր, որի լուծման ալգորիթմը պարունակում է ներդրուած ցիկլեր (Նկ. 71):

Այս ալգորիթմը կազմելիս մենք օգտագործել ենք ներքին եւ արտաքին ցիկլերը: Ներքին ցիկլի ղեկավարող փոփոխականը՝ j-է, իսկ արտաքին ցիկլինը՝ k-ն: Կառւեր է նշել, որ արտաքին ցիկլի ղեկավարող փոփոխականի ամէն մի արժէքի դէպքում ներքին ցիկլը լրիւ կատարւում է: Այսինքն ցիկլի մարմինը կազմող գործողութիւնը (Երդ բլոկ) կրկնում է տուեալ դէպքում ու ո անգամ:

Հարցեր եւ վարժութիւններ

1. Ո՞ր ալգորիթմներն են կոչում գծային:
2. Բերէք գծային ալգորիթմների օրինակներ:
3. Տրուած x-ի համար հաշուել Y ֆունկցիայի արժեքը՝

$$y = \begin{cases} 2x^2 + 3x - 1 & , \text{ եթե } -1 \leq x \leq 2 \\ 3^{x+1} & , \text{ եթե } 5 < x \leq 8 \\ 1 & , \text{ մնացած դէպքերում} \end{cases}$$

4. Տիրոյթը հարթութեան վրայ տրուած է հետեւեալ հաւասարումներով:

$$x=0, y=5=x \text{ եւ } y=x^3$$

Որոշել պատկանո՞ւմ է արդեօֆ, A(X, Y) կէտը տուեալ տիրոյթին:

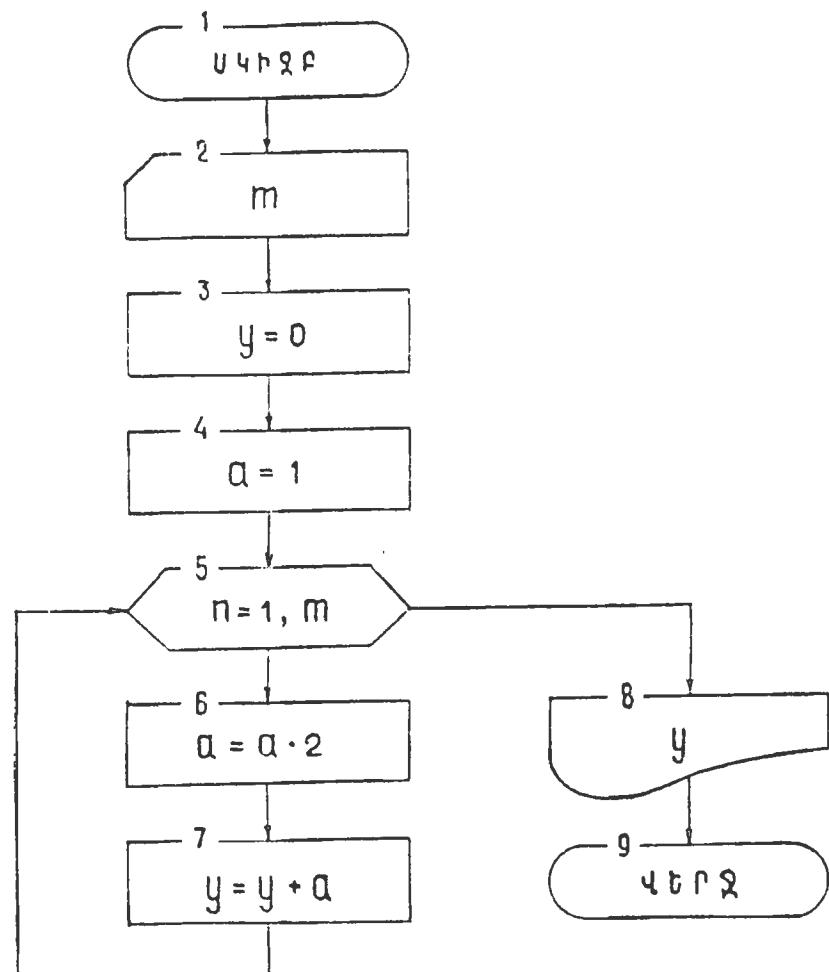
5. Եռանկիւնը հարթութեան վրայ տրուած է իր A, B, C գագաքների կորդինատներով: Որոշել գտնենո՞ւմ է տրուած D (X, Y) կէտը այդ եռանկիւն ներսում, քէ ոչ!

6. Հաշուել եւ տպագրել [1, 8] հատուածի վրայ և քայլով հետեւեալ ֆունկցիայի արժեքները:

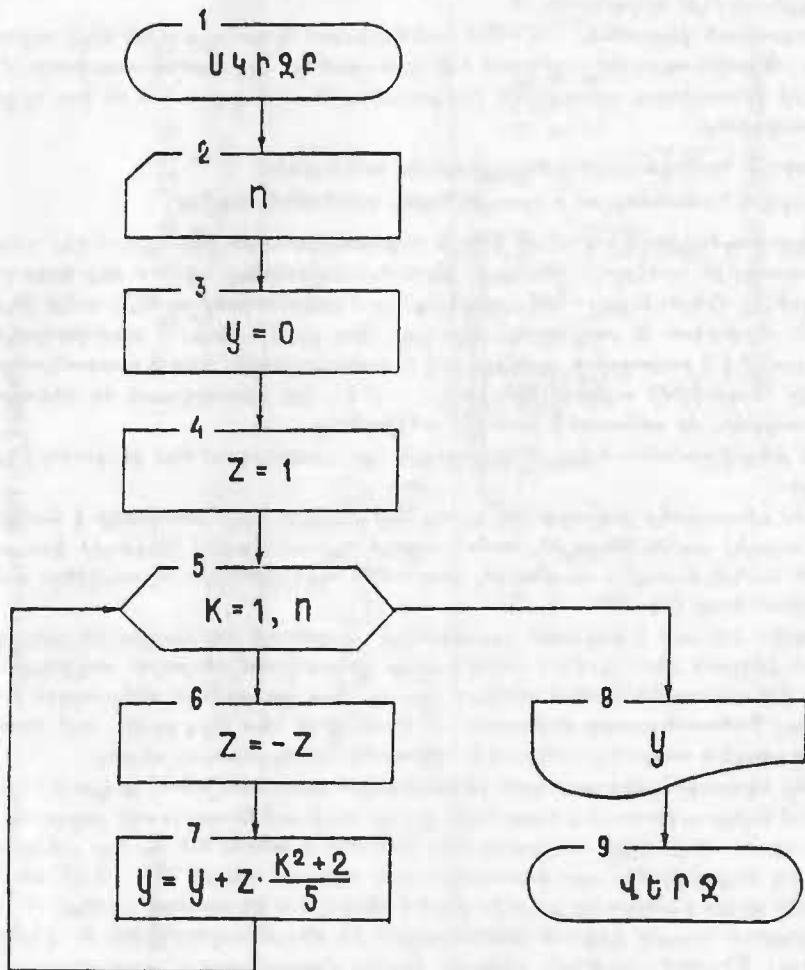
$$y = \begin{cases} \sqrt{x^2 + a^2} & , \text{ եթե } 1 \leq x < 2,5 \\ \operatorname{tg} x & , \text{ եթե } 2,5 \leq x \leq 4 \\ \sin \cos x & , \text{ եթե } 6 \leq x \leq 8 \end{cases}$$

ՀԱՇՈՒԵԼ $y = \sum_{n=1}^m 2^n$

$$2^{n+1} = 2^n \cdot 2$$

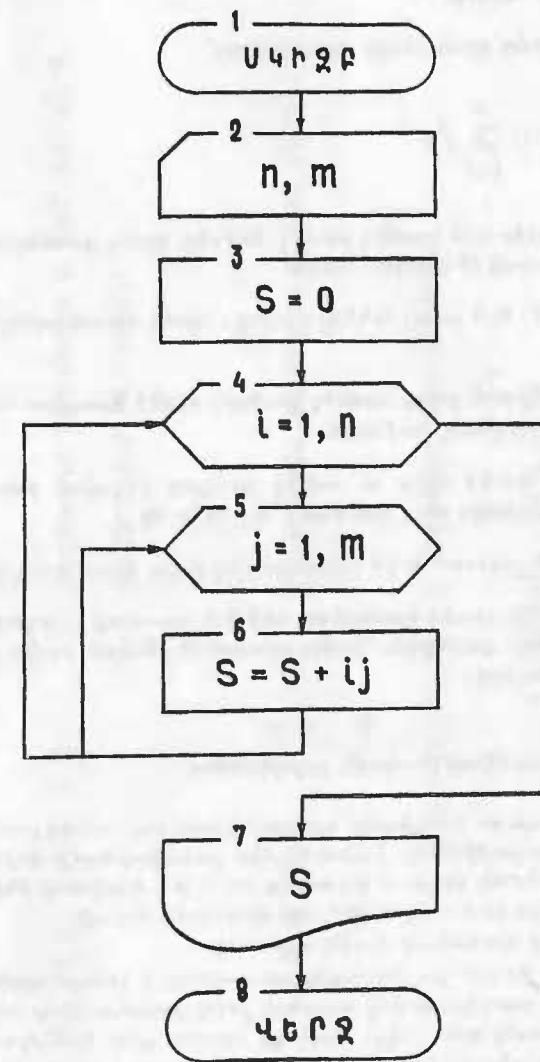


ՀԱՅՈՒԹԵԼ $y = \sum_{k=1}^n (-1)^k \frac{k^2+2}{5}$



ՀԱՅՈՒԹԵԼ ԿՐԿՆԱԿԻ ԳՈՒՄԱՐԸ

$$S = \sum_{i=1}^n \sum_{j=1}^m i j$$



7. Տրուած է ո թնական քիւը: Հաշուել
 $y=1 \cdot 2 + 2 = 4 + \dots + n(n+1) \dots 2n$

8. Որոշել $Y=f(x)$ ֆունկցիայի ամենամեծ արժեքը $[a, b]$ միջակայքում փոփոխելով x -ը ի քայլով:

9. Հաշուել գումարների արտադրեալ՝

$$Y = \prod_{j=1}^n (1 + j^2)$$

10. Հաշուել այն եռամբիշ քուերի քանակը որոնց քուանչամեների գումարը հաւասար է տրուած M թնական քուին:

11. Գտնել այն բոլոր երկնիշ քուերը, որոնց քուանչամեների գումարը բաժանում է 13ի:

12. Գտնել այն բոլոր եռամբիշ քուերը, որոնք հաւասար են իրենց քուանչամեների խորանարդների գումարին:

13. 1022 քուին ազից ու ճախից կցագրել մէկական քուանչան այնպէս, որ ստացուած վեցամիշ քիւը բաժանուի 7ի, 8ի և 9ի:

14. Գտնել տրուած N -ին չգերազանցող բոլոր պարզ քուերը:

15. Թիւր կոչում է կատարեալ, եթէ այն հաւասար է իր բոլոր բաժանարարների (բացի իրենից) գումարին: Գտնել տրուած M թնական քուին չգերազանցող բոլոր կատարեալ քուերը:

5. 5. Զանգուածների մշակման ալգորիթմներ

Նատ յաճախ են հանդիպում այնպիսի խնդիրներ, որոնց լուծման համար հարկ է լինում գործողութիւններ կատարել մեծ քանակութեամբ նոյնատիպ տուեալների նկատմամբ: Նման դէպում նպաստակայարմար է մշակաւոր ինֆորմացիան ներկայացնել գծային կամ ուղղանկիւն աղիւսակների տեսքով:

Իսկ ի՞նչ է նշանակում մշակել աղիւսակը:

Կախուած խնդրի պայմաններից անհրաժեշտ է լինում որոշակի տարր, դեռեւ նոր տարրեր, փոխել տարրերի յաջորդականութիւնը եւ այլն: Իսկ ի՞նչպէ՞ս կարելի է կազմել աղիւսակը, որպէսզի յարմար լինի նրա մշակումը: Նպաստակայարմար է մէկ անուան տակ ձեւակերպել (նկարագրել) տուեալների ամբողջ խումբը, նրա իւրաքանչիւր տարրին համապատասխանեցնելով մէկ կամ երկու կարգահամարներ (ինդեքսներ) այնպէս, որ բոլոր տարրերն ունենան նոյն քանակութեան կարգահամարներ, որոնցով միարժեքօրէն որոշում է տուեալ տարրի համարը այդ

աղիւսակում: Դրա չնորհիւ հնարաւորութիւն ենք ստանում ցանկացած տարր առանձին գիտարկելու:

Դիտարկենք հետեւեալ խնդիրը (նկ. 74):

Մենք արդէն ծանօթ ենք գումարը կամ արտադրեալը հաշուելու ալգորիթմներին: Փոփոխելով 1 ինդեքսը 1ից մինչեւ ո կազմում ենք ցիկլ, որի միջոցով հնարաւոր է գառնում կուտակել վեկտորի բաղադրիչների գումարը ((ա) ալգորիթմում) կամ արտադրեալը ((բ) ալգորիթմում):

Մանրամասն վերլուծենք (նկ. 75) առաջարկուած երկու ալգորիթմների աշխատանքը: Առաջին ալգորիթմում բացի 1 փոփոխականից, որը համապատասխանում է վեկտորի դիտարկուող բաղադրիչից կարգահամարին օգտագործուած են եւս երկու փոփոխականներ՝

Ե-արդէն համեմատուած բաղադրիչներից ամենամեծը.

Կ-արդէն համեմատուած բաղադրիչներից ամենամեծի համարը:

Ծիկլ կազմակերպելուց առաջ կ եւ կ փոփոխականներին վերացրում ենք առաջին բաղադրիչից արժեքը եւ համարը: Համեմատութիւնները սկսում ենք երկրորդ բաղադրիչից մինչեւ ո-դզը: Եթէ պարզուեց, որ յ փոփոխականի արժեքն աւելի փոքր է, քան հերթական ան բաղադրիչից արժեքը (Երդ բլոկ), ապա յ փոփոխականին վերագրուում է ա բաղադրիչից արժեքը, իսկ կ փոփոխականին՝ նրա Յ կարգահամարի արժեքը: Ալգորիթմի աշխատանքն աւարտուում է, եթէ զիտարկուած են վեկտորի բոլոր տարրերը եւ գտնուած է նրանցից ամենամեծը:

Այս խնդրի լուծման երկրորդ ալգորիթմը (բ) առաջարկում ենք վերլուծել ինքնուրոյն:

Այժմ դիտարկենք վեկտորի կարգաւորման խնդիրը, որը յանախակի է հանդիպում տարրեր քոմիտերային համակարգերի աշխատանքում: Վեկտորի կարգաւորման համար կարելի է առաջարկել բաղմաթիւ ալգորիթմներ: Դիտարկենք, օրինակ, հետեւեալը (նկ. 76):

Խնդիքս երեւում է բերուած ալգորիթմից, կազմուած են նիրուուած ցիկլեր: Ներքին ցիկլում (5-7 բլոկեր) յաջորդաբար դիտարկուում են լույս տարրերը եւ գտնուում է ամենամեծ արժեքն ունեցող տարրը: Այդ ալգորիթմին մենք արդէն ծանօթ ենք: Գտնուած տարրը փոխատեղաւում է առաջինի հետ (Երդ բլոկ), որի հետեւանքով առաջին տեղուում է ամենամեծ արժեք ունեցող տարրը:

Մենք նկարագրենք արտաքին ցիկլի առաջին կատարումը $k=1$ դէպօւմ: $k=1$ դէպօւմ նորից աշխատում է ներքին ցիկլը, դիտարկուում են աղիւսակի բոլոր տարրերը, սկսած երկրորդից եւ նրանց մէջ գտնուում է ամենամեծ արժեքը ունեցողը (Երկրորդ մաքսիմում), որը փոխատեղաւում է երկրորդ տարրի հետ: Նոյն ձեւով արտաքին ցիկլի կատարումը շարունակուում է մինչեւ կ-ի վերջնական արժեքը (ն-1): Այդ դէպօւմ ներքին ցիկլում համեմատուում են միայն նախակերպին եւ վերջին տարրերը: Անհրաժեշտութեան դէպօւմ նրանք փոխատեղաւուց յետոյ կարգաւորումը աւարտուում է:

Ուշագործիւն գարձնենք 8րդ բլոկում ընդգրկուած գործողութիւնների վրայ, որոնց միջոցով իրագործուում է վեկտորի երկու տարրերի արժեքների փոխատեղումը: $a_k=a_1$ վերագրումը կատարելուց առաջ անհրաժեշտ է յիշել Կ-րդ տարրի

ԶԱՆԳՈՒԱԾԲ ՄԻԵՒՆՈՅՆ ԱՆՈՒՆՈՎ, ԲԱ.ՅՅ ՏԱՐՐԵՐԸ ԻՆԴԵՔՍՆԵՐՈՎ ՆՇԱՆԱԿՈՒԱԾ ՀԱՄԱՍԵՐ ՏԱՐՐԵՐԻ ԿԱՐԳԱԿՈՐՈՒԱԾ ԱՄԲՈՂՋՈՒԹԻՒՆ Է:

ՕՐԻՆԱԿ՝

$$1) \quad \begin{matrix} 5 & 3,5 & 0 & 125,4 & -7,15 \\ X_1 & X_2 & X_3 & X_4 & X_5 \end{matrix}$$

. | □ ○ △ △ ◇

$$2) \quad a \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7$$

$$3) \quad \left(\begin{array}{ccccc} 1 & 2,3 & -0,52 & 1 & 1 \\ 2,1 & 5 & 6,32 & 0 & X_{11} \\ -3 & 1 & -77 & 2,2 & X_{12} \\ \end{array} \right) \rightarrow \left(\begin{array}{cccc} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \end{array} \right)$$

ՏՈՂԻ ՀԱՄԱՐԸ ՍԵՍՆ ՀԱՄԱՐԸ

Նկ. 72

ԶԱՆԳՈՒԱԾԻ ՀԱՓԱԿԱՆՈՒԹԻՒՆԸ ՈՐՈՇՆՈՒՄ Է ԻՐ ՏԱՐՐԻՆ ԿԻՑ ԻՆԴԵՔՍՆԵՐԻ ՔԱՆԱԿՈՎ.

ՕՐԻՆԱԿ՝

$$1) \quad X(x_1, x_2, \dots, x_n)$$

ՄԻՉԱՓԱՆԻ ԶԱՆԳՈՒԱԾ Է, ՈՐԸ ԱՅԼ ԿԵՐՊ

ԱՆՈՒԱՆՈՒՄ ԵՆ ՎԵԿՏՈՐ:

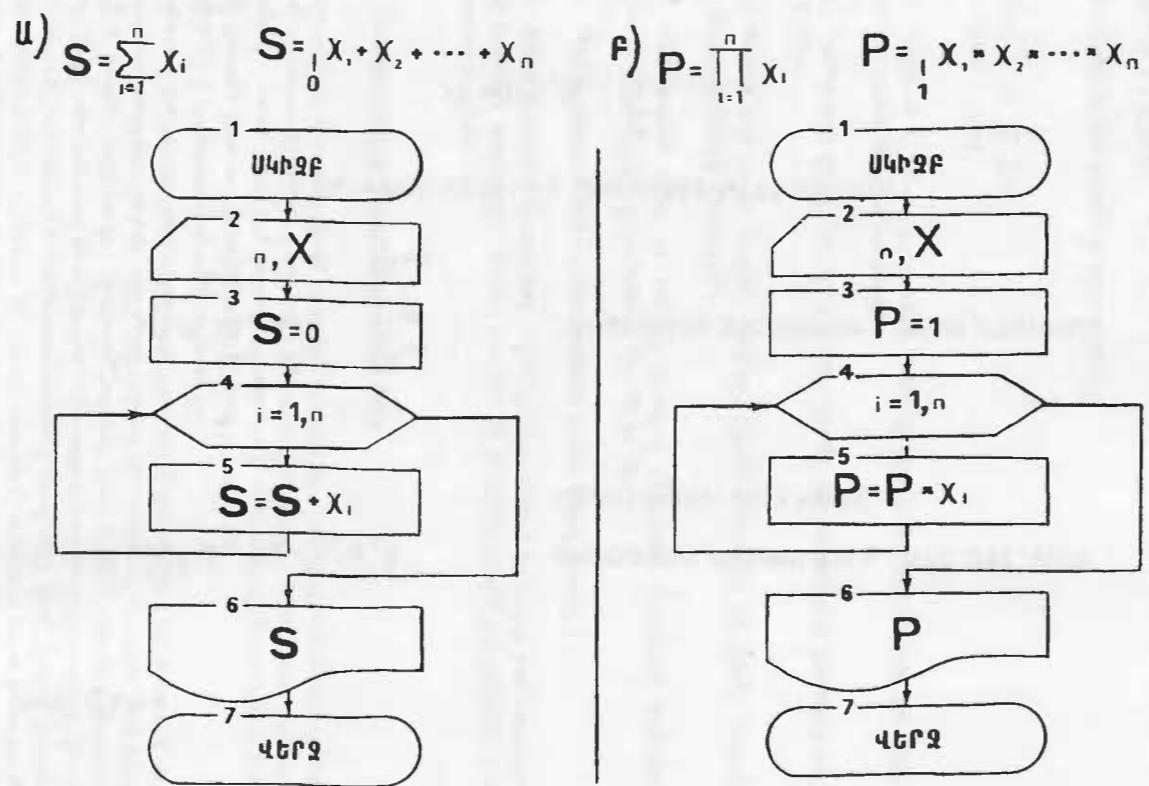
$$2) \quad A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

ԵՐԿՉԱՓԱՆԻ ԶԱՆԳՈՒԱԾ Է, ԿԱՄ ՄԱՏՐԻՑԱ:

ՄԱՏՐԻՑԱՆ ԿԱՐԵԼԻ Է ՆՇԱՆԱԿԵԼ ՆԱԵՒ ԱՅՍՊԻՍ.

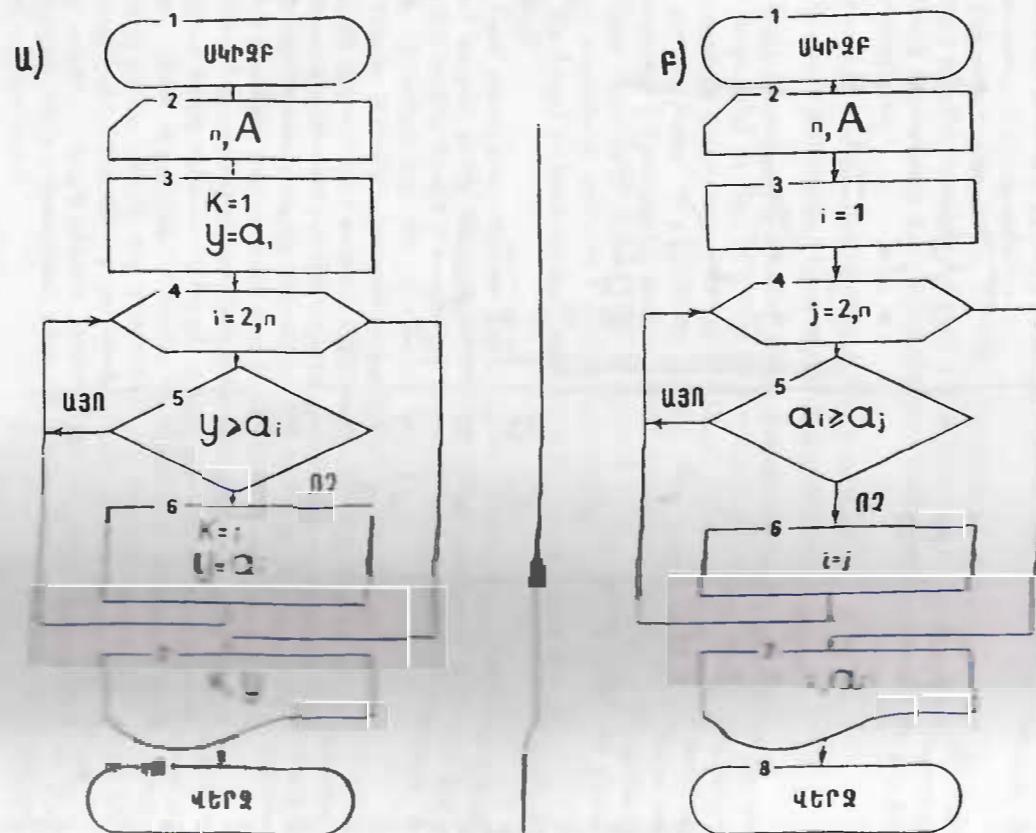
$$A = \|a_{ij}\|, \quad i = \overline{1, n}; \quad j = \overline{1, m}:$$

ՀԱՅՈՒԵԼ $X(x_1, x_2, \dots, x_n)$ ՎԵԿՏՈՐԻ ԲԱՂԱԴՐԻՉՆԵՐԻ ԳՈՒՄԱՐԸ ԵՒ ԱՐՏԱԴՐԵԱԾ:



նկ. 74

ՈՐՈՇԵԼ $A(a_1, a_2, \dots, a_n)$ ՎԵԿՏՈՐԻ ԱՄԵՆԱՄԵՇ ԲԱՂԱԴՐԻՉԻ ՀԱՄԱՐԸ ԵՒ ԱՐԺԵՔԸ:



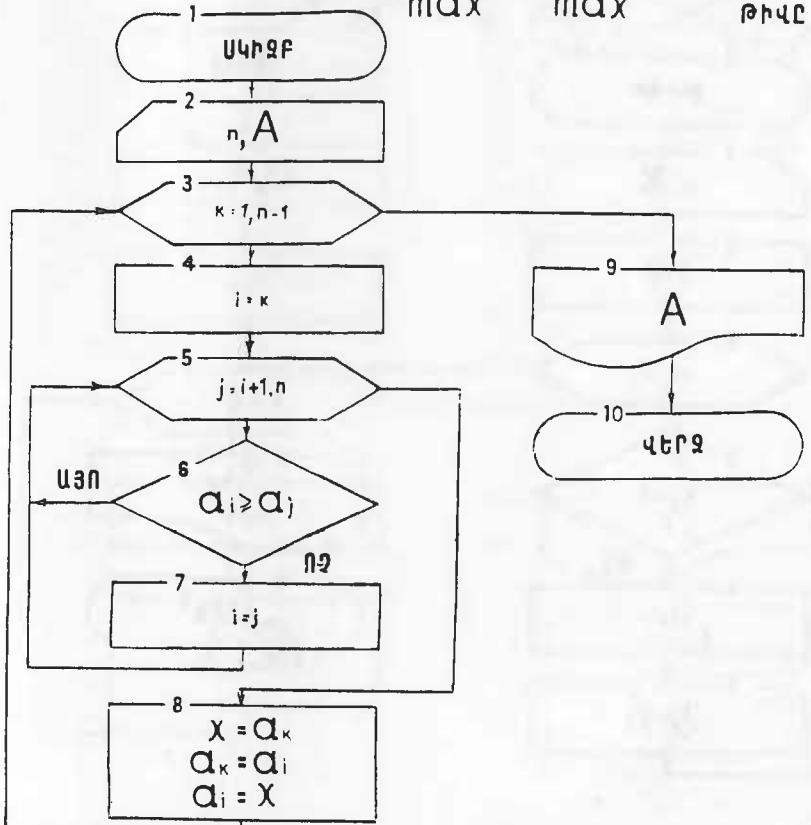
նկ. 75

ԿԱՐԳԱԽՈՐԵԼ $A(a_1, a_2, \dots, a_n)$ ՎԵԿՏՈՐԻ ԲԱՂԱԴՐԻՉՆԵՐԸ ԸՍՏ
ՆՈՒԱԶՄԱՆ:

I ԵՂԱՆԱԿ

Ա) $a_1, a_2, \dots, a_i, \dots, a_k, \dots, a_n$

ԱՌԱՋԻՆ
max ԵՐԿՐՈՐԴ
max ԱՄԵՆԱՓՈքՐ
թիվը



Նկ. 76

"Հին" արժէքը, որպէսզի այն "չկորչի": Դա կատարւում է $x=a_k$ վերագրման միջոցով: Իսկ $a_i=x$ գործողութիւնը առահօգուում է $a_k=x$ "հին" արժէքի վերագրումը a_i փոփոխականին:

Ներկայացնենք նաև այս խնդրի լուծման երկրորդ ալգորիթմը, որի մանրամասն վերլուծութիւնը առաջարկում ենք կատարել ինքնուրոյն (նկ. 77):

Նշենք, որ օ փոփոխականը (*յայտանիշը*) նախատեսուած է ներքին ցիկլի աւարտից յետոյ վեկտորի տարրերի կարգաբարուածութիւնը ստուգելու համար (8րդ րոկ): Եթէ $p=0$, ապա կարգաւորումը աւարտած է, հակառակ դէպում ցիկլը (4-7 բլոկեր) պէտք է կատարուի նորից: Բացատրէք ինչու:

Դիմարկենք յաջորդ խնդրը (նկ. 78):

Այստեղ յ փոփոխականը նախատեսուած է ստացուող յ վեկտորի հերթական կարգամարը հաշուելու համար: Ցիկլի մէջ յաջորդաբար զիտարկւում են x վեկտորի բոլոր տարրերը: Զրոյի հաւասար տարրերը անտեսում են (5րդ րոկ), իսկ միւսները զրանցում են յ աղիւսակում, ստանալով հերթական կարգահամարը (6րդ, 7րդ րոկ):

(նկ. 79) բերուած ալգորիթմը վերլուծենք ինքնուրոյն:

Ինչի՞ համար է նախատեսուած յ փոփոխականը եւ ի՞նչ արժէքներ է այն ընդունում ցիկլի կատարման ընթացքում:

Այժմ դիմարկենք մատրիցների (ուղղանկիւն աղիւսակ) մշակման խնդիրներ (նկ. 80):

Սկսենք նրանից, որ մատրիցի տարրերը ներածելուց առաջ անհրաժեշտ է ներածել նրա տողերի (ո) եւ սիլուների (ո) քանակը (2րդ րոկ):

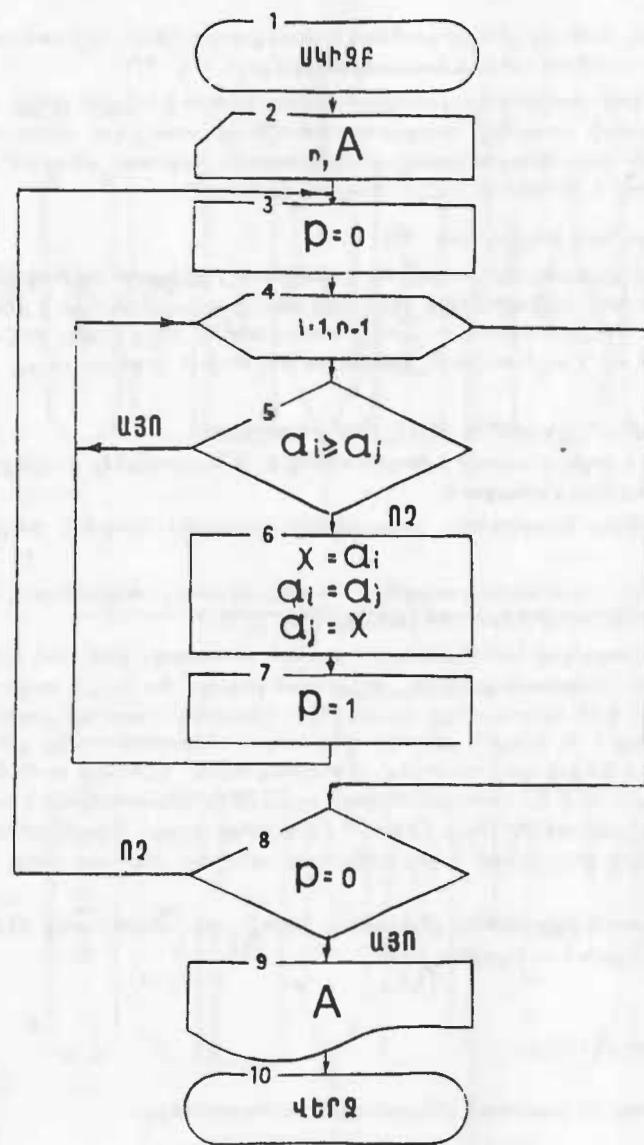
Նախապէս վերագրելով $s=0$ (գումարի հաշւման ժամանակ) կամ $p=1$ (արտադրեալ ալգորիթմ) կազմակերպում ենք ներդրուած ցիկլեր: Դա թոյլ է տախիս ընդհանուր գումարը կամ արտադրեալը կուտակելիս ընդգրկել մատրիցի բոլոր տարրերը: Արտաքին եւ ներքին ցիկլերի դեկապրոց փոփոխականների ընտրութիւնը կախուած է խնդրի պայմաններից: Սասնաւորապէս, գումարը պահանջում է հաշուել տող եւ համապատասխան ալգորիթմը պարունակում է արտաքին ցիկլ ըստ յ փոփոխականի, որը ֆիկսում է մատրիցի տողը: Ներքին ցիկլի աշխատելու ժամանակ փոփոխելով յ կարգահամարը ստանում ենք 1րդ տողի բոլոր տարրերը:

(նկ. 81) բերուած ալգորիթմը վերլուծենք ինքնուրոյն: Համեմատէք վեկտորի տարրերի կարգաբարման ալգորիթմի հետ:

Հարցեր եւ վարժութիւններ.

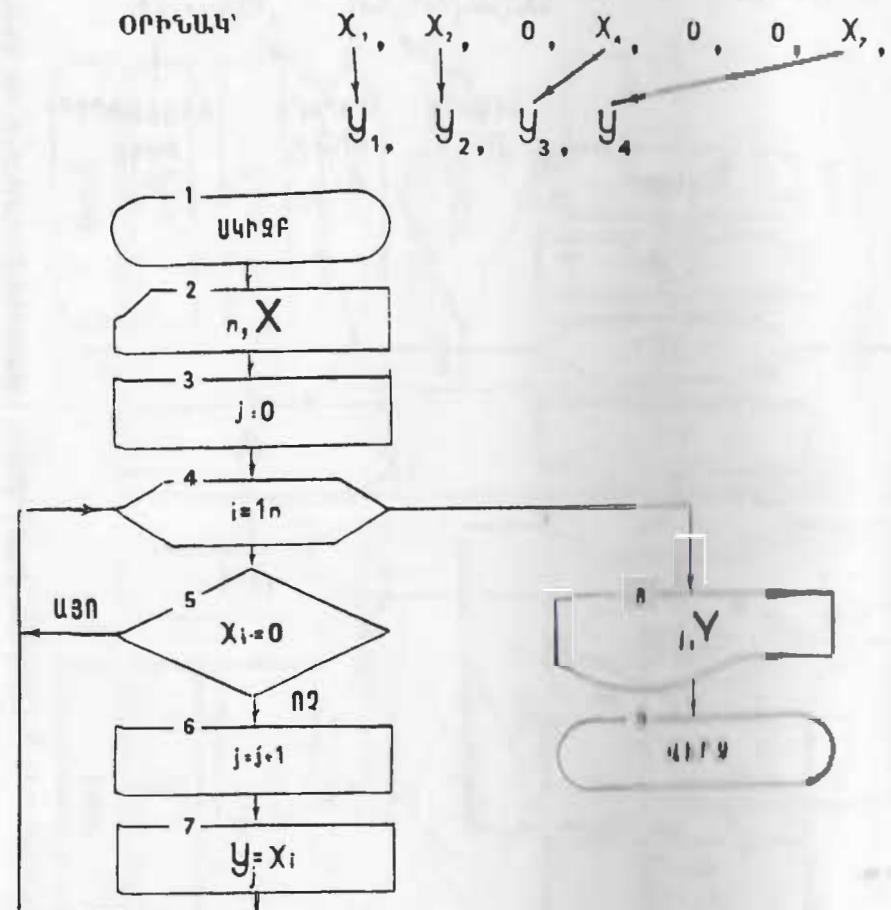
1. Ինչի՞ համար են կազմւում ինֆորմացիոն զանգուածները:
2. Բերէք գծային աղիւսակների (վեկտորների) օրինակներ:
3. Բերէք ուղղանկիւն աղիւսակների օրինակներ:

Եղանակ (ՊՂՊՁԱԿԱԽՈՐ ՄԵԹՈԴ)



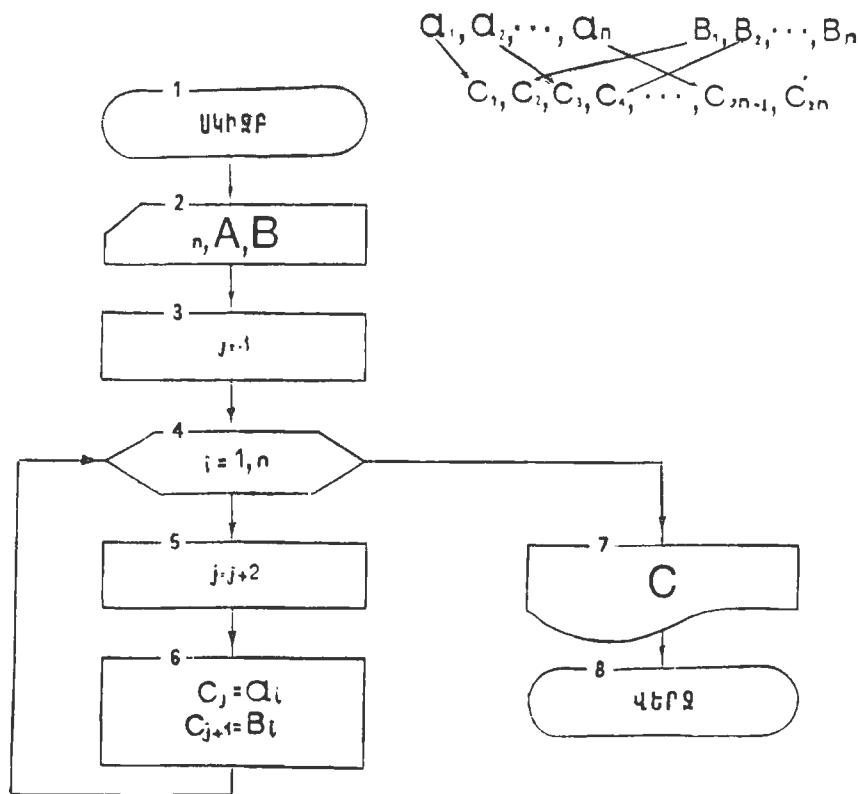
Նկ. 77

ՏՐՈՒԱԾ $X(x_1, x_2, \dots, x_n)$ ՎԵԿՏՈՐԻՑ ՀԵԹԱՑՆԱԼ ԶՐՈԱԿԱՆ ԲԱ-
ՂԱԴՐԻՑՆԵՐԸ:



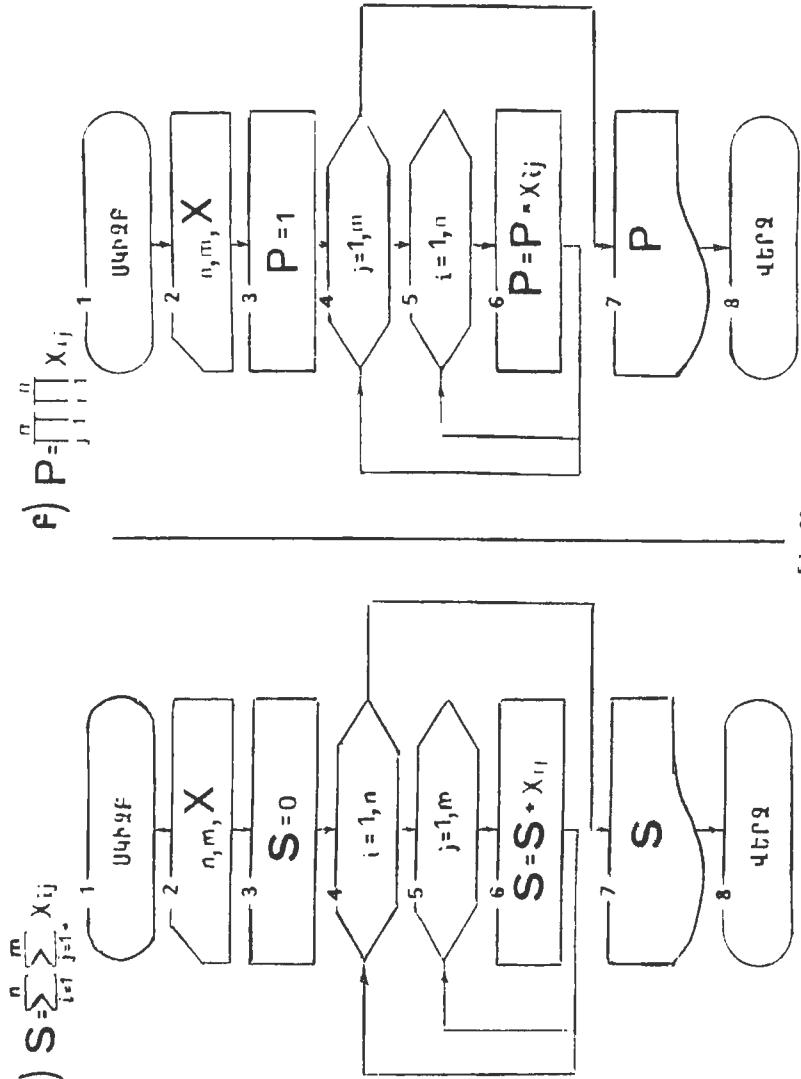
Նկ. 78

ԿԱՌՈՒՑԵԼ $C(c_1, c_2, \dots, c_n)$ ՎԵԿՏՈՐԸ, ՈՐԻ ԲԱՂԱԴՐԻՉԵՐԸ ՍՏԱՑ-
ԻՈՒՄ ԵՆ ՏՐՈՒԱԾ $A(a_1, \dots, a_n)$ և $B(b_1, \dots, b_n)$ ՎԵԿՏՈՐՆԵՐԻ ԲԱ-
ՂԱԴՐԻՉԵՐԻՑ ՅԱԶՈՐԴԱՀԱՎԱՅ:



ՀԱՅՈՒԹԵԼ $X = |x_{ij}| ; i=1, n; j=1, m$ ՄԱՏՐԻՑԱՅԻ ԷԼԵՄԵՆՏՆԵՐԻ S ԳՈՒՄԱՐԸ (ՏՈՂ ԱՄ-
ՏՈՂ) ԽԱՐՍԱԳՐԵԱԱՐ (ՄԻՒՆ ԱՅ ՄԻՒՆ):

$$u) S = \sum_{i=1}^n \sum_{j=1}^m x_{ij}$$



4. Հաշուել $A(a_1, a_2, \dots, a_n)$ եւ $B(b_1, b_2, \dots, b_m)$

Վեկտորների սկալար արտադրեալը՝

$$y = \sum_{i=1}^n a_i b_i$$

5. Հաշուել $X = \prod a_{ij}$; $i=1, n$; $j=1, m$ մատրիցի դրական, բացասական եւ զրոական տարրերի քանակը:

6. Հաշուել $x = \prod a_{ij}$; $i=1, n$; $j=1, m$ մատրիցի գլխաւոր անկիւնագծից մերժել գողուող տարրերից ամենամեծը: Ցիշեցմենք, որ տրուած մատրիցի գլխաւոր անկիւնագծի տարրերն են $a_{11}, a_{22}, \dots, a_{nn}$:

7. Փոխատեղել $A = \prod a_{ij}$, $i=1, n$; $j=1, m$ մատրիցի k -րդ եւ l -րդ տողերը:

8. Տրուած է ո ամբողջ թիւը: Կառուցել A վեկտորը հետեւեալ բաղադրիչներով՝ $n, n(n-1), n(n-1)(n-2), \dots, n!$

9. Տրուած է ո ամբողջ թիւը: Կառուցել $(n \times n)$ կարգի B մատրիցը հետեւեալ բաղադրիչներով՝

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 2 & 2 & \dots & 2 \\ 0 & 0 & 3 & \dots & 3 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & n \end{pmatrix}$$

10. Տրուած է $A = \prod a_{ij}$, $i=1, n$; $j=1, m$ մատրիցը: Կառուցել $B(b_1, b_2, \dots, b_k)$ վեկտորը, որտեղ.

$$\text{ա) } b_i = \sum_{j=1}^m a_{ij}; \quad i=1, n$$

$$\text{բ) } b_i = \max(a_{ij}); \quad i=1, n; j=1, m$$

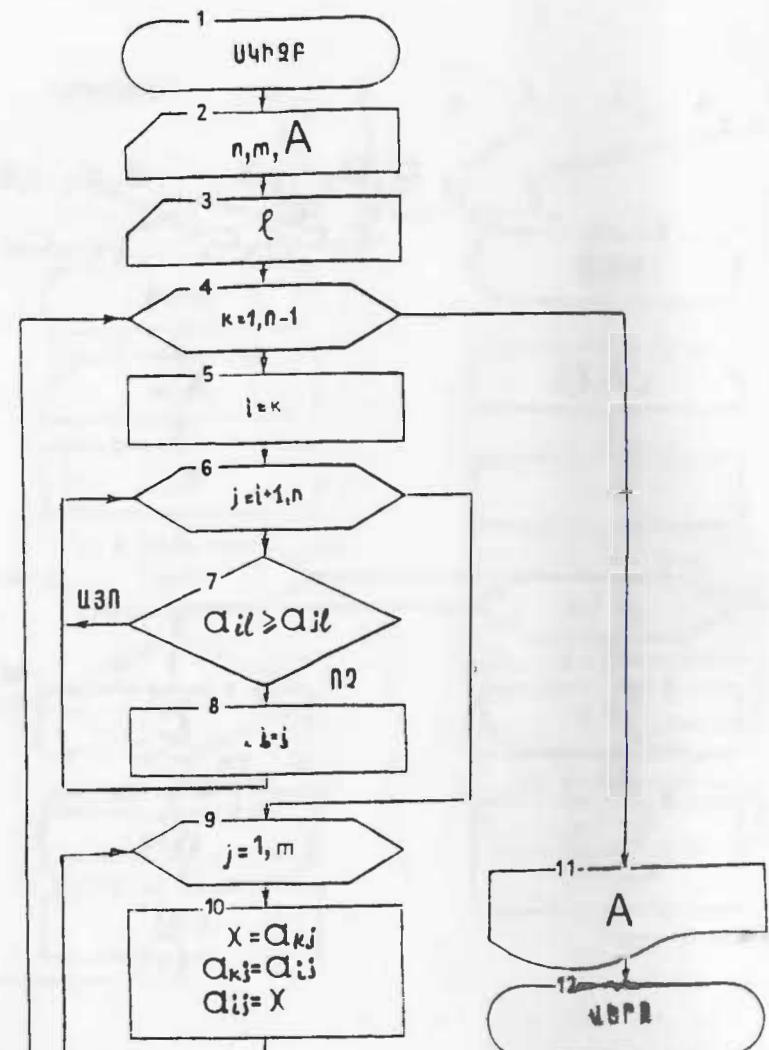
գ) b_j հաւասար է A մատրիցի j -րդ սեան դրական տարրերի քանակին:

11. Գտնել $X(x_1, x_2, \dots, x_n)$ վեկտորի դրական տարրերից ամենափոքը:

12. Գտնել $Y(y_1, y_2, \dots, y_m)$ վեկտորի բացասական տարրերից ամենամեծը:

13. Տրուած են K բնական թիւը եւ $X(x_1, x_2, \dots, x_n)$ վեկտորը, որի տարրերը բնական թուեր են: Կառուցել $Y(y_1, y_2, \dots, y_m)$ վեկտորը, որտեղ y_i -ն հաւասար է $k-i$ եւ x_i -ի ամենափոքք ընդհանուր բազմապատճիկին:

ԿԱՐԳԱԽՈՐԵԼ $A = |a_{ij}|$; $i=1, n$; $j=1, m$: ՄԱՏՐԻՑԱՅԻ ՏՈՂԵՐԸ ԸՍՏ
I-րդ ՄԵԱՆ ԷԼԷՄԵՆՏՆԵՐԻ ԱՃՄԱՆ:



14. Բնական M թիւը սրուած է իր երկուական Ալշերի $A(a_0, a_1, \dots, a_n)$ գծային աղիւսակի տեսմով.

$$M = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2 + a_0, \text{ որտեղ}$$

$$a_i = 0 \text{ կամ } a_i = 1 \quad (i=0, 1, \dots, n):$$

Սուանալ $M+1$ թուի երկուական Ալշերի Յ աղիւսակը:

15. Ներածել $X(x_1, x_2, \dots, x_m)$ իրական բուերի աղիւսակը: Արտածել $Y(y_1, y_2, \dots, y_s)$ գծային աղիւսակը, որը ստացւում է X աղիւսակից՝ նրա ամենամեծ տարրը հանելով:

6. ՔՈՄՓԻԿԹԵՐՆԵՐԻ ՄՐԱԳՐԱԿԱՆ ԱՌԱՋՐԱՎԱՐՈՒՄԸ

6. 1. Բարձր մակարդակի ծրագրաւորման լեզուներ

Մենք արգեն զիտենք, որ ծրագրաւորման լեզուն հանդիսանում է ալգորիթմի ներկայացման այն եղանակը, որն իրաղործեում է հաշուիչ մեքենայի կողմից: Ցանկացած էջՄ-ի աշխատանքը կայանում է ծրագրի կառարման մէջ: Խսկ ծրագիրը իրենից ներկայացնում է մեքենայական հրամանների համախմբութիւն, որոնք տեղաւորում են էջՄ-ի յիշողութիւնն մէջ: Հետեւարար, հաշուիչ մեքենայի հետ համագործակցելու համար անհրաժեշտ է ուսումնասիրի նրա հրամանների լեզուն կամ ընտրել որեւէ լեզու՝ միջնորդ՝ հասկանալի հետ էջՄ-ի եւ մարգար կողմից:

Մեքենայական հրամանները լրենցից ներկայացնում են 0 եւ 1 թուանշաններից կազմուած երկուական թուերի յաջորդականութիւններ: Եթէ ծրագիրը ծաւալուն է, ապա նրա զրաւումը մեքենայական լեզուով չափազանց ծանր է եւ ճանճրապի: Այդ պատճառով էլ լայն տարածում են ստացել այսպէս կոչուած ալգորիթմական ծրագրաւորման լեզուները, որոնց օգտագործումը աւելի հեշտ է ընական է զարգնում ծրագրաւորման պրոցեսում:

Հաշուիչ մեքենաները օժտուած են յատուկ թարգմանիչներով (ծրագրերով), որոնք թարգմանում են այս կամ այն ալգորիթմական լեզուով զրուած ծրագիրը մեքենայականի: Այդ թարգմանիչը ծրագրելու լինում են երկու տիպի՝ կոմպիլատորներ և ինտերպրետատորներ:

Կոմպիլատորը կարգում է ծրագրաւորման լեզուով զրուած տեքստը սկզբից մինչեւ գերջ, կազմելով համարժեք ծրագիր՝ մեքենայական լեզուով, որը հետագայում եւ օգտագործում է հաշուարկների համար: Այդպիսի ծրագրերում թեկու մէկ նշանի փոփոխութիւն կատարելու անհրաժեշտութեան զէպքում ստիպուած ենք լինում ամուղջ ծրագիրը նորից թարգմանել կոմպիլատորի միջոցով: Դա, ի հարկէ, որոշ անյարմարութիւն է առաջացնում ծրագրի ճշգրտման պրոցեսում:

Ինտերպրետատորը կարգում է ծրագիրը մաս մաս, անմիջապէս թարգմանելով եւ կատարելով համապատասխան գործողութիւնները: Այս զէպքում, անկասկած, աւելի յարմար է զաւուած ծրագրի ճշգրտումը, սակայն նուազում է խոնդիր լուծման արտգութիւնը, քանի որ ցիկլերի կատարման ժամանակ կրկնուող հրամանները բազմաթիւ անվամ թարգմանում են:

Ինչպէս անսնում ենք, եւ կոմպիլատորները եւ ինտերպրետատորները ունեն իրենց առանձիւթիւններն ու թերութիւնները: Նշենք, որ երկխօսային լեզուների համար, որոնց թուին են պատկանում օրինակ բէյսիկը, Բապիրան, սովորաբար կազմում են ինտերպրետատորներ, խսկ ֆորտրանի, Ալգորի, Պասկալի համար՝ կոմպիլատորներ:

Ամենապարզ թարգմանիչ ծրագիրը՝ ասկեմբլերն է: Այն ձեւափոխում է այսպէս կոչուած ասեմբլերի լեզուով գրուած ծրագիրը մեքենայական լեզուի: Գործնականօրէն ասեմբլերի լեզուի ցանկացած հրաման ունի իրեն համապատասխանը մեքենայական լեզուում: Սահմանակի հրամաններում երկուական կողերի փոխարէն օգտագործուած են գործողութիւնների կրճատ անուանումները, իսկ յիշողութեան հասցէները ներկայացուում են ծրագրաւորողին յարմար սիմպոլիկ անուններով:

Իսկ ի՞նչ է նշանակում բարձր մակարդակի ծրագրաւորման լեզուներ: Այդ լեզուները յարմար եւ մատչելի լինելով ծրագրաւորողի համար, միաժամանակ բարդ են էջՄ-ի տեսակիտից (այստեղից է գալիս “բարձր մակարդակ” անուանումը), քանի որ նրանցից մեքենայական լեզուին անցնելու համար պահանջւում են լրացուցիչ ծրագրային միջոցներ:

Տարածուած բարձր մակարդակի ծրագրաւորման լեզուներից մէկը Ֆորտրանն է, որը ստեղծուել է 1954 թուականին Ամերիկայի Միացեալ Նահանգներում: Ծրագրաւորման այս լեզուն ստեղծողները նախատեսել էին այն գիտական եւ տիեսնիկական հաշուարկներ կատարելու համար: Դրա հետ է կապուած Ֆորտրան բառի ծագումը՝ FORmula TRANslation (բանաձեւերի թարգմանութիւն) արտայայտութեան յապաւումը: 60ական, 70ական թուականներին Ֆորտրանը իր զանազան տարրերակներով (Ֆորտրան-II, Ֆորտրան-IV, Ֆորտրան-77 եւ այլն) համարում էր ամենատարածուած ծրագրաւորման լեզուն:

Հարկ է նշել նաեւ Ալգորիթմուն (ALGORITMIC LANGUAGE - ալգորիթմական լեզու), որը ստեղծուել է 1960 թուականին եւ մեծ տարածում գտել:

Ալգորիթման վրայ է ստեղծուել Պասկալ ծրագրաւորման լեզուն (յայտնի գիտնական Բէզ Պասկալի պատուին) 1970 թուականին Յիւրիիսում (Զուիցերիա) պրոֆեսոր Նիկառու Վիրտի կողմից: Պասկալ լեզուն շատ յարմար է ինֆորմացիոն-որոնողական համակարգերի նախագծման համար:

Վերջին տարիներին լայն տարածում գտաւ ծրագրաւորման Սի լեզուն, որը հզօր գործիք է հանդիսանում տարրեր ոլորտների քոմֆիւթերային համակարգեր ստեղծող ծրագրաւորողների համար: Սի լեզուով է, օրինակ, գրուած UNIX օպերացիոն համակարգի ծրագրերի մեծամասնութիւնը: Միւս կողմից Սի-ն լայնօրէն կիրառում են քոմփիւթերային մուլտիպլիկացիոն ֆիլմերի ստեղծման համար: Սի լեզուն ստեղծել է 1972 թուականին ամերիկացի Դեննիս Ռիթչին:

Ներկայում լայն կիրառութիւն են գտել ծրագրաւորման տասնեակ լեզուներ, օրինակ Բեյսիկ, Ֆորտրան, Քորլ, Ալգա, Պրոլոգ, Լիսպ, Սի եւ այլն, իսկ լեզուների ընդհանուր քանակը մի քանի հազար է:

6. 2. Ծրագրաւորման Բեյսիկ լեզուն

BASIC

«BEGINNER'S ALL PURPOSE SYMBOLIC INSTRUCTION CODE»

ՍԿՍՆԱԿՆԵՐԻ ՀԱՄԱՐ

ՆԱԽԱՏԵՍՈՒԱԾ ՍԻՄՊՈԼԻԿ ՀՐԱՄԱՆԳՆԵՐԻ ԲԱԶԱՆՊԱՏԱԿ ԼԵԶՈՒ

Ստեղծուել է 1965 թուականին յայտնի մաթեմատիկոս Ճոն Քեմենիրի կողմից:

Բեյսիկը իրաւացիօրէն համարւում է ամենատարածուած եւ մատչելի ծրագրաւորման լեզուն, որը նախատեսուած է մարդու եւ հաշուիչ մեքենայի միջեւ երկխօսութիւն կազմակերպելու համար:

Իրօք, Բեյսիկ լեզուն նախատեսուած էր սկսնակների համար, սակայն լայն տարածում գտաւ չնորհիւ իր պարզութեանը եւ մատչելիութեանը: Այսօր Բեյսիկը ինտերապետասորներ գոյութիւն ունեն զրեթէ բոլոր մինի եւ միկրո մեքենաները համար: Կարելի է նշել Բեյսիկ լեզուի չուրչ 50 զանագան տարրերակ, որոնք ընդունել են միւս ալգորիթմական լեզուներից վերցուած բազմաթիւ ծրագրաւորման էֆեկտիւ միջոցներ (օրինակ՝ Բեյսիկ-80, Բեյսիկ-Ալգաթ, Բեյսիկ-Իղոտ, MSX-BASIC, BASICA եւ այլն): Այսպիսով, մենք ունենք հիմա ոչ թէ մի Բեյսիկ լեզու այլ՝ այդ անունով կոչուող միահարար երկխօսային լեզուների մի ամենող լսումք:

Այդ պատճառով էլ յաջորդ բաժիններում նկարագրուած է Բեյսիկ լեզուն, որի ուսումնասիրումը թույլ կու տայ ձեզ անկացկան չփուել ցանկացած մակնիշի էջՄ-ի հետ, կատարելով անհրաժեշտ հաշուարկները: Մենք ընարելի ենք BASICA ընդլայս նուած տարրերակը, որը ընձեռնում է մեծ հնարաւորութիւններ եւ ընդունուած է IBM PC, EC-1841, Պրավեց եւ այլ ժամանակակից մեքենաների վրայ:

Ինչպէս արդէն նշուել է, ինտերապետատորը ծառայում է Բեյսիկ լեզուով զրուած ծրագրի հրահանգների թարգմանման եւ կատարման համար: Այն կարող է աշխատել երկու ոեժիմով՝ անմիջական եւ ծրագրային: Անմիջական ոեժիմում ինտերապետատորը կատարում է առանձին հրահանգներ, որոնք արտուի են առանց համարակալման ի տարրերութիւն ծրագրային համարակալուած հրահանգների:

Օրինակ՝ $3,5 + \sin 1,2$

արտայայտութեան արժէքը հաշուելու համար բաւական է ներտածել ստեղծնաշարին հետեւեալ հրահանգը.

PRINT 3,5+SIN(1,2)

Մեքենան անիմջապէս կը պատասխանի՝

4,43204

Այստեղ PRINT ծառայողական բառը նշանակում է տպիլ, այսինքն գրուած հրահանգը մեկնարաւանում է այսպէս՝

հաշուել եւ տպիլ $3,5 + \sin 1,2$ արտայայտութեան արժէքը:

Օգտագործելով PRINT հրահանգը անմիջական ոեժիմում կարելի է ինտերապետատորի միջոցով կատարել հզօր միկրոհաշուարկէի բոլոր գործողութիւնները: Աւելի մանրամասն մենք կը քննարկներ PRINT հրահանգը յաջորդ բաժնում, իսկ այստեղ նշենք, որ անմիջական ոեժիմում կարելի է օգտագործել Բեյսիկի հրահանգների մեծ մասը:

Աւելորդ չենք համարում յիշեցնել, որ ծրագրաւորման լեզուի ուսուցումը անհնարին է առանց էջՄ-ի հետ աշխատելու: Բոլոր բերուող օրինակները, վարժութիւնները եւ խնդիրները առաջարկում ենք կատարել հաշուիչ մեքենայի միջոցով՝ ուշադիր հետեւելով նրա պատասխաններին եւ շտկելով հնարաւոր միավաները:

Ուշադիր դարձենք նաեւ լեզուի գերականութեան վրայ, որի չնչին խախտումը բերում է սխալների եւ որպէս հետեւանք՝ չաշխատող ծրագրի:

6. 3. Բեյսիկ լեզուի կառուցուածք

Ծրագրաւորման լեզուով կազմուած ծրագիրը իրենից ներկայացնում է տրուած ալգորիթմն իրականացնող հրահանդիների յաջորդականութեան:

Հրահանգը, իր հերթին, նշանների համախումբ է, որը ցոյց է տալիս, թէ ինչ պործողութիւն է պէտք կատարել որոշակի արգումենտների նկատմամբ:

Իմաստակենք Բեյսիկ լեզուով գրուած մի ծրագիր, որը կազմուած է առանձին հրահանգներից (նկ. 82):

Ինչպէս տեսնում ենք Բեյսիկ լեզուով գրուած ծրագիրը կազմուած է տաղերից: Իւրաքանչիւր տող սկսում է զրական ամբողջ թուրից՝ տողի համարից: Հարկ է նշել, որ էջՄ-ը տեղադրում է ներածուած ծրագրի տողեւը յիշողութեան մէջ համարների աճման կարգով: Դա թոյլ է տալիս ծրագրի ներածման ընթացքում փոփոխութիւններ կատարել՝ աւելացնել նոր միջանկեալ տողեր, հեռացնել աւելորդ եւ սփառ հաւաքած հրահանգները: Այդ պրոցեսը կոչում է ծրագրի խմբագրում եւ որպէսզի այն յարմար լինի ծրագրային տողերը համարակալում են ոչ թէ հերթականութեամբ (1, 2, 3, ...), այլ որոշակի ինահրուալով, ինչպէս մեր օրինակում: Նշենք նաև, որ մէկ ծրագրային տողում կարելի է աելացրել մի քանի հրահանգ, զրանք իրարից անջատելով երկու կէտով (բերուած օրինակում զա 70րդ տողն է):

Ներկայացուած ծրագրով կարելի է հաշուել չարգումենտի սինուաը, կոսինուաը, տանգենսը եւ կոսանգենսը:

10 եւ 20րդ տողերում օդասագործուած REM հրահանգը նախատեսուած է մեկնարանութիւնների (REMARK-մեկնարանութիւն) համար: Այդ հրահանգը հաշուիչ մեքնայի կողմից չի նշանակում:

INPUT X հրահանգը (30րդ տող) նախատեսուած է արգումենտի արժէքի ներածման համար: 40, 50, 60, 70 համարներով տողերում զրուած են վերադրման հրահանգներ, որոնց միջոցով կատարում են անհրաժեշտ հաշուարկները (LET թարգմանում է զիցուք, թող):

Ստացուած արժէքները արտածում են PRINT հրահանգի միջոցով, որը զրուած է 80րդ տողում:

Ծրագրիրը էջՄ-ի յիշողութեան մէջ ներածելուց եւ խմբադրելուց յետոյ այն պէտք է աշխատացնել (կատարել): Դա կարելի է անել RUN հրամանի միջոցով, որը

10 REM ՏՐՈՒԱԾ ԱՐԳՈՒՄԵՆՏԻ ԵՌԱՆԿԻԼՆԱՉԱՓԱԿԱՆ

20 REM ՖՈՒՆԿՑԻԱՆԵՐԻ ՀԱՇՈՒԱՐԱՐ

30 INPUT X

40 LET Y=3, 14*X/10

50 LET S=SIN(Y)

60 LET C=COS(Y)

70 LET T=S/C : LET P=1/T

80 PRINT X; Y; S; C; T; P

90 STOP

նկ. 82

ստիպում է էջՄ-ին կատարել յիշողութեան մէջ տեղաւորուած ծրագիրը՝ սկսած տմբնափոքը համարն ունեցող տողում գրուած հրահանգից:

INPUT հրահանգի կատարման համար անհրաժեշտ է ներածել չ արդումենափ արժէքը (մեր խնդրում անկիւնը պէտք է տրուի աստիճաններով): Էկրանի վրայ յայտնուած է "?" նշանը, որով մերհեան ներածման հրահանգի կատարման ազդանշանն է տարիս:

7 35

Թիւր ներածելուց յետոյ կը ստանանք էջՄ-ի պատասխանը՝

35 0, 6105556 0, 5733228 0,8193296 0, 6997462 1, 42909

Դա PRINT հրահանգի կատարման արժիւնքն է:

Չ արդումենափ այլ արժէքի զէսքում հաշուարկները կրկնելու համար բաւական է նորից տալ RUN հրամանը եւ հարցական նշանի երեւալուց յետոյ ներածել չ-ի արժէքը՝

RUN

7 45

45 0, 785 0, 7068252 0, 7073883 0, 999204 1, 000797

6. 4. Բեյսիկի այրուբեթը

Ինչպէս եւ ցանկացած լեզու, Բեյսիկը նոյնակս ունի իր այրուբենը, այսինքն՝ այն թոյլատրելի նշանների համախումբը, որոնցով կազմւում են հրահանգները: Դրանց մէջ մտնում են՝

1. Լատինական այրուբենի տառերը՝

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2. Թուանշանները՝

0 1 2 3 4 5 6 7 8 9

3. Թուաբանական գործողութիւնների նշանները՝

+ գումարում

- հանում

* բազմապատկում

/ բաժանում

\ ամբողջ բաժանում

^ սստիճան բարձրացնելու գործողութիւն

4. Յարաբերութեան նշաններ՝

- < փոքր
- > մեծ
- = հաստար, կամ վերագրման նշան

5. Յատուկ նշաններ՝

- տասնորդական կէտ
- , ստորակէտ
- ; կէտ-ստորակէտ
- : ապաթարց
- ? Հարցական նշան
- " չակերտներ
- (ձախ փակագիծ
-) աջ փակագիծ

բացակ

\$ տեքստային փոփոխականների յայտարարման նշան
! սովորական ճշտութեան փոփոխականների յայտարարման նշան
% ամբողջ փոփոխականների յայտարարման նշան
համար կամ կրկնակի ճշտութեան փոփոխականների յայտարարման նշան
: երկու կէտ

- ընդգծման նշան

Իրաքանչիւր նշան ունի իր կոդը, որը ներկայացւում է երկուական, տասական կամ տասնվեցական թուով:
Օրինակ՝ A նշանի կոդը
2-ական 1000001
10-ական 65
16-ական 41

6. 5. Հաստատուններ եւ փոփոխականներ

Մարզի կատարման ընթացքում գործողութիւններ են կատարուում որոշակի մեծութիւնների նկատմամբ, որոնք լինում են երկու տեսակի՝ հաստատուն եւ փոփոխական:

Հաստատուն է կոչուում է այն մեծութիւնը, որի արժեքը ծրագրի կատարման ընթացքում չի փոխվում:

Գոյութիւն ունեն երկու տիպի հաստատուններ՝ թուային եւ տեքստային:

Տեքստային հաստատունը կամայական նշանների յաջորդականութիւնն է՝ վերցրած չակերտների մէջ:

Օրինակ՝ "Երեւան-90"

"Արցախ"

"Ինֆորմատիկա եւ հաշուողական տեխնիկա"

Թուային հաստատուններն իրենց հերթին բաժանուում են ամբողջ եւ իրական տիպերի:

Ամբողջ հաստատունները [-32768,+32767] միջակայքում գտնուող ամբողջ թուերն են, որոնք կարող են ներկայացնել տասական, ութական եւ տասնվեցական համակարգերում:

Օրինակներ.

տասական համակարգում՝

26	-374	989	0	-49
----	------	-----	---	-----

ութական համակարգում թուից առաջ անհրաժեշտ է գրել &0 նշանները՝

&0347	-&0176	&01234
-------	--------	--------

տասնվեցական համակարգում թուից առաջ գրում են &H նշանները՝

&H1AB	-&H7C	&H429
-------	-------	-------

Առաջարկում ենք ինքնուրոյն փոխակերպել բերուած ութական եւ տասնվեցական թուերը տասական համակարգի:

Իրական հաստատունները կարող են ներկայացնել եզակի եւ կրկնակի ճշտութեամբ (տասնորդական կէտից յետոյ համապատասխանարար Յ եւ 16 նիշերի ճշտութիւն): Միւս կողմից, իրական հաստատունները կարող են ներկայացնել սովորական կամ էքսպոնենցեալ տեսքով:

Առաջինները-ըլրական կամ բացասական իրական թուեր են, որոնք պարունակում են տասնորդական կէտ:

Օրինակ՝ 37, 268	-4, 18,	0, 6719
-----------------	---------	---------

Էքսպոնենցեալ տեսքով հաստատունները ներկայացւում են որպէս տասնորդական կոտորակ (մանտիս)՝ բազմապատկուած 10-ի համապատասխան աստիճանով՝ 10-ի փոխարէն օգտագործում է Ե տառը:

Օրինակ՝

6, 28E3	-0, 437E-2,	24, 163E1
---------	-------------	-----------

(6280)	(-0, 00437)	(241, 63)
--------	-------------	-----------

Կրկնակի ճշտութեամբ ներկայացուող հաստատունները գրառուում են նոյն եղանակով, միայն Ե տասի փոխարէն (էքսպոնենցեալ տեսքով ներկայացման ժամանակ) օգտագործում է D տառը:

Օրինակ՝

3, 1342678924,	-0, 236D-3,	5, 27681923D5
----------------	-------------	---------------

Փոփոխական է կոչուում այն մեծութիւնը, որի արժեքը ծրագրի կատարման ընթացքում կարող է փոփոխուել:

Ծրագրի կազման ժամանակ փոփոխական մեծութիւնների համար ներածուում են յատուկ նշանակումներ, որոնք կոչուում են փոփոխականների անուններ: Բեյսիկ-ում փոփոխականները ընութագրուում են անունով (Նոյնարկող), արժեքով եւ տիպով:

Փոփոխականների արժեքները կարող են վերագրել ծրագրաւորողի կողմից կամ ծրագրի կատարման ընթացքում հաշւումների արդիւնքների հիման վրայ: Նախքան

Փոփոխականին արժեք վերապրեր, այն բնպունտում է համար 0-ի (տեքստային փոփոխականը՝ “” -դաստիլ տաղ)։

Փոփոխականների անունները կարող են լինել ցանկացած երկարութեան։ Դրանք կարող են պարունակել լուսինական տառեր կամ թուանշաններ, բնդ որում, առաջին նշանը պէտք է լինի տառ։

Փոփոխականի տիպը նշելու համար թոյթառուում է օգտագործել նաև մի շաբթ յառուկ նշաններ (%, #, \$ և այլն)։

Փոփոխականի անունը չի կարող լինել կամ պարունակել թիյուիկ-ի սրեւէ ծառայուական բառ եւ այլ նախագիս առանձնազուտած կառուցուածք։

Փոփոխականները կարող են ունենալ թուային կամ աեքստային արժեքներ։

Տեքստային փոփոխականների անունները առաջաւում են \$ նշանով։
Օրինակ՝ A\$=“ըսմիկթեր”։

Թուային փոփոխականների անունները կարող են ցոյց տալ նրա տիպը։ Դրա համաց անունների վերջում դրում են համարեալ նշաններ։

% - ամբողջ փոփոխականների համար

! - սպորական ճշտութեան փոփոխականների համար

- կրկնակի ճշտութեան փոփոխականների համար

Օրինակ. A1% = 4358; X! = 1,076531.9 և այլն։

Փոփոխականների տիպերը կարող են յայտարարուել նախագիս DEF հրահանդի միջոցով։

DEFINT B, C

Այս հրահանդի միջոցով ծրագրում օգտագործուագ բոլոր փոփոխականները, որոնց անունները սկսուամ են B և C ասուերով յայտարարում են ամբողջ արողի (INTEGER - ամբողջ)։

DEFSNG X

X տառով սկսուաղ անուններ ունեցող բոլոր փոփոխականները յայտարարում են իրական արողի ճշտութեամբ (SINGLE - միայնակ)։

DEFDBL A-D

A-ից մինչեւ D (ըստ լասմական այրութեանի հերթականութեան A, B, C, D) տառերով սկսուող անուններ ունեցող բոլոր փոփոխականները յայտարարում են իրական արողի կրկնակի ճշտութեամ (DOUBLE E - կրկնակի)։

DEFSTR Y, Z - տեքստային փոփոխականների (STRING - տող) յայտարարում։

6. Զանգուածներ

Փոփոխականների զանգուածը - դա միասիս մեծութիւնների համախումը է կամ աղիւակ, որոնք ունեն միեւնային անուանումը։

Զանգուածի իրաքանչիւր տարրի վիճակ են զանգուածի անուանման միջոցով փակազծերում նշելով հերթական համարը (համարները), որոնք ամբողջ թուեր են

կամ արտայայտութիւններ եւ հանդէս են դարիս որպէս կարգահամարներ։ Զանգուածի տարրերը պէտք է ունենան այնքան կարգուածի 7րդ տարրը, իսկ B(2, 3)ր ցոյց է տալիս երկշափանի Յանգուածի 2րդ տարի 3րդ տարրը։

Բեյսիկ յեղուում զանգուածները նկարագրուում են DIM հրահանդի միջոցով (DIMENSION - չափականութիւն)։

DIM r₁, r₂, ..., r_k

որտեղ r_i (i=1, 2, ..., k) նկարագրման բաղադրիչներ են։

Իրարանչիւր բաղադրիչ որոշում է զանգուածի անունը (կամ անուն եւ արողի), ինչպէս նեաւ նրա կարգահամարների վերին սահմանները։ Կարգուածմարների վերին սահմանները արում են հասաստուն բնական թուերով, որոնք չեն դերազանցում 255ին։ Կարգահամարների սարորին սահմանը ընդունում է համար 0-ի։ Եթէ զանգուածի DIM հրահանդի նկարագրուած չէ, ապա նրա բոլոր կարգահամարների վերին սահմանը ընդունում է համար 10-ի։

Օրինակներ։

ա) 10 DIM K% (20)

Այս զանգուածի տարրերն են K%(0), K%(1), ..., K%(20)։

Բոյլտարուում է նաև K%(m+n) դրասումը, որտեղ m-n և n-p բնական թուեր են։

բ) 10 DIM P(17,40)

Երկրաչափանի զանգուած է որի տարրերը կրկնակի ճշտութեան թուեր են եւ որոշում են P(0,23), P(4,30) և այլ կարգահամարներով։

Տարրերի քանակը՝ (17+1)*(40+1)=18*41=738

գ) 10 DIM R\$(30)

Միաչափ զանգուած է, որի տարրերը տեքստային հաստատուններ են։

DIM հրահանդի կատարման ժամանակ ֆիքսուում է համարատափան զանգուածների առկայութիւնը եւ թուային զանգուածի տարրերին վերագրուում է 0-ի համար սկզբնական արժեքը, իսկ տեքստային զանգուածներին՝ զատարկ տող։

Օպերատու յիշողութիւնը նախագիս յայտարարուած զանգուածներից աղատիկու համար օգտագործուում է ERASE (քննիլ) հրահանդի։ ERASE հրահանդի հեռացուած զանգուածները կարելի է յայտարարել նորից, փոխելով նրանց բնութագրերը (արողը, չափականութիւնը, կարգահամարների սահմանները)։

Օրինակ։

10 DIM X(20), Y(40)

...

110 ERASE X, Y

120 DIM X(5,30), Y\$(60)

...

6. 7. Ստանդարտ ֆունկցիաներ

Բաղմաթիւ ծրագրերում հարկ է լինում կատարել այնպիսի գործողութիւններ, որոնք կապուած են տարրական ֆունկցիաների օգտագործման հետ: Օրինակ՝ հաշուել որեւէ արգումենտի քառակուսի արմատը, սինուսը, կոսինուսը, լոգարիթմը, հաշուել տեքստային հաստատունի նիշերի քանակը, փոխակերպել թիւը ամբողջ տիպի եւ այլն: Այդպիսի ֆունկցիաների հաշվման ծրագրերը բարդ չեն եւ ընդունելի են թիւյիկում որպէս բաղադրիչ մաս: Դրանցից իւրաքանչիւրն ունի իր անուանումը (օրինակ՝ SIN, COS, SQR, LEN\$ եւ այլն), որի միջոցով ծրագրաւորովը հնարաւորութիւն է ստանում զիմել անհաջեց ֆունկցիային եւ կատարել համապատասխան գործողութիւնները փակաց ծերում նշուած արգումենտների նկատմամբ:

Հիմնական ստանդարտ ֆունկցիաները կարելի է բաժանել հետեւեալ խմբերի:

ա) մաթեմատիկական ֆունկցիաներ.

բ) տեխստային տուեալների մշակման ֆունկցիաներ.

գ) տուեալների տիպերի փոխակերպման ֆունկցիաներ:

Մաթեմատիկական ֆունկցիաներ

Որպէս մաթեմատիկական ֆունկցիայի արգումենտ կարող է հանդէս գալ ցանկացած թուաբանական արտայայտութիւն, որը գրում է կը դարձերի միջեւ: Բյուկում լողացիուած են հետեւեալ մաթեմատիկական ֆունկցիաները.

