

Стоян Айков

**ВЪВЕДЕНИЕ
В ПРОГРАМИРАНЕТО
НА ПЕРСОНАЛНИТЕ
МИКРОКОМПЮТРИ
ПРАВЕЦ-82**

**ВИСШ ИКОНОМИЧЕСКИ ИНСТИТУТ "КАРЛ МАРКС" — СОФИЯ
ОТДЕЛЕНИЕ ЗА ПОВИШАВАНЕ КВАЛИФИКАЦИЯТА НА КАДРИТЕ**

проф. д.ик.н. СТОЯН АНГЕЛОВ АЙКОВ

**ВЪВЕДЕНИЕ
В ПРОГРАМИРАНЕТО
НА ПЕРСОНАЛНИТЕ
МИКРОКОМПЮТРИ
ПРАВЕЦ-82**

ПЪРВА ЧАСТ

СОФИЯ, 1984

В записките се разглеждат основните въпроси на технико-експлоатационните възможности и програмирането на персоналните микрокомпютри П Р А В Е Ц - 82. Разгледани са основните оператори за съставяне на програми при използване на програмния език Б Е Й С И К. Записките са предназначени за всички специализанти от О П К К при ВИИ "Карл Маркс".

ВЪВЕДЕНИЕ В ПЕРСОНАЛНИТЕ МИКРОКОМПЮТРИ

През последните години тенденцията при производство на ЕИТ е преминаването към микрокомпютеризация. Като резултат от тази тенденция в края на 70-те години започва производството и използването на т.нар. персонални микрокомпютри. Основното им предназначение е решаването на научноизследователски, научно-технически и икономически задачи. Внедряването им дава възможност да се автоматизират процесите по обработка на икономическа и научно-техническа информация.

В нашата страна са произведени и се прилагат в практиката персоналните микрокомпютри ИМКО-2 и ПРАВЕЦ-82. Освен тях в практиката намират приложение и такива персонални компютри като ЕИПЪЛ, ЛВМ-РС, ЛВМ-ХТ, ІЕ-800 и др.

В конфигурацията на персоналния компютър се включват: клавиатура /кирилица и латиница/, дисплей /черно-бял или цветен/ с различни формати в зависимост от големината на символите /24 реда по 80 знака в ред, 24 реда по 40 знака и др./, запаметяващи устройства на гъвкави магнитни дискове с капацитет от 250-500 Кбайта /при една или две повърхности/, фиксиран магнитен диск, тип УИНЧЕСТЪР с капацитет 10 до 20 Мбайта, печатащи устройства /широк печат - 80 до 180 знака в ред/, работещи със скорост от 50-100 знака в секунда, централен процесор, оперативна памет с капацитет до 640 Кбайта, помощна памет с капацитет 40 Кбайта, комуникационен адаптер за предаване на информацията по канали за връзка.

Все по-широко в конфигурацията на персоналните компютри

намират приложение запаметяващите устройства, тип УИНЧЕСТЪР. За тях е характерно следното: дисковете по правило не могат да се сменят, тъй като заедно с главите за запис и четене са поставени в херметично затворен корпус и образуват т.нар. информационен модул. Магнитните глави са с олекотена конструкция Ø, лежат върху повърхността на неподвижните дискове и "плават" на твърде малко разстояние над тях при завъртането им. Въздухът в информационните модули непрекъснато се филтрира с цел елиминиране попадането на частици прах върху повърхността на дисковете. Тези дискове имат диаметър 180 mm. Капацитетът на един "Мини Уинчестър" диск е в границите от 2-190 Мбайта, но преобладават дисковете с капацитет от 6 до 20 Мбайта.

В по-голямата част от персоналните компютри се използват такива езици за програмиране като БЕЙСИК, ПАСКАЛ, АСЕМБЛЕР, КОБОЛ, ФОРТРАН и др.

Като недостатък на приложението на персоналните компютри може да се посочи необходимостта управленският работник в зависимост от програмното осигуряване на компютъра да може да програмира работата му при знаене на един от посочените по-горе алгоритмични езици. Това обстоятелство затруднява приложението на тези машини. С цел да се елиминира този недостатък през последните години започна да се провежда усилена научноизследователска работа по усъвършенствуване диалога между управленския работник и машината. В това отношение вече има и постигнати успехи. Например в сравнение с прилаганите персонални компютри, усъвършенствуваните осигуряват опростена работа на оператора с него.

Използването на тези персонални компютри дава възможност управленският работник без предварителна подготовка в областта на програмирането само за един ден да го усвои и използва в своята

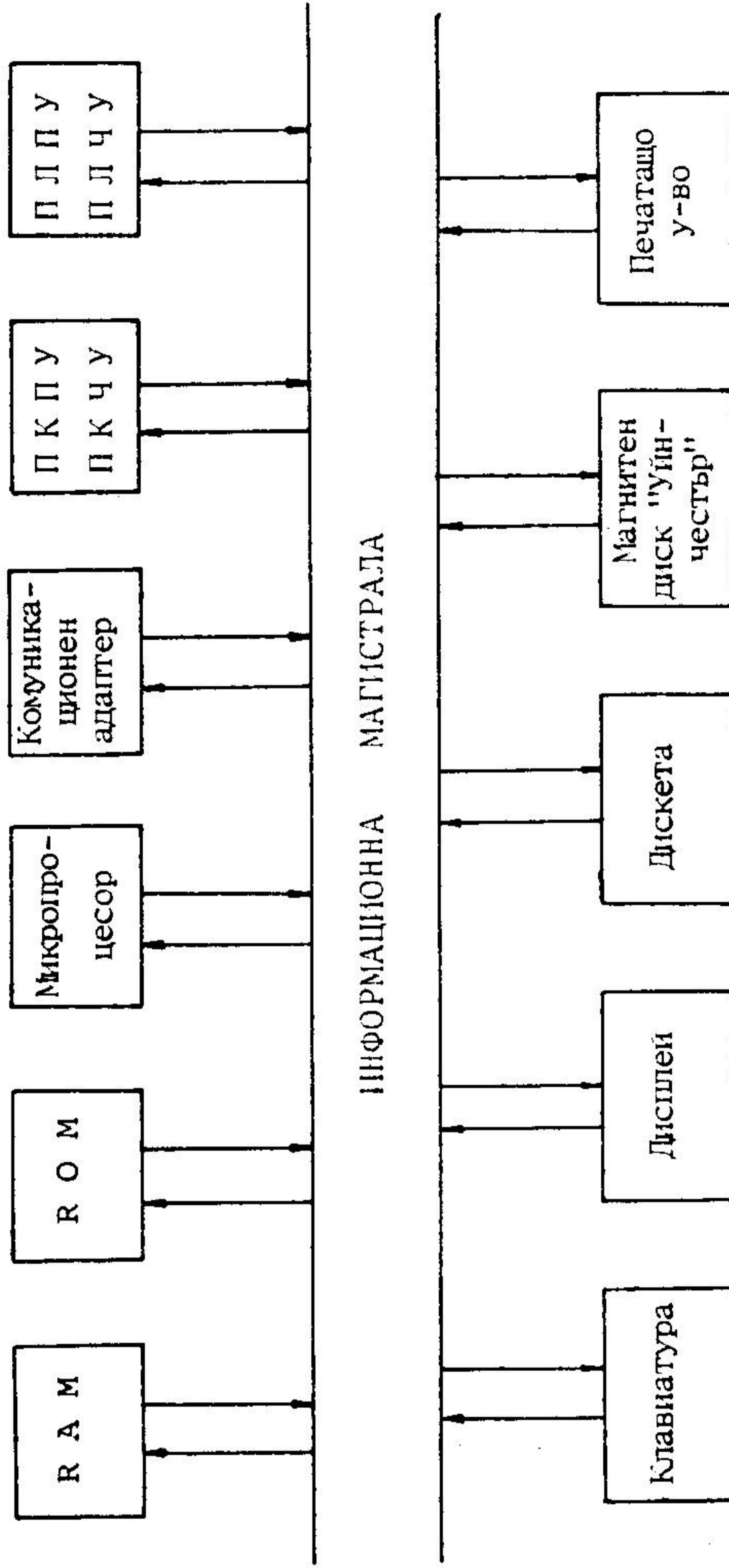
практика, докато за другите персонални компютри е необходимо да се извършва подготовка в срок от един-два месеца.

На схемата от фиг. 1 е показана принципната блокова схема на съвременен микрокомпютър. Основните устройства, изграждащи микрокомпютъра, са: микропроцесор, постоянно запаметяващо устройство /ПЗУ - ROM, PROM и R PROM - за съхранение на програми; памет с произведен достъп /RAM/ за съхраняване на данни /тази памет се нарича още оперативна/; клавиатура /цифрова и текстова/ за въвеждане на информацията; дисплей /за осъществяване на диалога човек-компютър/; гъвкави магнитни дискове /флопи дискове/; минимагнитни дискове; минимагнитна лента; касета; печатащо устройство; перфолетни или перфокартни четящи или перфориращи устройства и устройство за свързване на микрокомпютъра с голяма ЕИМ.

Постоянната памет ROM служи за записване на програмата за действие на микропроцесора, определящи функционирането на микропроцесорното устройство. Записът на програмата в ROM се осъществява още при производството на самите схеми и впоследствие не може да бъде променена, т.е. тя е енергонезависима /не се влияе от изключване на захранването/.

В програмируемата постоянна памет PROM информацията се записва след изработката на схемите по електрически път. Внасянето на информация в схемите PROM води до трайни изменения в тях и в следствие съдържанието им не може да бъде променено/ те са енергонезависими/.

Репрограмируемата постоянна памет R PROM позволява многократно записване и изтриване на информацията по електрически път. Тази памет се използва при реализирането на единични микропроцесорни устройства или при производството на малки серии.



Фіг. 1

Паметта с произволен достъп RAM се нарича оперативна памет или памет с възможност за четене и запис. При нея записът на командите или данните /думите/ се извършва по време на действието на микропроцесорното устройство. Тази памет е енергозависима /т.е. изключването на захранването води до изтриването на записаната в нея информация/.

Информационната магистрала осигурява връзката и обменът на информация между всички възли на микропроцесорната /в случая микрокомпютърната/ конфигурация.

През 1981-82 год. Института по техническа кибернетика и роботика при БАН разработи български персонален микрокомпютър ИМКО-1 и ИМКО-2 /индивидуален микрокомпютър/. По-късно основното му производство започна в Приборостроителния завод в гр. Плевен, като същите са съвместими с ИМКО-2 и носят названието ПЛЕВЕН-82.

При работа на персоналния микрокомпютър /ПК/ Плевен-82 както много подобни ПК в света се използва програмният език БЕЙСИК. Този език е разработен през 1965 г. от сътрудници на Дортмундския колеж /САЩ/ по поръчка на фирмата "Дженерал Електрик". Наименованието на езика БЕЙСИК произлиза от първите букви на английските думи: Beginner's, All - purpose, Symbolic, Instruction Code /многоцелеви език от символически команди за начинаещи/. Този език е бил замислен като прост и удобен за ползвателите го, служещ за решаване на задачи с малък обем входна информация в диалогов режим. Понастоящем БЕЙСИК е един от широко разпространените езици при обработка на информацията в пакетен режим и режим на времеделене. Освен в ПК езикът БЕЙСИК намира приложение в програмното осигуряване и на много мини ЕИМ /СМ-3, СМ-4 и др./ и големи ЕИМ.

Популярността на този език се състои в простотата при него-

вата реализация и поради голямата му близост с алгоритмичния език ФОРТРАН.

Персоналният микрокомпютър ПРАВЕЦ-82 обработва информацията на основата на разширен БЕЙСИК, което осигурява допълнителни възможности на машината. Въвеждането на програмата /команди и оператори/ и входната информация първоначално се въвежда чрез клавиатурата. По-нататък информацията и програмите могат да се съхраняват върху флопидискове и касети.

2. ТЕХНИКОЕКСПЛОАТАЦИОННИ ВЪЗМОЖНОСТИ НА ПМК ПРАВЕЦ-82.

Универсалният ПМК ПРАВЕЦ-82 е предназначен за индивидуално използване от различни по квалификация и професия потребители.

Основната конфигурация на ПМК се състои от: изчислителен блок, клавиатура, видеомонитор и битов касетофон. В разширената му конфигурация освен посочените устройства се включват още: минифлопидисково устройство, печатащо устройство, цветен монитор, ръчки за игри, платки за увеличаване паметта, плотери и др.

Изчислителният блок представлява една голяма платка, състояща се от около 90 различни интегрални схеми. Сърцето на ПМК представлява една по-голяма схема, наречена микропроцесор. С помощта на нея се извършват основните аритметични и логически операции и координиране работата на останалите интегрални схеми. Използуваният микропроцесор в Правец-82 /6502/ може да адресира 65536 осембитови клетки и да изпълнява 56 инструкции.

Микрокомпютърът има както почти всички ЕИМ вътрешна /оперативна/ памет и външна памет.

Оперативната памет е изградена на основата на голям брой интегрални схеми, съдържащи повече от 700 000 транзистора. Запаметяването от транзисторите на информация се извършва, като отделните букви и цифри се кодират в двоична система /наличието на напрежение "1", а липсата на напрежение "0"/. Например при натискане на клавиша "А" напрежение се явява в първия и в предпоследния проводник, който свързва клавиатурата с изчислителния блок / $A=01000001$ /, за клавиша "Б" - напрежение се появява в първия, във втория и в седмия проводник / $B=01000011$ / и т.н.

Както се вижда изразяването на буквите и цифрите се извършва с двоични цифри /битове/.

Всички транзистори в ПМК са разпределени в групи, наречени клетки на паметта. От своя страна всяка клетка от паметта съдържа 8 самостоятелни участъка, като всеки участък може да съдържа 0 или 1. Например ако в дадена клетка трябва да се запише буквата "Б", напрежение ще има само в първи, втори и седми участък /фиг. 2/. По този начин във всяка клетка се нанасят всички букви и цифри.

Необходимо е да се посочи, че ако дадена клетка съдържа само един участък /т.е. 1 бит/ възможните състояния са $2^1 = 2$; когато клетката съдържа два участъка /т.е. два бита/, възможните състояния са $2^2 = 4$; когато клетката съдържа три участъка възможните състояния са $2^3 = 8$; при четири участъка те са $2^4 = 16$; при пет участъка те са $2^5 = 32$ и т.н.

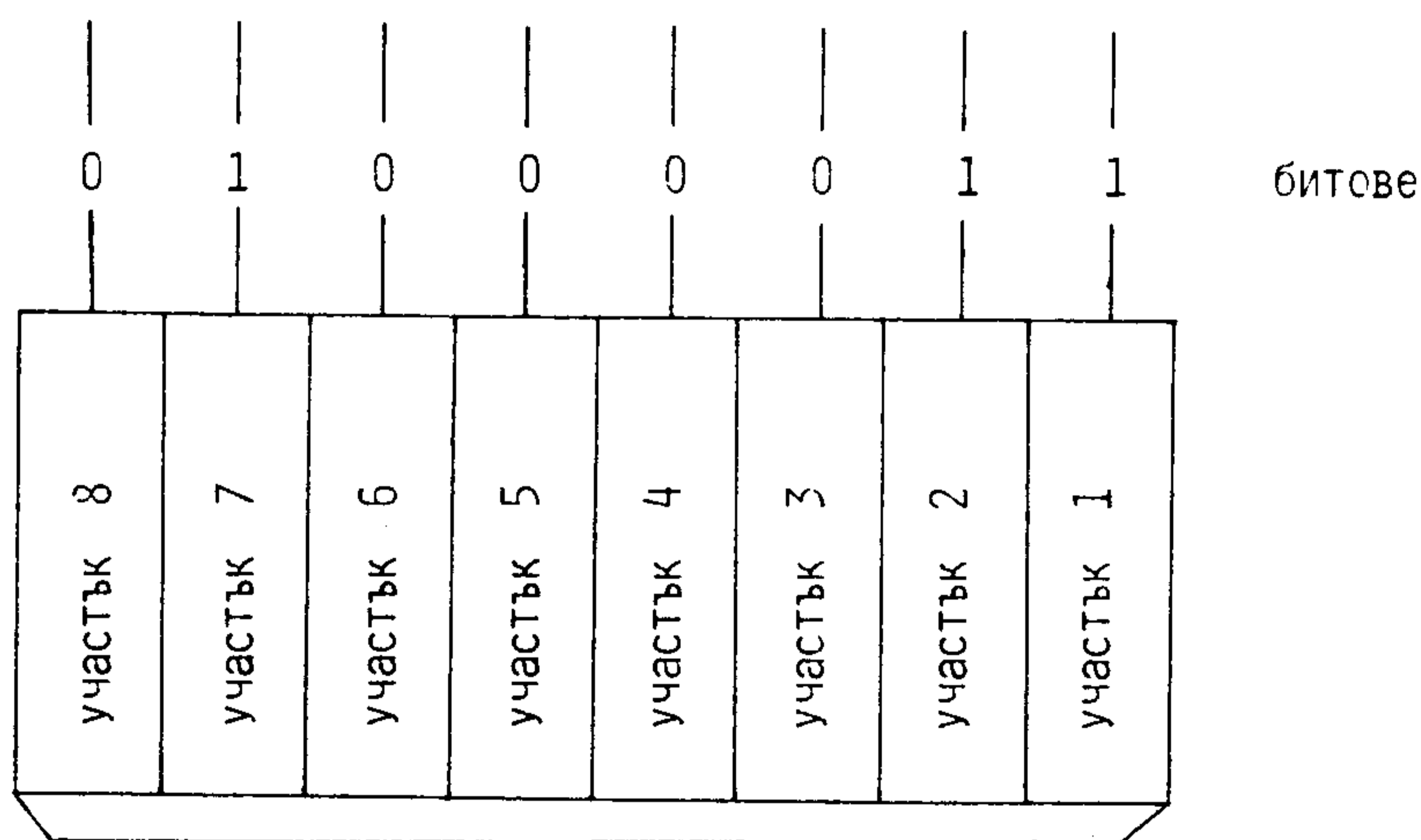
При ПМК ПРАВЕЦ-82 клетките на паметта са изградени от 8 участъка /т.е. 8 бита/ и възможните състояния са $2^8 = 256$.

В практиката при използването на ПМК често се използва единицата за информация-байт /1 байт = 8 бита/, фиг. 2.

Съхраняването на информацията в ПМК става в отделни клетки, поради което байтът е най-малкото количество информация, която може да се адресира. Това означава, че когато чрез клавиатурата се нанесе една буква при нейното съхраняване същата заема памет един байт, т.е. една клетка. Както бе посочено по-горе възможните състояния на клетката са 256 /от 0 до 255/, което означава, че в дадена клетка могат да се адресират до 256 различни символи.

Следващата по-голяма информационна единица за памет е 1 Кбайт /килобайт, Кбайт/, при което 1 Кбайт = 1024 байта.

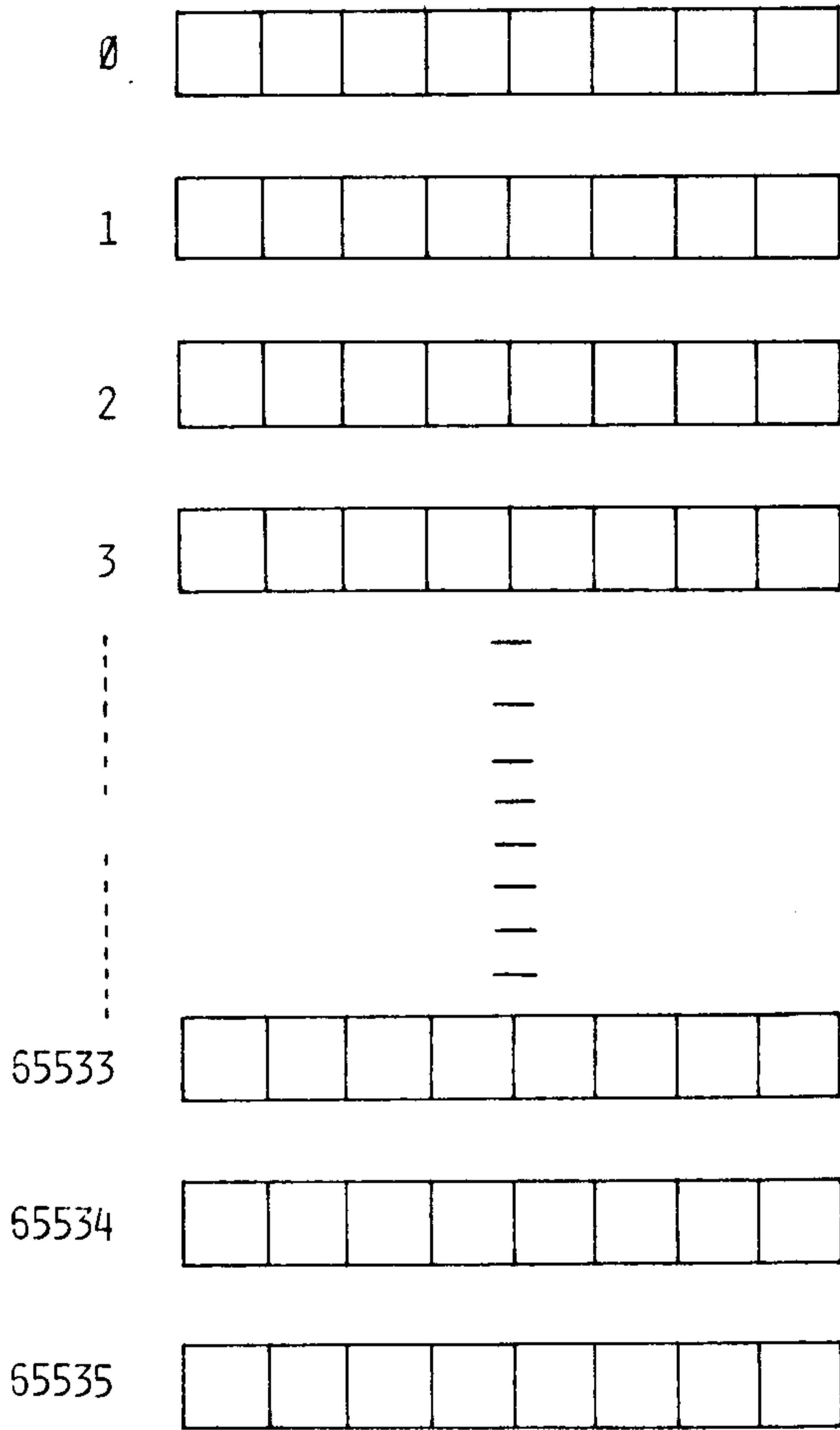
Общата вътрешна памет на ПРАВЕЦ-82 е 65536 клетки /или байта/, което е равно на 64 Кбайта. Всяка клетка в паметта на ПМК има свой номер, наречен адрес. По този начин адресът на първата клетка е 0, а на последната - 65535 /фиг. 3/.



клетка
/1 байт/

Фиг. 2

К Л Е Т К И



Фиг. 3

Вътрешната /оперативната/ памет на ПРАВЕЦ-82 се разделя на три групи:

1. Свободно адресируема памет /RAM/. Тук се включват клетките, в които потребителят може да записва или да чете информацията. Тази част на паметта още се нарича памет с произволен достъп и има капацитет 48 Кбайта.

Ако общата оперативна памет на ПМК може да се оприличи на една книга с 256 страници, като всяка страница съдържа памет по 256 байта, то свободноадресируемата памет обхваща страници от 0 до 191 /фиг. 4/. В действителност първите 8 страници от тази памет се използват от ПМК за други цели, при което БЕЙСИК-програмите започват да се разполагат след адрес на клетка 2048 /8 стр. по 256 байта = 2048 байта/ - свободните клетки са 47 100. Тъй като паметта RAM е енергоуязвима след изключване на електрическия ток записаната от потребителя информация се изтрива.