ABS (X) - բացարձակ արժեք $|X|$,

INT (X) - ամբողջ մաս (X -ին չդերազանցող ամենամեծ ամբողջ թիւը) $[X]$,

FIX (X) - X թիւն առանց կոտորակային մասի,

SQR (X) - քառակուսի արմատ \sqrt{X} ,

SIN (X) - սինուս $\sin X$ (X -ը ռադիաներով),

COS (X) - կոսինուս $\cos X$ (X -ը ռադիաներով),

TAN (X) - տանգենս $\tan X$ (X -ը ռադիաներով),

ATN (X) - արկտանգենս $\arctan X$,

LOG (X) - բնական լոգարիթմ $\ln X$ ($X > 0$),

EXP (X) - էքսպոնենտ e^X ,

RND (X) - պատահական թիւ $(0, 1)$ միջակայքում,

SGN (X) - թուի նշանը

$$\text{SGN}(X) = \begin{cases} 1, & \text{եթե } X > 0 \\ 0, & \text{եթե } X = 0 \\ -1, & \text{եթե } X < 0 \end{cases}$$

Օրինակներ.

PRINT ABS (-4, 26)

4, 26

PRINT INT (2, 67), INT (-2, 67)

2 -3

PEINT FIX (2, 67), FIX (-2, 67)

2 -2

Ինչպէս տեսնում ենք, INT եւ FIX ֆունկցիաները արդումենափակ դրական արժեքների դէքստրում ուղարկում տալիս են նոյն արդիւնքները, իսկ բացասական արժեքների դէքստրում - տարբեր բացատրենք ինչու:

PRINT SQR (2)

1, 414214

PRINT COS (1, 8)

-0, 227202

PRINT ATN (1, 2)

0, 8760581

PRINT LOG (4,3)

1, 458615

PRINT EXP (3)

20, 08554

PRINT SGN (-4)

-1

PRINT RND (5)

0, 8688611

Տեխստային տուեալների մշակման ֆունկցիաներ

Այս ֆունկցիաները կարող են ունենալ մի քանի արգումենտ ($թուային եւ տեքստային մեծութիւններ$):

MID\$ (X, I, J)$ ֆունկցիայի արդիւնքն է $X$$ տեքստային մեծութեան I-րդ դիրքից սկսած J քանակով սիմպոլների տողը (middle - մէջտեղ): I-ն եւ J-ն [1, 255] միջակայքում գտնուող թուեր են: Եթէ I-ն կամ J-ն կոտորակային թիւ է, ապա վերցւում է նրա ամբողջ մասը:

Օրինակ՝ **PRINT MID\$ ("մատենադարան", 7, 3)**

դար

Եթէ J արգումենտը չի նշուած, կամ I-րդ նշանից յետոյ յ-ից պակաս նշաններ են մնացել $X$$ տեքստային մեծութեան մէջ, ապա MID\$ ֆունկցիան առանձնացնում է I-ից սկսած լուլով մնացած նշանները:

Օրինակ՝ PRINT MID\$ ("փաստաթուղթ", 6)
թուղթ
PRINT MID\$ ("տարեղարձ", 5, 10)
դարձ

Եթէ 1-ն X\$ տեքստային մեծութեան նշանների քանակից մեծ է, ապա MID\$ ֆունկցիայի արդիւնքը զատարկ առղ է:

LEFT\$ (X\$, J) ֆունկցիայի արդիւնքն է X\$ մեծութեան առաջին նշանից սկսած J քանակով նշաններից կազմուած առղը (left - ձախ): Այսինքն LEFT\$ ֆունկցիան հանդիսանում է MID\$ ֆունկցիայի մասնաւոր դէպքը եւ նրա միջոցով կարող է որոշուել այսպէս:

LEFT\$ (X\$, J)=MID\$ (X\$, 1, J)

Օրինակ՝
PRINT LEFT\$ ("գայլաճուկ", 4)
գայլ
PRINT MID\$ ("գայլաճուկ", 1, 4)
գայլ

RIGHT\$ (X\$, J) ֆունկցիայի արդիւնքն է X\$ մեծութեան վերջին նշանից սկսած J քանակով նշաններից կազմուած առղը (right - աջ): Այս ֆունկցիան նոյնպէս կարող է փոխարինուել MID\$ ֆունկցիոնուով:

RIGHT\$ (X\$, J)=MID\$ (X&, K+1, 1),

×որտեղ K-ն X\$ տեքստային մեծութեան նշանների քանակն է (X\$-ի երկարութիւնը):

Օրինակ՝
PRINT RIGHT\$ ("ծրագիր", 3)
ծրագիր
PRINT MID\$ ("ծրագիր", 4, 3)
ծրագիր
PRINT RIGHT\$ ("գայլաճուկ", 3)
Ճուկ

LEN (X\$) ֆունկցիայի միջոցով կարելի է որոշել X\$ տեքստային մեծութեան երկարութիւնը (նշանների քանուկը) (length - երկարութիւն):

Օրինակ՝
PRINT LEN ("երկարութիւն")
10

STRING (n, m) ֆունկցիան նախառեսուած է ու հաս միեւնոյն նշաններից բաղկացած առղ ձեւաւորելու համար (string - առղ):

Օրինակ՝

PRINT STRING (6, 67)
CCCCCCC

STRING ֆունկցիան յարմար է օգտագործել նաեւ հետեւեալ տեսքով՝ STRING\$ (n, x\$), որտեղ x\$-ի առաջին նշանը հանդիսանում է որպէս ձեւաւորուող տողի կրկնուող նշան:

Օրինակ՝
PRINT STRING\$ (5, "+")
+++++
PRINT STRING\$ (7, "- * /")

SPACE\$ (n) ֆունկցիայի միջոցով կարելի է ձեւաւորել ու հաս բացակից բաղկացած առղ (space - բացակ): Այս ֆունկցիան հանդիսանում է STRING\$ ֆունկցիայի մասնաւոր դէպքը եւ նրա միջոցով կարող է որոշուել այսպէս:

SPACE\$ (n)=STRING\$ (n, " ")

Օրինակ՝
PRINT "/" SPACE\$ (10) "/"
/ /
ASC(X\$) ֆունկցիայի արժէքը X\$ տեքստային մեծութեան առաջին նշանի կողն է:
Օրինակ՝
PRINT ASC ("A")
A
PRINT ASC ("3")
51

Տուեալների տիպերի փոխակերպման ֆունկցիաներ

VAL (X\$) ֆունկցիան թուանշաններով զրուած X\$ տեքստային մեծութիւնը ներկայացնում է որպէս տասական թիւ (value - արժէք):

Օրինակ՝
PRINT 3+VAL ("412, 3")
412, 3

STR\$ (X) ֆունկցիան տրուած թիւը (x) ներկայացնում է որպէս նշանների շղթայ (տեքստային մեծութիւն):

Օրինակ՝
PRINT LEN (STR\$ (-46, 4))
5

Նկատենք, որ STR\$ ֆունկցիան հանդիսանում է VAL ֆունկցիայի հակագարձը.
PRINT VAL (STR\$ (5, 3))

CHR\$ (X) ֆունկցիայի արժէքն այն նշանն է, որի կողը հաւասար է x-ի (character - սիմվոլ):

Օրինակ՝

PRINT CHR\$ (82)

R

PRINT CHR\$ (&H3C)

<

CHR\$ ֆունկցիան հանդիսանում է ASC ֆունկցիայի հակառարձը.

PRINT ASC (CHR\$ (70))

70

OCT\$ (X) ֆունկցիան փոխակերպում է թիւը ութական համակարգի:

Արդիւնքը տեքստային հաստատուն է:

Օրինակ՝

PRINT OCT\$ (124)

174

HEX\$ (X) ֆունկցիան փոխակերպում է թիւը տասնվեցական համակարգի: Արդիւնքը տեքստային հաստատուն է:

Օրինակ՝

PRINT HEX\$ (251)

FB

6. Արտայայտութիւններ

Արտայայտութիւնները իրենցից ներկայացնում են որոշակի արժեքի հաշվան կանոններ, կազմում են հաստատուններից, փոփոխականներից եւ ֆունկցիաներից գործողութիւնների նշանների եւ փակագծերի օգնութեամբ:

Բեյսիկում ընդունած են թուարանական եւ տեքստային արտայայտութիւններ:

Թուարանական արտայայտութիւններ

Թուարանական են կոչւում այն արտայայտութիւնները, որոնց արդիւնքում ստացւում է թուային արժեք:

Օրինակ՝

$$-\frac{b - \sqrt{b^2 - 4ac}}{2a}$$

Արտայայտութիւնը Բեյսիկ լեզուի կանոններով կը գրուի
այսպէս.

$$(-B - SQR(B^2 - 4*A*C)) / (2*A)$$

Գործողութիւնների կատարման հերթականութիւնը թուարանական արտայայտութեան մէջ հետեւեալն է՝

1. Փունկցիաների արժեքների հաշում,
2. աստիճան բարձրացման գործողութիւն,
3. բազմապատկում եւ բաժանում,
4. գումարում եւ հանում.

Նշուած կարգի փոփոխման համար օգտագործում են փակագծեր: Փակագծերում լողակուած գործողութիւնները կատարում են առաջին հերթին, իսկ փակագծերի ներսում պահպանում է վերը նշուած առաջնահերթութիւններից Բեյսիկում լողակուած մեղ քաջ յայտնի թուարանական գործողութիւնները.

Բացի նշուած մեղ քաջ յայտնի թուարանական գործողութիւնները՝
ամբողջ բաժանում (առանց մնացորդի) $\setminus (x \setminus y)$,
մնացորդի որոշումը MOD (X MOD Y)

Օրինակ՝

PRINT 20/8

2, 5

PRINT 20\8

2

PRINT 20 MOD 8

4

Տեխստային արտայայտութիւններ

Թուարանական արտայայտութիւններից բացի Բեյսիկում ընդունած են նաև տեքստային արտայայտութիւններ: Մենք արդէն ծանօթ ենք տեքստային մեծութիւններին, ուստի նշենք, որ նրանց միջեւ կատարում է միայն զծալըման (գումարման) գործողութիւն:

Օրինակ՝

A\$ = "Արցախ"

PRINT A\$ + "Հայաստան"

Արցախ Հայաստան

R\$ = "Դասարան"

B\$ = "9-րդ"

C\$ = "10-րդ"

PRINT B\$ + R\$

9-րդ դասարան

PRINT C\$ + R\$

10-րդ դասարան

6. 9. Յարաբերութիւններ

Յարաբերութիւններն օգտագործում են թուային, կամ տեքստային մեծութիւնների համեմատման համար: Յովզորերութեան արժեքը կարող է լինել «true» - ճշմարիտ, եթէ յարաբերութիւնը աելի առնի, կամ «false» - կեղծ, եթէ այն տեղի չունի:

Յարաբերութիւնների նշաններն են:

=	հաւասար,
<>	անհաւասար,
<	փոքր,
>	մեծ,
<=	փոքր կամ հասասար,
>=	մեծ կամ հասասար:

Օրինակ. $a > b$

$a - b <= c * d$

$W\$ < > "D"$

Եթե միեւնոյն արագոյացաւմնեան մէջ հանդէս են զոլիս և թուարանական գործողութիւններ եւ յարաբերութիւններ, ապա տապային հերթին կատարում են թուարանական գործողութիւնները:

Տեքստային մեծութիւնները համեմատում են նիշ՝ առ նիշ՝ ձախից աջ: Դա կատարում է համապատասխան կոդերի համեմատութեան միջոցավ: Մեծութիւնները համարում են հաւասար, եթէ նրանց բարը նիշները համբնենում են: Եթէ որոշ կոդեր տարրերում են, ապա փոքր է համարում այն տեքտային մեծութիւնը, որի կազմում առաջին հանդիսած տարրերուն սիմվոլի կողքը փոքր փոքր է: Եթէ մի տեքտային մեծութեան նիշների քանակը փոքր է միանից, ապա այն համարում է փոքր: Նշենք որ համեմատական ժամանակ դիտարկում են նաև բացակները:

Օրինակներ.

"BC" > "BA"

"PROGRAM" = "PROGRAM"

"WK&" > "WK !!"

"AXYZ" < "BAAA"

"r" > "R"

6. 10. Տրամաբանական գործադրութիւններ

Տրամաբանական գործադրութիւնները կապակցում են տարրեր յարաբերութիւնները: Տրամաբանական գործադրութիւնները կարող է լինել ճշմարիտ (1) կամ կեղծ (0): Արագոյացաւթիւններում արագարանական գործադրութիւնները կատարում են թուարանական գործադրութիւններից յիտոյ:

Տրամաբանական գործադրութիւնների արժեքները հաշվառում են որոշակի կանոններով:

Ստորեւ բերում են համապատասխան աղիւսակները՝ ըստ գործողութիւնների առաջնահերթութեան:

NOT (Ժխտում)

+---+-----+

I X I NOT X I

+---+-----+

I O I 1 I

+---+-----+

AND (Եւ, կոնյիւնիցիա)

+---+-----+

| X | Y | X AND Y |

+---+-----+

| 0 | 0 | 0 |

+---+-----+

| 0 | 1 | 0 |

+---+-----+

| 1 | 0 | 0 |

+---+-----+

| 1 | 1 | 1 |

+---+-----+

OR (կամ, դիզիմֆղիա)

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

XOR (քաշառող կամ)

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

EQV (համարժելութիւն)

x	y	x EQV y
0	0	1
0	1	0
1	0	0
1	1	1

IMP (իմպլիկացիա)

x	y	x IMP y
0	0	1
0	1	1
1	0	0
1	1	1

ՕՐԻՆԱԿՆԵՐ

NOT A>3

X<-4 OR X>3

Y>=6 AND Y<=12

Հարցեր եւ վարժութիւններ

- Ի՞նչի՞ համար է նախատեսուած իմասնապրետատորը:
- Ի՞նչ է նշանակում կոմպիյուտոր:
- Ի՞նչ տարբերութիւն կայ իմասնապրետատորի եւ կոմպիյուտորի միջև:
- Ի՞նչ է նշանակում բարձր մակարդակի ծրագրաւորման լեզու:
- Ի՞նչ է նշանակում BASIC լայպուամբ:
- Կարեի՞ է արդեօք մէկ տողի վրայ գրառել Ինյուիկի մի քանի հրահանգ:
- Ի՞նչպէ՞ս են համարակարուում Ինյուիկ ծրագրի տողերը:
- Ի՞նչպէ՞ս կարեի է աշխատցնել Անրածուած Ինյուիկ ծրագիրը:
- Ի՞նչ տարբերութիւն կայ իմասնապրետատորի անմիջական եւ ծրագրային ոեժինների միջև:
- Որո՞նք են Ինյուիկի այրութեան յատուկ նշանները:
- Ընտրեմք հետեւեալ հաստատութեներից Աիշտ գրուածները:

37	-47, 36	0, 17E+2
"BASIC"	5, 68E-1, 5	7B
&HAA	"լեզու"	ծրագիր
-3	65%	16!
5, 13 #	0, 33 (3)	11*E-2

- Որտե՞ղ է օգտագործում ! Աշանի:

- Ներկայացրեմ հետեւեալ հաստատութեները Եխպոնենցեալ տեսքով.

-367, 284	167	33, 444
-0, 0000001	-2000	-0, 000864
100001	0, 123456	54, 17

- Հետեւեալ փոփոխականների անուանութեներից որո՞նք են Աիշտ գրուած:

A+B	A&	4X
Y6! #	-K%	Z4&
DIM	AI PMA	njd

- Բացատրեմ DEFINT D-II հրահանգի օգտագործման նպատակը:

- Քանի՞ տարբ ունի W գանգուածը, որը նկարագրուած է DIM W (36, 18) հրահանգով:

- Ի՞նչի՞ համար է նախատեսուած ERASE հրահանգը:
- Օգտուելով Բյուիկի ստանդարտ ֆունկցիաներից անմիջական ոեժիմում հաշուել հետեւեալ արտայայտութիւնների արժեքները:

$\sin 40^\circ$	$\cos 1, 2 + \sin 1, 3$	$\operatorname{tg} 35^\circ$
$\operatorname{arctg} 0, 9 + \sqrt{4, 6}$	$\operatorname{arcsin} 0, 6$	$(\arccos 0, 3)^3$
$13, 51$	$\log_2 6$	$\sqrt[3]{3, 5^2 + 6, 1^3}$
$e^i + \ln 6$	$\lfloor -61, 4 \rfloor$	$\sin 1, 6 \cos 20^\circ$
- Օգտագործելով MID\$, RIGHT\$, LEFT\$, Փունկցիաները, "պարկապուկ" բառից ստանալ "կապ", "պար", "պարկ" եւ "պգուկ" բառերը:
- Զեւաւորել 34 աստղամշերից բաղկացած տողը:
- Որոշել Q, q, +, -, F, f նշանների տասական կոդերը օգտագործելով ASC փունկցիան:
- Օգտագործելով CHR\$ փունկցիան գտնել այն նշանները, որոնք ունեն 70, 72 եւ 74 տասական կոդերը:
- Փոխակերպել ուրական համակարգի հետեւեալ տասական թուերը՝ 26, 79, 158, 429, 6314:
- Փոխակերպել տասնվեցական համակարգի հետեւեալ տասական թուերը՝ 256, 327, 1555, 4333:
- Գրառել Ինյուիկի լեզուի կանոններով հետեւեալ թուարանական արտայայտութենները:
 - $\sqrt[5]{\cos x} + \sqrt[5]{\sin x}$
 - $I \operatorname{sincos} x I - \log, y$
 - $\frac{a-b}{3ab}$
 - $2 \operatorname{asin} 2b + \ln(a+b)$
 - $Ix-yI + \operatorname{arctg} x^2 - e^{i \ln x}, \operatorname{arctg} \sin x$
- Ի՞նչ է նշանակում գումարել երկու տեխնային մեծութիւն:
- Որո՞նք են յարաբերութիւնների նշանները:
- Դուք հետեւեալ տրամարանական արտայայտութիւնների արժեքները:
 - NOT A>1 OR X>A AND X<=3 ,
b_pt, A=2, X=1
 - NOT A>1 XOR X>A AND X<=3 ,
b_pt, A=2, X=1
 - X>2 AND X<6 EQV ABS(Y)<10 ,
b_pt, X=4, Y=-5

6. 11. Բեյսիկի հիմնական հրահանգները

Նախորդ բաժիններում մենք ծանօթացանք Բեյսիկ լեզուի կառուցուածքի հետ, ուսումնասիրեցինք գործածուող օբյեկտները (հաստատուններ, փոփոխականներ, զանգուածներ եւ այլն):

Իսկ ինչպէ՞ս կարելի է վերագրել փոփոխականին որեւէ արժէք:

Ինչպէ՞ս փոխել այդ արժէքը:

Ինչպէ՞ս արտածել հաշուարկների արդիւնքները:

Ինչպէ՞ս փոխել հաշուողական պրոցեսի ընթացքը:

Այս հիմնական հարցերն են առաջանուած ցանկացած լեզուով ծրագրեր կազմելու ժամանակ: Ուստի մենք կ'ուսումնասիրենք Բեյսիկի այն հրահանգները, որոնց միջոցով հնարաւոր է կատարել նշուած գործողութիւնները:

6. 12. Արժեքների վերագրման հրահանգներ

Բեյսիկ լեզուն օժտուած է թուային եւ տեքստային փոփոխականներին որոշակի արժէքներ վերագրելու մեծ հնարաւորութիւններով:

Դիտարկենք այդ հնարաւորութիւնները:

1. LET հրահանգը (let - դիցուք, թող) նախատեսուած է փոփոխականին կամ զանգուածի տարրին արտայայտութեան արժէքի վերագրման համար:

Բերենք հրահանգի ընդհանուր տեսքը՝

LET փոփոխական = արտայայտութիւն

Վերագրման նշանի (=) ձախ կողմում LET ծառայողական բառից յետոյ զրուում է այն փոփոխականի կամ զանգուածի տարրի անուանումը, որի արժէքը անհրաժեշտ է տալ կամ փոխել: Վերագրման նշանի աջ կողմում զրուում է այն արտայայտութիւնը (թուարանական կամ տեքստային), որի միջոցով որոշուում է փոփոխականի նոր արժէքը:

Նշենք, որ թուային փոփոխականին չի կարելի վերագրել տեքստային արժէք, իսկ տեքստային փոփոխականին չի կարելի վերագրել թուային արտայայտութեան արժէք:

Օրինակներ՝

30 LET A=6, 12

50 LET X=2, 5*SIN (Y)

70 LET I% = I% + 1

90 LET T\$ = "գիրք"

LET ծառայողական բառը կարելի է չգրել.

Օրինակ՝

110 J% = 21

130 L = L + 1

Վերագրման նշանի իմաստը ընթուած հրահանգներում տարբերուում է մաթեմատիկայի հաւասարման նշանի իմաստից:

Օրինակ՝ K=K+1 նշանակում է հետեւեալը՝ K փոփոխականի լրիմացիկ արժէքին գումարուում է 1 (կատարուում է վերագրման նշանի աջ կողմում զրուած թուարանական արտայայտութեան հաշուարկը): Արդիւնքը ուղարկուում է նոյն ընթիւը, որն ունի K "հասցէն": Բնական է, որ K-ի սկզբնական արժէքը կը փոխարինուի նոր արժէքով:

Ցիշեցնենք, որ չատ կարեւոր է պահպանել հրահանգի քրահականութիւնը:

Բերենք սխալ գրուած վերագրման հրահանգների օրինակներ:

LET Z=X վերագրման նշանի ձախ կողմում չի կարող լինել հաստատուն.

LET R-V=K+L վերագրման նշանի ձախ մասում չի թոյյասորուում զրել արտայայտութիւնը.

LET A\$=B+C չեն համապատասխանում փոփոխականի եւ արտայայտութեան տիպերը (տեքստային փոփոխականին վերագրուած է թուային արժէքը):

DATA եւ READ հրահանգները (data – տուեալներ, read – կարդալ):

Փոփոխականներին արժէքներ վերագրելու համար յաճախ օգտագործուում է մի հնարաւորութիւն, որը հիմուած է տուեալների ըլոկ հասկացութեան վրայ:

Տուեալների ըլոկ անուաններ թուային կամ տեքստային հաստատունների կարգաւորուած յաջորդականութիւնը, որը ձեւաւորուում է ՀՀՄ-ի յիշողութեան մէջ DATA հրահանգի միջոցով՝

DATA արժէքների ցուցակ

Արժէքների ցուցակը կարող է պարունակել թուային կամ տեքստային հաստատուններ, որոնք իրարից առանձնացնելու են ստորակէտով: Մրտզրի կաստրման ժամանակ Բեյսիկի ինտերագրեսատորը հերթականութեալիր դիտարկում է ծրագրում հանդիպող բոլոր DATA հրահանգները եւ կազմակերպում տուեալների ըլոկ: Այս հրահանգը պատկանում է Հկատարուող հրահանգների թուին և կարող է տեղադրուել ծրագրի ցանկացած մասում, բայց պէտք է լինի տողի կամ միակ, կամ էլ վերջին հրահանգը:

Օրինակներ՝

10 DATA 25, -36, 3, յունուար, -11 մայիս

120 DATA 0, 5E-4, 16, 43, հոսանք

Տուեալների ըլոկից հաստատունները կարդալու եւ փոփոխականներին վերագրելու համար նախատեսուած է READ հրահանգը՝

READ փոփոխականների ցուցակ:

Փոփոխականների ցուցակում ընդգրկուած տարբերերը (անուանումները) իրարից անջատուում են ստորակէտով:

Օրինակ՝

20 READ X, Y, Q\$, Z, R\$

40 READ A, B, C\$

Մանրամասն նկարագրենք թէ ինչպէս են համագործակցում DATA եւ READ հրահանգները:

Տուեալների բլոկի համար նախասեռածած է հաշւման ցուցիչ, որը ծրագրի աշխատանքի ընթացքում յիշում է հերթական հաստատումի դիրքը (ծրագրի կատարման սկզբում ցուցիչը նշում է առաջին հաստատումի դիրքը): Կատարելով READ հրահանքը, ինտերպրետատորը "հանում է" տուեալների բլոկից արժեքը, որը որոշում է ցուցիչի միջոցով եւ վերադրում է այն READ հրահանքի ցուցակի առաջին փոփոխականին: Ընդ որում, ցուցիչը "անցնում է" յաջորդ հաստատումին, որի արժեքը համապատասխանաբար կր վերադրուի ցուցակի յաջորդ փոփոխականին, եթէ այն առևայ է:

Օրինակ՝

```
10 READ V, W, G$  
30 DATA 14, 4, -5, 25, դիլտեմբիր, 64  
....
```

10րդ առղում զրուած հրահանգի կատարումից յետոյ Ն փոփոխականին կը վերագրուի 14, 4 արժեքը, Վ-ին՝ -5, 25, իսկ G\$ տեքստային փոփոխականին՝ "զեկաեմբեր": Այս ամէնից յետոյ տուեալների բլոկում կը մնայ միայն 64 թիւը, որին եւ ուղղուած կը լինի ցուցիչը: Այդ արժեքը յաջորդ READ հրահանգի կատարման ժամանակ կարող է վերագրուել որեւէ փոփոխականին:

Նշենք նաեւ, որ տուեալների բլոկի սպասումից յետոյ ցուցիչի արժեքը որոշուած չէ:

Ցուցիչի վերքը կարելի է փոխել RESTORE հրահանգի միջոցով (restore - վերականգնել), որը թոյլ է տալիս մի քանի անդում օգտագործել DATA հրահանգով ստեղծուած տուեալների ցուցակը:

RESTORE n

որտեղ n-ը ծրագրի DATA հրահանգ սպարունակող ասդի համար է:

Այս հրահանգի կատարման արդինքն է ցուցիչի աեղափոխումը տուեալների բլոկի այն հաստատումի դիրքը, որն առաջինն է զրուած ծրագրի n-րդ առղի DATA հրահանգում: Եթէ RESTORE հրահանգում տողի համարը (n-ը) նշուած չէ, ապա ցուցիչը տեղափոխուում է ծրագրի առաջին DATA հրահանգի առաջին հաստատումի դիրքը (տուեալների բլոկը վերականգնուում է ամրողութեամբ):