2. Входно-изходна памет. Тази памет има капацитет 4 Кбайта и заема стр. 192-207 /фиг. 4/. Основно паметта е свързана с работата на клавиатурата, монитора и всички външни запамятаващи устройства /касетофони, минифлопи дискови устройства и др./

3. Постоянна памет - ROM. Този вид памет има капацитет 12 Кбайта и заема стр. 208-255 /фиг. 4/. В тази памет още по време на производството на ПМК са записани програми и данни, необходими за работата на машината. Но за разлика от свободноадресируемата памет /RAM/ тук записите са направени, така че информацията се запазва и при изключване на компютъра. Поради тази причина и тази памет се нарича ROM /Read Only Memory, памет само за четене/. В постоянната памет е записан интерпретаторът на езика БЕЙСИК, управляващата програма, наречена монитор, известен брой подпрограми и др.

№ на страницата	Вид на паметта	
0 1 2 3 4 5 190 191	Р А М 48 К байта	
192 193 206 207	ВХОДНО-ИЗХОДНА памет 4 К байта	64 К байта
208 209 254 255	Р О М 12 К байта	

Фиг. 4

От постоянната памет /12 Кбайта/ за интерпретатора на езика БЕЙСИК се заемат 10 Кбайта и за системния монитор - 2 Кбайта.

Въвежданата информация от клавиатурата, от оперативната памет или от външни запамятаващи устройства се нанася върху видеоекрана /видеомонитор/.

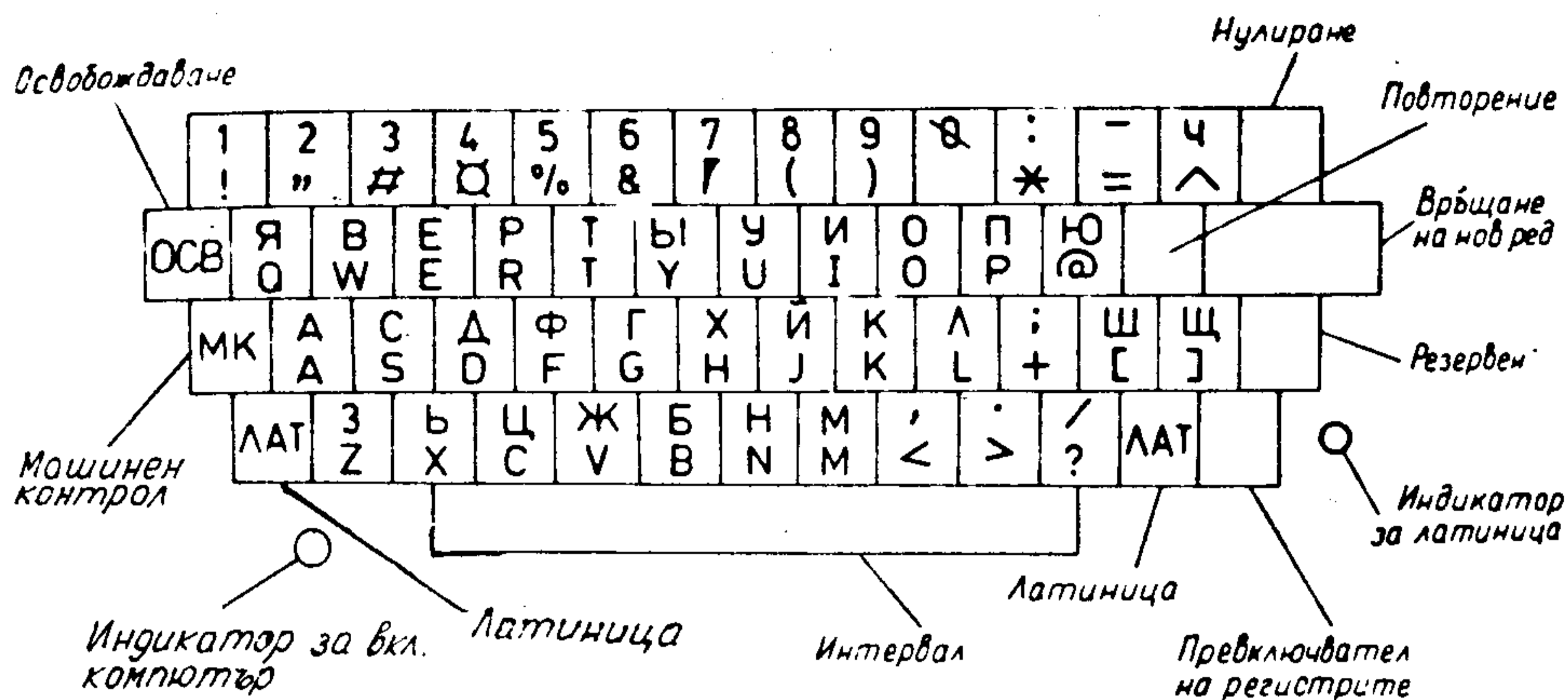
Видеомониторът представлява черно-бял екран /телевизионен приемник "София - 31"/. Изобразяването върху екрана се формира от 280/192 точки с размер 0,7 mm.

Върху екрана може да се нанесат 24 реда текст, като всеки ред съдържа 40 букви т.е. на един екран /един кадър/ максимално се нанасят 960 символа.

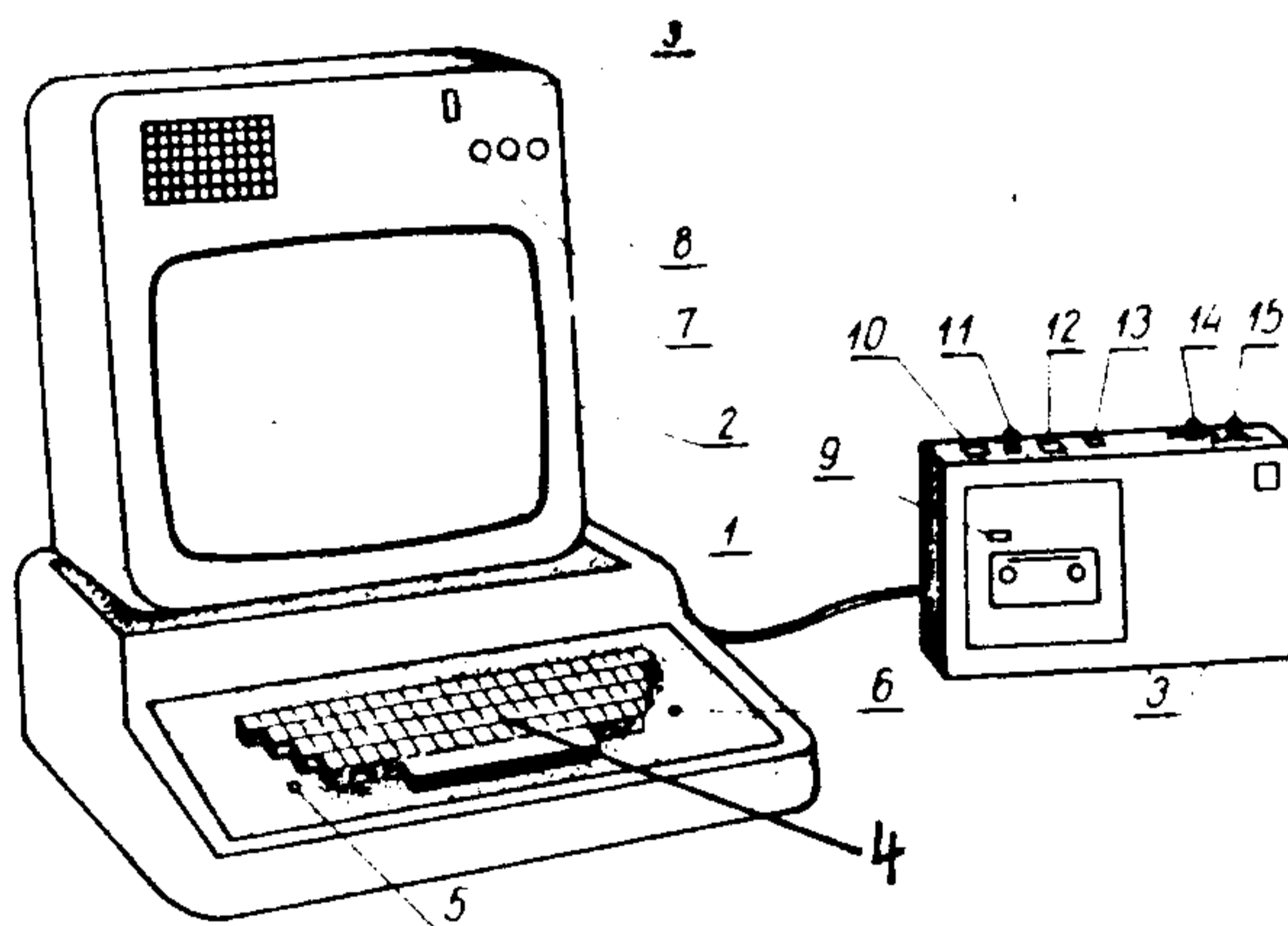
Ако се направи едно сравнение между възможностите на оперативната памет и видеомонитора трябва да се посочи, че в оперативната памет могат да се съхраняват около 50 кадъра /екрана/, което означава 25 машинописни страници.

Основно въвеждането на информацията в ПРАВЕЦ-82 се извършва с помощта на клавиатура. В ПМК е прието стандартното английско разположение на клавишите, при което символите са разпределени по два регистъра: горен - кирилица, препинателни знаци и цифри; долен - латиница и специални знаци. В основно положение при включване на ПМК клавиатурата е в горен регистър /кирилица/. Необходимо е да се има предвид, че в машината се използват само главни букви.

На фиг. 5 е показана клавиатурата на микрокомпютъра. Вижда се, че на всеки клавиш са означени по два символа. Както бе посочено, при включване на ПМК винаги се работи на горен регистър - кирилица. Това означава, че при задействуване /натискане/ на даден клавиш на екрана ще се появи символа означен на горната част на клавиша. Например при натискане на осмия клавиш от най-горния ред върху екрана ще



Фиг. 5



Фиг. 6

се появи цифрата "8", а не лява скоба /(/. Когато трябва да се работи в положение на долен регистър /латиница/ необходимо е да се държи непрекъснато натиснат в долно положение клавишът ЛАТ /клавишът ЛАТ се намира на две места - на четвъртия ред на клавиатурата на първо място и на същия ред на предпоследно място/. В случаите, когато е необходимо по-продължително време да се работи с латиница или използване на символите от долната част на всеки ред, трябва еднократно да се натисне жълтия клавиш /превключвател на регистрите/, намиращ се на последно място в четвърти ред. При натискане на този клавиш, долният регистър се включва за постоянна работа, а вдясно от него светва жълта индикаторна лампа /индикатор за латиница/. Изключването на регистъра за латиница става, като се натисне еднократно повторно жълтият клавиш, при което и контролната жълта лампа се изключва.

Предназначението на отделните символи, означени в клавишите в горен и долен клавиш се следните:

К л а в и ш и в п ъ р в и р е д

Горен регистър. В горния регистър /в горната част на клавишите/ са нанесени цифрите от 1 до 0 /нулата, за да се различава от буквата "0" е пресечена с наклонена черта/. След това са означенията за двоеточие, малко тире /това тире изпълнява функциите на знака "минус"/ и буквата "ч". В най-дясната част на този ред се намира клавишът за НУЛИРАНЕ /RESET, ресет/. При натискането на този клавиш се постига прекъсване изпълнението на програмата, като същата не се изтрива от паметта, а ПМК изчаква следващата команда.

Долен регистър. В първия ред, но при долен регистър върху клавишите са означени следните символи: удивителна, кавички, номер по ред /в действителност този символ прилича на дизел/, Знак за долар

/ като §/, търговско / &, съдружие/, апостроф, лява скоба, дясна скоба, звездичка /знак за умножение /, равно, знак за повдигане на степен /в произвежданите по-рано ПМК този знак, повдигането на степен става с буквата "ч"/.

К л а в и ш и в ъ в в т о р и р е д

Горен регистър. На първо място е клавишът "ОСВ", който се използва за специални цели /за някои от неговите функции ще бъде отделено място по-долу/. По нататък в горния регистър са разположени буквите от кирилицата /Я, В, Е, Р, Т, Ъ, У, Н, О, П и Ю/. Необходимо е да се посочи, че при първите модели ПМК от означението "ы" върху екрана се появява буквата "ъ". След това се намира клавишът ПОВТОРЕНИЕ /РЕПЕАТ, репийт/. Когато този клавиш се държи натиснат продължително време и едновременно с това се натисне клавишът "8", то върху екрана ще започне непрекъснато отпечатване на горната цифра. В най-дясната част на втория ред се намира един от най-често използваните клавиши - ВРЪЩАНЕ НА НОВ РЕД /RETURN, ритърн/. При натискане на този клавиш върху екрана не се явява знак, а нанесената информация от съответния ред постъпва в паметта и маркерът се появява на нов ред.

Долен регистър. В долния ред на долния регистър върху клавишите са означени латински букви. Символът "ø" се нарича "търговско ет".

К л а в и ш и в т р е т и р е д

Горен регистър. Първият клавиш е означен "МК" - машинен контрол и се използва за специални цели при едновременно натискане и на други клавиши /по-долу ще бъдат дадени примери за неговото използване/. В горния регистър на този ред са разположени букви от кирилицата и точка със запетая. Последният клавиш е резервен.

Долен регистър. В положение на долен регистър върху третия ред са разположени латински букви, знака "плюс", лява и дясна квадратна скоба.

К л а в и ш и в ч е т в ъ р т и р е д

Горен регистър. Предназначението на първия и последния клавиши от този ред бяха изяснени по-горе. В горния регистър на четвърти ред са разположени букви от кирилицата, запетая, точка и наклонена черта /наклонената черта се използва в качеството си на знак за деление, а точката за десетична запетая/.

Долен регистър. В долния регистър на четвърти ред са разположени букви от латиницата, математическите знаци: по-малко, по-голямо и знакът въпросителна.

В п е т и я р е д е разположен интервалният клавиш /SPACE, спейс/. С помощта на този клавиш се осигуряват интервали между отделните символи.

Запознаването и практическата работа с ПМК е немислимо без овладяване използването на отделните клавиши от клавиатурата.

В последните модели на ПМК ПРАВЕЦ-82 има някои изменения в клавиатурата. Така например в третия ред на предпоследното място има два символа "Э" и "\", освен това в четвъртия ред върху последните два клавиша са означени лява и дясна стрелка за местене на маркера върху екрана.

За да се започне работа с ПМК ПРАВЕЦ-82 е необходимо същият да бъде включен към електрическата мрежа. Най-напред е необходимо видеомониторът да се включи към изчислителния блок чрез коаксиален кабел. По-нататък изчислителният блок и видеомониторът се включват към мрежата с напрежение 220V с определени кабели. За включване на видеомонитора се натиска бутон 3 на фиг. 6 като се чака 8-10 се-

кунди, след което се натиска превключвателят на изчислителния блок, намиращ се от лявата задна страна. При това включване се чува кратък звук, светва индикаторната лампа от лявата страна на блока /бутон 5, фиг. 6/, а в горната част на екрана се появява надпис ПРАВЕЦ-82 /респ. ИИКО-2/. Едновременно с това в горния ляв ъгъл на екрана се появява квадратна скоба /мигащ правоъгълник, наречен маркер или още се нарича известяващ символ на БЕЙСИК/. Обикновено този маркер /курсор/ показва мястото, където върху екрана ще се появи всяка цифра, буква или знак, въведени от клавиатурата /4, фиг. 6/.

С цел по-добро запознаване с клавиатурата неколнократно нанесете всички символи от клавиатурата /горен и долен регистър/, като при допускане на грешки да не се правят опити за корекции. Практиката ще покаже, че след нанасяне на последния символ в даден ред без натискане на клавиша за обратен ред /RTN/ маркерът автоматично се пренася върху следващия ред на екрана. Ако продължаваме да нанасяме чрез клавиатурата символи, то постепенно информацията от първия /най-горния ред/ се заличава и се освобождава един ред отдолу на екрана. По този начин винаги /непрекъснато/ в 24-тия ред може да се насят символи.

Чрез бутони 7 и 8 /фиг. 6/ се регулира яркостта на изображението на символите върху екрана.

Ако след включване на ПМК с клавиатурата нанесем думите АЗ СЪМ СПЕЦИАЛИЗАНТ В ОПКК, на екрана ще се появи следното:

АЗ СЪМ СПЕЦИАЛИЗАНТ В ОПКК □

При натискане на клавиша за обратно връщане на нов ред /RETURN, накратко RTN/ на екрана ще се появи и следната информация:

]АЗ СЪМ СПЕЦИАЛИЗАНТ В ОПКК

? SYNTAX ERROR

]□

Както се вижда ПМК посочва, че има синтактическа грешка /този отговор е съпроводен с кратък звук/. Очевидно нанесените думи не са разбираеми за микрокомпютъра, поради това че не влизат в състава на неговите команди и оператори.

Необходимо е да се знае, че всички команди и оператори на езика БЕЙСИК се пишат само с латински букви /например HOME, LET, PRINT, RUN и др./ . След нанасянето на дадена команда обезателно трябва да се натисне клавишът RTN, при което се постига изпълнение на командата, а скобата и маркерът се появяват в началото на реда.

Изчистването на екрана на видеомонитора се извършва с командата HOME /хоум, у дома/.

Например, ако върху екрана има нанесена информация:

]ДОБЪР ДЕН

]ЗДРАВЕИТЕ

]56789012

]ABCDEFGHIJ

]

и искаме изчистване на екрана е необходимо да се нанесе командата HOME непосредствено до правоъгълната скоба. С натискане на RTN, екранът се изчиства и маркерът заедно със скобата се появява в горния ляв ъгъл.

Често при работа с микрокомпютъра ще се допускат грешки, които трябва да се коригират.

Тук ще бъде обърнато внимание на корекциите, които могат да се правят при работа на ПРАВЕЦ-82 не в програмен режим.

Например, ако върху екрана е нанесен операторът RON, а

трябва да бъде RUN /т.е. буквата O да се замени с u/. В този случай докато маркерът е на същия ред, корекцията се извършва така: клавиатурата се поставя в положение на горен регистър /кирилица/, с лявата ръка се натиска клавиша МК постоянно, а с дясната ръка чрез еднократно натискане на клавиша X, маркерът се мести вляво докато заеме положение върху буквата O. След това се освобождава клавишът МК и се натиска клавишът за ЛАТ, а с другата ръка се отпечатва буквата u. Ако дадени букви след корекцията трябва да бъдат изтрети, то е необходимо да се натиска клавишът за интервал.

Чрез този клавиш едновременно се заличават символите и маркерът се премества вдясно. В последните модели движението вляво и вдясно на маркера за коригиране се извършва с лявата и с дясната стрелка.

Когато грешката, открита в същия ред се последва от много букви и не е целесъобразно повторното им нанасяне, се използва клавишът у.

П р и м е р:

]АЗ СЪМ ТПЕЦИАЛИЗАНТ В ОПКК □

Както се вижда, допуснатата е грешка в думата "ТПЕЦИАЛИЗАНТ", като вместо С е отпечатана буквата Т. Придвижването на маркера отдясно вляво става чрез постоянно натискане на клавиша МК и едновременно с това натискане на клавиша X, докато маркера заеме място върху буквата Т, която се коригира с буквата С /кирилица/. Маркерът се намира върху буквата П. Необходимо е да бъдат повторени 16 букви. С цел да

се избегне това повторение с лявата ръка се натиска клавишът МК /при кирилица/, а с дясната с еднократни натискания на клавиша у, маркерът се премества надясно, докато се премине и последната буква /К/. По същия начин се постъпва, когато се коригират букви от латиница или цифри.

При работа с ПМК поради неправилно натискане на един или друг клавиш е възможно маркерът да изчезне от екрана. Компютърът в подобни случаи става неуправляем и не възприема подаваната чрез клавиатурата информация. За да се излезе от това положение трябва да се натисне червения клавиш за нулиране /рисет/, при което скобата и маркерът се появяват в горния ляв ъгъл на екрана.

3. ОСНОВИ НА ПРОГРАМИРАНЕТО НА ПМК ПРАВЕЦ-82

Персоналният микрокомпютър ПРАВЕЦ-82 работи при прилагане на три основни режима: директен, програмен и графичен.

При прилагането на директния режим ПМК работи като обикновен електронен калкулатор. Чрез този режим могат да се извършват такива аритметични операции като събиране, изваждане, умножение, деление, степенуване, коренуване, логаритмуване и др. Реализирането на този режим става като се нанесе математическият израз /например 25-14/ и се натиска клавишът RTN. По принцип директния режим е прост и не изисква прилагането на езика БЕЙСИК.

Пълните възможности на ПМК ПРАВЕЦ-82 се проявяват при прилагане на програмния режим. При този режим на работа за решаване на прости или по-сложни задачи е необходимо най-напред да се състави програма и след това се преминава към изпълнение на програмата.

За съставянето на програми при прилагане на програмния режим потребителят трябва да бъде запознат с основите на програмирането с използване на програмния език БЕЙСИК.

Изучаването на който и да е алгоритмичен език винаги се предшества от запознаването с набор от изобразителни средства. Върху символиката на БЕЙСИК не малко влияние е оказала ориентацията на езика за разговор с изчислителната машина.

В състава на езика БЕЙСИК се включват следните символи:

1. 26 главни латински букви - от А до Z;
2. 10 десетични цифри - от 0 до 9;
3. 4 препинателни знака: , /точка/, , /запетайка/, : /двоеточие/, ; /точка със запетайка/;
4. знаци за аритметични операции: + /плюс/, - /минус/, * /умножение/, / /деление/, ^ /степенуване/ и SQR /коренуване/;
5. Операции за отношения:
= /равно/, по-голямо > , по-малко < , > = /по-голямо или равно/, < = /по-малко или равно/ и < > /не е равно/.

3.1. Директен режим на работа

Характерно за работата при директния режим е, че информацията се появява само върху екрана без да постъпва в паметта.

При ПМК точката /. / не е знак за умножение, а десетична точка. Освен това двете точки /:/ не се използват като знак за деление, а имат специално предназначение.

Единственият оператор, който се използва при директния режим на работа с ПМК е операторът PRINT /печати, покажи на екрана/. Прилагането на този режим изисква нанасяне на горния оператор, след

това математическия израз и накрая се натиска клавишът RTN като резултатът се изчислява и се появява на следващия ред в началото.

Събиране. В резултат на посочените действия по-горе действието събиране на целите числа 12 и 26 се извършва по следния начин:

PRINT 12 + 26 - натиска се клавишът RTN

38 - резултат

Събирането на дробните числа 6.5, 14.15 и 9.225 се извършва така:

PRINT 6.5 + 14.15 + 9.225 - натиска се клавишът RTN

29.875 - резултат

След всяко изчисление мигащият маркер се появява на следващия ред, което показва, че ПМК е готов да поема следващите указания.

При разглеждането на отделните примери не се имат предвид предшестващите задачи. Всяка задача започва на изчистен екран. Известно е, че изчистването на екрана става с помощта на командата НОМЕ, която се нанася без номер.

Изваждане. Изваждането на числото 8 от 29 се извършва по следния начин:

PRINT 29 - 8 - натиска се клавиша RTN

21 - резултат

С използването на ПМК ПРАВЕЦ-82 могат да се получават в резултата и отрицателни числа. При необходимост трябва да се използват само малките скоби /на първия ред на клавиатурата/.

Да се извади от числото минус 17 числото минус 6:

PRINT -17 - /-6/ - натиска се клавишът RTN

-11 - резултат

Умножение. Умножението на числата 26 и 23 се извършва така:

PRINT 26 X 23 - натиска се клавишът RTN

598 - резултат

По-нататък при изчисленията е необходимо да се има предвид, че нулата пред десетичната точка може да не се пише - например числото 0,75 може да се напише и така ,75. Освен това с цел да се опростят обясненията след оператора PRINT и математическия израз винаги трябва да се натиска клавишът RTN, за да се извърши изчислението. В следващите примери това трябва да се подразбира без да се пояснява.

При умножение на отрицателни числа трябва да се използват малки скоби.

PRINT 26 X /-23/

-598 - резултат

Разбира се верен резултат ще се получи и ако се нанесе 26 -23, но този начин на изчисление не се препоръчва.

Деление. Да се раздели числото 18 на числото 2:

PRINT 18/2

9 - резултат

Да се раздели числото 19 на 2:

PRINT 19/2

9.5 - резултат

Да се раздели числото 18 на 3 - 3. Ако на екрана се нанесе 18/ (3-3) след натискане на клавиша RTN ще се получи съобщение, че делението се извършва на нула. С цел да не се получи това съобщение правилно е върху екрана да се нанесе следният математически израз:

PRINT 18/3-3

3 - резултат

В този случай най-напред 18 се разделя на 3 и се получава 6, а от 6 се изважда 3.

В същата последователност се извършват изчисленията и при необходимост от събиране. Например:

```
PRINT 18/3 + 3
```

9 - резултат

Както се вижда най-напред се извършва деление 18 на 3 като се получава 6 и към него се прибавя 3.

Степенуване. Когато едно число M трябва да се повдигне на квадрат по-добре е да се умножи на себе си, като на екрана се напише $M \times M$. В този случай квадратът на числото 276 може да се определи по следния начин:

```
PRINT 276 * 276
```

По същия начин може да се постъпи и когато се повдига на трета степен.

```
PRINT 276 * 276 * 276
```

Когато обаче числото M трябва да се повдигне на степен P , то това се постига чрез използване на буквата "C" /тази буква се различава от цифрата четири/. Например, ако числото 17 трябва да се повдигне на степен трета нанесената върху екрана информация ще изглежда така:

```
PRINT 17 C 3
```

Коренуване. При използването на езика БЕЙСИК за намиране на корен втори от M може да се използва означението `SQR /M/`. Например за изчисляване на $\sqrt{3}$ на екрана се написва:

```
PRINT SQR (3)
```


3.2. Програмен режим на работа

3.2.1. Основни въпроси. Оператори за присвояване и печат

Най-общо програмирането представлява съставянето на инструкции за ПМК, указващи как да бъдат решавани задачите. Създаденият набор от инструкции се нарича програма.

Всяка програма може да съдържа от една до няколко хиляди инструкции.

Основните възможности на ПМК ПРАВЕЦ-82 се проявяват при прилагането на програмния режим. Ето защо, за да бъде решена каквато и да е задача /по-проста или по-сложна/, трябва да бъде съставена и съответна програма. Както бе посочено по-горе за програмиране работата на ПМК ПРАВЕЦ-82 се използва езикът БЕЙСИК. В резултат на програмата, ПМК решава задачата и извежда решението на екрана /респ. върху табулограма или върху магнитен носител/.

Характерно за програмния режим е обстоятелството, че нанесената върху екрана програма се съхранява в паметта на ПМК /RAM/. Тази програма се съхранява в оперативната памет до момента на изключване на компютъра от електрозахранването или докато не се напише специална команда. Необходимо е да се отбележи, че съхраняваната в паметта програма може да бъде извикана на екрана и да бъдат коригирани някои нейни елементи.

При създаване на програми се използват типови оператори. Едни от тях започват с думите LET /нека/, а други с думата IF /ако/, PRINT /печат/, GOTO /премини към/ и др.

Най-напред ще се запознаем с оператора LET и номерата на редовете. За извършване събирането на числата 27 и 38 с помощта на

клавиатурата върху екрана може да се напише следната програма:

```
1Ø LET A = 27  
2Ø LET B = 38  
3Ø LET C = A + B  
4Ø PRINT C
```

} програма

5Ø END - край на програмата

RUN - команда за изпълнение на програмата

65 - резултат

]□ - ПМК очаква указания

Операторът LET се нарича оператор за присвояване, тъй като чрез него се присвояват съответстващи значения на различните променливи.

Представената програма се състои от пет реда, при което всеки ред има номер. Например 1Ø, 2Ø, 3Ø, 4Ø, 5Ø и т.н. Разбира се тази последователност не е задължителна, но целесъобразно е между отделните редове номерирането да е през 10. Това създава условия за коригиране или редактиране на програмата. Задължителни изисквания тук при програмирането са: да се използват само цели числа, да няма повторение между номерата и числата, и те да не са по-големи от 63999.

Както се вижда от програмата отделните редове съдържат думи, наречени оператори. Един от тези оператори е и операторът LET. Всеки оператор се пише на латиница.

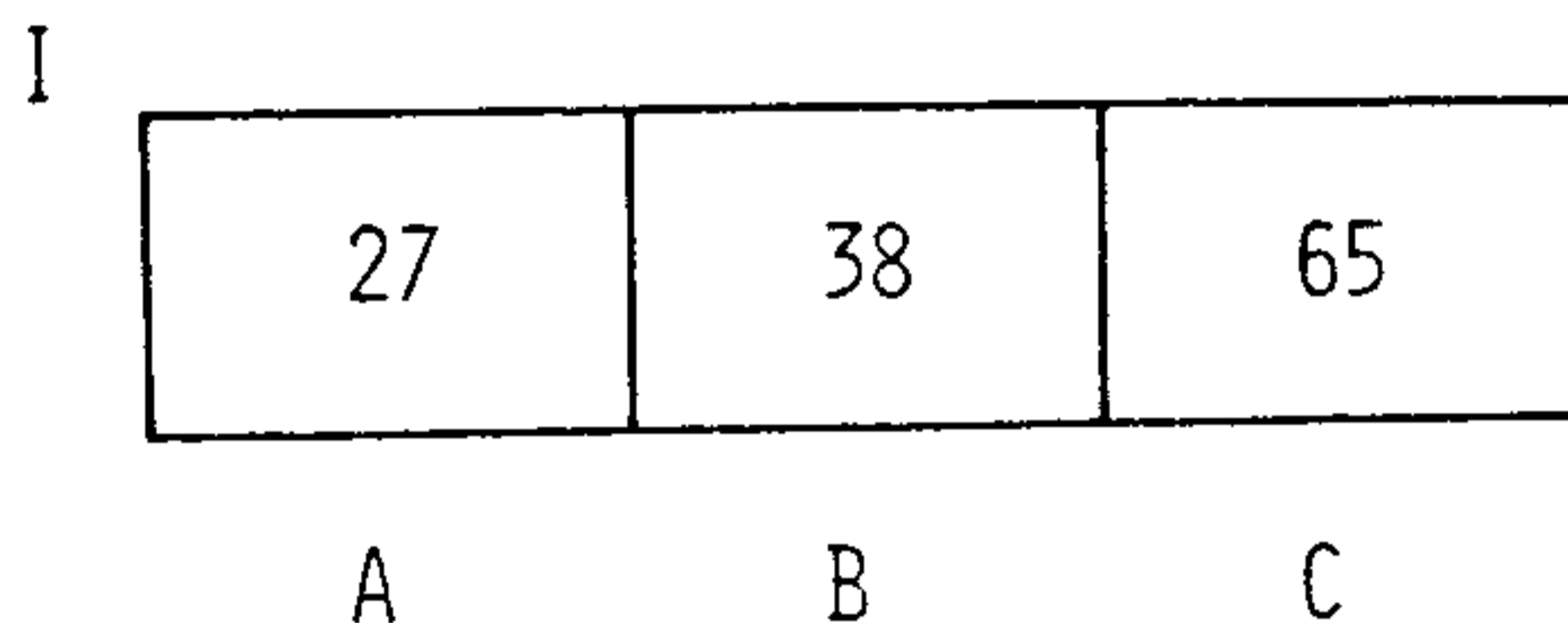
В програмата буквите A, B и C се наричат променливи, числата се наричат константи.

Операторът PRINT /принт, печатí, изобрази на екрана/ за краткост може да бъде заменен със знака "?".

Когато се напише LET A = 3, то се подава команда за ПМК да

избере клетка от паметта /всяка клетка от паметта, която компютърът счита за подходяща/ и записва в нея числото 27 /фиг. 7/. С написването на `LET B = 38` се подава команда в клетка В от паметта да се присвои числото 38, а при `LET C = A + B`, командата към ПК е следната.

1. Съберете съдържанието на клетки А и В.
2. Резултатът от събирането да се запише в клетка С /фиг. 7/.



Фиг. 7

Тъй като командата е `PRINT C`, то към компютъра се дава заповед съдържанието на клетка С да се изобрази на екрана. Ако командата беше `PRINT A, B, C` на екрана ще се отпечата съдържанието на трите клетки - 27, 38, 65 /съответно А, В и С/.

Наличието на оператора `END` /край/ съобщава на машината, че е необходимо след отпечатване на резултата да завърши изпълнението на програмата. Този оператор се записва с последния пореден номер на програмата.

За да започне изпълнението на програмата е необходимо да се напише командата `RUN` /рън, изпълни/. Същата се пише без номер в началото на реда.

До тук обаче, все още програмата не се изпълнява. За целта е необходимо да се натисне клавишът `RTN`.

С помощта на командата `RUN`, ПМК изпълнява следните основни действия:

1. Извършва контрол на верността на програмата /при откриване на грешки същите се отпечатват/;

2. При липса на грешки в програмата отпечатва резултатите. Като правило изпълнението на програмата започва по възходящ ред на номерата на редовете, като се започне от най-малкия /в нашия случай с номер 10/. Накрая на ред 40 компютърът среща оператора `PRINT C` и отпечатва на екрана 65.

По принцип накрая на всеки ред се натиска клавиш `RTN`, като след неговото задействуване информацията от всеки ред се въвежда в паметта на машината.

При необходимост съхранената в паметта програма може да се изведе върху екрана чрез командата `LIST` /лист, списък/. Тази команда се изписва в началото на екрана /задължително преди това трябва да е натиснат клавишът `RTN`/ без номер на началото на реда. Така след написването на командата `LIST` и след натискането на клавиша `RTN` на екрана се появява /се листва/ съхранената в паметта програма. Върху програмата могат да се правят корекции и редактиране на текста.

С написването на `RUN` и натискане на клавиша `RTN` започва изпълнението на програмата. Това може да се повтаря многократно. Ако при първоначалното писане на програмата вместо `PRINT` сме нанесли знака `?`, то при листването в програмата на екрана се изписва операторът `PRINT`.

В компютъра могат да се въведат две и повече програми. За да не се получи смесване на програмите и отпечатване на символи за грешка, необходимо е отделните програми да се записват с различни поредици от номера на началото на реда. Например дадена програма може да се

пише от номер 10 до 500, втора - от 600 до 1200 и т.н. Освен посоченото, с цел да се елиминира допускането на грешки в номерата на редовете, след като дадена програма е написана и изпълнена същата да бъде изтрита от паметта на компютъра. Ето защо преди започване писането на всяка следваща програма трябва чрез оператора NEW /ню, нов, у дома/ да се изчиства оперативната памет. Тази команда се изписва в началото на реда с последващо натискане на клавиша RTN.

За изчистване едновременно и паметта и екрана на ПК в началото на екрана /без номер на реда/ се изписват командите HOME: NEW и се натиска клавишът RTN.

В езика БЕИСИК двете точки служат за отделяне на две или повече команди или оператори, написани на един ред.

Вече е известно, че компютърът изпълнява определена програма не по наредбата /последователността/ на редовете, а по възходящия ред на техните номера.

Ако на дадената по-горе програма се разместят редовете, но се запазят техните номера резултатите също ще бъдат верни.

```
20 LET B = 38
```

```
30 LET C = A + B
```

```
10 LET A = 27
```

```
40 PRINT C
```

```
50 END
```

```
RUN
```

```
65
```

```
] □
```

Резултатът е верен, тъй като след натискане на клавиша RTN компютърът започва работа най-напред от ред 10, след това от ред 20, 30 и т.н.

Това подреждане на програмата по номера на редовете е особено важно в случаите, когато отделни части от програмата трябва да бъдат коригирани и редактирани.

Например при писане на горната програма може да се получи следното:

```
1Ø LET A = 27
2Ø LET B = 38
3Ø PRINT C
4Ø END
```

Както се вижда пропуснат е операторът `LET C = A + B`. В случая пропуснатият оператор се пише в края на програмата като му се нанася такъв номер на реда, с който до попадне на съответното му място. За горната програма посоченият оператор трябва да попадне между редове 2Ø и 3Ø. Очевидно тук може да се постави произволен номер между 21 и 29. Обикновено, за да има резерви преди и след този оператор, целесъобразно е да му се присвои номер 25. В програмата допълнението ще има следния вид:

```
1Ø LET A = 27
2Ø LET B = 38
3Ø PRINT C
4Ø END
25 LET C = A + B
```

Ако след всичко това се нанесе `LIST` и се натисне `RTN`, програмата ще бъде коригирана и подредена по следния начин:

```
1Ø LET A = 27
2Ø LET B = 38
25 LET C = A + B
3Ø PRINT C
4Ø END
```

Когато програмата е на екрана и се напише операторът RUN резултатите се появяват, както се вижда, непосредствено под нея. В случаите, когато е необходимо резултатите да се отпечатат върху нов кадър, то трябва екранът чрез командата HOME да се изчисти. По-нататък на изчистения екран чрез командата RUN и клавиша RTN ще се появи резултата.

При някои системи персонални микрокомпютри програмирането на езика БЕЙСИК изисква задължително във всеки ред наличие на оператора LET. При други системи, какъвто е и ПМК ПРАВЕЦ-82, операторът LET може да не е задължителен.

Например програмата:

```
100 LET A = 34
110 LET B = 16
120 LET C = A * B
130 PRINT C
140 END
```

е еквивалентна на програмата:

```
100 A = 34
110 B = 16
120 C = A * B
130 PRINT C
140 END
```

Наред с посоченото по-горе при някои системи, каквато е и ПРАВЕЦ-82, не е необходимо употребата на оператор END.

Посочените до тук примери показват използването на БЕЙСИК само за една аритметична операция. В следващата програма е дадено решението на една комбинирана задача при използване на четирите аритметични действия.


```

11Ø LET A = 5
12Ø LET B = 7
13Ø LET C = A + B
14Ø LET D = A - B
15Ø LET A = 2
16Ø LET E = A * B
17Ø LET B = 8
18Ø LET F = A/B
19Ø PRINT A, B, C
20Ø PRINT D, E, F,

```

В програмата има осем оператора за присвояване LET. След на-
насяне на командата RUN и натискане на клавиша RTN компютърът из-
вършва следното:

1. На променливата A се присвоява значението 5;
2. На променливата B се присвоява значението 7;
3. На променливата C се присвоява значението 12 /A+B/;
4. На D се присвоява -2 /A-B/.

Посоченото до тук в паметта на машината се вижда на фиг. 8.

5	7	12	-2
A	B	C	D

Фиг. 8

5. По нататък на A се присвоява 2, при което съдържанието на
A не води до изменение на съдържанието на C или D /фиг. 9/.

2	7	12	-2
A	B	C	D

Фиг. 9

6. На E се присвоява значението $14 / A \times B /$.

7. Изменя се съдържанието на клетка B на 8, което не указва влияние на съдържанието на другите клетки.

8. На F се присвоява значението $0,25 / A/B /$, фиг. 10.

2	8	12	-2	14	.25
A	B	C	D	E	F

Фиг. 10

9. Накрая в два реда се отпечатва /изобразява на екрана/ съдържанието на клетки A, B, C, D, E, F /респ. 2, 8, 12, и на втория ред: -2, 14, .25/.

Тук е мястото да се обърне внимание на начините за отпечатване на резултата. Когато се отпечатва само една величина същата се изразява на първа позиция /в нашия случай 2/. Когато обаче резултатите са два, три и повече, то стойностите им се отпечатват съответно на 17-та и 33-та позиция /т.е. през 16 позиции/. При повече от три резултата, например четвъртия, машината автоматично отпечатва на първа позиция от следващия ред. Следователно при ПМК ПРАВЕЦ-82 всеки ред може да съдържа не повече от три величини, тъй като на един ред има 40 позиции.

Поради това, че в горната програма трябва да се отпечатат

шест резултата, то в два последователни реда се пише операторът PRINT. Разделянето след този оператор на величините със запетайка /A, B, C и D, E, F/, осигурява отпечатване на тяхното съдържание на три места през 16 позиции.

1	17	33
2	8	12
-2	14	.25

Ако обаче вместо запетайка се постави точка и запетайка /A; B; C и D; E; F/ числата се отпечатват едно до друго вляво на екрана.

2812
-214.25

Освен по посочения до тук начин на съставяне на програмите като за всяка променлива се отделя самостоятелен ред възможно е същите да се изписват и в един ред.

Например програмата:

100 A = 144

110 B = 12

120 C = A/B

130 ? A, B, C

Може да се напише и по следния начин:

100 A = 144: B = 12: C = A/B: ? A, B, C.

В тази програма всеки оператор се отделя от другия чрез двоеточие.

Тази програма може да се нанесе и по следния начин:

100 A = 144: B = 12: C = A/B

110 ? A, B, C

Като недостатък на този начин на писане на програми може да се посочи трудността на коригирането на грешки.

С цел намаляване времето за писане на програми може да се съкращава броят на операторите и редовете. Например горната програма може да се състави и по следния начин:

```
100 A = 144: B = 12
```

```
100 ? A/B
```

Както се вижда тук операторът PRINT има по-сложни функции, като осигурява не само отпечатване на резултата, но и извършване на аритметичната операция /в случая деление/.

Във всяка програма, променливата на която не е дадена стойност и се иска отпечатване на нейното съдържание, то се получава нула /0/.

Още в началото бе посочено, че променливите се означават с една латинска буква /A, B, C, D, E и т.н./, с две латински букви /AB, AC, AD, BL, BM и т.н./ или с латинска буква и до нея цифра /A 1, A 2, A 3, A 4, B 1, B 2, B 3, B 4 и т.н./.

Съставянето на по-сложни програми изисква анализиране състоянието на клетките на присвоените променливи.

Това в известна степен се изяснява в дадените по-долу програми:

Програма №1

```
100 A = 10
```

```
110 B = 20
```

```
120 C = A * B
```

```
130 A = 5
```

```
140 ? A, B, C
```


$$15\emptyset B = B + 1\emptyset$$

$$16\emptyset C = A \times B$$

$$17\emptyset ? A, B, C$$

$$18\emptyset A = A + 1$$

$$19\emptyset B = B + 1$$

$$20\emptyset ? A, B, C$$

Да анализираме изпълнението на всеки един от операторите.

Например включително до ред 13 \emptyset състоянието на клетки A, B и C е показано на фиг. 11. Съдържанието на C е резултат от умножението на A по B /1 \emptyset X 2 \emptyset /, докато след ред 13 \emptyset състоянието на клетка A се изменя от 10 на 5 /фиг. 12/.

10	2 \emptyset	2 $\emptyset\emptyset$
A	B	C

Фиг. 11

С ред 14 \emptyset същата информация се появява на екрана, съответно на 2-ра, 17-та и 33-та позиция. След преминаване на ред 15 \emptyset и съдържанието на клетки A, B и C е показано на фиг. 12.

5	3 \emptyset	15 \emptyset
A	B	C

Фиг. 12

При ред 15 \emptyset към първоначалното състояние на клетка B /2 \emptyset / се прибавя 10 /B + 1 \emptyset / и се извършва умножението на A с B /5 по 3 \emptyset /, Посоченото съдържание в клетките се появява на екрана /ред 17 \emptyset /.

В редове 18 \emptyset и 19 \emptyset към старото съдържание машината прибавя

по единица и се получават новите значения на $A/A = A + 1/$ и $B /B = B + 1/$.

Например старите съдържания на А и В бяха съответно 5 и 30 /фиг. 12/, а сега вече са 6 и 31 /фиг. 13/.

6	31	150
---	----	-----

А В С

Фиг. 13

Програма №2

$$100 \text{ A} = 20$$

$$110 \text{ B} = 2 \times \text{A}$$

$$120 \text{ C} = 5 \times \text{B}$$

$$130 \text{ M} = \text{A}$$

$$140 \text{ P} = \text{C} - \text{B}$$

$$150 \text{ ? A, B, C, M, P}$$

При изпълнение на тази програма ПМК извършва следното:

1. На променливата А се присвоява 20.
2. На променливата В се присвоява два пъти по А /2 по 20/.
3. На променливата С се присвоява пет пъти по В /200/.
4. На променливата М се присвоява съдържанието на А /20/.

Текущото значение на А е равно на 20, поради което и на М се присвоява същото значение.

5. На променливата Р се присвоява съдържанието на С /200/ минус съдържанието на В /20/, т.е. на Р се присвоява 160.

Съдържанието на променливите А, В, С, М и Р е показано на фиг. 14.

20	40	200	20	160
A	B	C	M	P

Фиг. 14

Програма №3

```

110 A = 2
120 B = 4
130 C = 5
140 M = 10
150 E = A + B + 15
160 P = A - 7 - C
170 X = B * 4 * M
180 H = A/2/M
190 I = A * 3
200 ? A, B, C
210 ? M, E, P
220 ? X, H, I

```

С изпълнението на оператор PRINT на екрана в три реда се получават следните резултати: 2, 4, 5, 10, 21, -10, 160, .1 и 8.

Програма №4

```

110 A = 2
120 B = 4
130 C = 5
140 D = 10
150 E = A + B - C - D - 3
160 F = B - A + C - D - 3

```

$$17\emptyset G = A \times B / C \times 2$$

$$18\emptyset H = B/C \times A \times D \times 3$$

$$19\emptyset I = A \div B \div 2$$

$$20\emptyset ? A, B, C$$

$$21\emptyset ? D, E, F$$

$$22\emptyset ? G, H, I$$

След стартиране на горната програма върху екрана се получава следното:

2	4	5
10	-12	-6
3.2	48	256

В тази програма особено внимание заслужават операторите в редове 150, 160, 170, 180 и 190. По-долу на табл. 1 са показани изчисленията, които трябва да се извършат.

Когато даден математически израз съдържа само операциите събиране и изваждане, то те се изпълняват в тази последователност, в която се срещат символите на операциите при преглеждане на ПМК в реда отляво надясно. Същото е и когато даден математически израз съдържа само операциите умножение и деление - те се изпълняват в последователност, в която се срещат символите за операциите в реда като от ПМК се преглеждат отляво надясно /р. 170 и 180/. Когато математическите изрази съдържат само операциите повдигане на степен /р. 190/, то те се изпълняват в същата последователност, в която се срещат символите за операциите при преглеждане на ПМК отляво надясно. Например:

$$400 A = 2 \div 3 \div 4, \text{ което означава}$$

410 PRINT A

$$A = 2 / 3^4 / = 4096$$

Т а б л и ц а 1

Номер на реда	Оператори на БЕИСИК	Математически изчисления
15Ø	E = A + B - C - D - 3	E=2+4-5-1Ø-3 = -12
16Ø	F = B - A + C - D + 3	F=4-2+5-1Ø+3=Ø
17Ø	G = A * B / C * 2	G= 2*4/5*2 = 3.2
18Ø	H = B / C * A * D * 3	H=4/5*2*1Ø*3 = 48
19Ø	I = A ^ B ^ 2	I = 2/4^2/ = 256

Когато в един математически израз се използват всички пет аритметични операции ПМК изпълнява същите в следната последователност:

1. Повдигане в степен.
2. Умножение и /или/ деление.
3. Събиране и /или/ изваждане.