Օրինակ՝

```
10 DATA 11, 15, 20, 30, 40  
20 READ X, Y  
30 RESTORE  
40 READ A, B, C  
....
```

20րդ տողի հրահանգի կատարումից յետոյ X=11; Y=15: Այնուհետեւ 30րդ տողում զրուած RESTORE հրահանգը կր տեղափոխի ցուցիչը տուեալների բլոկի սկիզբ եւ 40րդ տողում զրուած READ հրահանգի կատարումից յետոյ A, B, C փոփոխականները համապատասխանաբար կը ստանան 11, 15, 20 արժեքները:

Եթէ հեռացնենք ծրագրից 30րդ տողը, ապա՝

X=11; Y=15; A=20; B=30; C=40

INPUT հրահանգը (input - մուտք) նախատեսուած է տուեալները ստեղնաշարից ներածելու համար:

Այս հրահանգի ընդհանուր տեսքը հետեւեան է՝

INPUT "տեքստ"; փոփոխականների ցուցակ

Ցուցակում նշուած փոփոխականները իրարից անջատուում են ստորակէտով:

INPUT հրահանգի կատարման ժամանակ ընդհանուր է ծրագրի աշխատանքը եւ տեսատիպի էկրանին արտածում է չակերաների մէջ զրուած հաղորդագրութիւնը (եթէ այն առկայ է) եւ հարցական նշանը (?): Մրագրի կատարումը չի չարունակուի մինչեւ ստեղնաշարից չներածուեն ցուցակում նշուած բլոկը փոփոխականների արժեքները: Ներածման ժամանակ տուեալներն առանձնացուում են ստորակէտով: Նրանց քանակը, մուծման հերթականութիւնը եւ ամպերը պէտք է խիստ համապատասխան ցուցակում նշուած փոփոխականների քանակին, հերթականութիւնը եւ ափակերին: Ներածման աւարտից յետոյ արժեքները յաջորդարար վերագրուում են ցուցակում նշուած փոփոխականներին, որից յետոյ շարունակուում է ծրագրի աշխատանքը: Այսինքն, եթէ ներածումը զե՞ն չի աւարտուել, կարելի է շտկել թոյլ արուած սիրալները (փոփոխականներին գես չեն վերագրել համապատասխան արժեքները):

Օրինակ՝

```
10 INPUT "Յերածէք տարին, ամիսը եւ օրը"; G, M$, D  
RUN
```

Ներածէք տարին, ամիսը եւ օրը? 1990, սեպտեմբեր, 17

Այս փոքրիկ ծրագրի կատարումից յետոյ (այն, ինչպէս տեսնում էք, լաղկացած է ընդամենը մէկ հրահանգի) G փոփոխականին կը վերագրուի 1990 արժեքը, M\$ տեքստային փոփոխականին՝ սեպտեմբեր, իսկ D-ին 17:

10 INPUT A, B, X%

RUN

? -3, 47, 65, 36, 10

Այս ծրագրի արդինքն է՝

A=-3, 47 B=65, 36 X%=10

INPUT հրահանգը շատ է կիրառուում այնպիսի ծրագրերում, որոնք նախատեսուած են բազմաթիւ տարբերակների հաշուարկման համար նախնական տուեալների փոփոխման պէտքուում: Եթէ այդպիսի ծրագրում նախնական տուեալների ներածումը կազմակերպենք LET կամ DATA եւ READ հրահանգների միջոցով, ապա հերթական տարբերակի հաշուարկից առաջ ստորագրուած կը լինենք փոփոխութիւն կատարել ծրագրի LET կամ DATA հրահանգներում, որպէսզի փոփոխականներին վերագրենք նոր արժեքներ: Իսկ եթէ օգտագործենք INPUT հրահանգը, ապա նախնական տուեալների փոփոխութեան համար ամենեւին կը չի պահանջուի ծրագրային փոփոխութիւն (կէՄ-ի ազդանշանից (?)) յետոյ մենք ուղղակի ստեղնաշարից կը ներածենք նոր տուեալները), որն, անկասկած, աւելի յարմար է:

Միւս կողմից, եթէ ներածուող տուեալները շատ են (օրինակ, ինֆորմացիոն զանուածների մշակման ինդիքներում), յարմար է կազմակերպել նրանց մուտքը DATA եւ READ հրահանգների միջոցով:

Ինչպէս տեսնում ենք այս կամ այն ներածում օգտագործման ուրուտները բաղմազան են եւ կապուած են կոնկրետ տուեալների խնդրի պայմանների հետ:

58 մեխակ
60 PRINT 58; "մեխակ"
RUN

58 մեխակ

Ինչպէս տեսնում ենք բերուած օրինակներից, տարրերն իրարից բաժանում են ստորագէտով կամ կէտ-ստորակէտով:

Եթէ տորրերի միջև գրուած են ստորակէտներ, ապա համապատասխան արդզինքները էկրանի վրայ արտադում են իրարից հեռու: Էկրանի տաղր այդ գլուխում բաժանում է պայմանական զոտիների, իրաքանչիրը 14 նիշ լայնութեամբ: Հերթական տարրին արտածեում է սկսած համապատասխան զոտու առաջին նիշից: Եթէ տարրերն իրարից առանձնացւում են կէտ-ստորակէտով, արդիւնքները տպում են կողք կողքի (մէկ նշանի հետաւորութեամբ):

PRINT ծառայողական բառի փոխարէն կարելի է օգտագործել ? նշանը:
Օրինակ՝

10 INPUT X\$
20 Y=LEN(X\$)
30 ? X\$; "բառը պարունակում է"; Y; "նառ"
RUN
?ապատութիւն

ապատութիւն բառը պարունակում է 0 տառ:

Եթէ ցուցակի վերջին տարրից յետոյ դրուած է ստորակէտ կամ կէտ-ստորակէտ, ապա յաջորդ PRINT հրահանգով արտածուազ տուեալները կը տպուեն նոյն տողում, եթէ, ի հարկէ, այդ տողը դու չի լրացի:

Օրինակ՝
60 PRINT X, Y
90 PRINT "աւարտ"

Հրահանգներով կատարուազ գործուալիթիւնները համարժէք են
100 PRINT X, Y, "աւարտ"

Հրահանգի կատարմանը:

PRINT հրահանիվ հետ կիրասաւմ է TAB(0) ֆունկցիան, որի միջոցով անցում է կատարուամ տողի Ա-րդ դիրքին: Ա-ը կարող է իրենից ներկայացնել թուաբանական արտայայտութիւն: Այդ գէտքում սկզբում հաշւընում է արտայայտութեան արժեքը և վեց տուական մասը, այնուհետեւ ակօսիչը (տեղացոյց) տեղափոխում է ընթացիկ տողի Ա-րդ դիրքը և կատարուամ է տուեալների արտածումը այդ դիրքից սկսած:

Օրինակ՝
10 INPUT X, Y
20 PRINT TAB(X); X
30 PRINT TAB(Y); Y
RUN
?5, 15
5

15

Ինչպէս տեսնում ենք X փոփոխականի արժեքը (5) տողի 5րդ դիրքում, իսկ Y փոփոխականի արժեքը (15) - 15րդ դիրքում:

LOCATE հրահանգի միջոցով (locate - տեղադրել) կարելի է արտածել ելքային տուեալների էկրանի կամայական դիրքում:

Հրահանգի ընդհանուր տեսքն է՝

LOCATE m, n, k

որտեղ ո-ը տողի համարն է, 1≤n≤25

մ-ը սեան համարն է

կ-ն ընդունում է 0 կամ 1 արժեք

եթէ կ=1, ապա ծրագրի կատարման ժամանակ ակօսիչը երեւում է էկրանի վրայ, իսկ եթէ կ=0 - չի երեւում:

Օրինակ՝

20 LOCATE 12, 30

30 PRINT "Հաշուարկիչ"

Ծրագրի կատարուամից յետոյ էկրանի կենարունում (12րդ տողի 30րդ դիրքից սկսած) կը գրուի "Հաշուարկիչ" բառը:

Ակօսիչի ընթացիկ դիրքը որոշելու համար նախատեսուած են POS (position - պոզիցիոն) և CSRLIN (cursor line - ակօսիչի տողը) ֆունկցիաները:

POS(0) ֆունկցիայի արդիւնքում ստանուած ենք այն սեան համարը, որտեղ դունում է ակօսիչը: Այս ֆունկցիայի արդումենաը չի ազդում արդիւնքի վրայ և կարող է լինել կամայական թիւ:

Համապատասխանաբար CSRLIN ֆունկցիայի արդիւնքում ստանուած ենք այն տողի համարը, որտեղ դունում է ակօսիչը: Այս ֆունկցիան օգտագործուած է առանց արդումենաի:

Օրինակ՝

10 LOCATE 10, 12

20 PRINT 15, "Քոմիլիւրեր"

30 M=CSRLIN

40 N=POS(0)

50 PRINT "Ակօսիչը գտնուում է"; M; "Տողի"; N; "դիրքում"

RUN

15 Քոմիլիւրեր

Ակօսիչը գտնուում է 10 տողի 38 դիրքում

Ինքնուրոյն վերլուծէք այս ծրագրի աշխատանքը եւ մեկնաբանէք արդիւնքները:

PRINT USING հրահանգը թոյլ է տողիս կազմակերպել տուեալների արտածուամբ ըստ նմուշաբանի, որի միջոցով կարելի է նկարագրել տաղուող տարրերի ներկայացման ձեւը, զբաղեցուող դաշտի իրկարութիւնը, ասանորդական նիշերի քանակը եւ այլն:

Հրահանգի ընդհանուր տեսքը հետևեալն է՝

PRINT USING նմուշաբան; տարրերի ցուցակ

Նմուշաբանը իրենից ներկայացնուած է տեքստային հաստատուած կամ փոփոխական, որը սահմանուած է արտածման ձեւու:

Օրինակ՝

PRINT USING "##, ##"; -4. 2764

-4. 276

Այստեղ նմուշատիպը պարունակում է # եւ . նշանները։ Ակնյայտ է, որ կէտը որոշում է տասնորդական կէտի դիրքը, իսկ իւրաքանչիւր # նշանը յատկացնում է մէկ նիշ թուանշանի, թուի նշանի կամ բացակի համար։

Բերուած օրինակում թուի ամբողջ մասին (նշանի հետ միասին) յատկացնում է 2 նիշ, իսկ կոտորակային մասին՝ 3։

? ? ? ? նշանները սահմանում են թուի արտածումը էքսպոնենցիալ տեսքով։

Օրինակ՝

10 X\$="##, ##, ????"

20 PRINT Using X\$; 28

30 PRINT USING X\$; -367

40 PRINT USING X\$; 4418, 539

RUN

2, 800E+01

-3, 670E+02

4, 419E+03

Եթէ արտածուող թիւը չի տեղաւորւում նմուշատիպով սահմանուած դաշտում, ապա թուից առաջ դուրս է բերում % նշանը։

Օրինակ՝

10 Y\$="###"

20 PRINT USING Y\$; 97

30 PRINT USING Y\$; -721

40 PRINT USING Y\$; 16738

RUN

97

%-721

% 16738

Տեսատարին արժէքների նկարագրման համար օգտագործում են հետեւալ նշանները՝

? - արտածուում է տողի միայն առաջին սիմվոլ։

& - արտածուում է ամբողջ տողը։

\ \ Հատ բացակ\ - արտածուում են տողի n+2 սիմվոլներ սկսած առաջինից։

Օրինակ՝

10 Z\$="Ազատութիւն"

20 PRINT USING "!"; Z\$

30 PRINT USING "\\"; Z\$

40 PRINT USING "\\\"; Z\$

50 PRINT USING "&"; Z\$

RUN

Ա.

Ա.զ

Ա.զատ

Ազատութիւն

Տուեալների արտածումը տպող սարքի վրայ

Նկարագրուած PRINT եւ PRINT USING հրահանգերի միջոցով կարելի է արտածել ելքային տուեալները տեսատիպի կերտուի վ/ և կայու շատ յաճախ անհրաժեշտ է լինում ստանալ ինդքի լուծման արդիքները, թղթի վրայու Այդ նպատակին են ծառայուալ LPRINT եւ LPRINT USING յու այ գերեւ պրոցեսուագործման կանոնները ոչնչով չեն տարբերւում համապատասխան ՊRINT եւ PRINT USING հրահանգների օգտագործման կանոններից։

Օրինակ՝

10 INPUT A, H

20 S=A*H/2

30 PRINT "Ի՞մքը="; A, "բարձրութիւնը "; H, "մակերեսը "

40 LPRINT "Ի՞մքը="; A, "բարձրութիւնը "; H, "մակերեսը ", S

RUN

?6, 4

Ի՞մքը=6 բարձրութիւնը=4 մակերեսը=1

Այս ծրագրի միջոցով, ներածելով եռանկեան կիմքի և բարձրութիւն երկարութիւնները, ստանում ենք մակերեսի արժեքը։ Յօրդ տողուած դրույուած PRINT հրահանգը արտածում է տուեալները էկրանի վրայ, իսկ 40րդ տողի ՊRINT հրահանգի միջոցով այդ նոյն տուեալները տպում են թղթի վրայ էթիւ, ի հարկէ, միացած է նաեւ տպող սարքը։

Մինչեւ այժմ դիտարկուած թեյսիկ լիդուի հրահանգների միջոցով կարելի է կազմել գծային ծրագրեր։ Մասնաւորապէս, մենք կորու հետ կոյսիկ այն ինդքի լուծման ծրագրերը, որի ալգորիթմը բերուած է նկ. ԱՅ (տես նկ. ԱՅ)։

ԽՍԴԻՐ

Կազմել ծրագիր ա, բ, ս կողմեր ունեցուի եռամկեան բարձրութիւններ հաշուելու համար։

Ծրագիր

10 INPUT "Ենթածենք եռամկեան կողմեր", A, B, C

20 P=(A+B+C)/2

30 T=SQR (P*(P-A)*(P-B)*(P-C))

40 H1=2*T/A

50 H2=2*T/B

60 H3=2*T/C

70 PRINT "բարձրութիւնները="; H1, H2, H3

80 STOP

Նկ. 83

Հարցեր եւ վարժութիւններ

1. Որո՞նք են արժեքների վերագրման հրահանգները Բեյսիկում:
2. Հետևեալ վերագրման հրահանգներից լմտրէք նիշտ գրուածները՝
 - ա) 10 LET Y=X+Y
 - բ) 10 LET A="շենք"
 - գ) 10 LET B\$="աշակերտ"+ "ղպրոց"
 - դ) 10 LET C\$="Քաղաքագլուխ"- "գլուխ"
 - ե) 10 K\$="հզօրութիւն"+ "123, 96"
 - զ) 10 LET L\$="հոսանք"+4
3. Ի՞նչ արժեքներ կը ստանան X եւ Y փոփոխականները
10 READ A, B, C
20 RESTORE
30 READ X, Y
40 DATA 1, 2, 3, 4, 5, 6
հրահանգների կատարումից յետոյ:
4. Ի՞նչ արժեք կը ստանայ Q փոփոխականը
10 DATA 15, 27, 38, 60
20 READ Q, W, Q, R
հրահանգների կատարումից յետոյ:
5. Տրուած են եռամկեան երեմ՝ կողմերի երկարութիւնները՝ a, b, c; Կազմել ծրագիր այդ եռամկեան միջնագիծը հաշուելու համար: Ցուցում՝ միջնագիծը հաշուելու համար կտրելի է, օգտուել հետեւեալ բանաձեւերից.
 $m_a = 0, 5 \sqrt{2b^2 + 2c^2 - a^2}$
 $m_b = 0, 5 \sqrt{2a^2 + 2c^2 - b^2}$
 $m_c = 0, 5 \sqrt{2a^2 + 2b^2 - c^2}$
6. Տրուած է շրջանագծի երկարութիւնը: Հաշուել շրջանի մակերեսը: Արդիւնքները տպի թղրի վրայ:
7. Գտնել A(X1, Y1) և B(X2, Y2) ծայրակետեր ունեցող հատուածի երկարութիւնը: Արդիւնքը տպի Յ տասմորդական նիշի թշութեամբ:
8. Մեկնարանութիւնները Բեյսիկ ծրագրում

Մրագիրը մատչելի դարձնելու նախապայմաններից մէկը նրա մէջ մեկնարանութիւնների օգտագործումն է, որը կատարում է REM հրահանգի միջոցով (remark – մեկնարանութիւն):
Մեկնարանութիւնը մի տեքստ է, որը անդադրում է ծրագրում՝ կողաւորման որոշակի հայեցակերպ պարզաբանման համար: Հաշուիչ մեքենայի կողմից REM

հրահանգը չի մշակուում, մեկնարանութիւնները ծրագրի աշխատանքի վրայ չեն ազդուում:

Օրինակ՝

10 REM Թառակուսի հաւասարումների

20 REM լուծման ծրագիր

...

50 REM Դիսկրիմինանտի հաշուարկը

...

80 REM Սրմատմերի հաշուումը

...

Ինչպէս տեսնուում ենք, մեկնարանութիւնների միակ նպատակը ծրագրի ճիշտ հասկցուումն է: Դրանք կարող են տեղադրուած լինել ծրագրի ցանկացած մասուում:

6. 15. STOP եւ END հրահանգները

STOP հրահանգը լնդհատում է ծրագրի աշխատանքը (stop – կանդնիր), առաջներով համապատասխան հաղորդագրութիւնը էկրանի վրայ: Այս հրահանգը կարող է տեղադրուել ծրագրի ցանկացած մասուում: Այն չափ յաճախ օգտագործուում է ծրագրերի ճշգրտման նպատակով:

Եթէ ծրագրի աշխատանքն լնդհատուել է STOP հրահանգով, ապա այն կարելի է շարունակել CONT հրահանգի միջոցով (continue – շարունակել):

END հրահանգը ազդաբարում է ծրագրի աւարտը (end – վերջ): Սովորաբար, այն տեղադրում է ծրագրի վերջուում:

Նշենք, որ STOP եւ END հրահանգները ամենեւին կը պարագիր չեն Բեյսիկ – ծրագրուում:

6. 16. Ղեկավարման հրահանգներ

Ինչպէս արդէն նշել ենք, Բեյսիկ լեզուով գրուած ծրագիրն իրենից ներկայացնում է հրահանգների յաջորդականութիւն, որոնք հաշուիչ մեքենան կատարուում է յաջորդաբար, համաձայն նրանց առղերի համարների: Սակայն բազմաթիւ խնդիրներում անհրաժեշտութիւն է առաջանուում, կափուած որոշակի պայմանի բաւարարումից, կրկնել կամ լշջանցել հրահանգների որեւէ խումբը: Դա վերաբերուում է ճիշդաւորուած եւ ցիկլային ալգորիթմներին ու ծրագրերին:

Բեյսիկ լեզուն օժտուած է ղեկավարումը ծրագրի այս կամ այն մասին փոխակցելու մի քանի հնարաւորութիւններով, որոնք թոյլ են տալիս փոփոխել ծրագրի կատարման բնական ընթացքը:

Անպայման անցման հրահանգ

Առանց որեւէ պայման ստուգելու ղեկավարումը ծրագրի ցանկացած տողին փոխանցելու համար նախատեսուած է GO TO հրահանգը (go to – գնալ դէպի)

GO TO n

որտեղ ուշ ծրագրի այն տողին համարն է, որին անցում է կատարուում (դրական ամբողջ թիւ):

GO TO Հրահանգը հնարաւորութիւն է տալիս անմիջապէս անցնել ծրագրի նշուած տողին: Մրագրի աշխատանքը շարունակւում է յաջորդաբար սկսած այն տողից, որին անցում է կատարուել:

Օրինակ՝

30 GO TO 80
40
50
60
70
80
.
120 GO TO 40

Նշենք, որ GO TO Հրահանգը կարելի է օգտագործել նաև անմիջական ռեժիմում ծրագրի ցանկացած տողից աշխատանքը սկսելու համար (RUN հրամանի փոխարէն):

Հնտրութեամ հրահանգ

Որոշ ծրագրերում ճիւղաւորումը կազմակերպելու համար նկատակայարմար է օգտագործել փոխարկիչ, որի գործողութիւնը Բեյսիկում կատարում է ընտրութեան Հրահանգը՝

ON A GO TO n_1, n_2, \dots, n_k

կամ

ON A THEN n_1, n_2, \dots, n_k

(ON ... GO TO- այս դէպօւմ գնալ դէպի), որտեղ A-ն թուարանական արտայայտութիւն է, n, n, \dots, n - ծրագրի տողելի համարներն են (դրական ամբողջ թուեր):

Այս Հրահանգում կարելի է օգտագործել եւ՝ GO TO եւ՝ THEN (ապա) ծառայողական բառերը:

ON GO TO Հրահանգը կատարում է հնտեւեալ հերթականութեամբ՝ սկզբում հայտուում է A թուարանական արտայայտութեան արժէքը եւ վերցւում նրա ամբողջ մասը՝ ու եթէ $m=1$, ապա դեկավարումը փոխանցւում է ո համարն ունեցող տողին, եթէ $m=2$, ապա՝ ո - տողին եւ այլն: Նշենք, որ եթէ $m < 1$ կամ $m > k$, ապա կատարում է ON GO TO Հրահանգին անմիջապէս յաջորդող Հրահանգը:

Օրինակ՝

60 ON B+C THEN 200, 150, 300, 20

70

Տուեալ դէպօւմ փոխարկիչը աշխատում է այսպէս.

Եթէ $B+C=1$, ապա անցնել 200րդ տողին,

Եթէ $B+C=2$, ապա անցնել 150րդ տողին,

Եթէ $B+C=3$, ապա անցնել 300րդ տողին,

Եթէ $B+C=4$, ապա անցնել 20րդ տողին,
Եթէ $B+C<1$ կամ $B+C>4$, ապա անցնել 70րդ տողին:
Կազմենք հնտեւեալ խնդրի լուծման ծրագիրը (նկ. 84):

ԽՆԴԻԲ

Կազմել ծրագիր, որը հնարաւորութիւն է տալիս Անբաժեկայի 1-ից մինչև 7 թուերը, ստանալ ելքում շաբաթուայ համապատասխան որի եթերարթի, երեքշաբթի եւ այլն:

Ծրագիր

```
10 INPUT N
20 ON N GO TO 50, 60, 70, 80, 90, 100, 110
30 PRINT "շաբաթը"; N; "օր չունի"
40 GO TO 10
50 C$="Երկուշաբթի"; GO TO 120
60 C$="Երեքշաբթի"; GO TO 120
70 C$="չորեքշաբթի"; GO TO 120
80 C$="հինգշաբթի"; GO TO 120
90 C$="ուրբաթ"; GO TO 120
100 C$="շաբաթ"; GO TO 120
110 C$="կիրակի"
120 PRINT C$
130 STOP
RUN
?9
շաբաթը 9 օր չունի
?5
ուրբաթ
```

Նկ. 84

Այս ծրագրում օգտագործուած ON GO TO Հրահանգը թիւ (տալիս բար ներածուած N-ի արժէքի անցում կատարել այն տողին, որոկ կազմականին վերագրուում է համապատասխան անուանումը): Եթէ յետոյ անցում է կատարում արտածման PRINT հրահանգին (120րդ տող): Եթէ ունի, որ $N < 1$ կամ $N > 7$ դէպօւմ կատարում է 30րդ տողի գործողութիւնը (Եթէ 1ի վրայ առաջանում է "շաբաթը ... օր չունի" հաղորդագրութիւն) եւ այնուահան: Եթէ ում (հստացւում INPUT հրահանգին (առաջարկւում է կրկնել ներածուած շաբաթը մինչև):

Պայմանակիր հրահանգ

Այս Հրահանգն օգտագործուում է ծրագրի ճիւղաւորման համար (Այս Հրահաւորութիւն է տալիս անցում կատարել ծրագրի այս կամ այն ժամանակի պայմանի բաւարարութիւն):

Բերենք պայմանակիր անցման հրահանգի ընդհանուր տեսքը.