П р и м е р:

$$5ØØ P = A - B / C ^ 3 * D + E / F * G ^ 2$$

При изпълнение на програмата ПМК преглежда математическия израз отляво надясно и изчислява C на степен 3 и запомня резултата. По-нататък ПМК изчислява G на степен 2 и също запомня резултата.

От гледище на ПМК задачата сега може да се представи по следния начин:

$$5ØØ P = A - B / \textcircled{1} + E * \textcircled{2}$$

Тук с 1 и 2 са дадени изчислените значения при степенуването.

По-нататък ПМК повторно преглежда същия математически израз и изпълнява всички аритметични операции умножение и /или/ деление, които машината среща при преглеждане отляво надясно.

Значението на B /но не $-B$ / се дели на ① и получаваме ③.

Значението на ③ се умножава на D и получаваме ④.

Значението на E се дели на F и получаваме ⑤.

Значението на ⑤ се умножава на ② и получаваме ⑥.

След горните изчисления задачата има следния вид:

$$P = A - ④ + ⑥$$

И накрая ПМК преглежда математическия израз трети и последен път. При това положение от A се изважда ④, получава се ⑦ и ⑥ се събира със ⑦, като полученият краен резултат се присвоява от P .

Контролни въпроси и задачи

1. Какво представлява номер на ред и за какво се използва?
2. Кой е минималният и максималният номер на реда?
3. Защо редовете трябва да се номерират с промеждутъци?
4. Може ли в една програма да има редове с еднакви номера?
5. За какво служи командата HOME?
6. За какво служи командата NEW?
7. За какво служи командата RUN?
8. Кой е операторът за присвояване?
9. Пред кои от операторите във въпроси 5, 6 и се поставя номер на реда?
10. Кои са символите /знаците/ за аритметичните операции - събиране, изваждане, умножение, деление, степенуване и коренуване?
11. Какво е съдържанието на M при LET $M = 5.6 + 4.2$?
12. Разгледайте оператора 20 LET $M = A * B$
Какво е значението на M , ако $A = 7$, а $B = 8$?

13. След коя команда ПМК проверява програмата за наличие на грешки?

14. Има ли грешки и какви са те в следната програма?

1Ø PRINT C

2Ø LET A = 4

3Ø LET B = 7

4Ø LET C = A Ж B

15. Разгледайте програмата:

1ØØ LET A = 12

11Ø LET B = 8

12Ø LET C = A Ж B

13Ø LET A = 5

14Ø LET B = 4

15Ø PRINT A, B, C

Какви числа компютърът отпечатва върху екрана?

16. Съставете програма, при която:

1. Да се присвои значение 16 на клетка от паметта, означена с X,

2. Да се присвои значение 17 на клетка от паметта, означена с Y,

3. Да се умножи значението на клетка X и Y и резултатът да се присвои от клетката M,

4. Да се отпечата съдържанието на клетки X, Y и M.

17. Да се състави програма, изпълняваща следните действия:

1. Да се присвои на клетка A значение 625,

2. Да се присвои на клетка B значение 5,

3. Да се раздели A на B и резултатът да се присвои от клет-

ка C,

4. Да се извади от клетка А съдържанието на клетка С и резултатът да се присвои от клетка М.

5. Да се отпечата съдържанието на клетки А, В, С и М.

18. Възможно ли е при ПМК ПРАВЕЦ-82 да не се използва операторът LET?

19. Позволява ли ПМК ПРАВЕЦ-82 при завършване на програмата да не се използва операторът END?

20. Кой оператор се използва с цел прехвърляне програмата от паметта на машината върху екрана?

21. Колко числа могат да се отпечатат на един ред върху екрана, ако отделните променливи се разделят със запетайка?

22. Как се отпечатват числата, ако се отделят с точка и запетайка?

23. Как ще се отпечатат четири резултата при команда PRINT A, B, C, D?

3.2.2. Начини за коригиране на програми

По-горе бяха дадени примери за извършване корекции на програми в случаите, когато курсорът се намира на същия ред, където е и грешката. Тук ще бъде разгледан въпросът за корекции, когато грешката е открита по-късно и трябва курсорът /маркерът/ да се върне на съответния ред.

Ако е съставена програмата:

1Ø A = 48

2Ø B = 3Ø

3Ø C = A - B

4Ø PRINT C

и откриваме, че е допусната грешка в ред 20, където числото 30 трябва да се коригира на 38. Най-лесният начин в този случай е накрая на програмата да се повтори ред 20 с коригираната информация. Например 20 В = 38. След което програмата се листва /LIST/ и на екрана се получава:

10 А = 48

20 В = 38

30 С = А - В

40 PRINT C

Коригирането на подобни грешки може да се извършва и по друг начин, като маркерът с помощта на определени клавиши от клавиатурата се мести вляво, вдясно, нагоре и надолу.

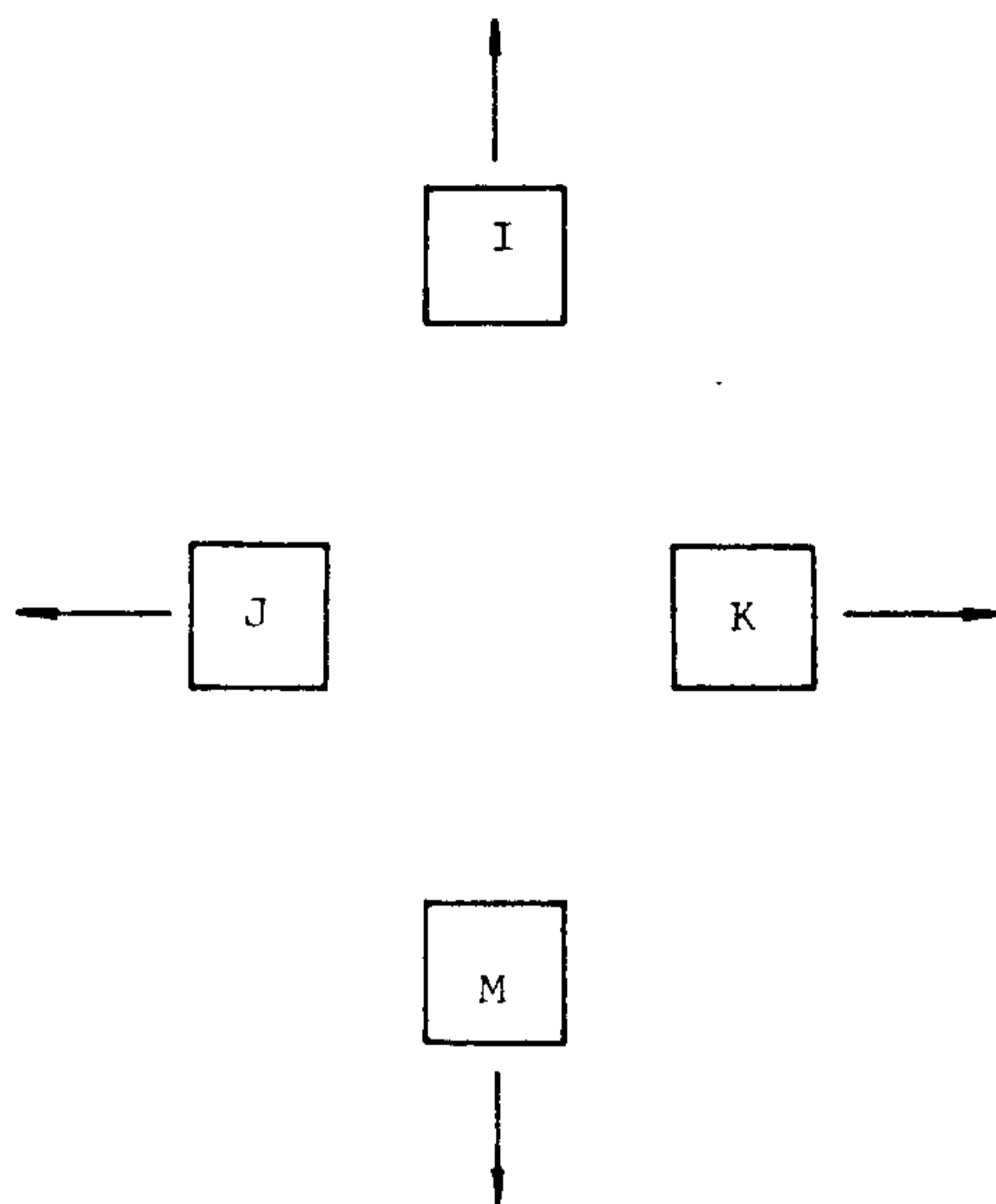
Това движение на маркера до произволна точка се извършва по следния начин:

1. Клавиатурата се превключва на долен регистър /латиница/.
2. Еднократно се натиска клавишът ОСВ, при което персоналният микрокомпютър преминава в т.нар. режим "освобождение". С натискането на този клавиш се изменя значението на почти всички други клавиши. И така при режим ОСВ клавишите I, J, M и K вече не печатат букви, а преместват маркера в четири посоки /така, както е показано на фиг. 14 а/. Преминаването на маркера през букви и цифри не води до тяхното изтриване.

3. С помощта на тези четири клавиша маркерът задължително се премества върху първата цифра на номера на реда, в който ще се извършва корекцията.

4. Преминава се в горен регистър /кирилица/ и постоянно се държи натиснат клавишът МК, като при постоянни натискания на клавиша У, намиращ се до буквата И, маркерът се премества надясно по реда.

Това местене продължава до попадане върху символа, който трябва да се коригира. След това клавишът МК се освобождава.



Фиг. 14 а

5. Извършва се корекцията /с кирилица или латиница/. В случаите, когато след коригирания символ има следващи символи /цифри или букви/, които е необходимо да се запазят, чрез клавиша МК и У маркерът се премества вдясно докато се премине след последния символ и едва тогава се натиска клавишът `RTN`. Ако символите след корекцията трябва да се елиминират, това се постига чрез непрекъснато натискане на интервалния клавиш.

6. За проверка правилността на корекцията се написва командата `LIST` и натиска клавишът `RTN`. Върху екрана се появява коригираната програма. Ако има друга грешка коригирането ѝ се извършва по посочения начин.

Разгледаният начин за коригиране на грешки ще бъде илюстриран със следния пример:

$1\emptyset\emptyset A = 3$

11Ø В = 5

12Ø С = 7

13Ø D = 12

14Ø Е = А Ж В

15Ø F = А Ж В Ж С

16Ø G = А Ж В Ж С Ж D

17Ø ? А, В, С

18Ø ? D, Е, F

19Ø ? G

В програмата е открито, че в ред 13Ø съдържанието трябва да бъде не 12, а 9.

За извършване на корекцията клавиатурата се превключва на латиница, еднократно се натиска клавишът ОСВ и чрез клавиша I маркерът се предвижва до ред 13Ø, като чрез клавиш J маркерът се предвижва вляво, така че да бъде върху цифрата 1. След това клавиатурата се премества в горен регистър /кирилица/ и постоянно се държи натиснат клавишът МК, а чрез клавиш У маркерът се премества надясно върху единицата на числото 12. Освобождава се клавишът МК и се натиска клавишът 9, като с това на мястото на 1 се отпечатва 9, а маркерът се премества върху цифрата 2. Тъй като същата трябва да се елиминира, се натиска еднократно интервалният клавиш и цифрата 2 се изтрива. Накрая се натиска клавишът RTN, при което маркерът се премества в началото на ред 14Ø. Независимо от това върху ред 14Ø се напечатва LIST и се натиска RTN. В резултат на това върху екрана се появява коригираната програма.

Когато в дадена програма трябва да се изтрие определен ред, то това може да се постигне като в края на програмата се написва само неговият номер. Така например, ако в горната програма трябва да

се елиминира ред 140, то под номер 190 се написва 140. Ако след това се нанесе LIST и RTN, то върху екрана се получава следното:

```
100 A = 3
110 B = 5
120 C = 7
130 D = 9
150 F = A * B * C
160 G = A * B * C * D
170 PRINT A, B, C
180 PRINT D, E, F
190 ? C
```

Както се вижда, в програмата ред 140 е изтрит. Тук трябва пак да се обърне внимание, че когато в първоначално създаваната програма за отпечатване на резултатите за по-голяма краткост вместо PRINT се напише ?, то при команда LIST на екрана се появява PRINT.

Във връзка с изтриването на отделни редове БЕЙСИК създава още една възможност. Например, ако трябва да се изтрият няколко последователни реда от някоя дълга програма, то се използва командата DEL /дилит, изтривам/.

Командата се пише на нов ред без номер и след нея се нанасят номерата на редовете, които трябва да се изтрият.

Например, ако в нашата по-горе зададена програма трябва да се изтрият редове 100, 110, 120 и 130 се постъпва по следния начин:

```
DEL 100, 130
```

Натиска се клавишът RTN и редове от номер 100 до номер 130 се изтриват.

При съставяне на програми често между номерата на редовете и променливата, между променливата и знака за равенство или друг знак

и т.н., може да има интервал повече от една позиция. Това не оказва отрицателно влияние върху качеството на програмата. Независимо от това обаче целесъобразно е да се пишат програмите така, както се пише с пишеща машина.

3.2.3. Отпечатване на български текст

При използването на ПМК в практиката често ще бъде необходимо чрез клавиатурата да се въвежда и изразява върху екрана български текст. Такъв текст може да се въвежда чрез касетофона или флопи дисковото устройство върху касета или дискета. Използването на български текст е необходимо и при извеждане на резултатната информация чрез печатащо устройство върху табулограми.

Например при разработването на делови управленски игри задаването на въпросите и получаването на отговорите трябва да бъде на български език при използване на кирилица. Същото се отнася при означаване на информацията на екрана върху предварително проектирани първични документи, наименованията на табулограмите и много други изисква използването на български текст.

За отпечатване на български текст се използва операторът PRINT, при което текстът се поставя в кавички.

Ако на екрана се нанесе текстът:

```
200 PRINT "Аз съм специалист в ОПКК"
```

и програмата се стартира /т.е. нанася се RUN и се натиска клавишът RTN/върху екрана ще се изобрази същият текст, но без номера на реда, без оператора за печат и без кавичките - Аз съм специалист в ОПКК.

Текстът се разполага в лявата горна част на екрана /на първия ред/, ако разбира се преди това е използван операторът HOME.

Когато е необходимо даден текст да се използва като заглавна част, например в първия ред, то това може да се постигне така:

200 ? " СПЕЦИАЛИЗАНТ"

Както се вижда първите кавички се поставят непосредствено след оператора за печат, а след това чрез интервалния клавиш по желание маркерът се премества вдясно и тогава се изписва съответния текст / в случая СПЕЦИАЛИЗАНТ/. Компютърът взема предвид кавичките и горния текст ще се отпечати например в средата на екрана.

Освен по посочения начин определен текст може да бъде разположен по желание на потребителя на определено място в хоризонтален и вертикален разрез. Това разположение на текста може да се постигне при използване на операторите HTAB /хоризонтален табулатор/ и VTAB /вертикален табулатор/. Непосредствено след тези оператори се нанасят цифрите чрез които се определя хоризонталното и вертикалното положение на текста.

Например, за нанасяне текст на първичен документ "Складова разписка", при което отпечатването да започне от 10-та позиция на екрана, програмата ще има следния вид:

200 HTAB 10: ? "СКЛАДОВА РАЗПИСКА"

За да се отпечата на средата на екрана /след 10-та позиция, т.е. първата буква да се нанесе от десетия интервал/ е необходимо да се нанесе командата RUN и да се натисне клавишът RTN. Очевидно за изпълнението на всяка програма /вкл. и при отпечатване на български текст/ е необходимо винаги да се изписва командата RUN и да се натисне клавишът RTN. В специалната литература нанасянето на командата и натискането на клавиша е прието да се нарича СТАРТИРАНЕ. По-ната-

тък при разглеждането на отделните въпроси този процес ще се отбелязва като стартиране.

И така при стартиране горният текст ще се означаи без кавички и ще започне от 10-та позиция.

Ако е наложително даден текст да се разположи не само от средата на определена позиция, но и някъде по средата на екрана, то трябва да се използва операторът VTAB. Следователно, ако горният текст трябва да се отпечата на 12-ти ред от 10-та позиция, програмата ще има следния вид:

```
200 VTAB 12: HTAB 10: ? "СКЛАДОВА РАЗПИСКА"
```

След стартиране на програмата текстът ще се отпечата в средата на екрана /от 10-та позиция на 12-ти ред/. Известно е, че VTAB не може да бъде повече от 24 и HTAB - не повече от 40. Необходимо е също така да се има предвид, че след един оператор PRINT /т.е. между кавичките/ не могат да се напишат повече от 239 символа /цифри и букви/. В случаите, когато ще се пише текст с по-голям брой символи трябва да се използва и съответен брой оператори PRINT.

В зависимост от това дали даден текст трябва да започва не от края, а с няколко интервала, то преди PRINT се поставя HTAB и броя на интервалите, а при без позиции в началото се изписва номера на реда, оператора за печат, кавички и веднага текста.

При необходимост от интервали между редовете /например между заглавието и текста/ в програмата се изписва оператора за печат без информация след него. Например при програмата:

```
200 PRINT
```

```
210 PRINT
```

на екрана два реда се оставят свободни.

В дадената по-долу програма се вижда приложението на оператора за печат в различни случаи:

- 100 ? 25 * 25
- 110 ? 10 * /-300/
- 120 ? -3000
- 130 ? 4000
- 140 ? "4000"
- 150 ? 10, 200, 3000
- 160 ? "10", "200", "3000"
- 170 ? "КОНСТАНТИНОВ"

При стартиране на екрана се получава следното:

```
625
-3000
4000
4000
10          200          3000
10          200          3000
Константинов
```

3.2.4. Програмиране при използване на оператори

IF, THEN, GOTO, FOR и NEXT

Последователността от действия, които трябва да се извършат при решаването на една задача се нарича алгоритъм.

Всяка икономическа или каквато и да е друга задача се решава при спазването на определена последователност, т.е. при прилага-

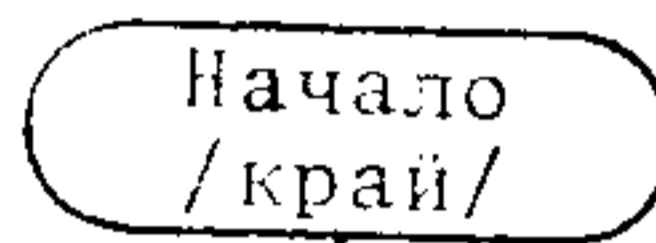
нето на даден алгоритъм.

Даден алгоритъм може да се изрази с помощта на текст или чрез използването на блокови схеми.

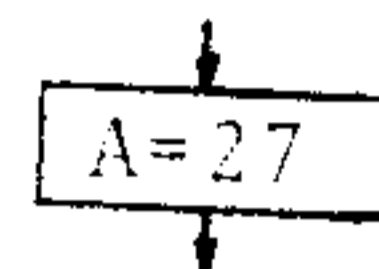
За съставянето на блокови схеми се използват следните основни символи:

1. Блок за начало

/блок за край/



2. Функционален блок



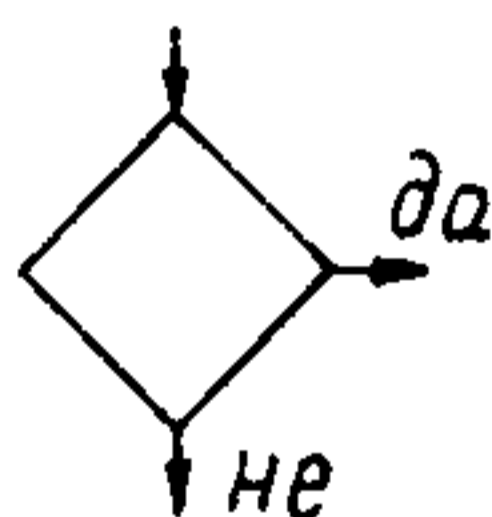
С помощта на този блок се описват различните действия, които се извършват от ПМК. Например тук се посочват извършването на такива действия като, какво се присвоява на една или друга променлива, какви аритметични операции трябва да се извършат $C = A * B$, и др.

3. Блок за въвеждане или извеждане на информацията -





Например, чрез оператора READ се осигурява въвеждане на информацията, а чрез оператора PRINT - извеждане на информация.

4. Логически блок



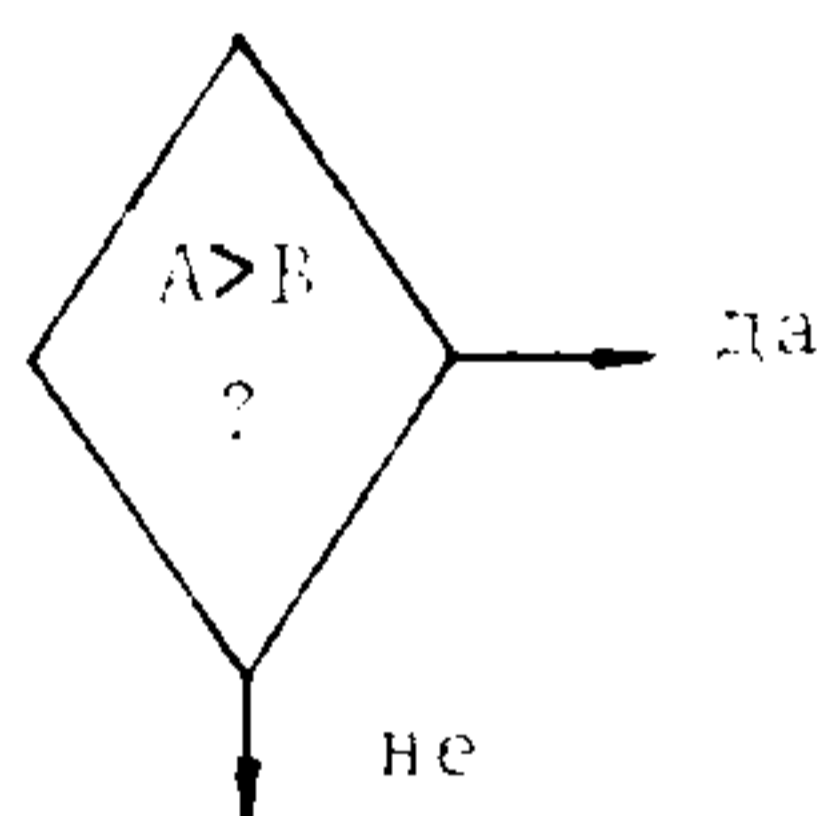
Когато в програмата се изисква проверка на някакво условие, то за означаване на това действие се използва логическият блок /ромб/. Обикновено от ромба излизат две стрелки - "да" и "не". Когато дадено условие е изпълнено, движението става по посока на стрелката "да", а когато условието не е изпълнено - по посока на стрелката "не".

Всяка блокова схема трябва да завършва с блок за край /както за начало/. При достигане на този блок се извършва спиране на действието.

Последен символ, използван при съставяне на блокови схеми - това е съединителя, Той представлява от себе си неголям кръг  или . Този съединител има предназначение да съкрати в блоковата схема броя на съединителните линии.

Така дадените символи, имащи различни геометрични фигури, осигуряват простота на разбирането на действията, които ще се извършват от ПМК.

Например при използването на логическия блок /фиг. 15/ може лесно да се разбере, че се проверява условието: А по-голямо ли е от В? Ако е "да", то програмата следва по едно направление /клон/, а ако е "не" - по друго. В случаите, когато на А е присвоено 12, а на В - 7, при проверка на горното условие посоката ще бъде "да", а ако А = 26, а В = 37 - посоката ще бъде към "не".



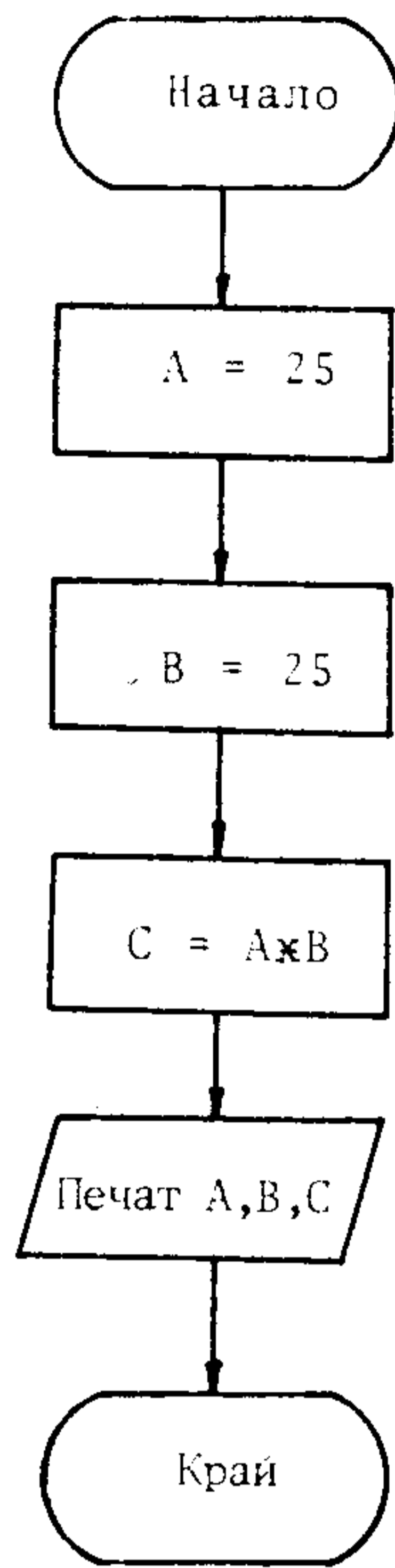
Фиг. 15

ПРИМЕР ЗА БЛОКОВА СХЕМА

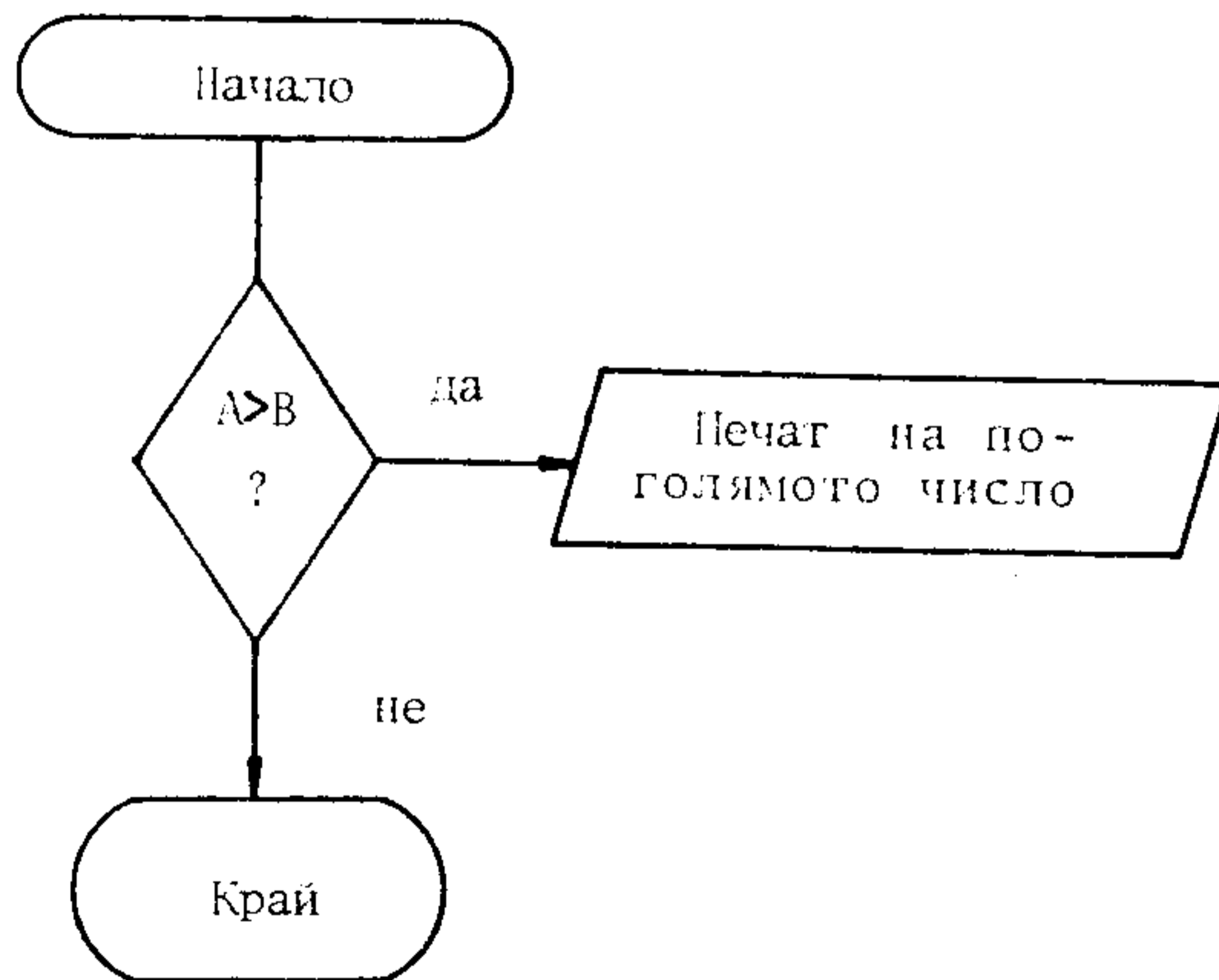
Ако ПМК трябва да извърши действието умножение блоковата схема ще има вид, показан на фиг. 16. В БЕЙСИК не се използва оператор за начало.

На блоковата схема са дадени три блока и един блок за извеждане /отпечатване/ на резултата.

В следващата блокова схема /фиг. 17/ е дадено отпечатването на числото, което има по-голяма стойност.



Фиг. 16



Фиг. 17

Големите възможности на ПМК се дължат на бързината на извършваните аритметични действия /200 000 операции в секунда/ и способността му да извършва повтарящи се действия. Или както посочва Т.Уорт, ако премахнете от ЕИМ тази способност, то в нея почти нищо не остава.¹

За извършване на повтарящи се действия може да се използва операторът `goto` /`гоу ту`, `върви към`, `премини към`/. Този оператор се пише с номер на реда и след него трябва да се нанесе число, което да укаже на ПМК към кой ред на програмата да се премине.

П р и м е р:

```
200 PRINT "ИКОНОМИСТ"
```

```
210 GOTO 200
```

При нанасяне и стартиране на тази програма /`RUN` и `RTN`/ ПМК ще започне работа от ред 200 и ще отпечата върху екрана думата ИКОНОМИСТ. След това ще премине към ред 210 от където операторът `goto` го изпраща отново към ред 200. При това положение ПМК наново отпечатва думата ИКОНОМИСТ и преминава към ред 210. Чрез оператора `goto` отново се преминава към ред 200 и т.н. На екрана на ПМК се отпечатва една след друга думата ИКОНОМИСТ /24 реда/, т.е. върху целия екран. Процесът не спира и се счита, че програмата се зацикля. За да се излезе от това положение трябва да се натисне червеният клавиш за нулиране /клавишът на последно място в първия ред на клавиатурата /фиг. 5/.

Операторът `goto` се нарича още оператор за безусловен преход.

За да се повторят действията от ПМК се използва и операторът `IF` `THEN` / `иф` ... `ден`, `ако` `тогава`/. Този оператор се пише с номер на реда, като след `IF` следва някакво условие,

¹Т. У о р т, Програмиране на языке БЕЙСИК, Машиностроение, М., 1981, с. 82.

а след THEN се пише номерът на даден ред. При положение, че дадено условие се изпълнява, ПМК преминава на посочения ред, ако обаче условието не се изпълнява ПМК преминава на следващия ред. Очевидно посоката, по която ще се движи ПМК зависи от това дали дадено условие се изпълнява или не. Поради тази причина операторът IF THEN се нарича още оператор за условен преход. Чрез този оператор ПМК може да "взема решения".

В условието след IF могат да се използват означенията: =, >, <, > =, < = и < > /не неравно/.

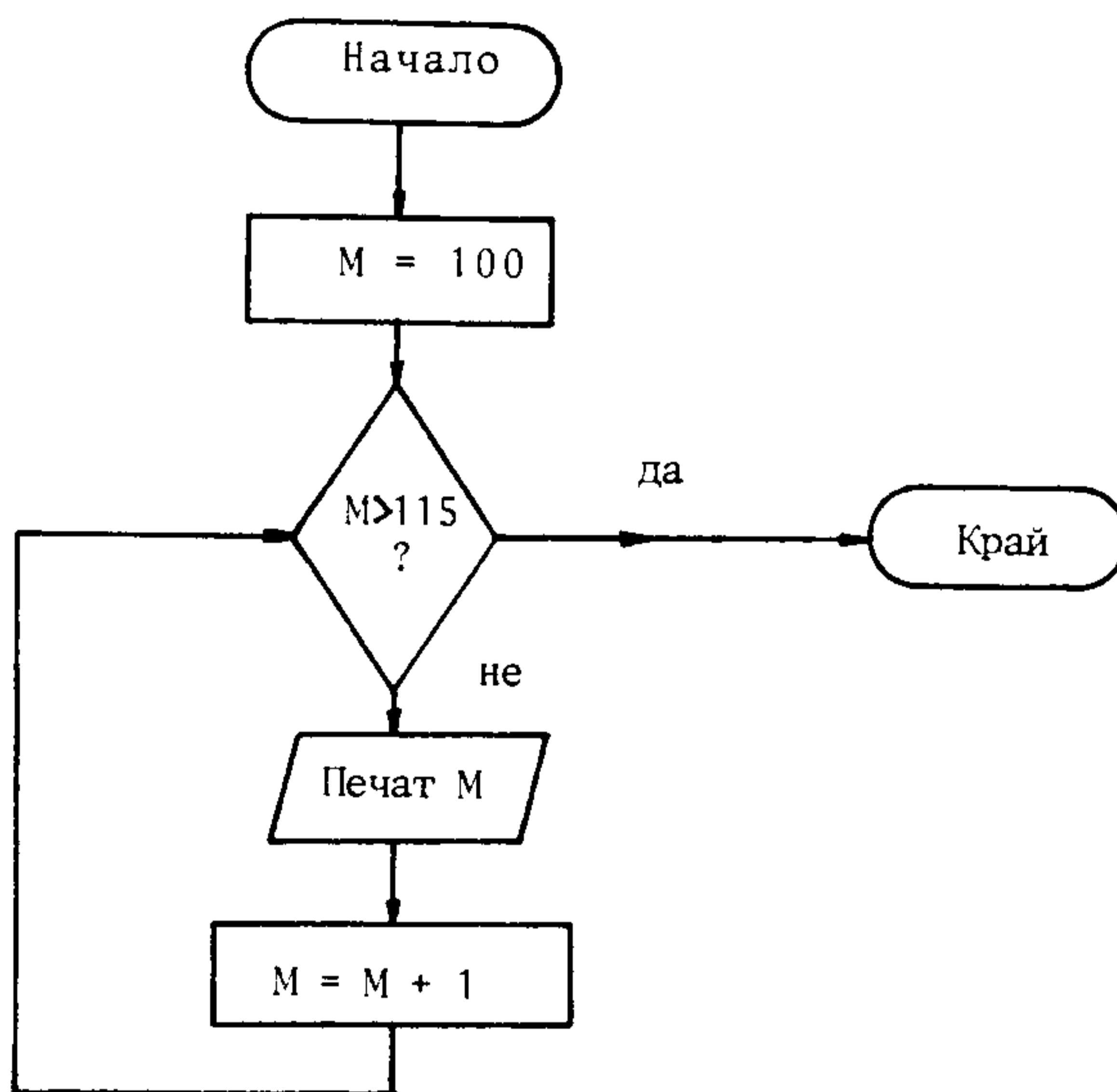
Алгоритъмът, при който ПМК трябва да отпечата табелните номера на работниците от номер 100 до 115 във вид на текст може да се изрази по следния начин:

1. Да се започне с числото 100.
2. Да се отпечата числото 100.
3. Да се премине към числото 101.
4. Да се отпечата числото 101.
5. Да се премине към числото 102.
6. Да се отпечата числото 102.
-
-
29. Да се премине към числото 114.
30. Да се отпечата числото 114.
31. Да се премине към числото 115.
32. Да се отпечата числото 115.
33. Да се премине към числото 116.
34. Да се спре.

Както се вижда тук, след като се започне с номера на работник 100, действията се повтарят. Получава се следният цикъл: отпечат-

ване на 100, прибавяне на 1, отпечатване на 101, прибавяне на 1, отпечатване на 102, прибавяне на 1 и т.н., докато се стигне до отпечатване номера на последния работник - 115 и спиране. За да се получи спиране, за отпечатване вкл. и на 115-ти номер на работника, а следващите да не се отпечатват, трябва да се проверява кога числото M , което при всеки цикъл нараства с 1, ще стане 116, за да изведе действието извън цикъла и го насочи към края на програмата за спиране /тази проверка в блоковата схема се извършва от логическия блок/.

На фиг. 18 е дадена блокова схема на алгоритъма на горната програма. С M е означен началният номер на първия работник /100/. Понататък постепенно M ще приема стойности 101, 102, 103, 104, и т.н.



Фиг. 18

След като на променливата M се присвои 100 се отива в логическия блок, където се прави проверка: числото M /в случая $M=100$ /

по-голямо ли е от 115? Ако не е /в нашия случай не е, тъй като $M=100$ / преминава се в следващия блок, където указанието е да се отпечата съдържанието на M /100/. В следващия блок /функционален/ указанието е $M = M + 1$, т.е. M да бъде равно на $100 + 1$ /101/. Оттук отново се отива в логическия блок и се прави проверка $101 > 115$. Отговорът естествено е "не" и пак се отива надолу, отпечатване, пак прибавяне на 1 / M вече е равно на 102/ и отново в логическия блок. По посочения начин ПМК отпечатва в една от последните си итерации 114 и в следващия блок се прибавя 1, като M става 115. В логическия блок се прави проверка $M > 115$, т.е. $115 > 115$ и понеже отговорът е "не" ПМК отпечатва последния желан номер на работник /115/, а в следващия блок M получава стойност 116 / $115 + 1$ / и за последен път отива в логическия блок /шестнадесети цикъл/ и се прави проверка $116 > 115$. Отговорът е "да" и се отива в блока за край, при което програмата завършва своята работа.

И така в крайна сметка програмата за отпечатване номерата на работниците от 100 до 155 ще има следният вид:

```

200 M = 100
210 IF M > 115 THEN 250
220 PRINT M
230 M = M + 1
240 GOTO 210
250 END

```

След стартиране на така написаната програма ПМК ще отпечата числата от 100 до 115 едно под друго върху екрана.

По-долу са дадени примери за използване на оператора IF:

```

600 IF M = P THEN 3600
610 IF T > 74 THEN 3800
620 IF 26 < B THEN 1200

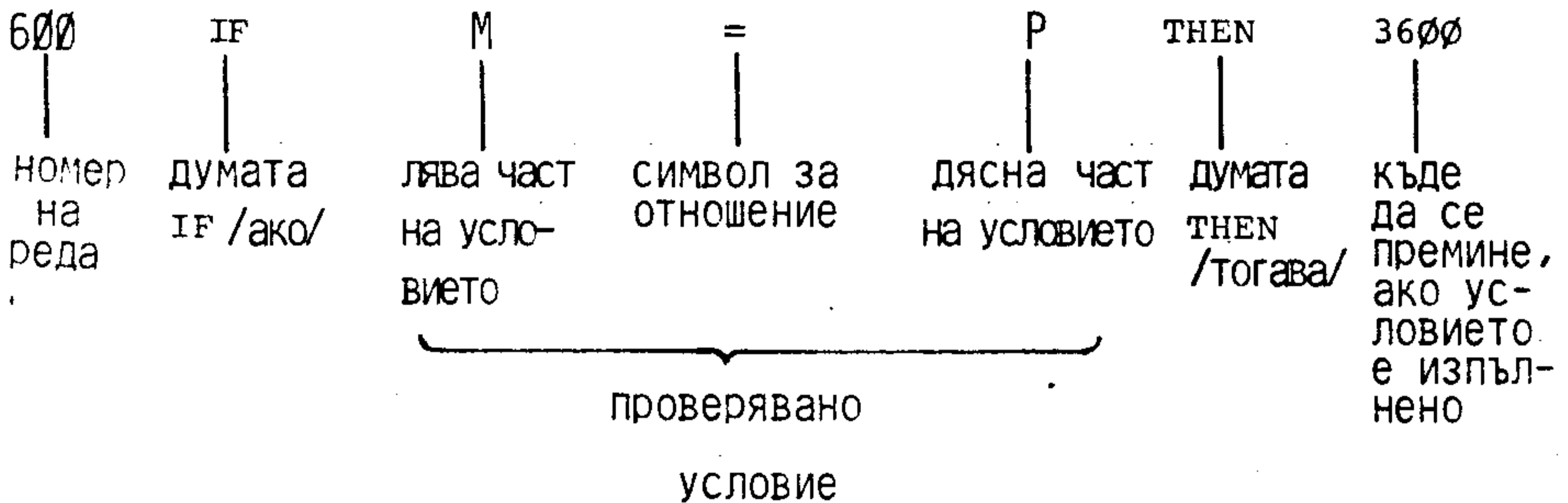
```

630 IFX <= /P-M/ X THEN 4200

640 IF /A - B/ Ж /C + T/ > = 36 THEN 4300

650 IFM <> K THEN 4400

Дадените примери показват, че операторът IF се състои от 7 части /фиг. 19/.



Фиг. 19

В дадения пример, ако $M = P$ ще се осъществи преход към ред 3600 и програмата ще продължи да се изпълнява от оператора, намиращ се в ред 3600. Ако M не е равно на P , програмата преминава към следващия оператор /ред 610/. От примерите се вижда също така, че лявата или дясната част в оператора IF до символа за отношението може да бъде число, променлива или математически израз.

Програмата, при която ще се отпечатват числата от 1 до 22 и техните квадрати може да има следния вид:

210 M = 1

220 IF M > 22 THEN 260

230 ? M, M Ж M

240 M = M + 1

250 GOTO 220

260 END

При стартиране на тази програма върху екрана се получават следните резултати:

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100
11	121
12	144
13	169
14	196
15	225
16	256
17	289
18	324
19	361
20	400
21	441
22	484

Програмата за отпечатване на числата от 1 до 22 и корен втори от тях може да се състави така:

$$210 \text{ M} = 1$$


```

22Ø IF M > 22 THEN 26Ø
23Ø ? M, SQR /M/
24Ø M = M + 1
25Ø GOTO 22Ø
26Ø END

```

След стартиране на горната програма върху екрана ще се отпечатат едно след друго числата от 1 до 22 и съответните изчисления от корен втори.

Програмата за отпечатване на числата от 100 до 500 и техните квадрати може да изглежда така:

```

21Ø M = 1ØØ
22Ø IF M > 5ØØ THEN 26Ø
23Ø ? M, M * M
24Ø M = M + 1
25Ø GOTO 22Ø
26Ø END

```

Характерното в тази програма е, че след нейното стартиране, получените резултати няма да се съберат на екрана. Числата отдолу нагоре ще започнат да се движат до последния си ред. Следователно върху екрана ще се фиксират последните 24 числа /т.е. резултатите/. Разбира се изпълнението на програмата може да бъде спряно по желание и по-рано.

Например в положение на кирилица се държи постоянно натиснат клавишът МК и еднократно се задействува клавишът с буквата С /намира се на третия ред вдясно от буквата А/. По този начин се спира изпълнението на програмата. Ако трябва отново да започне изпълнението на програмата, то без да се отпусне клавишът МК, повторно се натиска клавишът С. За нейното повторно /следващо/ спиране пак се натиска клавишът С и т.н.

Една програма за изчисляване на сумата $S = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + \dots + 100$, като се отпечата полученият резултат има следния вид:

```
200 S = 0 : K = 1
210 IF K > 100 THEN 250
220 S = S + K
230 K = K + 1
240 GOTO 210
250 PRINT S
```

В случая променливата S представлява сума от събирането на числата последователно от 1 до 100, поради което в началото ѝ се дава стойност 0. За да бъде решена задачата е необходимо на S да се дава при всеки цикъл $s+k$. А тъй като на K е присвоено 1 /ред 200/ още по време на първия цикъл на S ще се присвои 1. В началото K е равно на 1 /ред 200/ и се увеличава с 1 при всеки цикъл $/K = K + 1/$. Когато K нарасне на повече от 100, цикълът се е изпълнявал 100 пъти и сумата на числата от 1 до 100 вече е изчислена. Тази именно сума $/s/$ ще бъде отпечатана съгласно оператора в ред 250 /5050/.

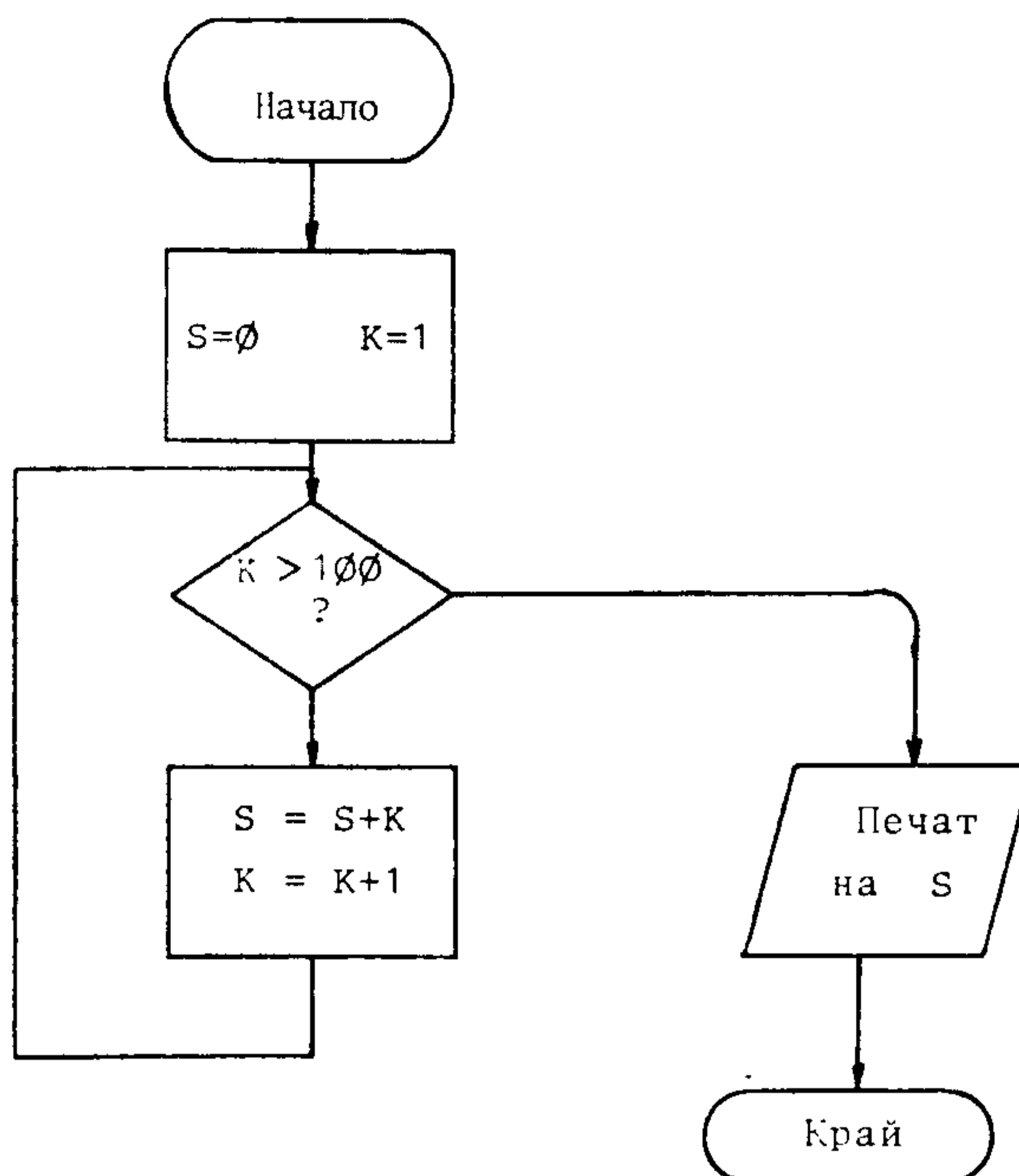
Блоковата схема на алгоритъма на тази задача има вид, показан на фиг. 20.