IF պայման GO TO տողի համար ELSE հրահանգների շարք
կամ

IF պայման THEN տողի համար կամ հրահանգների շարք
ELSE հրահանգների շարք

Այսուհետ պայմանը իրենից ներկայացնում է տրամաբանական արտայայտութիւն, որն ընդունում է TRUE (ճշմարիտ) կամ FALSE (կեղծ) արժեքները։ Կախուած տրամաբանական արտայայտութեան արժեքից (ի՞ - եթէ), ծրագրի աշխատանքը շարունակում է THEN (կամ GO TO) ճիւղով (եթէ պայմանը բաւարարում է) կամ ELSE (else - այլ կերպ) ճիւղով (եթէ պայմանը չի բաւարարում)։

Օրինակ՝

50 IF A>B THEN U=A ELSE U=B

եթէ A>B, ապա U=Ա, Հակառակ դէպքում U=B։

Այլ կերպ ասած, Ս փոփոխականը ընդունում է A եւ B մեծութիւններից փոքրա-
դոյնի արժեքը։

IF հրահանգը կարող է օգտագործուել նաև առանց ELSE ճիւղի։ Այդ դէպքում,
եթէ պայմանը չի բաւարարում, ծրագրի աշխատանքը շարունակում է պայմանա-
կիր հրահանգին անմիջապէս յաջրգոյն տողից։

Անդրագաւաններ թափ բացարձակ մեծութեան հաշւման ալգորիթմին, որը մենք
արդէն քննարկել ենք (նկ. 57)։

Ինչպէս տեսնում էք բերուած է երկու համարժեք ծրագիր (նկ. 85)։

ԽԵՌԻՐ

Կազմել ծրագիր բայցարձակ մեծութեան հաշուարկի համար $Y = |X|$

Ծրագիր 1

```
10 INPUT X
20 IF X>=0 THEN Y=X ELSE Y=-X
30 PRINT Y
40 END
```

Ծրագիր 2

```
10 INPUT X
20 Y=X
30 IF X<0 THEN Y=-X
40 PRINT Y
50 END
```

Նկ. 85

Առաջին ծրագրում կիրառուելով IF-THEN-ELSE կառուցուածքը (20րդ տող)
ստուգում ենք $X > 0$ պայմանը եւ Y փոփոխականին վերագրում համապատասխա-
նաբար X կամ $-X$ արժեքը։

Երկրորդ ծրագրում նախօրօք Y -ին վերագրուում է X -ի արժեքը (20րդ տող)։
Այնուհետեւ ստուգում է $X < 0$ պայմանը. եւ եթէ այն բաւարարուում է, ապա Y -ի
արժեքը փոխուում է ($Y=-X$), իսկ եթէ չի բաւարարուում՝ Y -ի արժեքը չի փոխուում
եւ մնում է հաւասար X -ի։ Նշենք նաև, որ այսուհետ ԻF հրահանգը օգտագործուած
է առանց ELSE ճիւղի, որը տուեալ դէպքում աւելորդ է։

Դիտարկենք հետեւեալ ծրագիրը (նկ. 86)։

ԽՆԴԻՐ

Հաշուել յ-ի արժեքը՝ x-ի տրուած արժեքի դէպքում։

$$y = \begin{cases} \sin x^2, & \text{եթէ } |x| > 2 \\ \cos x^2, & \text{հակառակ դէպքում} \end{cases}$$

Ծրագիր 1

```
10 INPUT X
20 IF X < -2 GO TO 40
30 IF X <= 2 GO TO 60
40 Y=COS (X*X)
50 GO TO 70
60 Y=SIN (X*X)
70 PRINT X, Y
80 END
```

Ծրագիր 2

```
10 INPUT X
20 IF ABS (X) <= 2 THEN Y=SIN (X*X) ELSE Y=COS (X*X)
30 PRINT X, Y
40 END
```

Նկ. 86

Խնդրի լուծման ալգորիթմը բերուած է նկ. 58։ Այդ ալգորիթմին համապատաս-
խանում է առաջին ծրագիրը, որտեղ չի օգտագործուում ABS ֆունկցիան։ Ակնյալու է,
որ երկրորդ ծրագիրը, որտեղ օգտագործուած եւ ԻF հրահանգի Ճիւղերը աւելի կարճ է եւ հականալի։

Թաջորդ խնդրի (նկ. 87) համապատասխան ալգորիթմը բերուած է նկ. 59։

ԽՆԴԻՐ

Կազմել ծրագրի տրուած 3 քուերից ամենափոքրը որոշելու համար։

Ծրագիր

```
10 INPUT A, B, C
20 U=A
30 IF A < B THEN U=A
40 IF U > C THEN U=C
50 PRINT "Փիգրագոյն թիւը="; U
60 END
```

Նկ. 87

Հարցեր եւ վարժութիւններ

- Ինչպէս են տրում մեկնաբանութիւնները Բեյսիկ ծրագրում:
 - Ինչ արժեք պէտք է ներածուի, որպէսզի հետեւեալ ծրագրի արդիւնքում արտածուի "Եիշտ է" հաղորդագրութիւնը:
- 10 INPUT N
20 M=ABS (N-4)
30 ON M GO TO 40, 50, 60, 70
40 PRINT "Ախալ է": GO TO 80
50 PRINT "Մոտածիր": GO TO 80
60 PRINT "Եիշտ է": STOP
70 PRINT "Եիշտ չէ"
80 PRINT "Փորձիր նորից": GO TO 10

- Կազմել ծրագիր ֆունկցիայի արժեքները հաշուելու համար՝

$$y = \begin{cases} x^2 + 4, & 5x - 3 \\ \log x, & 4 < x < 10 \\ 4, & \text{հակառակ դեպքում} \end{cases}$$

$$p). \quad y = \begin{cases} \sin^2 x + \cos x, & \text{եթե } |x| \leq 11 \\ 1, & 5^{12} \\ 6x, & \text{հակառակ դեպքում} \end{cases}$$

$$q). \quad y = \begin{cases} xa, & \text{եթե } x^2 + a^2 \leq 1 \\ x + \frac{x+a}{x-a}, & \text{եթե } x^2 + a^2 > 1 \text{ և } x \leq 0 \\ \sqrt{2x} + \frac{2x^2 + a}{3x - xa}, & \text{մնացած դեպքերում} \end{cases}$$

- Տիրոյիր հարրութեամ վրայ տրուած է հետեւեալ հաւասարումներով՝

$$x=0, \quad y=5-x, \quad y=x^3;$$

Որոշել, պատկանո՞ւմ է տրուած A (x₀, y₀) կետը նշուած տիրոյիրն թէ ոչ:

- Հարրութեամ վրայ տրուած է 3 կետ A, B եւ C իրենց կոորդինատներով: Որոշել կազմո՞ւմ են նրանք եռանկիւթի եթէ ոչ: Եթէ այո, ապա հաշուել այդ եռանկիւթեամ:

- Տրուած են x₁, x₂, x₃, y₁, y₂, y₃ իրական թուերը: Պարզել պատկանո՞ւմ է արդիօք կոորդինատների սկզբանկետը (x₁, y₁), (x₂, y₂), (x₃, y₃) գագաթներով եռանկիւթեամ:

- Տրուած են a, b, c, d թուերը: Եթէ նրանք բոլորն ել դրական են, ապա հաշուել եւ տպել նրանց միջին երկրաչափականը, իսկ եթէ նրանցից բեկուզ մեկը բացասական է՝ միջին բուարամեականը:

- Տպել տրուած a, b, c, d, e իրական թուերից ամենամեծի արժեքը:

- Տրուած են x, y, z իրական թուերը: Եթէ x ≤ y ≤ z, ապա իւրաքանչիւր թիւը փոխարինել դրանցից ամենամեծով, եթէ x > y > z, ապա թուերը չփոխել, հակառակ դեպքում բոլոր թուերը փոխարինել իրենց բառակուսիներով:

17. Ցիկլերի կազմակերպման հրահանգներ

Ենչպէս արդէն նշուել էր, ցիկլ է կոչում հրամանների այնպիսի յաջորդականութիւնը, որը ծրագրի կատարման ընթացքում կրկնում է մի քանի անգամ:
Ցիկլ կարելի է կազմակերպել պայմանակիր հրահանգի միջոցով:

ՕՐԻՆԱԿ՝

10 INPUT A, B, H

20 X=A

30 Y= SIN (X) + COS (X)

40 PRINT X, Y

50 X=X+H

60 IF X <= B GO TO 30

70 STOP

30, 40 եւ 50րդ տողերում գրուած հրահանգները կրկնուում են քանի որ ու X ≤ B: Այդ հրահանգները կազմում են ցիկլ:

Բեյսիկ լեզուն օժտուած է յատուկ հրահանգներով ցիկլերի կազմակերպման համար, որոնք բաժանուում են երկու խմբեր՝

1. Ըստ զեկավարող փոփոխականի փոփոխման (FOR-NEXT ցիկլեր),

2. Ըստ պայմանի (WHILE-WEND ցիկլեր):

FOR-NEXT ցիկլի կառուցուածքը հետեւեալն է՝ *

FOR X=K₁ TO K₂ STEP K₃

• • • • • • •

NEXT X

FOR, TO, STEP եւ NEXT ծառայողական բառերը համապատասխանաբար թարգմանուում են՝ համար, մինչեւ, քայլ եւ յաջորդ:

X-ը ցիկլի փոփոխականն է, K_i-ը X-ի սկզբնական արժեքը, K_{i+1}-ը՝ X-ի սահմանային արժեքը, իսկ K_{i+2}-ը՝ քայլը, որով X-ը փոխւում է K_{i+1} մինչեւ K_{i+2}:

FOR-NEXT ցիկլը կատարւում է հետեւեալ կերպ:

1. Ցիկլի փոփոխականին (X-ին) վերագրւում է սկզբնական արժեքը (K_i):
2. Իրագործուում է ցիկլի մարմինը (կատարւում են մինչեւ NEXT հրահանգը գրուած բոլոր հրահանգները):
3. X-ին վերագրւում է X+K, արժեքը եւ այն համեմատւում է K_{i+1} հետ:
4. Եթէ X≤K_{i+1}, ապա պրոցեսը կրկնուում է 2րդ կտրից սկսած (K_{i+1}<0 դէպքում ստուգուում է X≥K_{i+2} պայմանը):
5. Հակառակ դէպքում կատարւում է անցում: NEXT-ին անմիջապէս յաջորդող հրահանգին:

Նշենք, որ նոյն ցիկլի մէջ FOR եւ NEXT հրահանգները պէտք է ուղեկցուին միեւնոյն փոփոխականով:

Իսկ այժմ վերը նշուած ֆունկցիայի արժեքների հաշաման ծրագրը կազմենք FOR-NEXT կառուցուածքի օգնութեամբ:

```

10 INPUT A, B, H
20 FOR X=A TO B STEP H
30 Y=SIN (X)+COS (X)
40 PRINT X, Y
50 NEXT X
60 STOP
RUN
?1, 2, 0. 5

```

1	1, 38177
1. 5	1. 06823
2	0. 49315

Այս ցիկլը կատարւում է 3 անգամ՝ X փոփոխականի 1, 1. 5 եւ 2 արժեքների դէպքում: Քանի որ ցիկլի աարտի պայմանի ստուգումը կատարւում է ցիկլի վերջում, ուստի այն հրահանգները, որոնք կազմում են ցիկլի մարմինը, առնուազն մէկ անգամ կ'իրագործուեն K₁, K₂ եւ K, պարամետրերի ցանկացած արժեքների դէպքում: Դրա մէջ համոզուելու համար նորից կատարենք բերուած ծրագրը, այս անգամ ներածելով այնպիսի արժեքներ, որ A>B եւ H>0:

```

RUN
?2, 1. 5
2
0. 49415

```

K_i-ը, K_{i+1}-ը եւ K_{i+2}-ը կարող են լինել ոչ միայն հաստատուններ եւ փոփոխականներ, այլ նաև թուարանական արտայայտութիւններ:

Օրինակ՝

FOR J=R+2 TO R+10 STEP R/2

Եթէ քայլը՝ K_i=1, ապա այն կարելի է գնչել՝ | 0.0 | + | 0.0 |

Ցիկլերը կարող են ներդրուած լինել մէկը միասի մէջ և ապամ ներքին ցիկլերը պէտք է ամբողջութեամբ տեղաւորուած լինեն բայ սիկլերի ներսուած եւ ունենան տարբեր ղեկավարող փոփոխականներ:

Այժմ կազմենք մի քանի խնդիրների լուծման հրացրել կորիմները դիտարկուած են 4. 4 բաժնում:

Նկ. 92 դիտարկուած խնդրի լուծման ալգորիթմը բեր ա (1. 6 00)

Խնդիր

Հաշուել [0,5] հասուածի վրայ x=h քայլով հետեւեալ դունկցիայի արժեքները:

$$y = \begin{cases} 2x^2+x+1 & , \text{ եթէ } x \leq 2 \\ x^2 - 2 & , \text{ եթէ } 2 < x \leq 5 \end{cases}$$

Ծրագիր 1

```

10 INPUT "Աերածէք քայլը"; H
20 FOR X=0 TO 2 STEP H
30 Y=2*X^2+X+1
40 PRINT X, Y
50 NEXT X
60 FOR X=H+2 TO 5 STEP H
70 Y=SQR (X^2 - 2)
80 PRINT X, Y
90 NEXT X
100 STOP

```

Ծրագիր 2

```

10 INPUT "Աերածէք քայլը"; H
20 FOR X=0 TO S STEP H
30 IF X<=2 THEN Y=2*X^2+X+1 ELSE Y=SQR (X^2 - 2)
40 PRINT X, Y
50 NEXT X
60 STOP

```

Նկ. 92

Ներկայացուած առաջին ծրագրում կազմուած է երկու ցիկլ՝ փունկցիայի արժեքների տարբեր բանաձեւերով հաշւման համար: Նկատենք, որ երկրորդ ցիկլում (60րդ տող) X-ի սկզբնական արժեքը ընկունուած է 2+H, քանի որ X=2 դէպքում Y-ի արժեքը Y=2X²+X+1 բանաձեւով հաշւընում է առաջին ցիկլում:

Երկրորդ ծրագիրը պարունակում է մէկ ցիկլ (X-ը փոխում է 0-ից մինչեւ 5), որի մարմնում անհամեշտ բանաձեւի ընտրութիւնը կատարւում է IF-THEN-ELSE հրահանգի միջոցով:
Նկ. 93 և նկ. 94 բերուած են զումարի եւ արտադրեալի հաշւման ծրագրերը:

ԽՆԴԻԲ

$$\text{Հաշուել } \text{զումարը } y = i^2 \quad i=1$$

Մրագիր

```
10 REM թևական բուերի քառակուսիների
20 REM զումարի հաշուարկ
30 INPUT N
40 Y=0
50 FOR I=1 TO N
60 Y=Y+I^2
70 NEXT I
80 PRINT "ՊՈՒՄԱՐԸ="; Y
90 STOP
```

Նկ. 93

ԽՆԴԻԲ

Հաշուել $y=n!$

Մրագիր

```
10 REM ֆակտորիալի հաշուարկը
20 INPUT N
30 Y=1
40 FOR I=1 TO N
50 Y=Y*I
60 NEXT I
70 PRINT N; "ՓԱԿՏՈՐԻԱԼԸ="; Y
80 STOP
```

Նկ. 94

Ցիկլեցնենք, որ զումարը կամ արտադրեալը կուտակելու նպատակով ցիկլ կաղմակաբելոց առաջ, զումարի սկզբնական արժեքը ընդունում ենք 0, իսկ արտադրեալինը՝ 1:

WHILE-WEND ցիկլի կառուցուածքը հետեւեալն է՝
WHILE պայման (տրամաբանական արտայայտութիւն)

· · · · ·

WEND

Այսպիսի ցիկլի հրահանդների շարքը կատարւում է այնքան անգամ, որքան անհրաժշտ է, որպէսզի WHILE ծառայողական բառից յետոյ նշուած պայմանը դաղարի ճիշտ լինելուց (while - մինչ): Ի տարբերութիւն FOR-NEXT ցիկլի, այստեղ ստուգում է շարքի կատարումից առաջ, բայց ոչ կատարման լինթացքում: Դա նշանակում է, որ եթէ հենց սկզբից պայմանը տիղի չունի, ապա շարքը չի կատարւում ոչ մի անգամ:

WEND հրահանգը, որը տրում է տուանց պարամետրի, ցոյց է տալիս ցիկլի վերջը:

ՕՐԻՆԱԿ՝

$\text{Հաշուել } y=\sqrt{x}$ Փունկցիայի արժեքը տրուած $\epsilon > 0$ ճշտութեամբ:

Ի հարկէ, կարելի է օգտագործել SQR ֆունկցիան, սակայն մենք կը կազմենք ծրագիր, որը թոյլ կու տայ յաջորդաբար մօտենալ լուծմանը, կազմելով y_0 , y_1 , y_2 , ..., y_k , y_{k+1} ... յաջորդականութիւն, որտեղ y_{k+1} -ը աւելի քիչ է տարբերւում x -ից քան y_k -ն: Յաջորդականութիւնն հերթական անդամը կարելի է հաշուել

$$y_{k+1} = 0.5(y_k + x/y_k) \text{ բանաձեւով}$$

Որպէս առաջին անգամ կը վերցնենք $y_0=x/2$ արժեքը, իսկ հաշուարկները կը շարունակենք այնքան, մինչեւ չբաւարարուի տրուած ճշտութիւնը, այսինքն զանուի այս y_n -ը, որ $|y_{n+1}-y_n| < \epsilon$:

Յաջորդականութեան հերթական անդամը հաշուելու համար նպատակայարմար է կազմակերպել WHILE-WEND կառուցուածքի ցիկլ, որի աւարտի պայմանը հենց կը լինի $|y_{n+1}-y_n| < \epsilon$:

Նկատենք, որ FOR-NEXT կառուցուածքի ցիկլը հնարաւոր չէ կիրառել այս խնդիրի լուծման համար, քանի որ անորոշ է յաջորդականութեան անդամների քանակը: WHILE-WEND ցիկլը նախատեսուած է հիմնականում որպէս այսպիսի խընդիրների լուծման միջոց:

Բերենք առաջարկուած ինդրի լուծման ծրագիրը.

10 INPUT "Անրածեք արգումենտը եւ նշուութիւնը"; X, E

20 Y=X/2

30 Y0=X

40 WHILE ABS(Y-Y0)>E

50 Y0=Y

60 Y=(Y0+X/Y0)/2

70 WEND

80 PRINT "ԱՐՄԱՆ"; X; "="; Y

90 STOP

RUN

Ներածեք արգումենտը եւ ճշտութիւնը 73, 0.000001
արման 3=1. 732051

Նկատենք, որ Ե եւ ՅՕ փոփոխականներին մինչեւ ցիկլի մէջ մտնելը վերագրել ենք այնպիսի արժեքներ, որ 40րդ տողում զրուած պայմանը բաւարարուի եւ ցիկլը կատարուի:

Ստորեւ (նկ. 95, 96, 97, 98) բերուած են մի շարք խնդիրների լուծման ծրագրեր, որոնց ալգորիթմները գիտարկուած են նկ. 68, 69, 70, 71:

Առաջարկում ենք իրականացնել ծրագրերը ԷՀՄ-ի միջոցով եւ վերլուծել ստացուած արդիւնքները:

ԽՆԴԻՐ

Փոփոխելով x արգումենտի արժեքը հ ֆայլով |a, b| հատուածի վրայ, հաշուել $y = \sin(\cos(x))$ փունկցիայի դրական արժեքների գումարը (s) եւ այդ կետերի քանակը (k):

Սրագիր 1

```

10 INPUT "ԳԵՐԱՁԵՒ A, B, H"; A, B, H
20 IF A>B OR H<=0 THEN PRINT "Սխալ է"; GO TO 10
30 S=0
40 K=0
50 FOR X=A B STEP H
60 Y=SIN (COS (X))
70 IF Y>0 THEN S=S+Y ; K=K+1
80 NEXT X
90 PRINT "S=";S, "K=";K
100 STOP

```

Սրագիր 2

```

10 INPUT A, B, H
20 . . . . .
30 S=0
40 K=0
50 X=A
60 WHILE X<=B
70 Y=SIN (COS (X))
80 IF Y>0 THEN S=S+Y; K=K+1
90 X=X+H
100 PRINT "S="; S, "K="; K
120 STOP

```

Նկ. 95

ԽՆԴԻՐ

$$\text{Հաշուել } y = \sum_{k=1}^n (-1)^k \frac{k^2 + 2}{5}$$

$$2^{n+1} = 2^n \cdot 2$$

Սրագիր

```

10 INPUT M
20 Y=0
30 A=1
40 Y=0
50 A=A*2
60 Y=Y+A
70 NEXT I
80 PRINT "Y="; Y
90 STOP

```

Նկ. 96

ԽՆԴԻՐ

$$\text{Հաշուել } y = \sum_{k=1}^n (-1)^k \frac{k^2 + 2}{5}$$

Սրագիր

```

10 INPUT N
20 Y=0
30 Z=1
40 FOR K=1 TO N
50 Z=Z
60 Y=Y+Z*(K^2+2)/5
70 NEXT K
80 PRINT "Y="; Y
90 STOP

```

Նկ. 97

ԽՆԴԻԲ

$$\text{Հաշուել կրկնակի գումարը } S = \sum_{i=1}^n \sum_{j=1}^m ij$$

Ծրագիր

```

10 INPUT N, M
20 S = 0
30 FOR I=1 TO N
40 FOR J = 1 TO M
50 S = S + I * J
60 NEXT I
70 NEXT I
80 PRINT "ԳՈՒՄԱՐԸ="; S
90 STOP

```

Նկ. 98

6. 18. Զանգուածների մշակման ծրագրեր

ՄԵՆՔ արդյոք ծանօթ ենք! DIM հաստիկի հետ, որի միջոցով նկարագրուած են ինֆորմացիոն զանգուածները թեյսիկ ծրագրերում: Աւսումնասիրել ենք նաև զանգուածի առանձին տարրին գիտելու հնարաւորութիւնները նրա կարգահամարների օգնութեամբ: Այս պարագրաֆում կը պատարկենք մի շարք խնդիրների լուծման ծրագրեր, որոնք յանդում են մէկ կամ երկշափանի զանգուածների մշակմանը:

Նկ. 99 ներկայացուած խնդրի լուծման աղյուրիմը բերուած է նկ. 74:

Ուշադրութիւն դարձրէք վեկառորի տարրերի ներածման վրայ: Սկզբում անհրաժշտ է ներածել աարբերի բանակը (30րդ տող): Այնուհետեւ կազմակերպւում է ցիկլ, որի պարամետրը (I-ն) փոփոխուելով կը մինչի Ն սահմանում է ներածուող հերթական տարրի կարգահամարը: Իրադուժենք առաջին ծրագիրը, որը բերուած է նկ. 99:

RUN

Ներածէք տարբերի քանակը ? 5

X (1)=? 17

X (2)=? - 6

X (3)=? 4

X (4)=? - 3

X (5)=? 6

Գումարը = 18

Նկ. 100, 101, 102, 103, 104, 105 ծրագրերը բերուած են առանց մեկնաբանութեան, որովհետեւ այս խնդրների լուծման ալգորիթմները անհրաժեշտ վերլու-

ծութիւններով ներկայացուած են 75, 76, 77, 78, 79, 80, 81 նկարներում: Առաջարկում ենք իրականացնել ծրագրերը էջՄ-ի միջոցով եւ վերլուծել ստացուած արդիւնքները:

ԽՆԴԻԲ

Հաշուել X(x₁, x₂, ..., x_n) վեկտորի տարրերի գումարը եւ արտադրեալ:

$$S = \sum_{i=1}^n x_i \quad P = \prod_{i=1}^n x_i$$

Ծրագիր 1

```

10 REM վեկտորի տարրերի գումար
20 DIM X(100)
30 INPUT "Աերածէք տարրերի քանակը"; N
40 FOR I=1 TO N
50 PRINT "X ("; I; ")=";
60 INPUT X (I)
70 NEXT I
80 S=0
90 FOR I=1 TO N
100 S=S+X (I)
110 NEXT I
120 PRINT "ԳՈՒՄԱՐԸ="; S
130 STOP

```

Ծրագիր 2

```

10 REM վեկտորի տարրերի արտադրեալ
20 DIM X (100)
30 INPUT N
40 FOR I=1 TO N
50 INPUT X (I)
60 NEXT I
70 P=1
80 FOR I=1 TO N
90 P=P*X (I)
100 NEXT I
110 PRINT "ԱՐՏԱԴՐԵԱԼԸ="; P
120 STOP

```

Նկ. 99

ԽՆԴԻԲ

Որոշել $A(a_1, a_2, \dots, a_n)$ վեկտորի ամենամեծ տարրի համարը եւ արժեքը:

Մրագիր 1

```

10 REM վեկտորի մեծագոյն տարրի որոշում
20 DIM A(100)
30 INPUT N
40 FOR I=1 TO N
50 INPUT A (I)
60 NEXT I
70 K=1 ; Y=A (I)
80 FOR I=2 TO N
90 IF Y>=A (I) GO TO 120
100 K=I
110 Y=A (I)
120 NEXT I
130 PRINT "մեծագոյն տարրն է A ("; K;")="; Y
140 STOP

```

Մրագիր 2

```

10 REM վեկտորի մեծագոյն տարրի որոշում
20 DIM A (100)
30 INPUT N
40 FOR I=1 TO N
50 INPUT A (I)
60 NEXT I
70 I=1
80 FOR J=2 TO N
90 IF A (I)>=A (J) GO TO 110
100 I=J
110 NEXT J
120 PRINT "A ("; I;")="; A (I)
130 STOP

```

Նկ. 100

ԽՆԴԻԲ

Կարգաւորել $A(a_1, a_2, \dots, a_n)$ վեկտորի տարրերը ըստ նույզման:

Մրագիր 1

```

10 REM վեկտորի կարգաւորման
20 DIM A(100)
30 INPUT N "Յերածէք տարրերի բանակը"; N
40 FOR I=1 TO N
50 INPUT A (I)
60 NEXT I
70 K=1 ; TO N-1
80 I=K
90 FOR J=I+1 TO N
100 IF A (I) >=A (J) GO TO 120
110 I=J
120 NEXT J
130 X=A (K) : A (K)=A (I) : A (I)=X
140 NEXT K
150 REM կարգաւորուած վեկտորի արտածում
160 FOR I=1 TO N
170 PRINT A (I)
180 NEXT I
190 STOP

```

Մրագիր 2

```

10 REM Պղպջակի մեթոդ
20 DIM A (100)
30 INPUT N
40 FOR I=1 TO N
50 INPUT A (I)
60 NEXT I
70 P=0
80 FOR I=1 TO N-1
90 IF A (I)>=A (I+1) GO TO 120
100 X=A (I) : A (I) : A (I+1) : A (I+1)=X
110 P=1
120 NEXT I
130 IF P=1 GO TO 70
140 FOR I=1 TO N
150 PRINT A (I);
160 NEXT I
170 STOP

```

Նկ. 101

ԽԵՐԱԳԻՒԹ

Տրուած $X(x_1, x_2, \dots, x_n)$ վեկտորից հետացնել գրուական բաղադրիչները:

Մրագիր

```

10 DIM X (100), Y (100)
20 INPUT "N="; N
30 FOR I=1 TO N
40 INPUT X (I)
50 NEXT I
60 J=0
70 FOR I=1 TO N
80 IF X (I)=0 GO TO 110
90 J=J+1
100 X (J)=X (I)
110 NEXT I
120 FOR I=1 TO J
130 PRINT Y (I)
140 NEXT I
150 STOP

```

Նկ. 102

ԽԵՐԱԳԻՒԹ

Կառուցել $C(c_1, c_2, \dots, c_m)$ վեկտորը, որի տարրերը ստացւում են տրուած $A(a_1, a_2, \dots, a_n)$ և $B(b_1, b_2, \dots, b_n)$ վեկտորների տարրերից յաջորդելով միմեանց.