За автоматизиране на цикли при езика БЕЙСИК се използва специален оператор FOR ... NEXT /фор ... некст/. Този оператор съдържа брояч за итерации.

Със следващите няколко примера се илюстрират възможностите на този оператор.

Програма, при която да се отпечатаат числата от 1 до 22 и техните квадрати с използване на операторите FOR и NEXT, има следния вид:

```
200 FOR M = 1 TO 22
```



Фиг. 20

210 PRINT M, M * M

220 NEXT M

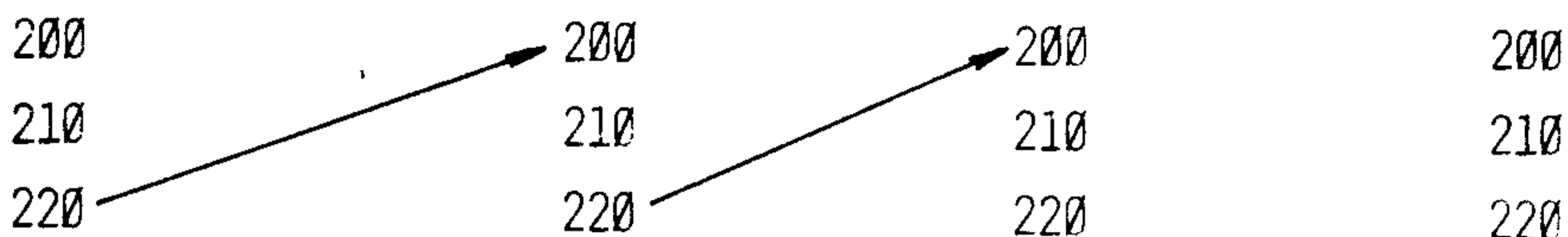
При стартиране на програмата на екрана ще се отпечатат всички числа от 1 до 22 и техните квадрати.

ПК тук извършва следното:

1. Установява началното значение на брояча /в дадения случай M/ на 1.
2. Извършва проверката дали броят е повече от 22?
3. Ако отговора е "не", изпълнява се оператор, образуващ тялото на оператора, след това броят се увеличава с единица и отново се извършва проверка на условието за край на цикъла. Ако условието

се изпълнява, програмата излиза от цикъла, а в противен случай програмата продължава работа,

Накрая се нанася `NEXT M` за означаване долната граница на цикъла. При горната програма със стартирането се преминава 22 пъти през трите реда:



Освен по посочения начин /отпечатване последователно на числата от 1 до 22 и техните квадрати/ при използването на оператора `FOR.... NEXT` може да се използва и определена стъпка, т.е. числата да се отпечатват през 0,5, през 1,5, през 3 и т.н.

На фиг. 21 е дадена блоковата схема.

Програмата, при която да се отпечатат числата от 1 до 22 и техните квадрати, при стъпка 0,5, може да се състави така:

```
200 FOR M = 1 TO 22 STEP 0.5
210 PRINT M, M * M
220 NEXT M
```

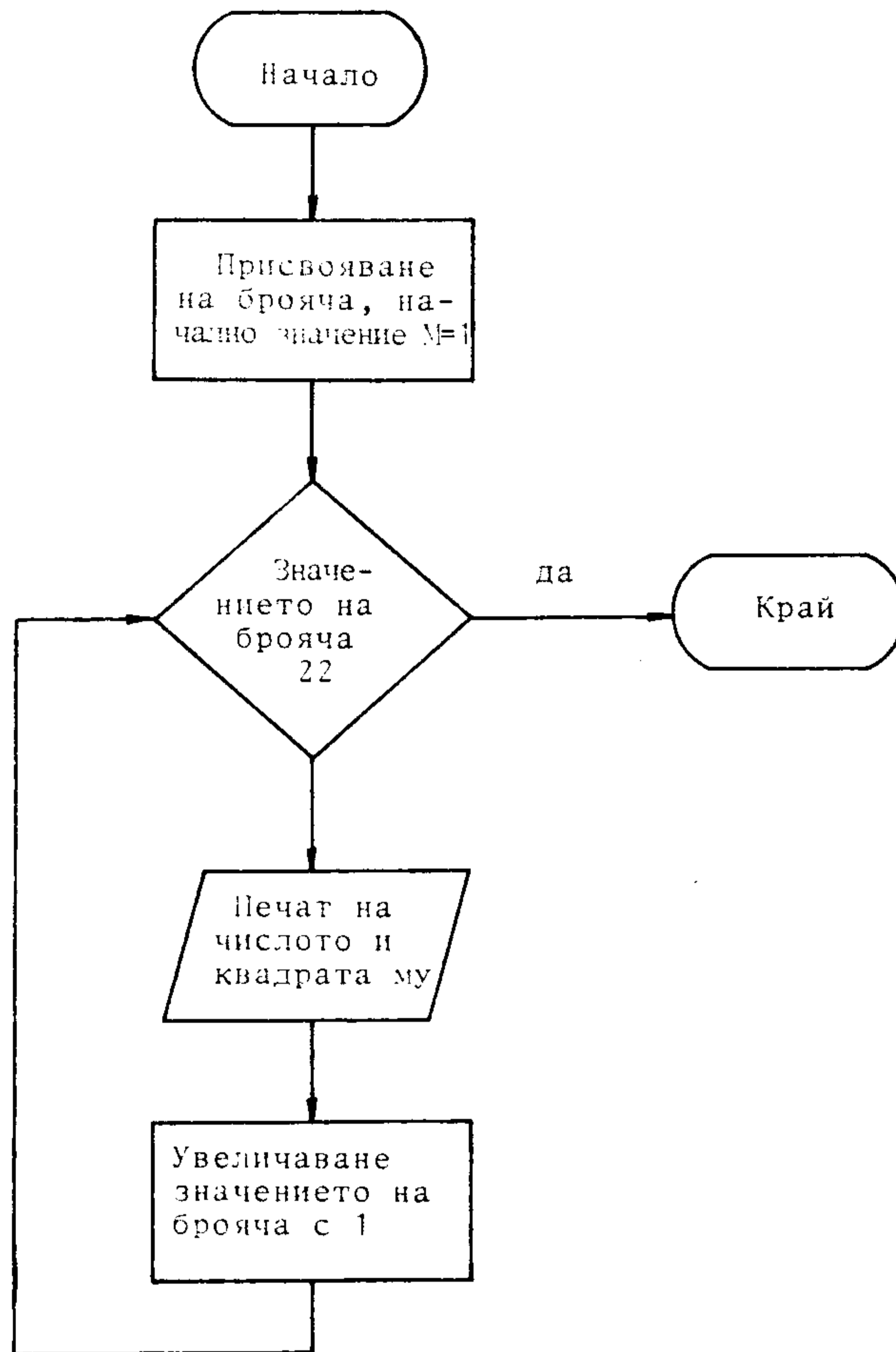
Със стартирането на тази програма на екрана ще се отпечатат числата 1, 1,5, 2, 2,5, 3, 3,5 и т.н. и техните квадрати.

Когато в първия ред липсва `STEP` това означава, че стъпката е 1.

Програма, при която текстът "ЗАВОД ВАСИЛ КОЛАРОВ" се отпечатва 10 пъти има следния вид:

```
200 FOR M = 1 TO 10
210 PRINT "ЗАВОД ВАСИЛ КОЛАРОВ"
220 NEXT M
```

При стартиране на тази програма върху екрана горният текст



Фиг. 21

ще се отбележи 10 пъти. Този пример няма практическо приложение, а има за цел да покаже възможностите на ПМК.

Една програма за изчисляване на сумата на всички цели числа от 1 до 100 изглежда така:

```
200 FORM = 1 TO 100
```

```
210 A = A + M
```

220 NEXT

230 PRINT A

В ред 200 при първия цикъл на M се дава 1, а на A - 0, то в ред 210 A ще бъде равна на 0 + 1. Във втория цикъл M = 2, A = 1, а при ред 210 A = 1 + 2, при третия цикъл - M = 3, A = 1 + 2 + 3, при четвъртия цикъл M = 4, A = 1 + 2 + 3 + 4 и т.н.

Контролни въпроси и задачи

1. Какво представлява цикъла?
2. Каква роля играе в цикъла операторът GOTO?
3. Как се спира програмата при зацикляне?
4. За какво служи в езика БЕЙСИК операторът IF?
5. Какви символи за отношение се използват в програмите, писани на езика БЕЙСИК?
6. Кой са трите основни съставни части на условията, проверявани с оператора IF?
7. Какво извършва програмата, ако условието проверявано в оператора IF се удовлетворява /т.е. приема значение "да", "истина"/ и когато не се удовлетворява /приема значение "не", "лъжа"/?
8. Какво представлява блоковата схема?
9. Какво означава операторът: 200 LET A = A + B?
10. Какво означава операторът: 200 LET A = A + 1?
11. За какво служи операторът FOR и NEXT?
12. Какво е предназначението на променливата, която се записва непосредствено след оператора FOR?
13. Какво начално значение може да се присвои на променливата, указана след оператора FOR?
14. За какво служи думата STEP?

3.2.5. Програмиране при използване на оператори за данни

Характерно при използването на ПМК, специализирани микро-компютри или терминали е работата в диалогов режим. В тези случаи при изпълнението на програмите се изисква еднократно или многократно участие на потребителя. Ако потребителят не вземе участие, ПМК не продължава по-нататъшната си работа. Диалоговото участие на потребителя се състои във въвеждане на данни, въвеждане на думи и др.

Въвеждането на данни по време на изпълнение на програмата може да се извърши с помощта на оператора `INPUT /въведи/`:

Пример 1

```
200 INPUT A
```

```
210 ? A Ж A
```

```
RUN
```

```
? 
```

При стартиране на програмата същата започва да се изпълнява като в ред 200 среща оператор `INPUT` и се отпечатва знака за въпрос. Това означава, че ПМК чака да бъдат въведени данни /в случая стойността на `A`/. Когато върху клавиатурата се нанесе числото 25, от компютъра се възприема като стойност на променливата след `INPUT`, т.е. на `A`. При натискане на клавиша `RTN` на екрана се появява резултатът, равен на 625. Изпълнението на тази програма изисква еднократно въвеждане на данни.

За многократно въвеждане на данни горната програма трябва да има следния вид.

Пример 2

```
200 INPUT A
```

```
210 ? A Ж A
```

```
220 GOTO 200
```

След стартиране на програмата на екрана се получава въпрос. Ако върху клавиатурата се нанесе числото 29 и се натисне клавишът RTN на екрана под въпроса ще се появи 841 и едновременно с това на екрана отново се получава въпрос. Това дава възможност отново да се въведе дадено число за А и т.н.

Освен въвеждането на едно число, операторът INPUT дава възможност за дефиниране на повече числа.

П р и м е р 3

```
200 INPUT A, B, C
```

```
210 PRINT A+A, B*B, SQR /C/
```

Със стартиране на горната програма върху екрана се появява въпросителна, която означава, че чрез клавиатурата трябва да се нанесат три числа. Ако се нанесат числата 10, 12 и 625, то на екрана под въпросителната ще се проявят числата 20, 144 и 25.

П р и м е р 4

```
200 INPUT P
```

```
210 IF P = 0 THEN 240
```

```
220 PRINT P, P Ч 3
```

```
230 GOTO 200
```

```
240 END
```

```
RUN
```

При стартирането на тази програма, след появата на въпросителна /под въздействието на оператор INPUT /се нанася дадено число. По-нататък действията са както бе обяснено по-горе. Особеността тук се състои в това, че програмата ще прекъсне своята работа тогава, когато се въведе числото 0. Разбира се работещият на машината може да избере и което и да е друго число, но числото трябва да се постави в началото след IF /ред 210/. Ако на въпроса след RUN на горната прог-

рама нанесем 12, 34, 67 на екрана се получава:

? 12	
12	1728
? 34	
34	39304
? 67	
67	300763

Към операторите за данни се отнасят и операторите DATA /дейта, данни// и READ /риид, четй/. С помощта на тези оператори се постига създаването на програми, които осигуряват решаването не на една, а на повече задачи. Изходните данни в тези случаи трябва да бъдат получавани от източника, който се изменя преди изпълнението на програмата.

П р и м е р 5

```
200 DATA 37
200 READ X
220 PRINT X, X2, X3
230 END
```

Същността на оператора READ се състои в това, че той чете значението на оператора DATA /данни/. Това означава, че значението на X ще бъде получена от оператора DATA. В случая от оператора DATA ще бъде прочетено значението 37. Програмата ще осигури напечатването на 37, а също така 37^2 и 37^3 /съответно 1369 и 50653/. При по-нататъшното изпълнение на програмата операторът DATA може да бъде изменен. Например операторът DATA сега може да съдържа числото 16 и програмата го чете като го присвоява на променливата X и отпечатва 16, 16^2 и 16^3 /като отпечатва 16, 256 и 4096/.

Значението на тези два оператора е голямо, особено що се отнася до решаването на икономически задачи.

Пример 6

```
200 DATA 4, 9, 15, 28, 45, 86, 116
210 READ B
220 PRINT B, B * B
230 GOTO 210
240 END
```

В този пример операторът DATA съдържа не едно, а набор от значения /седем числа/. При започване на работа, операторът READ чете число от оператора DATA и го присвоява на променливата B. По-нататък програмата отпечатва значението на променливата B и умножава B по B. След това се извършва преход към оператора READ и се повтарят действията. Първоначално от оператора DATA се чете числото 4, след това 9, 15 и т.н. Тъй като операторът DATA съдържа седем числа, то програмата осигурява отпечатването на осем реда. В момента на прочитане на всички числа от оператора DATA на екрана се появява съобщението ? OUT OF DATA ERROR IN 210, което показва, че изпълнението на програмата е завършено и същата е спряла на ред 210.

Изпълнението на програмата от пример 6 има следния вид:

4	16
9	81
15	225
28	784
45	2025
86	7396
116	14456

Във всяка отделна програма разположението на операторите DATA и READ може да бъде на произволно място, но DATA не може да

бъде след END. Наред с посоченото след READ може да има две и повече величини, а данните могат да се въведат чрез два или повече оператора DATA.

Операторът READ в една програма може да чете едновременно две или повече значения.

Пример 7

```
200 DATA 3, 7, 9, 4, 6, 10
```

```
210 READ A, B
```

```
220 PRINT A, B, A * B
```

```
230 GOTO 210
```

```
240 END
```

При стартирането на тази програма най-напред на променливите A и B се присвояват съответно значенията 3 и 7, след това 9 и 4 и накрая 6 и 10. На екрана ще се нанесат три реда със следното съдържание:

3	7	21
9	4	36
6	10	60
?	OUT OF DATA ERROR IN	210

Операторът DATA трябва да съдържа пълен набор от значения. Ако в горната програма операторът DATA съдържа още едно число /респ. едно число по-малко/ същото няма да се възприеме от променливата A и пак след третия ред ще има съобщение за спиране на програмата в ред 210.

Необходимо е да се отбележи, че значенията на оператора DATA се четат в този порядък в който се изпълняват в програмата от оператора READ.

Пример 8

```
200 DATA 2, 4, 5, 7, 8, 10, 13, 15, 20
210 READ A, B
220 READ C
230 PRINT A, B, C*2
240 GOTO 210
250 END
```

Значенията от оператора DATA ще бъдат присвоени от променливите A, B и C по следния начин: A→2; B→4; C→5; A→7; B→8; C→10; A→13; B→15; C→20.

След стартиране на програмата върху екрана на ПМК, ПРАВЕЦ-82 се получава следното:

```
2           4           25
7           8           100
13          15          400
?          OUT OF DATA ERROR IN           210
```

Използването на операторите DATA и READ дава възможност при решаване на икономически задачи да се извършва проверка на определени показатели /например дали дадени номера на работници са сортирани във възходящ ред/.

Пример 9

```
210 DATA 8, 10, 12, 30, 35, 40, 52, 64, 1000
220 READ C
230 READ B
240 IF B=1000 THEN 300
250 IF B>C THEN 280
260? "Номерата не са подредени"
270 GOTO 310
```



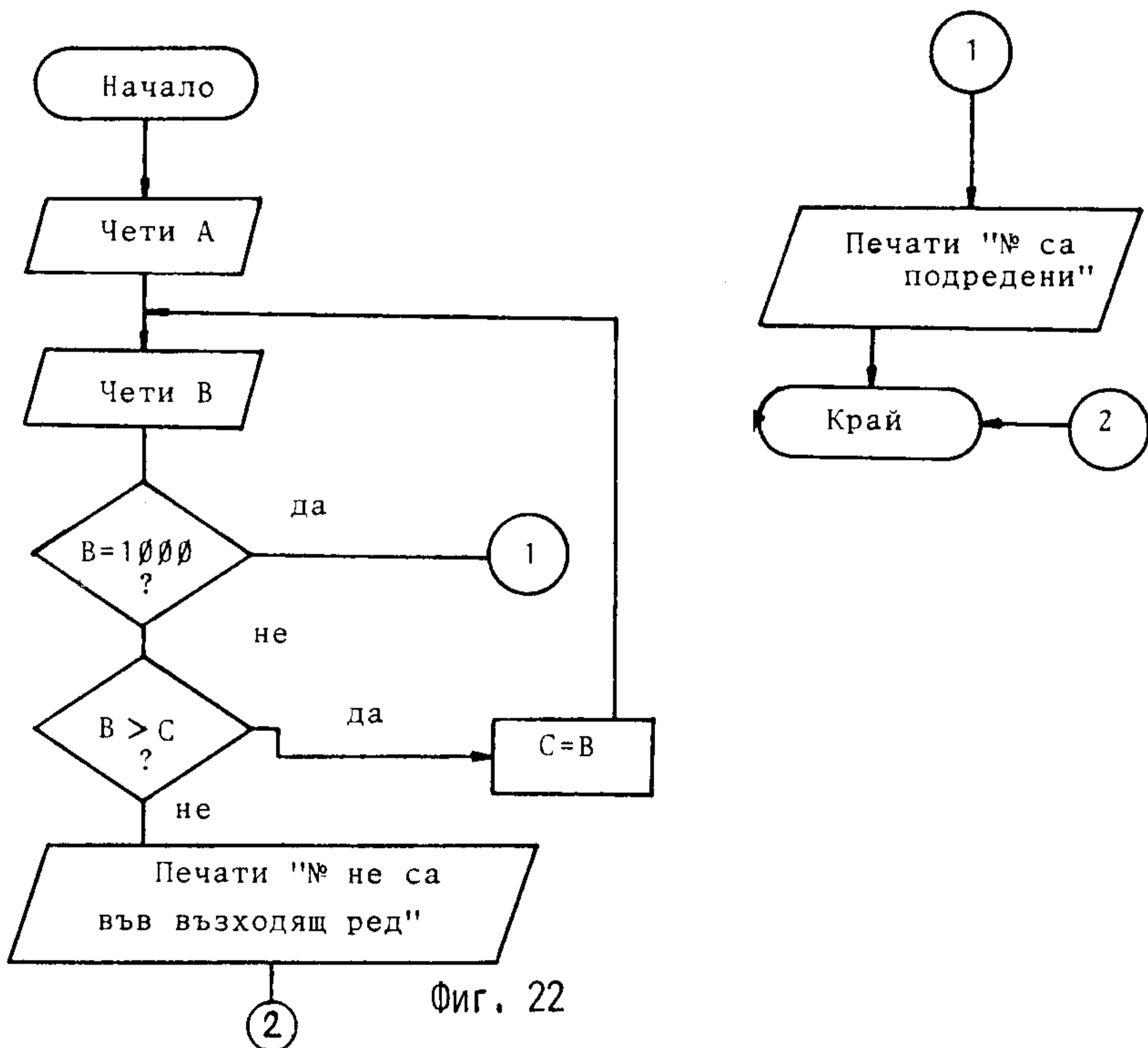
```

280 LET C=B
290 GOTO 230
300 ? "Номерата са подредени"
310 END

```

Числото 1000 е фиктивно. Когато същото бъде прочетено, то това е признак, че са обработени всички числа и трябва да се отпечата информацията от ред 260 /номерата са подредени/.

Алгоритъмът на пример 9 е представен на блоковата схема на фиг. 22.



Решението на задачата в пример 9 протича така: в началото на С и В се присвояват съответно стойностите 8 и 10. След това се проверява дали $B = 1000$ /фиктивното най-голямо число/. По-нататък се прави проверка дали В е по-голямо от с /ред 250/. Ако е "не" това означава, че числата не са подредени във възходящ ред и ще се отпечата

информацията от ред 260, а при "да" на с се присвоява съдържанието на В, т.е. дава се стойност 10. И тъй като от ред 290 програмата отива на ред 230 /READV/, то от DATA се чете следващото число след 10, а именно 12. По-нататък числото 12 се сравнява с 1000 и с с и понеже 12 е по-голямо от 10, на с се присвоява съдържанието на В /12/ и т.н., докато последното число от DATA /фиктивното число 1000/ се прочете от READ /ред 230/ и се присвои от В, като в ред 240 се проверява дали В е равно на 1000. Отговорът е "да" и компютърът се установява на ред 300 за отпечатване и номерата на работниците са правилно подредени.

Контролни въпроси и задачи

1. По какъв начин ПМК съобщава, че иска да се въведат числа?
2. Може ли чрез оператора INPUT да се въвеждат едновременно няколко числа? Как се отделят същите едно от друго?
3. Как може да се прекрати изпълнението на програма, която съдържа оператор INPUT?
4. Каква е разликата между операторите READ и INPUT?
5. В кой оператор е целесъобразно да се съхраняват данните в програмата?
6. Кой оператор трябва да се прилага, за да се четат данните от оператора DATA?
7. Къде може да бъде разположен операторът DATA на програмата при езика БЕЙСИК?
8. Какво съобщение отпечатва ПМК, когато в оператора DATA не остават числа за четене?

4. ГРАФИЧЕН РЕЖИМ НА РАБОТА

4.1. Режим на груба графика

Използването на персоналния микрокомпютър ПРАВЕЦ-82 дава възможност за конструиране, проектиране и изобразяване на екрана на различни фигури, схеми, графики и форми на първични и резултатни табулограми. За целта трябва да се прилага графичният режим на работа. Този режим има два варианта: режим на груба графика и режим на финна графика.

Режимът на грубата графика или още наричан графичен режим с ниска разделителна способност се характеризира с това, че чертаенето върху екрана се извършва чрез използването на едно правоъгълно "петно", което е голямо колкото размерите на маркера.

Екранът при този режим се разглежда като една координатна система, която по хоризонтала и вертикала съдържа по 40 правоъгълници /от 0 до 39/. Тази координатна система е показана на фиг. 23.

В така определеното поле могат да се чертаят различни груби фигури. Вътре в правоъгълника не може да се пише текст, а такъв може да се нанася в последните четири реда под правоъгълника.

В следващия пример е дадена програма за чертаене на малък правоъгълник.

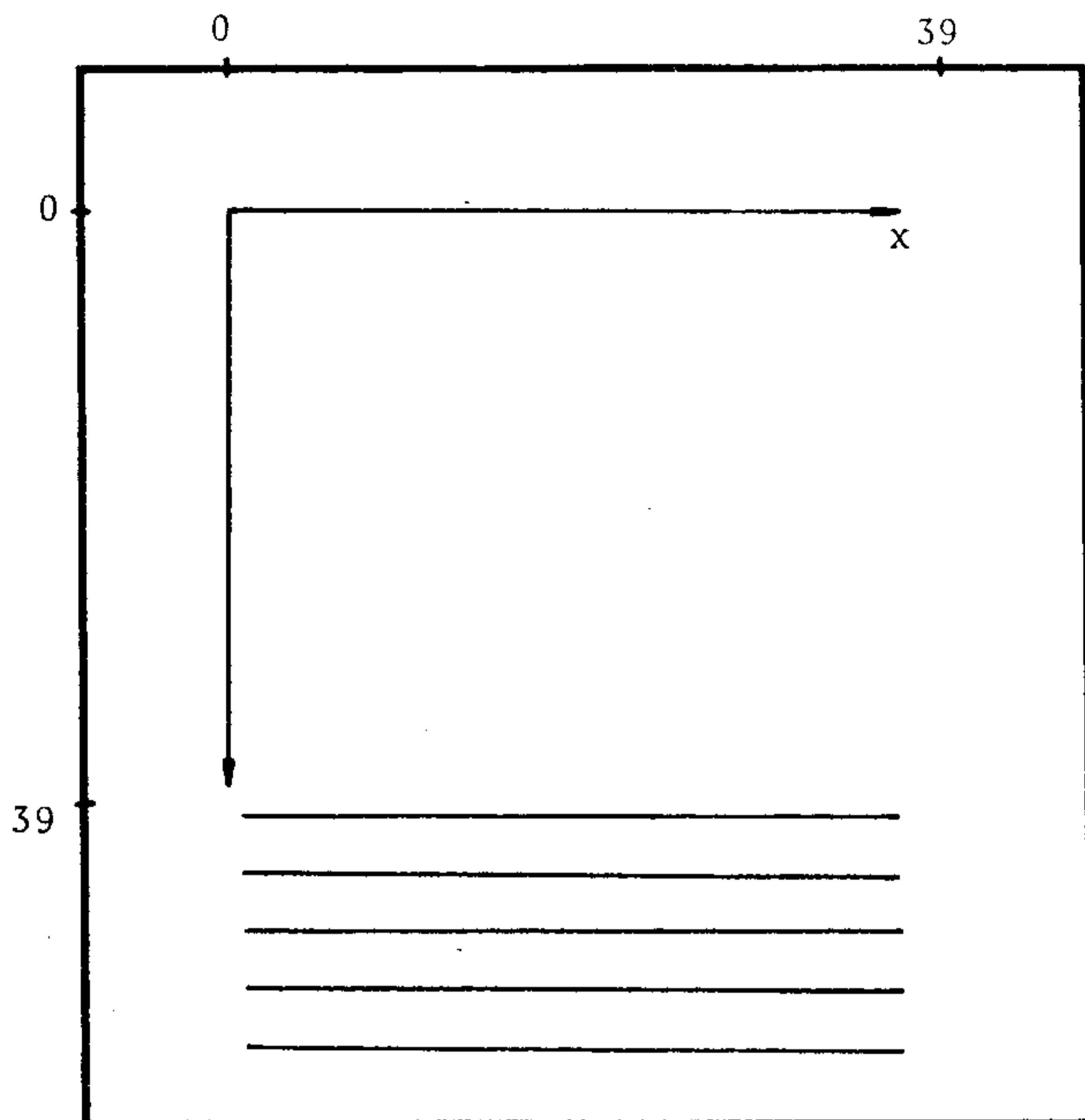
П р и м е р 1

200 GR

210 COLOR = 7

220 PLOT 20,22

В резултат на стартирането на програмата върху екрана се появява един малък правоъгълник.



Фиг. 23

Анализът на програмата показва следното: Чрез оператора `GR /графика/` се установява режим на груба графика. Вторият оператор, `COLOR /калър, цвят/` има предназначение да се даде цвета с който ще се чертае на екрана. При наличие на цветен екран за режим `GR` може да се чертае на 16 различни цвята. Конкретният цвят се определя от числото след оператора `COLOR` при използване на следните значения:

0 - черно	6 - тъмносиньо	12 - зелено
1 - мораво	7 - светлосиньо	13 - жълто
2 - тъмносиньо	8 - кафяво	14 - синьозелено
3 - пурпурно червено	9 - оранжево	15 - бяло
4 - тъмnozелено	10 - сиво	
5 - сиво	11 - розово	

Когато се чертае с черен цвят /0/ на екрана не се появява нищо. При черно-бял монитор при всички числа от 1 до 15 се получава образ, но най-добра е яркостта при числата 15, 7, 11, 13, 14 и 3.

Чрез оператора PLOT /плот, чертае/ от пример 1 се осъществява самото чертане и в този смисъл ред 220 означава: начертай малък правоъгълник в точка с координати X=20 и Y=22.

На последния ред от екрана може да се постигне листване на програмата при нанасяне на думата LIST и натискане на клавиша RTN.

Най-отдолу на екрана ще се появи програмата /по-точно на последните 2-3 реда от нея/. Както бе посочено по-горе при режим GR, текст може да има на последните 4 реда на екрана.

Появяването на цялата програма може да се постигне като най-напред се излезе от режим GR. Това се постига при нанасяне на командата TEXT. Чрез тази команда се осъществява изчистване на екрана от графиката и възвръща способността му върху него да се нанася текст.

За листването на програмата и нейното цялостно появяване върху екрана, в началото сега може без номер да се нанесе TEXT ; LIST и се натисне клавишът RTN.

Излизането от графичния режим може да се осъществи с помощта на клавиша за нулиране.

Изчистването на екрана и на паметта става по познатия ни начин /HOME и NEW/.

Начертаването върху екрана на правоъгълник може да стане със следната програма:

Пример 2

20 GR

21 COLOR = 11

22 PLOT 11,17

23 PLOT 13,25

С ред 210 се начертава малък правоъгълник в точка с координати $X=11$, $Y=17$ и в точка с координати $X=13$, $Y=25$. По посочения начин може да се чертаят малки правоъгълници в която и да е точка на екрана и в резултат на това да се създават различни фигури.

При този режим може да се чертае хоризонтална линия чрез оператора `HLIN` /хоризонтална линия/.

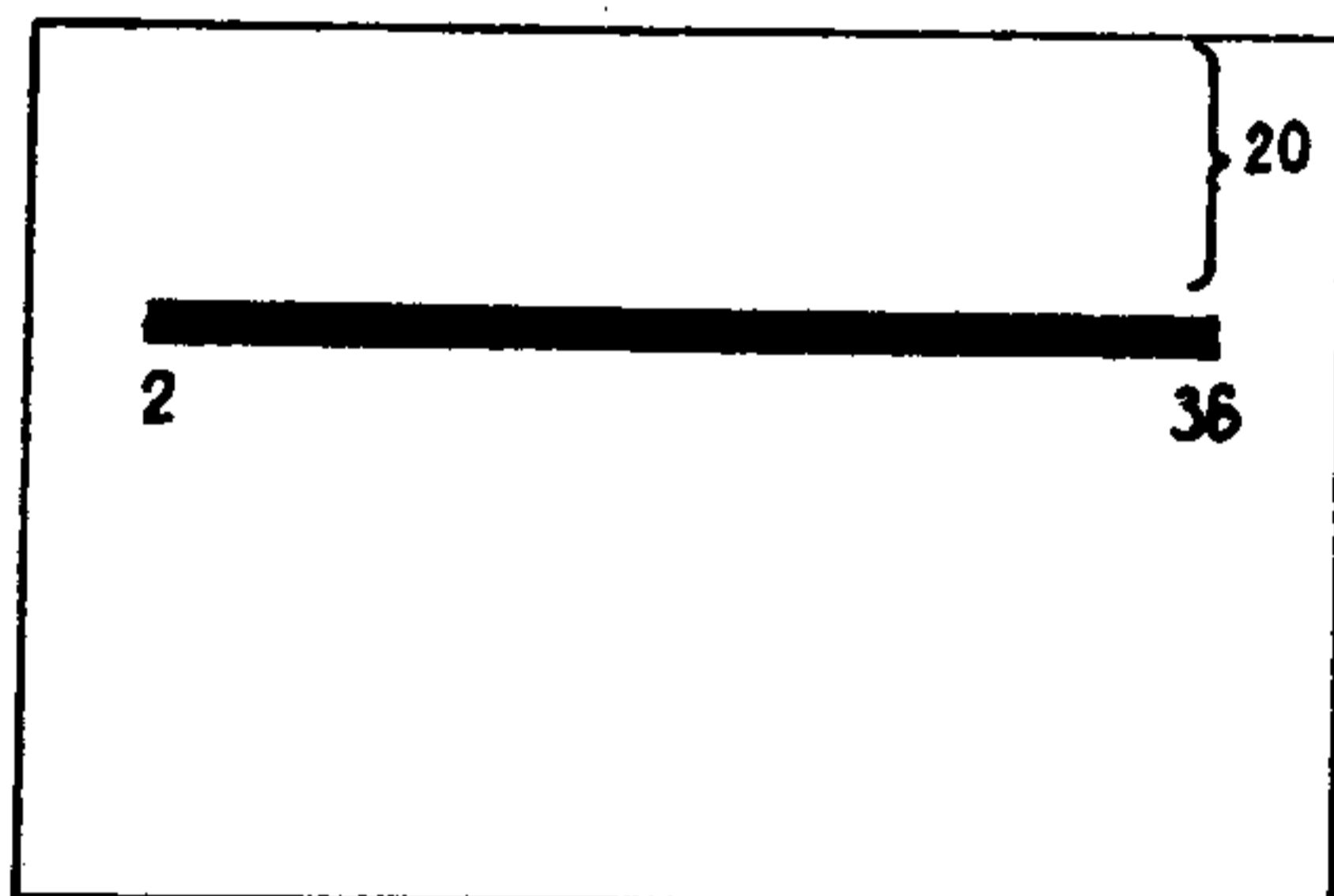
Пример 3

```
200 GR
```

```
210 COLOR = 11
```

```
220 HLIN 2,36 AT 20
```

С ред 220 се начертава хоризонтална линия от точка с абсциса $X=2$ до точка с абсциса $X=36$ на ред $Y=20$ /AT-ат, на/. При стартиране на програмата на екрана се появява схема, показана на фиг. 24.



Фиг. 24

1. Необходимо е да се има предвид, че стойностите на X и Y не могат да бъдат по-високи от 39 /т.е. между 0 и 39/.

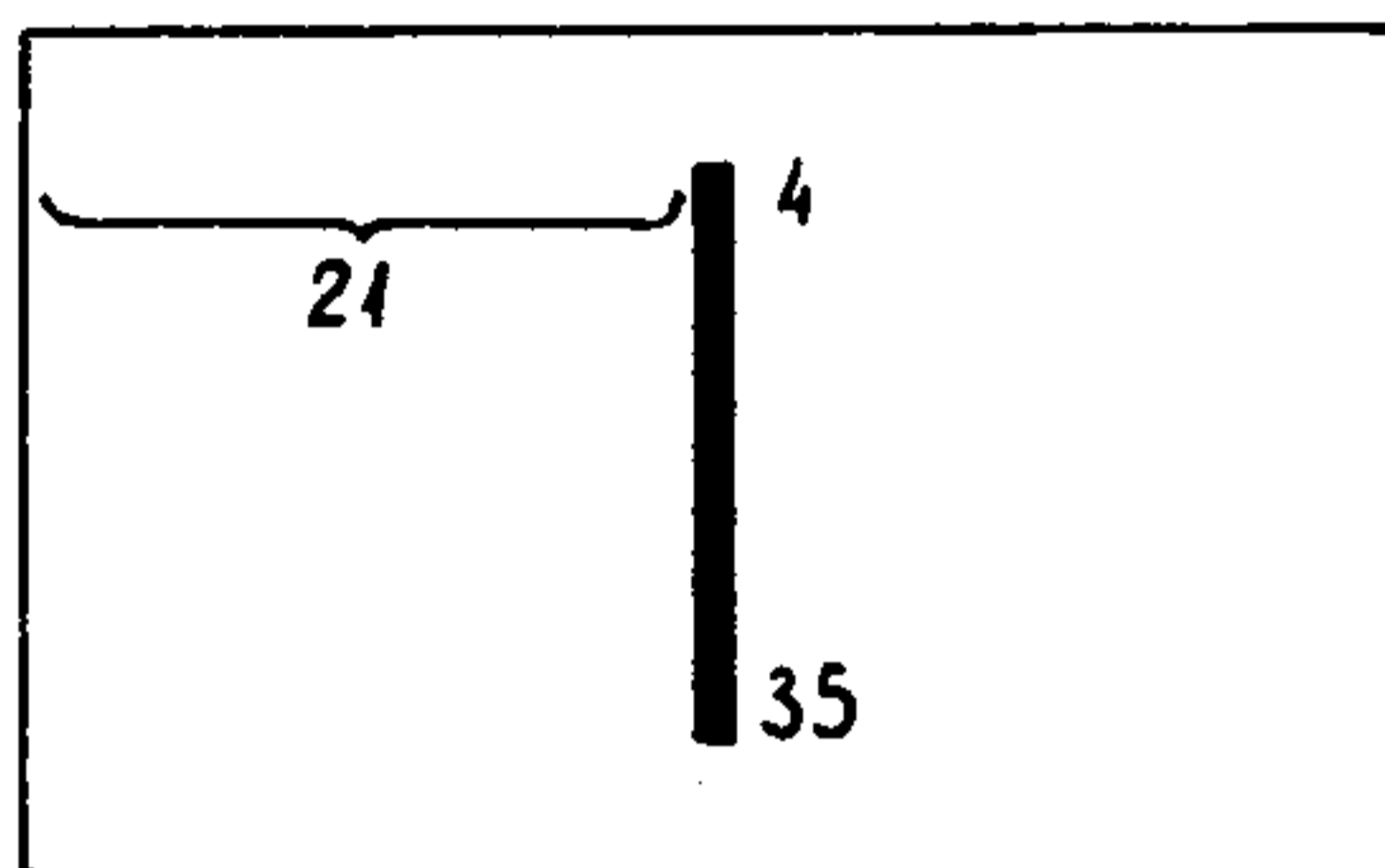
Начертаването на вертикална линия върху екрана може да се извърши с оператора `VLIN` /вертикална линия/.

Пример 4

```
200 GR:COLOR = 11
```

```
210 VLIN 4,35 AT 21
```

В ред 210 се осигурява начертаване на вертикална линия от точка с ордината $Y = 4$ до точка с ордината $Y = 35$ с абсциса $X = 21$. Върху екрана при стартиране на горната програма се получава линия, показана на фиг. 25.



Фиг. 25

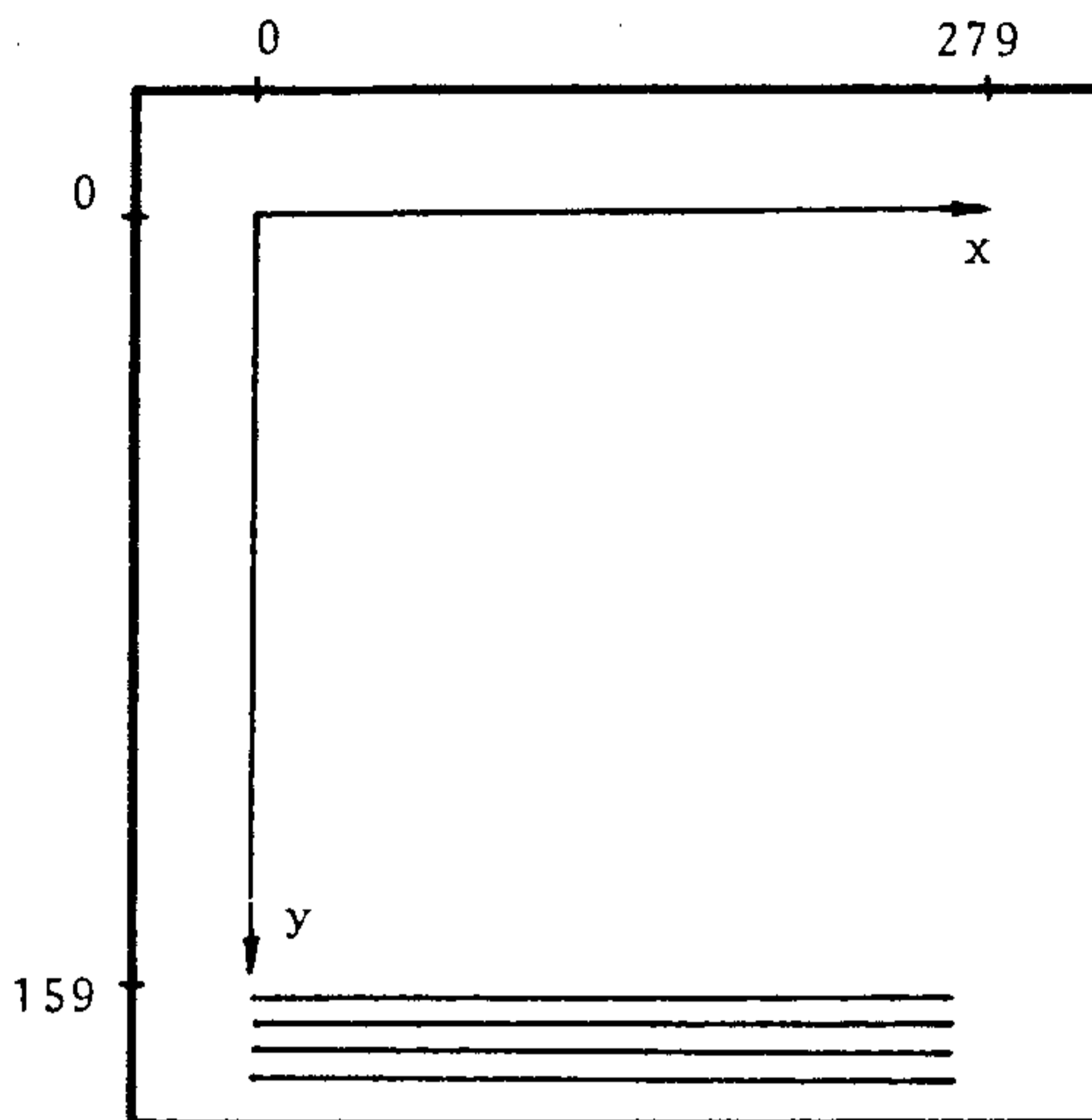
4.2. Режим на финна графика

Вторият режим се нарича графичен режим с висока разделителна способност. Този режим притежава по-големи възможности. Нанасянето на чертежи и фигури върху екрана се извършва с помощта на малки точки, като размерите на една точка са равни на сечението на електронния лъч на кинескопа.

Режимът с висока разделителна способност има две разновидности HGR и HGR2.

При първия режим /HGR - хай, график/ екранът може да се разглежда като координатна система /фиг. 26/, която по хоризонталата съ-

държа 280 точки /от 0 до 279/ и по вертикалата 160 точки /от 0 до 159/.



Фиг. 26

Вътре в тази координатна система могат да се чертаят най-различни фигури и схеми, но не може да се нанася текст. Текст може да се нанася само под координатната система в последните четири реда.

Пример 5. Да се начертае една хоризонтална черта по средата на екрана.

```
200 HGR
```

```
210 HCOLOR = 3
```

```
220 HPLOT 0,80 TO 279,80
```

За да се начертае линията, програмата трябва да се стартира /RUN и RTN/.

Чрез оператора HGR компютърът се въвежда в режим на фина графика. Операторът на ред 210 /HCOLOR хай-колър/ определя цвета на чертаене. Когато се прилагат черно-бели монитори за предпочитане е

след равенството да се нанесе цифрата 3. Когато обаче се прилагат цветни монитори могат да се използват цифрите от нула до 7 чрез които се определят следните цветове:

0 - черно I

1 - зелено

2 - синьо

3 - бяло I

4 - черно II

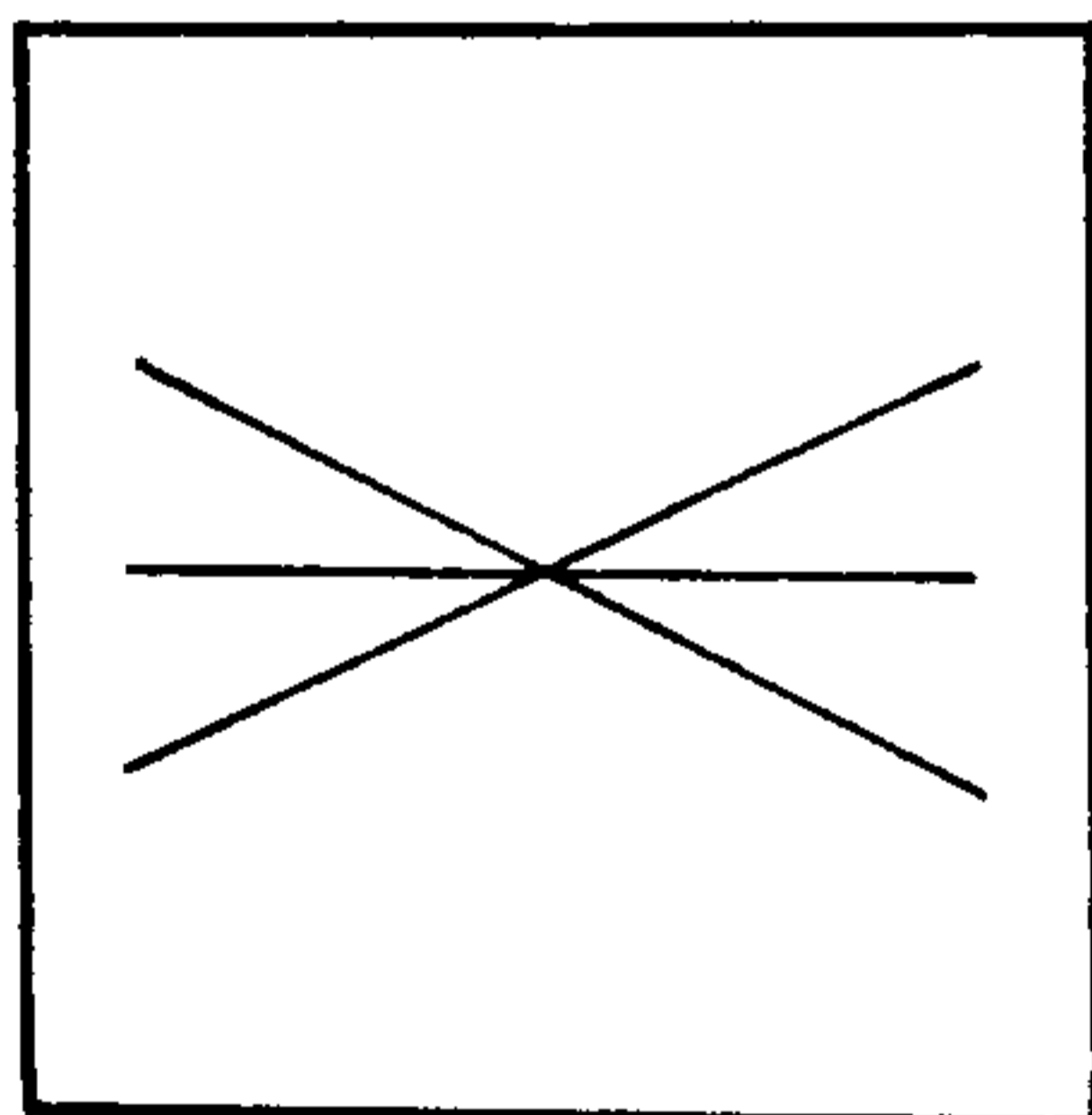
5 - зависи от монитора

6 - зависи от монитора

7 - бяло II

По-нататък чрез оператора HPLOT се извършва самото чертаене и в ред 220 значението на цифрите е следното: да се начертае линия през 2 точки, едната с координати $X=0$, $Y=80$, а другата с координати $X=279$, $Y=80$, т.е. линията е по средата на екрана от началото до края. Ако желаем да начертаем една линия в диагонал, операторът в ред 220 ще има следното съдържание: HPLOT 0,0 TO 279,159 и за следващия диагонал програмата може да се продължи като след 159 се нанася TO 0, 159 TO 279,0.