$(c_1=a_1, c_2=b_1, c_3=a_3, c_4=b_4, \dots, c_{m+1}=a_n, c_{2n}=b_n)$:

Մրագիր

```

10 DIM A (100), B (100), C (200)
20 INPUT "N="; N
30 FOR I=1 TO N
40 INPUT A (I), B (I)
50 NEXT I
60 J=1
70 FOR I=1 TO N
80 J=J+2
90 C (I)=A (I)
100 C (I+1)=B (I)
110 NEXT I
120 FOR I=1 TO 2*N
130 PRINT C (I);
140 NEXT I
150 STOP

```

Նկ. 103

ԽՆԴԻՐ

Հաշուել $X=\prod_{i=1}^n x_i$, $i=1, n$, $j=1, m$ մատրիցի տարրերի և գումարը (տող առ տող) եւ Պ արտադրեալը (սիւն առ սիւն):

$$S = \sum_{i=1}^n \sum_{j=1}^m x_{ij} \quad P = \prod_{j=1}^m \prod_{i=1}^n x_{ij}$$

Մրագիր 1

```

10 REM Մատրիցի տարրերի գումար
20 DIM X(50, 50)
30 INPUT "Անքանութեան N, M"; N, M
40 FOR I=1 TO N
50 FOR J=1 TO M
60 PRINT "X("; I; ", "; " J)=";
70 INPUT X (I, J)
80 NEXT J
90 NEXT I
100 S=0
110 FOR I=1 TO N
120 FOR J=1 TO M
130 S=S+X (I, J)
140 NEXT J
150 NEXT I
160 PRINT "Գումարը="; S
170 STOP

```

Մրագիր 2

```

10 REM Մատրիցի տարրերի արտադրեալ
20 DIM A (50, 50)
30 INPUT "Անքանութեան N, M"; N, M
40 FOR I=1 TO N
50 FOR J=1 TO M
60 INPUT X (I, J)
70 NEXT J
80 NEXT I
90 P=1
100 FOR J=1 TO N
110 FOR I=1 TO M
120 P=P*X (I, J)
130 NEXT I
140 NEXT J
150 PRINT "Արտադրեալը="; P
160 STOP

```

Նկ. 104

ԽՆԴԻՐ

Ապրաւորել $A=IIa_{ij}$, $i=1, n$, $j=1, m$ մատրիցի տողերը ըստ 1րդ սեան տարրերի անման:

Մրագիր

```

10 REM Մարիցի տողերի կարգաւորում
20 DIM A(50, 50)
30 INPUT "ԱԵՐԱԾԵԼ N, M"; N, M
40 FOR I=1 TO N
50 FOR J=1 TO M
60 INPUT A (I, J)
70 NEXT J
80 NEXT I
90 INPUT "ԱԵՐԱԾԵԼ L"; L
100 FOR K=1 TO N-1
110 I=K
120 FOR J=I+1 TO N
130 IF A (I, L)>=A (J, L) GO TO 150
140 I=J
150 NEXT J
160 FOR J=I TO M
170 X=A (K, J) : A (K, J)=A (I, J) : A (I, J)=X
180 NEXT J
190 NEXT K
200 FOR I=1 TO N
210 FOR J=1 TO M
220 PRINT A (I, J);
230 NEXT J
240 PRINT
250 NEXT I
260 STOP

```

Ակ. 105

6. 19. Օգտուողի կողմից սահմանուած ֆունկցիաներ

Ցածախ ծրագրում անհրաժեշտ է լինում հաշուել որեւէ ֆունկցիայի արժեքները մի քանի անգամ՝ արգումենտների տարրեր արժեքների դէպքում: Արագաւորման ԲԵՋԻԼ լեզում հարաւորութիւն է ընձեռնուած մէկ անգամ նկարագրել օգտուողի սահմանած ֆունկցիան եւ բազմիցս այն օգտագործել:

Օգտուողի կողմից սահմանուած ֆունկցիաների նկարագրումը կատարւում է DEF հրահանգի միջոցով (DEFINE - սահմանել), որի ընդհանուր տեսքը հետևեալն է:

DEF FN անուն (X₁, X₂, ..., X_n)=A
որտեղ "FN անուն" - ֆունկցիայի անուանումն է, որինակ՝ FNA, FNY եւ այլն, X₁, X₂, ..., X_n պարամետրերի ցուցակն է, A-ն այն արտայայտութիւնն է, որը սահմանում է ֆունկցիան նշուած պարամետրերի նկատմամբ:

Օրինակ՝

20 DEF FNK (A, B)=SIN (A)-(COS) B

Եթէ նկարագրուած ֆունկցիայի արժեքը պէտք է հաշուել երազրի որեւէ մասում, ապա կարելի է դիմել այդ ֆունկցիային իր անուանման միջոցով, փակագծերում նշելով արգումենտների այն կոնկրետ արժեքները, որոնք տեղադրուելու են վերագրման նշանի աջ մասում զրուած արտայայտութիւնն մէջ՝ ՀԱՅԹՎԱՊԱՍԽԱՆ պարամետրերի փոխարէն:

Օրինակ՝

100 U=FNK (0, 3, 0, 7)

Այս հրահանգի կատարումից յետոց Ս փոփոխականը կրու ստանայ SIN (0, 3) - COS (0, 7) արժեքը:

ԽՆԴԻՐ

$\zeta_{\text{աշուել}} \quad R = \sqrt{2x^3+x^2+\cos^2 x} + \sqrt{2y^3+y^2+\cos^2 y} = \sqrt{x^6+x^4+\cos^4 x}$
արտայայտութեան արժեքը տրուած x , y , եւ z -ի դէպքում:

Ինչպէս տեսնում ենք, արմատատակ արտայայտութիւնը երեք դէպքում էլ նոյնանման է: Այդ պատճառով յարմար է մէկ անգամ այն նկարագրել որպէս ֆունկցիա եւ երեք անգամ օգտագործել:

Մրագիր

10 DEF FNG (A)=SQR ((2*A^3+A*A+COS (A)^2)

20 INPUT X, Y, Z

30 R=FNG (X)+FNG (Y)+FNG (Z)

40 PRINT R

50 STOP

6. 20. ՍԱՐԱԾՐԱԳԻՐ

Տարբեր բնագաւառներից խնդիրներ լուծելիս յանձնի անհրաժեշտ է լինում մի քանի անգամ օգտագործել միեւնոյն ծրագրային րլուկը որոշակի պարամետրերի տարրեր արժեքների նկատմամբ: Նման դէպքերում ծրագրի հաւաքը փորձացնելու համար նպատակայարմար է այդ կրկնուող հաշուարկները տառանձնացնել իւ անհրաժեշտութեան դէպքում դիմել նրանց: Այդպիսի ծրագրային րլուկերն անուանում են ենթածրագրեր:

ԲԵՐՅԱԲԼՈՒՄ ԵՆԹԱԾՐԱԳԻՐԸ յատուկ նորասակով կազմուած հրահանգների համախումբ է, որին կարելի է դիմել ծրագրի ապրեր մասերից:

ԵՆԹԱԾՐԱԳՐԻ կատարման համար անհրաժեշտ է փոխանցել նրան ղեկավարումը, իսկ կատարումից յետոյ՝ շարունակի հրմական ծրագրի աշխատանքը: ԵՆԹԱԾՐԱԳՐԻ դիմելու համար նախատեսուած է GOSUB ո հրահանգը (go - զնալ, subroutine - ենթածրագրի):

Ո-ր ԵՆԹԱԾՐԱԳՐԻ մուտքը որոշագ առաջ համարն է: Նշենք, որ ԵՆԹԱԾՐԱԳՐԻ աշխատանքը կարելի է սկսել նրա ցանկացած տողից:

Այսպիսով, GOSUB հրահանգը ղեկավարում ի փոխանցում է ԵՆԹԱԾՐԱԳՐԻ ո համարն ունեցող տողին և աշխատանքը շարունակում է մինչեւ RETURN հրահանգին հանդիպելը (return - վերադարձ), որից յետոյ կատարում է վերագրձ ծրագրի այն հրահանգին, որը անմիջապէս յաջորդում է GOSUB ո հրահանգին: Խրաքանչիւր ԵՆԹԱԾՐԱԳՐԻ պէտք է ունենայ վերագրձի հրահանգը, ըստ որում մէկ ԵՆԹԱԾՐԱԳՐՈՒՄ կարող են հանդիպել մի քանի RENTURN հրահանգներ, եթէ դա պահանջում է նրա արամարանութիւնը:

Ամփոփելով առուածր, ենթածրագրի օգասպործումը կարելի է նկարագրել այսպէս՝

1. Ծրագրի կատարման ընթացքում հանդիպում է GOSUB ո հրահանգը՝ ԵՆԹԱԾՐԱԳՐԻ դիմելու համար:

2. ԵՆԹԱԾՐԱԳՐԻ դիմելուց առաջ յիշում է GOSUB ո հրահանգը պարունակող տողի համարը (վերագրձի հասցեն), որին անհրաժեշտ է ԵՆԹԱԾՐԱԳՐԻ կատարումից յետոյ ղեկավարումը հրմական ծրագրի ճիշտ կէալ փոխանցելու համար: Սրանով էլ ՀԵՆԳ ապրերում է GOSUB հրահանգը (ՅՈՒՇ-ից):

3. Կատարում են ԵՆԹԱԾՐԱԳՐԻ հրահանգները սկսած ո-րդ տողից:

4. Հանդիպելով RETURN հրահանգին, ենթածրագրիը վերագրձնում է ղեկավարումը GOSUB ո-ին անմիջապէս համեստ հրահանգին:

Լուծենք հետեւեալ խնդիրը (նկ. 106):

Ցանկացած եռանիշ թիւ կարելի է ներկայացնել ABC տեսքով, որտեղ A-ն, B-ն եւ C-ն նրա թուանշաններն են: Փափոխելով A-ն 1-ից մինչեւ 9, իսկ B-ն եւ C-ն 0-ից մինչեւ 9 ստանում ենք թոյոր եռանիշ թուերը, նրանցից իւրաքանչիւրի համար ստուգիով խնդրում առաջարկուած պայմանը (130րդ տող): Դրա համար անհրաժեշտ է բազմիցու հաջուել տարրեր թուերի ֆակտորեանները: Այդ պատճառով նպատակարմար է ֆակտորեալի հաջուարկը ձեւակերպել որպէս ԵՆԹԱԾՐԱԳՐԻ: ԵՆԹԱԾՐԱԳՐՈՐ սկսում է 200րդ տողից եւ հաջում է N-ի ֆակտորեալը, որը նշանակուած է յ:

ԵՆԹԱԾՐԱԳՐԻ դիմելուց առաջ N փոփոխականը ստանում է իր փաստացի արժէքը՝

20 N=A

.....

60 N=B

.....

100 N=C

ԽՆԴԻՐԸ

Դտմել այն եռանիշ թուերը, որոնք հաւասար են իրենց քուանչանների ֆակտորիալների գումարին:

Օրինակ

145=1!+4!+5!=1+24+120

Ծրագիր

```
10 FOR A=1 TO 9
20 N=A
30 GOSUB 200
40 Q=Y
50 FOR B=0 TO 9
60 N=B
70 GOSUB 200
80 T=Y
90 FOR C=0 TO 9
100 N=C
110 GOSUB 200
120 P=100*A+10*B+C
130 IF Q+T+Y=P THEN PRINT P
140 NEXT C
150 NEXT B
160 NEXT A
170 STOP
180 REM Փակտորիալի հաշուըման ենթածրագիր
200 Y=1
210 IF N=0 THEN RETURN
220 FOR I=1 TO N
230 Y=Y*I
240 NEXT I
250 RETURN
```

Նկ. 106

ԵՆԹԱԾՐԱԳՐԻ կատարումից յետոյ անմիջապէս օգտագործուում է (սուեալ դէպում յիշում) նրա արդիւնքում ստացուած ց փոփոխականը, այլապէս այն "կը կոչուի" (կը փոփարինուի նոր արժէքով) ԵՆԹԱԾՐԱԳՐԻ յաջորդ անգամ դիմելուց յետոյ:

40 Q=Y

....

80 T=Y

....

130 IF Q+T+Y=P THEN PRINT P

Մէկ ծրագիրը կարող է ունենալ մի քանի ենթածրագիր: Նեկավարումը ենթածրագրերից մէկին փոխանցելու համար կարելի է օգտագործել

ON A GOSUB n₁, n₂, ..., n_t

Հրահանգը, որը համանման մեզ արդէն ծանօթ ON ... GOTO հրահանգին:

ON GOSUB հրահանգը կատարում է հետեւեալ կերպ.

Ակցում հաշվում է A թուաբանական արտայատութեան արժէքը եւ վերցում նրա ամրող մասը: Եթէ այն հաւասար է 1, ապա ղեկավարումը փոխանցւում է n₁ համար ունեցող սողով սկսուող ենթածրագրին, եթէ այն հաւասար է 2, ապա՝ n₂ սողով սկսուող ենթածրագրին եւ այլն:

Խնդիրներ իմքնուրոյն լուծելու համար

1. [1, 8] հասուածի վրայ $\Delta x = h$ բայլով հաշուել եւ տպի հետեւեալ ֆունկցիայի արժէքները.

$$y = \begin{cases} \sqrt{x^2 + a^2} & , \text{ եթէ } 1 \leq x < +, 5 \\ \operatorname{tg} x & , \text{ եթէ } 2, 5 \leq x \leq 4 \\ \sin \cos x & , \text{ եթէ } 4 < x \leq 8 \end{cases}$$

2. Տրուած է ո թմական թիւը: Հաշուել $y = 1 + 2 + 3 + 4 + \dots + n(n+1) \dots 2n$ գումարը:

3. Որոշել $y = 4x^2 + 2, 5 \sin 3x$ ֆունկցիայի ամենամեծ արժէքը $[0, 5]$ միջակայքում, փոփոխելով x -ը $h=0, 1$ բայլով:

4. Հաշուել գումարների արտադրեալը՝

$$y = \prod_{i=1}^k \left(\sum_{j=1}^n j' \right)$$

5. Հաշուել այն եռանիշ թուերի քանակը, որոնց թուանշանների գումարը հաւասար է տրուած M թմական թուին:

6. Գտնել այն թուոր եռանիշ թուերը, որոնց եռանիշների գումարը թաժանաւում է 13ի:

7. Գտնել այն թուոր եռանիշ թուերը, որոնի հաւասար է ի իրենց թուանշանների գումարին:

8. 1022 թուին աչից եւ ձախից գծագրել մկանակ թուանից այնպէս, որ ստացուած վեցանիշ թիւը թաժանաւի 7ի, 8ի եւ 9ի:

9. Գտնել տրուած N-ին չգերազանցող թուոր կարգ թուերը:

10. Գտնել տրուած M թմական թուին չգերազանցող թուոր կատարեալ թուերը: Թիւը կոչւում է կատարեալ, եթէ այն հաւասար է իր թուոր թաժանշարթերի (թաժիցից) գումարին:

11. Հաշուել

$$z = \frac{2^{a^2+1} \sqrt[3]{\cos b}}{1 - 2^{b^2}} + \frac{\sqrt[3]{\cos(a+b)}}{2 \sqrt{2^{c^2+1}}} + 0, 1$$

արտայատութեան արժէքը սահմանելով երկու ֆունկցիոնի:

12. Հաշուել Y (y_1, y_2, \dots, y_n) զանգուածի այն դրական տարրերի միջին երկրաչափականը, որոնի քաւարարում են $y_i > a$ պայմաններ: Եթէ այդպիսի տարրեր չկան, տալ համապատասխան հանդրդագրութիւն:

13. Հաշուել A (a_1, a_2, \dots, a_n) եւ B (b_1, b_2, \dots, b_n) վեկտորների սկալար արտադրեալը՝

$$y = \sum_{i=1}^n a_i \cdot b_i$$

14. Հաշուել X (x_1, x_2, \dots, x_n) զանգուածի այն տարրերի համար, որոնց համար ամենամաս ամրող թիւն է հանդիսանում նը:

15. Տրուած է ո ամրող թիւը: Կառուցել A (a_1, a_2, \dots, a_n) վեկտոր հետեւեալ բաղադրիչներով՝

$$n, (n-1), n(n-1)(n-2), \dots, n!$$

16. Գտնել X (x_1, x_2, \dots, x_n) վեկտորի դրական տարրերից ամենափոփոքը:

17. Գտնել Y (y_1, y_2, \dots, y_n) վեկտորի թացանշանը տարրերից ամենամեծը:

18. Տրուած են K թեական թիւն ու $X(x_1, x_2, \dots, x_n)$ վեկտորը, որի տարրերը թեական թիւն են: Կառուցել $Y(y_1, y_2, \dots, y_n)$ վեկտորը, որտեղ y_i հաւասար է K -ի եւ X_i -ի ամենափոքր թեական թագմապատիկին:

19. Բնական M թիւը տրուած է իր երկուական նիշերի $A(a_0, a_1, \dots, a_n)$ գծային աղիւսակի տեսնով.

$$M = a_n - 2^n + a_{n-1} - 2^{n-1} + \dots + a_1 - 2 + a_0, \text{ սրանդ}$$

$$a_0 = 0 \quad \text{կամ} \quad a_0 = 1 \quad (i=0, 1, \dots, n):$$

Սուտայ $M+1$ թիւի երկուական նիշերի B աղիւսակը:

20. Գտնել $\Lambda(a_1, a_2, \dots, a_n)$ զանգուածի 3 ամենամեծ տարրերը:

21. Հաշուել $x=IIx_i II; i=1, n; j=1, m$ մատրիցի դրական, բացասական եւ գրոյական տարրերի քանակը:

22. Արտածել $A=IIa_j II; i=1, n; j=1, m$ մատրիցի գլխաւոր անկիւմագծի դրական տարրերը:

23. Որոշել $A=IIa_j II; i=1, n; j=1, m$ մատրիցի գլխաւոր անկիւմագծից ներքեւ գտնուող տարրերից ամենամեծը:

24. Փոխատեղել $A=IIa_i II; i=1, n; j=1, m$ մատրիցի K -րդ եւ 1-րդ տողերը:

25. Տրուած է ո ամրուց թիւր Կոռուցել $B=IIb II; i, j=1, m$ մատրիցը հետեւեալ բաղադրիչներով՝

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 2 & 2 & \dots & 2 \\ 0 & 0 & 0 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & n \end{pmatrix}$$

26. Տրուած է $X=IIx_i II; i, j=1, n$ մատրիցը: Գտնել նրա բոլոր տողերի մինիմալ տարրերը և փոխատեղի գլխաւոր անկիւմագծի տարրերի հետ:

27. Տրուած է $A=IIa_j II; i=1, n; j=1, m$ մատրիցը: Կառուցել $B(b_1, b_2, \dots, b_k)$ վեկտորը, որտեղ.

$$\text{ա) } b_i = \sum_{j=1}^m a_{ij}; \quad i=1, n$$

$$\text{բ) } b_i = \max a_{ij}; \quad i=1, n; j=1, m$$

$$\text{գ) } b_j \text{ հաւասար է } A \text{ մատրիցի } j\text{-րդ սեան դրական տարրերի քանակին:}$$

6. 21. Դրաֆիկական հրահանգներ

Բեյսիկ լեզուում ընդգրկուած են որոշակի հրահանգներ եւ ֆունկցիաներ, որոնց միջոցով հնարաւոր է տարրեր պատկերներ կառուցել զրաֆիկական տեստիպի: Էլերանի վրայ: Յանկացած պատկեր էկրանի վրայ կարելի է դիտարկել որպէս լուսաւորուած էկտերը համախումը (էկտերը հանգէս են գալիս որպէս պատկերի տարրեր):

Դրաֆիկական տեսաատիպի էկրանը վլտարկուած է որպէս ուղղանկիւն տղիսաակ, որի իւրաքանչիւր տարրը համապատասխանուած է որոշակի էկտերի: Ակնյայտ է, որ պատկերներ կառուցելու համար ծրագրաւորողն պէտք է "հասանելի" լինի էկրանի իւրաքանչիւր էկտեր (նա պէտք է հնարաւորութիւն ունենայ փոխել ցանկացած էկտեր զոյնը):

Տեսապիտի պատկերման հնարաւորութիւնները կախուած են էկտերի քանակից եւ չափսէրից: Ըստ պատկերուող էկտերի քանակի զրաֆիկական տեսաատիպը կարող է աշխատել բարձր եւ միջին թոյլատութեան (խոսցման) ոեժիւններուած:

Միջին խոտացման ոեժիւմում էկրանը բաժանուում է 320×200 էկտերի (200 տող, 320 սիւն), իսկ բարձր խոտացման ոեժիւմում՝ 640×200 էկտերի: Էկտերը համարակալուում են սկսած էկրանի վերին ձափ անկիւնից: Ըստ հորիզոնական առանցքի համարները փոխուում են 0-ից մինչեւ 319 (639), ըստ ուղղահայեաց առանցքի 0-ից մինչեւ 199: Այլ էկտեր ասած որպէս կոորդինատային սկզբնակտ (0, 0) ընդունուած է էկրանի վերին ձափ անկիւնը, իսկ ցանկացած էկտեր միարժէքօրէն որոշուում է իր (x, y) կոորդինատներով:

Օրինակ.

միջին խոտացման ոեժիւմում էկրանի 4 անկիւնների էկտերը ունեն հետեւեալ կոորդինատները՝

վերին ձափ անկիւն (0, 0),

վերին աջ անկիւն (319, 0),

ստորին ձափ անկիւն (0, 199),

ստորին աջ անկիւն (319, 199):

Կոորդինատների տրման այս ձեւը, երբ նրանք հաշւըւում են ըստ կոորդինատային սկզբնակտի, կոչւում է բացարձակ:

Ուոչ գէպերուած յարմար է օգտագործել էկտի յարաբերական կոորդինատները, որոնք արտուած են ըստ նախորդ արտածուած էկտի կոորդինատների: Նշենք, որ իւրաքանչիւր գրաֆիկական հրահանգի կատարումից յատոյ, յիշուում են ստեղծուած համապատասխան պատկերի վերջին էկտի կոորդինատները:

Յարաբերական կոորդինատները տրուում են STEP ծառայողական բառի միջոցով՝ STEP (x, y),

որտեղ x -ը եւ y -ը կարող են ընդունել նաեւ բացասական արժէքներ:

Օրինակ՝

եթէ վերջին արտածուած էկտի բացարձակ կոորդինատները հաւասար են (80, 100), ապա STEP (-20, 10)-ը սահմանում է (60, 110) էկտը:

Բոլոր գրաֆիկական հրահանգներուած կարելի է օգտագործել եւ բացարձակ յարաբերական կոորդինատները:

SCREEN հրահանգը

Այս հրահանգի միջոցով (screen - էկրան) կարելի է ընդունել էկրանի աշխատանքային ռեժիմը՝

SCREEN r, m

r պարամետրը կարող է ընդունել 0, 1 կամ 2 արժեքները:

Եթէ $r=0$, ապա ընտրում է այբբենական-թուային ռեժիմը, որտեղ գրաֆիկական հրահանգները չեն գործում (այս ռեժիմում էկրանի վրայ կարելի է արտածել միայն տեքստային ինֆորմացիա): Արտածուող տուեալների դիրքը սահմանելու համար կարելի է օգտագործել LOCATE հրահանգը, որը մենք ուսումնասիրել ենք:

$r=1$ - միջին խտացման գրաֆիկական ռեժիմ ($320*200$)

$r=2$ - բարձր խտացման գրաֆիկական ռեժիմ ($640*200$)

Նշենք, որ կախուած տեսատիպի հնարաւորութիւններից գրաֆիկական ռեժիմների քանակը կարող է հասնել 10ի:

m-ը ոչ պարտադիր պարամետր է, որը ընդունում է $m=0$ կամ $m \neq 0$ արժեք:

$m=0$ դէպքում միանում է գունաւոր պատկերման ռեժիմը,

$m \neq 0$ դէպքում անշատում է գունաւոր պատկերման ռեժիմը:

Եթէ m պարամետրը տրուած չէ, ապա համարում է $m=0$:

COLOR հրահանգը

Այս հրահանգի օգնութեամբ կարելի է փոփոխել կառուցուող պատկերների կամ արտածուող տուեալների գոյները (color - գոյն):

COLOR C1, C2, C3

C1, C2, C3 պարամետրերը կարող են ընդունել 0-ից մինչեւ 15 արժեքներ (իւրաքանչիւր գոյնին համապատասխանում է որոշակի կոդ):

C1-ը - պատկերման գոյնի կոդն է,

C2-ը - հիմնագոյնի (ֆոնի) կոդը,

C3-ը՝ էկրանի եղրաչերտի գոյնի կոդը:

Օրինակ՝

10 SCREEN 1

20 SCREEN 7, 1, 3

CLS հրահանգը

Այս հրահանգը գործում է բոլոր ռեժիմներում եւ թոյլ է տալիս մաքրել էկրանը (clear screen - մաքրել էկրանը):

Օրինակ՝

30 CLS

PSET եւ PRESET հրահանգները

Այս հրահանգների միջոցով կարելի է էկրանի կոմեյտիոն կամ ներկել որեւէ գոյնով (point reset - վերականգնել կէտը, point set - տեղադրի կամ):

PSET (X, Y), C

PRESET (X, Y), C

որտեղ (X, Y)-ը կէտի կոորդինատներն են, C-ն՝ բնարուած գոյնի կոդը:

Եթէ PSET հրահանգում բացակայում է C պարամետրը, ապ (X, Y) կոորդինատներով կէտը ներկւում է ընդունուած պատկերման գոյնով, իսկ թէ C պարամետրը չի նշուած PRESET հրահանգում՝ կէտը ներկւում է հիմնագոյնի համար էկրանի վրայ: Դա է միակ տարրերութիւնը PSET և PRESET էլ հրահանգների միջեւ:

Օրինակ՝

10 SCREEN 1

20 FOR X=50 TO 150

30 PSET (X, 50)

40 PSET (50, X)

50 PSET (X, X)

60 NEXT X

70 FOR X=150 TO 50 STEP-1

80 PRESET (X, 50)

90 PRESET (50, X)

100 NEXT X

Այս ծրագրի առաջին ցիկլը (20-60 տողեր) նախատեսուած է Ալ առուած գծելու համար, որոնք (50, 50) կէտը միացնում են համապատասխան գոյնը (150, 150), (50, 150) եւ (150, 150) եւ կէտերի հետ: Այնուհետեւ Յ գծերից երկուսը (հարիզոնականը եւ ուղղաձիգը) ջնջում են PRESET հրահանգի միջոցով (10-10 ասղեր):

LINE հրահանգը

Այս հրահանգն օգտագործում է գրաֆիկական ռեժիմում հատուածներ եւ ուղղանկիւններ պատկերելու համար:

LINE (X1, Y1) - (X2, Y2), C

Հրահանգի կատարումից յետոյ էկրանի վրայ գծում է հատուած (ՈՒԱ - պիծ), որը միացնում է (X1, Y1) եւ (X2, Y2) կոորդինատներ ունեցող կէտերը:

Ը պարամետրի միջոցով տրուում է պատկերի գոյնը իմէ (ամենամորը բաց առածքը, ապա ընդունում է COLOR հրահանգով սրացուող պատկերման գոյնը:

Օրինակ՝

10 SCREEN 1

20 LINE (50, 50) - (150, 50), 2

30 LINE (50, 50) - (50, 150), 2

40 LINE (50, 50) - (150, 150), 0

50 LINE (50, 50) - (50, 150), 0

Այս ծրագիրը համարժէք է նախորդ պարագրաֆում բերուած ծրագրին: Նշենք, որ 0-ն հիմնագոյնի կողն է, այդ պատճառով էլ 50 եւ 60րդ տողերում զրուած LINE հրահանդներով նշվուած են մինչ այդ գծուած հորիզոնական եւ ուղղաձիգ հատուածները:

LINE (X1, Y1) - (X2, Y2), C, B

Հրահանդ հնարաւորութիւն է տալիս գծել ուղղանկիւն, որի համար (X1, Y1) եւ (X2, Y2) կոորդինատներով կէտերը հանդիպակաց գագաթներն են, իսկ կողմերը զուգահեռ են կոորդինատային առանցքներին:

Օրինակ՝

10 SCREEN 1

20 LINE (0, 0) - (319, 199), 2, B

20րդ տողում զրուած հրահանդի միջոցով գծուած է էկրանի

LINE (X1, Y1) - (X2, Y2), C, BF

Հրահանդը թույլ է տալիս պատկերել ներկուած ուղղանկիւն:

Օրինակ՝

10 SCREEN 1

20 LINE (10, 20) - STEP (100, 50), 1, BF

30 LINE (10, 110) - STEP (100, 50), 2, BF

Առաջարկուած ենք ինքնուրոյն վերլուծել այս ծրագիրը:

CIRCLE հրահանգը

Հրահանդը նախատեսուած է գրաֆիկական ռեժիմում շրջանագիծ, էլիպս կամ աղեղ գծելու համար (circle - շրջան): Հրահանդի ընդհանուր տեսքը հետեւելուն է.

CIRCLE (X, Y), R, C, I., B, K

Որտեղ (X, Y)-ը շրջանագծի կենտրոնի կոորդինատներն են, R-ը՝ շառավիղը (էլիպսի հորիզոնական կիսառանցքը), C-ն՝ գովնոր, L-ը եւ B-ն՝ աղեղի ծայրակէտերին համապատասխան շառավիղ վեկտորների կազմած անկիւնները X-առանցքի դրական ուղղութեան հետ, K-ն՝ էլիպսի ուղղաձիգ և հորիզոնական տրամադերի երկարութիւնների յարաբերութիւնը:

Պարամետրերից իւրաքանչյուրը կարող է լինել թիւ, թուային փոփոխական կամ թուային արտայայտութիւն: C, L, B, K պարամետրերից ցանկացածը կարելի է բաց թողնել: Ընդ որում պէտք է նկատի ունենալ, որ եթէ բաց թողնուող պարամետրերին յաջորդող պարամետրերից զոնի մէկը զրուելու է, ապա պէտք է պահպանի համապատասխան քանակի ստորակէտները: L-ը եւ B-ն կարող են փոփոխուել (-2π, 2π) միջակայքում:

Պարամետրերի բացակայութեան դէպքում ընդունուած է:

L=0,

B=2π,

K=1,

Օրինակ՝

10 SCREEN 1

20 CIRCLE (50, 60), 40

30 CIRCLE (150, 60), 40, , 0, 3. 14

40 CIRCLE (250, 60), , 3. 14, 6. 28

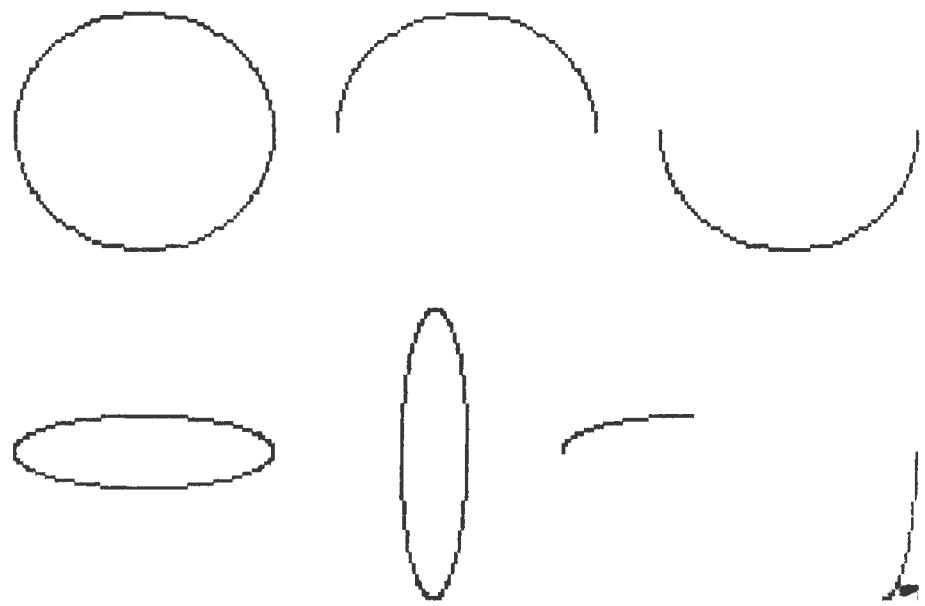
50 CIRCLE (50, 150), 40, , , 1/4

60 CIRCLE (140, 150), 40, , , 4

70 CIRCLE (220, 150), 40, , 1. 57, 3. 14, 1/4

80 CIRCLE (280, 150), 40, 4. 71, 6. 28, 4

Այս ծրագրի կատարումից յետոյ էկրանին կը ստանանք հետեւեալ պատկերը (նկ. 107):



Նկ. 107

PAINT հրահանգը

Այս հրահանգը նախատեսուած է գրաֆիկական ռեժիմում տրուած կէտի շրջակայքը ներկելու համար: Հրահանգի ընդհանուր տեսքն է՝

PAINT (X, Y), C1, C2

(X, Y)-ը այն կէտի կոորդինատներն են, որտեղից սկսում է ներկումը,

C1-ը ներկման գոյնի կողն է,

C2-ը եղրագծի գոյնի կողն է:

Որպէս եղրագիծ կարող է հանդէս գալ C2 կող ունեցող գոյնի կամայական փակ կոր: PAINT հրահանգի կատարման ժամանակ ներկում է սահմանափակուած էլլուանի մասը: Նշենք, որ առաջինը ներկում է (X, Y) կոորդինատներով կէտը, այնուհետեւ "ներկը տարածում է" բոլոր ուղղութիւններով մինչեւ եղրագիծը:

Եթէ C2 պարամետրը բաց է թողնուած, ապա որպէս եղրագիծ է ընդունում C1 կող ունեցող գոյնի փակ կորը:

Օրինակներ՝

10 SCREEN 1

20 CIRCLE (150, 100), 50, 2

30 PAINT (150, 100), 2

Ներկում է շրջանագծով եղրափակուած տարածութիւնը, քանի որ (150, 100) կոորդինատներով կէտը գտնուում է շրջանի ներսում:

10 SCREEN 1

20 CIRCLE (150, 100), 50, 2

30 PAINT (210, 120), 2

Ներկում են շրջանագծից դուրս գտնուող բոլոր կէտերը, որովհետեւ (210, 120) կոորդինատներով կէտը գտնուում է շրջանագծից դուրս:

DRAW հրահանգը

Այս հրահանգի միջոցով (draw - նկարել) կարելի է կառուցել տարրեր պատկերներ, օգտագործելով գրաֆիկական արտածման յատուկ լեզուն, որի հրամանների յաջորդականութիւնը գրում է DRAW ծառայողական բառից յետոյ:

DRAW նշանների տող

Նշանների տողն իրենից ներկայացնում է տեքստային արտայայտութիւն, կազմուած գրաֆիկական արտածման հրամաններից: Այդ հրամանները կարելի է բաժանել երկու խմբի՝ շարժման եւ զեկավարման:

Շարժման հրամանները թոյլ են տալիս գծել տրուած երկարութիւն ունեցող հատուածներ որոշակի ուղղութեամբ: Իւրաքանչիւր հրամանի կատարումից յետոյ յիշում է վերջին արտածուած կէտը, որը եւ ընդունուում է որպէս յաջորդ հրամանով հատուածի սկզբնակէտ:

Հստ ուղղութիւնների օգտագործում հն հասկեալ յարժման բամանները:

Un - շարժում դէպի վերեւ

Dn - շարժում դէպի ներքեւ

Ln - շարժում դէպի ձախ

Rn - շարժում դէպի աջ

En - շարժում անկիւնագծով վերիւ և աջ

Fn - շարժում անկիւնագծով ներքիւ և աջ

Gn - շարժում անկիւնագծով ներքիւ և ձախ

Hn - շարժում անկիւնագծով վերիւ և ձախ

Mx, y - շարժում դէպի (X, Y) կոորդինատներով կէտը

Այստեղ ո-ը հատուածի երկարութիւնն է արտայայտուած կէտերով: (X, Y) կոորդինատները կարող են մորուել որպէս բացարձակ կոմ յարարելական եթէ Տ-ից առաջ դրում է "+" կամ "-" նշանը, ապա կոորդինատները ամերում են յարարելական: Հակառակ դէպում կոորդինատները համարեամ են բացարձակ:

Օրինակ՝

10 SCREEN 1

20 PSET (0, 0), 2

30 DRAW "M10, 20R30D30L30U30"

(0, 0) կոորդինատներով կէտը (էկրանի վերին մայուսիկի միացում է ուղիղ գծով (10, 20) կոորդինատներով կէտի հետ և զծում է քառուկուսի, որի կողմը հաւասար է 30:

Նշուած բոլոր հրամաններում ո, X, Y պարամետրերը կարող են լինել փոփոխականներ: Այդ դէպում փոփոխականի անուանումից առաջ առ բաժիշտ է զնի "=". Նշանը, իսկ յետոյ՝ ";"

Օրինակ՝

T=15

DRAW "D=T"

Շարժման հրամանների հետ կարող են օգտագործուել Ա և Ա նախածանցները՝

B - շարժուել առանց գնելու

N - շարժման գործողութիւնը կատարելուց յետոյ վերադառնութ սկզբնակէտին:

Օրինակ՝

10 SCREEN 1

20 DRAW "BM50, 60E40F40NM50, 60"

Գնելում է եռանկիւն սկսած (50, 60) կոորդինատներով կէտից:

Դեկավարման հրամանների միջոցով կարելի է պատել պատկերը, այն մեծացնել, փոքրացնել, փոխել գոյնը եւ այլն՝

Cn - սահմանել գոյնը (ո-ը գոյնի կողն է),

An - սահմանել պատման անկիւնը (ո-ը կարող է լինել 0, 1, 2 կամ 3): Այս հրամանի կատարումից յետոյ բոլոր կառուցուղղ գները կը պատառ են Ա. 90° անկիւնով:

Տու - սահմանել մասշտաբը: Մասշտարի գործակիցը հաւասար է $n/4$, որտեղ n -ը կարող է փոփոխութել 1-ից մինչեւ 255: Նարժման հրամաններում նշուած արժեքները բազմապատկւում են այդ գործակցի վրայ և եթէ $n < 4$, ապա արժեքները համապատասխանաբար նուազում են, իսկ եթէ $n > 4$ ՝ աճում:

XY\$; - կատարել $Y\$$ տեքստային փոփոխականով տրուած հրամանները:

Օրինակներ:

10 SCREEN 1 : CLS

20 B\$="R20113F3G3"

30 DRAW "BM120, 100XB\$;"

(120, 100) կոորդինատներով կէտից սկսած զծում է հորիզոնական դիմ (R20), որի աջ ծայրին նկարում է սլաք (կատարում են $B\$$ տեքստային փոփոխականին վերագրած հրամանները):

10 SCREEN 1 : CLS

20 A\$="D10F2R2E2U10D10F2"

30 DRAW "BM30, 50XA\$;"

40 DRAW "BM80, 50S2XA\$;"

50 DRAW "BM130, 50S8XA\$;"

60 DRAW "BM180, 50S16XA\$;"

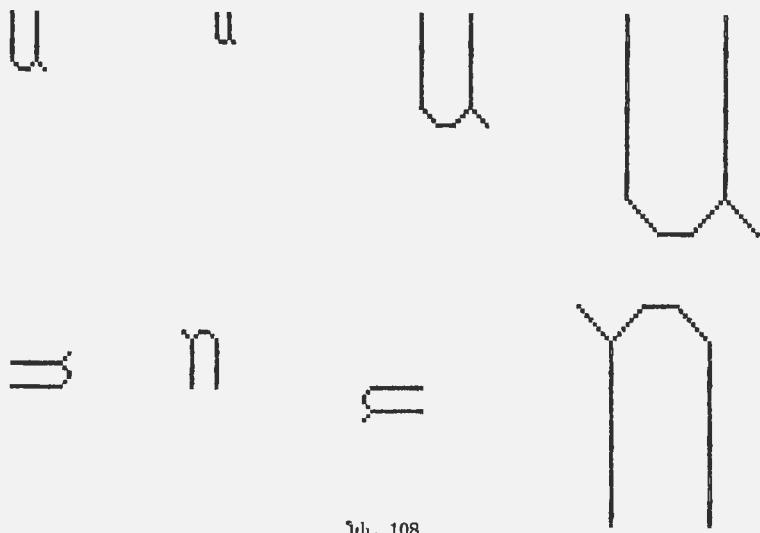
70 DRAW "BM30, 150S4A1XA\$;"

80 DRAW "BM80, 150A2XA\$;"

90 DRAW "BM130, 150A3XA\$;"

100 DRAW "BM200, 180S16A2XA\$;"

Այս ծրագրը պատկերում է էկրանի վրայ "Ա" տառը փոքրացրած, մեծացրած, պատահ 90, 180, 270 (նկ. 108):



Նկ. 108

Հարցեր եւ վարժութիւններ

1. Ինչպէս են տրում կէտի բացարձակ եւ յարաբերական կորորինատները:
2. Կարիլի՞ է օգտագործել COLOR հրահանգը այրենական - բուային ոեժիմում:
3. Կառուցել եռանկիւն, որի զագարներն են (50, 60), (100, 80), (80, 100) կոորդինատներով կէտերը:
4. Կառուցել հաւասարակողմ եռանկիւն եւ նրան ներգծուած շրջանագիծ:
5. Կառուցել շրջանագիծ եւ նրան արտագծուած հառակուսի:
6. Կառուցել ներկուած վեցանկիւն (120, 100), (140, 120), (140, 140), (120, 160), (100, 140), (100, 120) գագարներով:
7. Կառուցել կոորդինատային առանցքները: Ակզրնակէտը տեղադրել էկրանի մէջտեղում:
8. Նկարել "ա", "Բ", "բ", "Գ", "շ", "Ռ", "Ռ" տառերը:

6. Աշխատանք ֆայլերի հետ

Ինֆորմացիոն, գործավարման եւ այլ աւտոմատացուած համակարգերի աշխատանքներին ծանօթանալիս, նախորդ բաժիններում արդէն հանդիպել ենք Փայլի հանկացութեանը:

Փայլն իրենից ներկայացնում է տուեաների համախումբ, որը պահում է արտաքին սարքի վրայ:

Փայլ (file) բառն ինֆորմատիկայի ոլորտ է թափանցել գործավարման ոլորտից, նշանակում է քարտարան՝ համասեռ փաստաթղթերի հաւաքածու: Քարտարանները, ինչպէս զիտենք, օգտագործում են զրադարաններում, հաշուապահութիւններում, կադրերի բաժիններում եւ այլ բնագաւառներում: էջՄ-ի միջոցով նոյնպէս կարելի է ստեղծել քարտարաններ (Փայլեր) եւ այն սարքերի վրայ, որոնք կարող են օգտագործուել որպէս ինֆորմացիայի կրիչներ: Դրանց թուին են պատկանում արտաքին յիշող սարքերը (մագնիսական սկաւառակներ, ժապաւէչներ), տպող սարքը, տեսատիպը եւ այլն:

Փայլերը տարրերելու համար նրանցից իւրաքանչիւրին վերագրում է անուանում: Այդ անուանումը բաղկացած է երկու մասից:

"սարքաւորման անուն: Փայլի անուն"

Առաջին մասը իրենից ներկայացնում է այն սարքի պայմանական անունը, որի վրայ ստեղծում է փայլը (անունը պէտք է աւարտուի ":" նշանով):

Օրինակ՝

KYBD: - ստեղնաշար,

LPT1: - տպող սարք,

SCRN: - տեսատիպ,

A: - տուաքին մագնիսական սկաւառակ,

B: - երկրորդ մագնիսական սկաւառակ:

Ֆայլի անունն իր հերթին կարող է բաղկացած լինել երկու մասից, որոնք իրարից անջատում են ". " նշանով (կէտով):

Օրինակ՝
PROG. BAS

Կէտոյ յետոյ գրւում է մէկ, երկու կամ երեք նշան, որոնք նախատեսուած են ֆայլի տիպը ցոյց տալու համար: Այս հարաւորութիւնը սովորաբար օգտագործւում է ֆայլերի դասակարգման նպատակով: Բերուած օրինակում BAS տիպը օգտագործուած է Բեյսիկ լեզուով կազմուած ծրագիր պարունակող ֆայլի համար: Ֆայլի տիպը կարելի է եւ չնշել:

Բերենք ֆայլերի անունների օրինակներ.

BAR. DAT XYZ CLASS. 1

Ֆայլերի մշակման համար Բեյսիկում նախատեսուած է 4 գործողութիւն՝ ֆայլի բացում,
տուեալների ընթերցում,
տուեալների գրանցում,
ֆայլի փակում:

Ֆայլի բացման հրահանգ

Յանկացած ֆայլի հետ կատարուող աշխատանք սկսում է ֆայլի բացման հրահանգից՝

OPEN "ֆայլի անուանում" FOR
 { INPUT
 OUTPUT
 APPEND } AS # n
 (open - բացել, append - կցել)

Այս հրահանգը կատարում է հետեւեալ գործողութիւնները:

1. Ընտրուած ֆայլին վերագրում է որոշակի համար (n)՝ դրական ամբողջ թիւ, որը յետագյուում օգտագործւում է տուեալների ընթերցման եւ գրանցման հրահանգներում ֆայլի անուանման փոխարէն:

2. Սահմանում է ֆայլի մշակման ռեժիմը՝

FOR INPUT - ֆայլը բացուած է միայն տուեալների ընթերցման համար (նրանք պէտք է ներածուեն էջՄ-ի յիշողութեան մէջ),

FOR OUTPUT - ֆայլը բացուած է միայն տուեալների գրանցման համար (նրանք պէտք է արտածուեն էջՄ-ի յիշողութիւնից),

FOR APPEND - ֆայլը բացուած է լրացուցիչ տուեալներ գրանցելու համար (նրանք գրանցուում են ֆայլի վերջում):

Օրինակներ՝

10 OPEN "A: TEXT. DAT" FOR INPUT AS #3

Այս հրահանգով բացւում է "TEXT. DAT" անունով ֆայլը և սկսածակի վրայ եւ նրան վերագրում է 3 համարը:

20 OPEN "LPT1:" FOR OUTPUT AS #2

Այստեղ չի նշուած ֆայլի անունը, որովհակա սպազ ուարթաւորման վրայ կարող է կազմակերպել միայն մէկ ֆայլ, որը կազմուած է ձրագրի արտածման հրահանգներում նշուած տուեալներից:

Տուեալների ընթերցման եւ գրանցման հրահանգներ

Ֆայլում գրանցուած տուեալների ընթերցման համար (լինթերցում տակելով նկատի ունենք տուեալների ներածումը էջՄ-ի օպերատու յիշողութեան մէջ) օգտագործւում են INPUT # n եւ LINE INPUT # n հրահանգները, որոնց ընդ անուր տիսքն է՝

INPUT # n, փոփոխականների ցուցակ

LINE INPUT # n, փոփոխական

որտեղ n-ը նախօրօք բացուած ֆայլի համարն է:

Այս հրահանգների կատարման կանոնները համապատասխանում են INPUT եւ LINE INPUT հրահանգների կատարման կանոններին, որոնք Պանդաման նկարագրուած են 5. 12 պարագագաֆում: Կարելի է ասել, որ մտցնիսական սկսածակից տուեալների ներածումը համանման է տուեալների ներածմանը առկինաշարից:

Օրինակ՝

10 OPEN "A: XX. YY" FOR INPUT AS #1

20 INPUT # 1, C, W, U, V

Եթէ A սկսածակի վրայ գոյութիւն ունի XX. YY անունով ֆայլ, որտ 20րդ տողում դրուած հրահանգի կատարումից յետոյ այդ ֆայլում գրանցուած 4 իրական թուերի արժէքները կը վերագրուեն համապատասխանաբար C, W, U, V փոփոխականներին:

Տուեալների գրանցման համար օգտագործւում են մեկ քայլ ծանօթ PRINT եւ PRINT USING հրահանգները՝

PRINT # n, տարրերի ցուցակ

PRINT # n USING նմուշատիպ; տարրերի ցուցակ

Թիշեցնենք, որ եթէ տարրերն իրարից բաժանուում են ստորակէտով կամ կէտսորակէտով, ապա համապատասխան արտածուող արժէքների միջին գրուում են բացակներ (բայց ոչ ստորակէտեր): Այս հանգամանքը պէտք է հաշուի առնեկ յանուագէս տեքստային տուեալների գրանցման եւ ընթերցման ժամանակը (Բանն այն է, որ մէկ PRINT հրահանգով որեւէ ֆայլի մէջ գրանցուած երկու տեքստային փոփոխականները ընթերցման ժամանակ գիտարկուում են որպէս մէկ փոփոխական (մէկ տող):

Օրինակ՝

10 OPEN "A: RR. TXT" FOR OUTPUT AS #2

20 X\$= "PROGRAM"

30 Y\$= "BASIC"

40 Y\$=PRINT # 2, X\$, Y\$,

Այս հրահանգների կատարումից յիսոյ ֆայլի մէջ կը գրանցուի

PROGRAM BASIC

առողք: Ընթերցման գործողութեան կատարման ժամանակ այդ բոլոր նշանները կը դիտարկուին որպէս մէկ տող՝

10 OPEN "A: RR. TXT" FOR INPUT AS #2

20 INPUT #2, X\$

20րդ տողում զրուած հրահանգի կատարումից յիսոյ

X\$="PROGRAM BASIC"

Փոխենք 40րդ տողի հրահանգը՝

40 PRINT #2, X\$; ","; Y\$

Մենք աւելացրեցինք բաժանիչը (ստորակէտ), որը կը գրանցուի երկու փոփոփ-խանների արժէքների միջև՝

PROGRAM, BASIC

Այս առողք պարունակում է երկու առանձին տեքստային մեծութիւն, որոնք

20 INPUT #2, X\$, Y\$

Հրահանգի կատարման ժամանակ կը վերագրուին X\$ եւ Y\$ տեքստային փոփոխաններին:

Նշենք, որ թուային առանձիների դրանցման դէպքում լրացուցիչ բաժանիչներ կարելի է չօգտագործել, որովհետեւ բացակները թուային տուեալների ներածման ժամանակ դիտարկում են որպէս բաժանիչներ:

Ֆայլի փակման հրահանգ

CLOSE #n
(close - փակել)

Նշենք, որ մէկ հրահանգով կարելի է փակել մի քանի ֆայլ՝

CLOSE #n₁, #n₂, #n₃,

Եթէ CLOSE հրահանգը տրուած է առանց որեւէ ֆայլի համարի, ապա փակւում են բոլոր բացուած ֆայլերը:

Հարցեր եւ վարժութիւններ

1. Ինչպէ՞ս է կազմւում ֆայլի անուանումը:

2. Ո՞րն է ֆայլի բացման հրահանգը:

3. Գտէք սխալը հետեւեալ ծրագրում:

10 OPEN "B: PR. RP" FOR INPUT AS #1

20 PRINT #1, 1, 2, 3, 4

30 CLOSE #1

40 END

4. Գրանցէք "NAME. FAM" ֆայլում ճեր անունը, ազգանունը, ծննդեան թիւը եւ հասցեն: Կազմէք ծրագիր, որը լեբեցում է այդ տուեալները մագմիսական սկաւառակից եւ արտածում տեսատիպի էկրանի վրայ:

5. "R.R. PP" ֆայլում գրանցուած է 100 հատ իրական թիւ: Գտնել այդ բուերի գումարը եւ արտադրեալը:

Տպագրութիւնը աւարտեցաւ 1991ի Սեպտեմբերին
Ս. Ղազարի Միհրանի Տպարանին մէջ, Վենետիկ

Finito di stampare nel mese di settembre 1991
presso la Tipo-Litografia Armena di S. Lazzaro, Venezia