При стартиране на програмата заедно с пример 5 ще се получи следната схема /вж. фиг. 27/.



Фиг. 27

Действително в този си вид схемата няма никакво значение, тъй като с това посочваме само част от възможностите на ПМК при работата в режим на финна графика.

Крайните точки на линиите, които се чертаят могат да се задават отдолу нагоре, отгоре надолу, отляво надясно или отдясно наляво.

За да се листва дадена програма, когато на екрана има дадена фигура трябва предварително да се излезе от режим HGR. Това се постига чрез нанасяне на командите TEXT:LIST и нанасяне на клавиша RTN, независимо, че на екрана не се вижда скобата и маркерът.

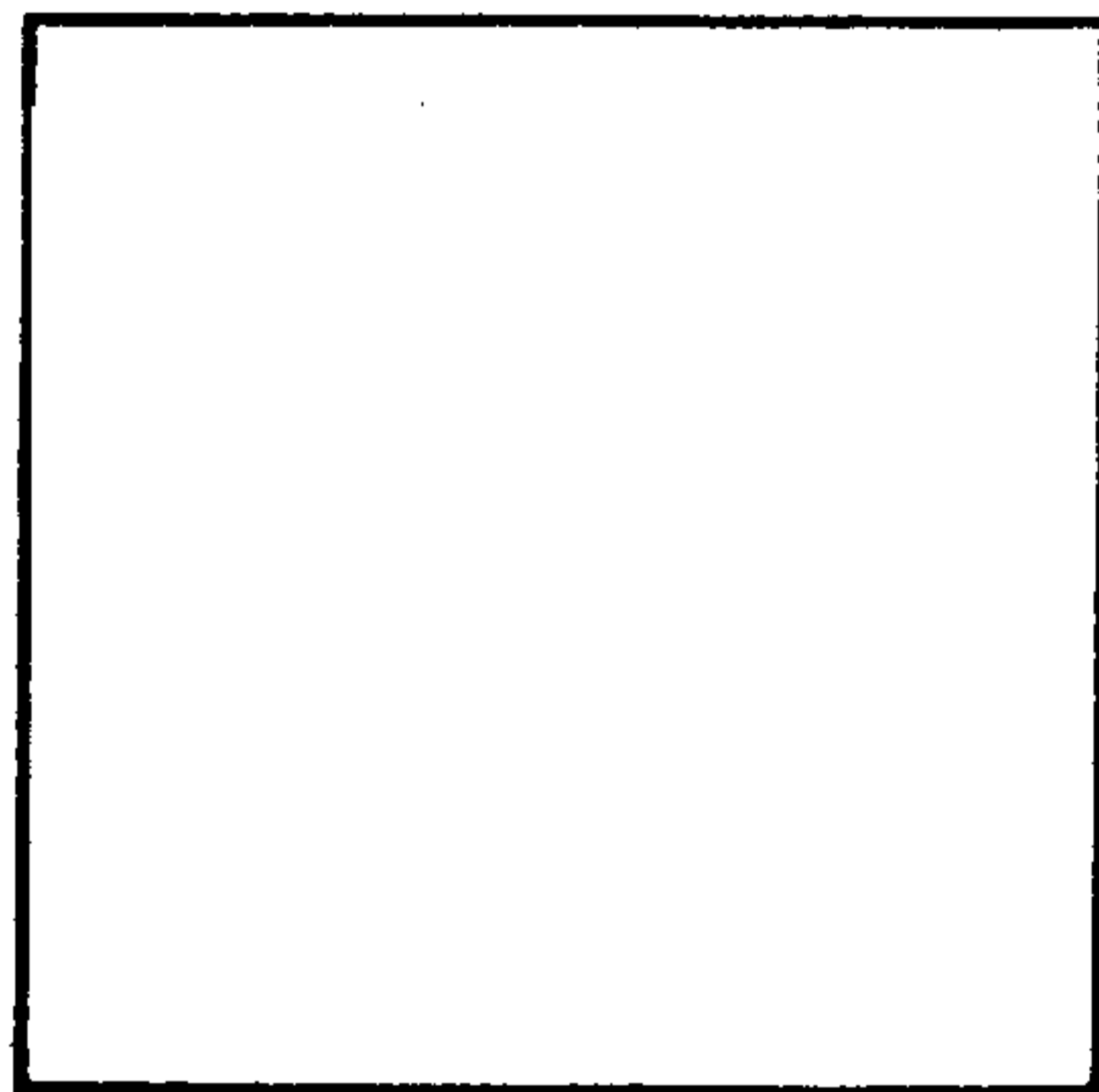
Пример 6. Да се състави програма за начертаване на четириъгълник.

```
200 HGR
```

```
210 HCOLOR = 3
```

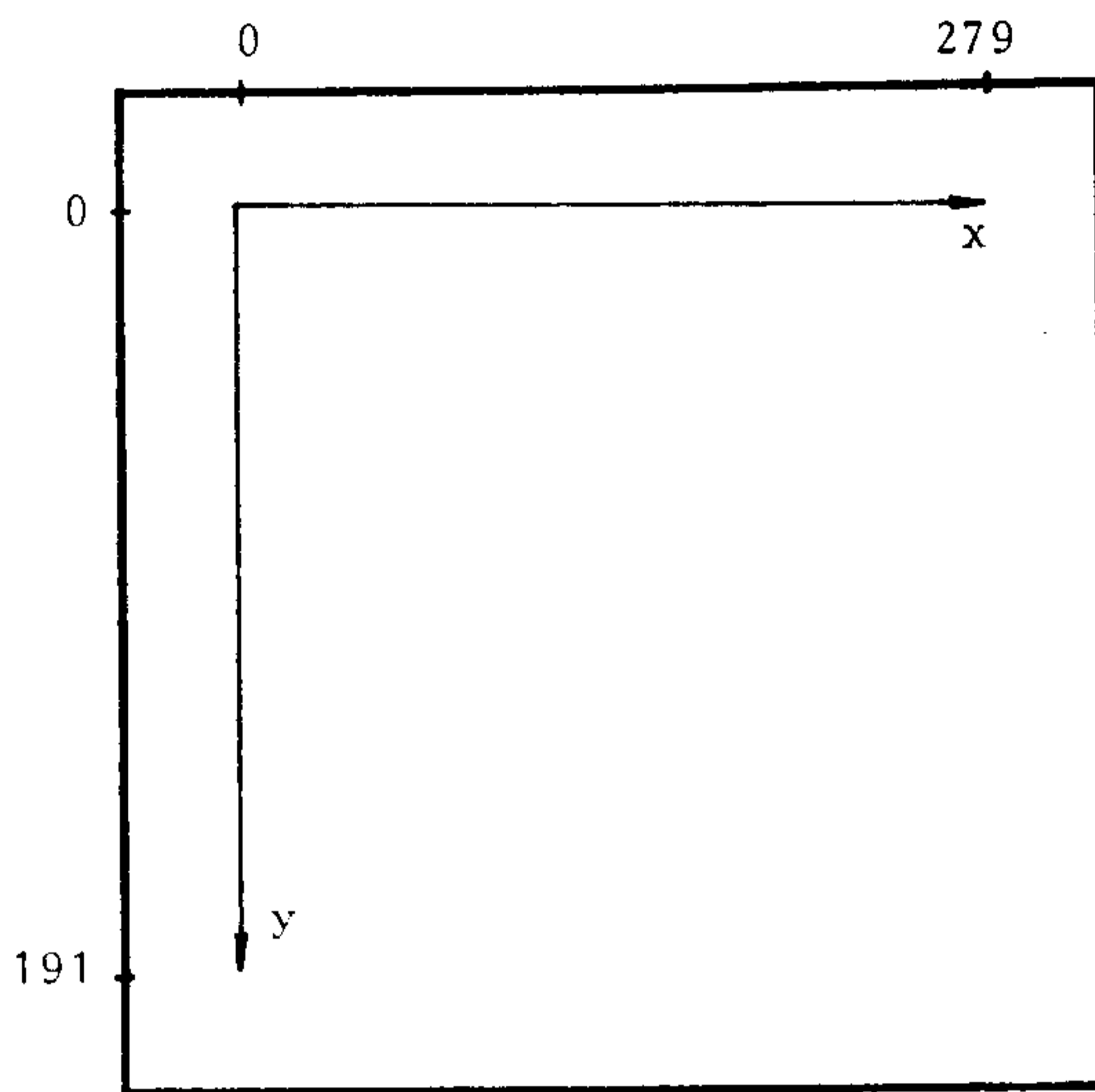
```
220 HPLOT 69,09 TO 209, 09 TO 209,149 TO 69,149 TO 69,09
```

При стартиране на екрана се появява фигура, която е изобразена на фиг. 28.



Фиг. 28

Втората разновидност за работа в режим на финна графика е HGR2. При тази разновидност екранът трябва да се разглежда, като координатна система съдържаща по хоризонтала също както първата разновидност - 280 точки /от 0 до 279/, но по вертикалата - точките са 192 /от 0 до 191/, фиг. 29.



Фиг. 29

Характерно за този режим, е че най-отдолу на екрана липсват 4-те реда за нанасяне на текст.

Излизането от графичен режим освен по посочения по-горе начин може да се осъществи и чрез натискане на червения клавиш /нулиране/.

Всички други видове работи, които могат да се извършват при първата разновидност са валидни и тук.

Пример 7. Да се състави програма за форма на първичен документ, който да има правоъгълна форма, таблична антетка с три графи:

```
200 HGR2 : HCOLOR = 3
```

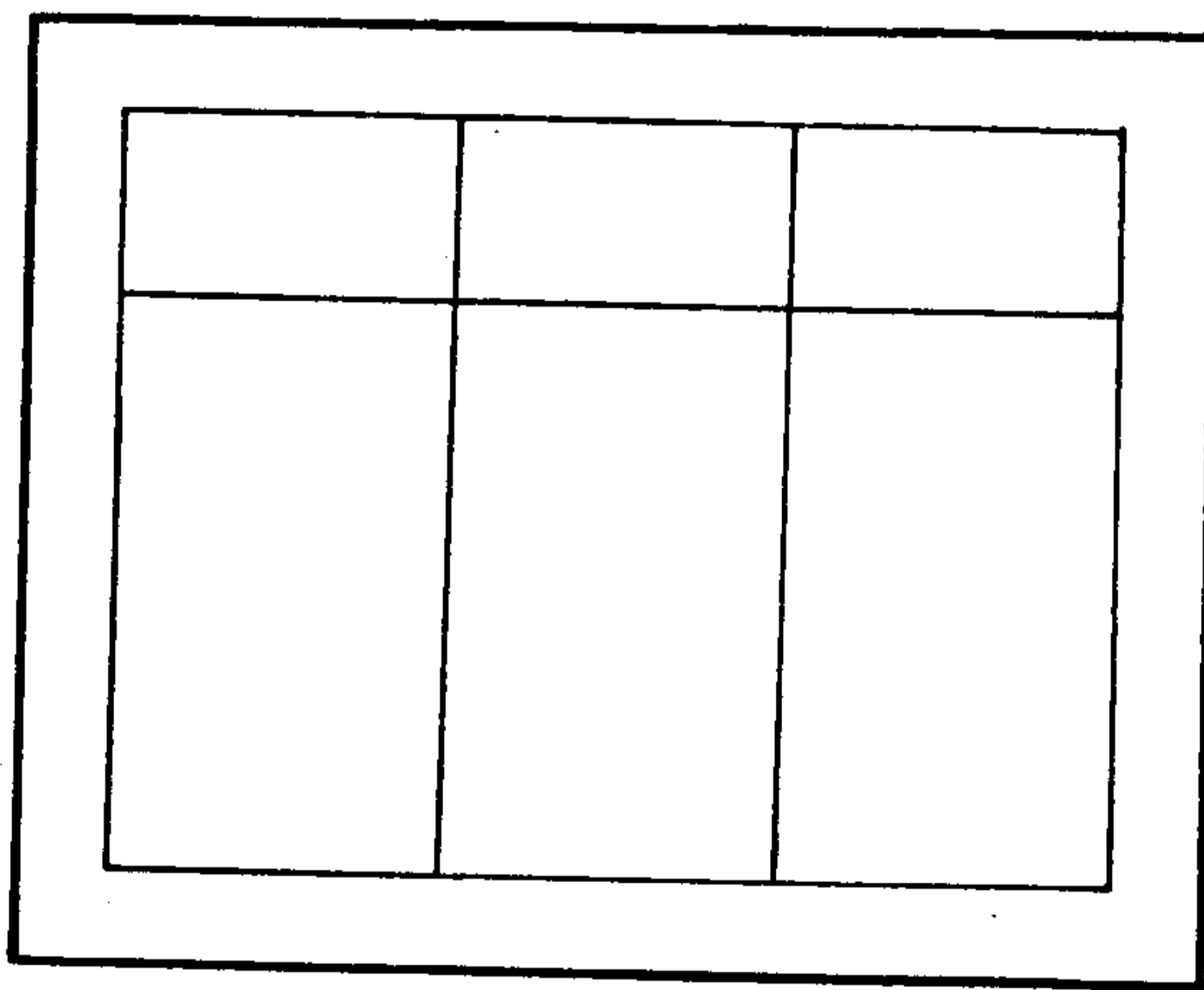
```
210 HPLOT 20,10 TO 270,10 TO 270,191 TO 20,191 TO 20,10
```

```
220 HPLOT 20,50 TO 270,50
```

```
230 HPLOT 100,10 TO 100,191
```

```
240 HPLOT 180,10 TO 180,191
```

Със стартиране на програмата върху екрана се появява формата на първичен документ /вж. фиг. 30/.



Фиг. 30

С ред 210 се начертава външната част на документа, с ред 220 - табличната част, с ред 230 - разграничаването на първата графа

и с ред 240 - разграничаването на втората и третата графа.

Контролни въпроси и задачи

1. В какви видове графични режими може да работи ПМК ПРАВЕЦ-82?
2. Какво е характерно за режима на груба графика?
3. Какви са размерите на координатната система на грубата графика?
4. Може ли вътре в полето на координатната система да се пише текст?
5. Къде на екрана може да се пише текст?
6. Какво е значението на оператора GR?
7. Какво е значението на оператора COLOR?
8. Колко цветове може да има цветен монитор при груба графика?
9. Какво е предназначението на оператора PLOT?
10. Какво е предназначението на операторите HLIN и VLIN?
11. За да се листва програмата как се излиза от режим GR?
12. Съставете програма при режим GR за чертане на триъгълник.

13. Съставете програма за начертване на правоъгълна рамка в максималните размери на екрана при режим на груба графика.

14. Какви са размерите на координатната система при режим на финна графика? Какви са двете разновидности?

15. Как могат да се задават крайните точки при режима на финна графика?

16. Може ли при режим HGR да се пише текст и къде?

17. Начертайте триъгълник при режим HGR.

18. Какви са разликите между режимите HGR и HGR2?

19. Начертайте първичен документ, който да има на екрана следната форма:

20. Начертайте първичен документ със следната форма:

21. Начертайте първичен документ със следната форма:

22. По какъв начин се излиза от режим HGR2 за листване на програмата?

5. ЗАПАМЕТЯВАЩИ УСТРОЙСТВА

В началото при разглеждането на техникоексплоатационните възможности на ПМК бе отбелязано, че съхраняването на програми и данни за един по-продължителен период от време може да се извърши при използване на магнитолентови и флопидискови запаметяващи устройства. Известно е, че въведената в оперативната памет информация /програми и данни/ след изключването на ПМК се изтрива. Естествено нейното запазване може да стане при нанасяне върху външна памет. Използуването на външна памет дава възможност не само да се пренася дадена информация от оперативната памет, но и обратно, при необходимост определена програма или данни да се пренасят от външните запаметяващи устройства в оперативната памет или върху екрана, или пре-записване на информацията от един носител върху друг.

5.1. Работа с магнитолентови запаметяващи устройства

Персоналният микрокомпютър ПРАВЕЦ-82 може да се свърже с касетофон "Унитра" /фиг.6/.

Нанасянето на информацията от ПМК в касетофона се извършва със скорост 1500 бита в секунда. В тези касетофони се използват обикновени касети с времетраене $3600 \text{ s} / 2 \times 30 = 60 \text{ min}$ /. При това положение върху една касета теоретически може да се нанесе информация с обем от $660 \text{ Кбайта} / 3600 \text{ s} \times 1500 \text{ бита} = 5\,400\,000 \text{ бита} = 675\,000 \text{ байта} = 600 \text{ Кбайта}$ /. .

Практически обаче между отделните задачи /между отделните програми и масиви от данни, между отделните редове/ има празни пространства. Освен това, за да се разграничава една програма от друга програма може в началото да има звуков, за да нанася наименованието на програмата. Всичко това значително намалява площта на използване на магнитната лента. Например една касета може да осигури съхраняването на около 200 Кбайта информация или в една касета могат да се нанесат 5-10 програми /това зависи от дължината на програмите/.

Запис на програми и данни от ПМК върху магнитна лента

Понастоящем все още сравнително широко приложение при ПМК ПРАВЕЦ-82, като външно запаметяващо устройство намира касетофонът.

Записването на информация /програми и данни/ от ПМК върху касетите става по следния начин:

1. На касетофона се поставя касета, като лентата се поставя в начално положение.

2. Нанася се програмата върху екрана /ако програмата е голяма тя постъпва и в оперативната памет/ и на края при наличие на скоба и маркер се записва без номер командата `SAVE` /сейф, запазвам/.

3. Касетофонът, вж. фиг. 6, се включва на запис при натискане на бутони 10 и 13.

4. Натиска се клавишът RTN на ПМК и в този момент от екрана изчезва маркерът. След изтичане на няколко секунди ПМК издава кратък звук, което е начало на записа.

5. Записването на информацията продължава от 15-20 s до 1-2 min /това зависи от дължината на програмата/. След свършване на записа ПМК отново издава звук като маркерът и скобата отново се появяват на екрана. След появата на скобата и маркера, което е признак за край на записа, касетофонът чрез бутон 12 може да бъде изключен.

Запис на програми и данни от касетофон в ПМК

Когато дадена програма е записана върху касета често се налага нейното въвеждане в оперативната памет и последващо изпълнение. Пренасянето на информация от касетофона в оперативната памет се осъществява по следния начин:

1. Касетата се поставя на касетофона като предварително се определя мястото на необходимата за прехвърляне в компютъра програма. Регулаторите на касетофона /бутони 14 и 15 на фиг. 6/ се поставят в средно положение, т.е. след като се включи ПМК и на горния ляв ъгъл на екрана се появи скоба и маркер без номер, след което се нанася командата LOAD /лоуд, зареждам/.

3. Касетофонът се включва на възпроизвеждане при натискане на бутон 10.

4. Натиска се клавишът RTN, при което маркерът изчезва от екрана. След няколко секунди се издава звук, което показва, че ПМК е започнал пренасянето на информацията от касетата в оперативната памет.

5. И тук записът продължава /в зависимост от дължината на програмата/ от 15 - 20 секунди до 1-2 минути. След завършване на записа ПМК отново издава кратък звук и на екрана се появяват скобата и маркерът. По-нататък касетофонът чрез бутон 12 може да се изключи.

В случаите, когато на екрана на ПМК се появи съобщение за грешка записването на програмата трябва да се повтори.

Контролни въпроси и задачи

1. Какви запамятаващи устройства могат да се включат към ПМК?
2. Какво е предназначението на външните запамятаващи устройства?
3. С каква скорост се записва информацията върху касета?
4. Каква команда се използва за запис на информация от ПМК върху касета?
5. След натискането на кой клавиш на ПМК мигачият маркер изчезва?
6. В кой момент ПМК издава кратък звук, което е указание, че записът е започнал?
7. Коя е командата за запис на информация от касети в оперативната памет на ПМК?
8. Нанася ли се номер пред командите SAVE и LOAD.
9. Нанесете програмата върху екрана и запишете същата върху касета:

11Ø DATA 6, 12, 7, 29, 26, 19, 95, 4, 3

12Ø READ A, B

13Ø LETD = A Ж B

14Ø READ C


```
15Ø LET E = C43
16Ø PRINT A, B, C
17Ø PRINT D, E
18Ø GOTO 120
19Ø END
```

10. Пренесете горната програма от касетата на ПМК.

5.2. Работа с магнитнодискови запаметяващи устройства

5.2.1. Общи сведения

Към ПМК ПРАВЕЦ-82 може да се свържат на работа едно или две флопидискови устройства. В редица предприятия и организации се използват флопидискови устройства от други персонални микрокомпютри /например тези от ПМК APPLE II и др./

Приложението на флопидискови устройства позволява да се използват различни дискети /едностранни и двустранни и такива с различна плътност на записа/. Работата с дискови устройства изисква запознаване на потребителя ѝ с командите на дисковата операционна система и техникоексплоатационните възможности на дискетата.

Използуването на система от команди създава условия потребителят да осъществява диалог с ПМК като чете или записва информация от дискетата. Нанасянето на информацията върху дискетата се извършва чрез организиране на тази информация в масиви и на всеки масив се дава име, а по-нататък потребителят се обръща към този масив чрез това име.

Управлението на масивите се извършва чрез специална програма наречена дискова операционна система. За работа с ДОС се използ-

зуват някои команди, с което се разширяват възможностите на езика БЕЙСИК. Естествено за работа с ДОС-команди ПРАВЕЦ-82 трябва да е в режим на работа с БЕЙСИК.

Имената на масивите могат да бъдат с дължина от 1 до 30 символа, като всяко име трябва да започва задължително с буква /латиница/. В името могат да бъдат включени различни символи, използвани от клавиатурата и нанесени върху клавиатурата /с изключение на запетайка/. Ако името е с дължина повече от 30 символа, автоматично се намаляват до 30.

Сравнително често в практиката се използват едностранни дискети /ИЗОТ-5253Е/. Всяка дискета има по 35 писти и всяка писта е разделена на 16 части, наречени сектори /номерирани от 0 до 15/. На един сектор може да се нанесе 256 байта информация.

От всички 35 писти за потребителя са достъпни 31 писти.

Писти 0, 1, 2 и 11-та са за служебна информация, и те са недостъпни за потребителя.

Писти 0, 1 и 2 съдържат дисковата операционна система.

Информация за съдържанието на дискетата се съдържа в 11-та писта /отнася се за инициализирани дискети/. В тези писти ДОС записва за всеки масив информация за това какъв е типа на масива, колко сектора заема същия, името му и къде е записан. При подаване на ДОС-командата CATALOG част от тази информация се показва на екрана. Като цяло всяка дискета има по 496 сектора като в отделния сектор може да се съдържа 256 байта информация или 124 Кбайта / $496 \times 256 = 126976$ байта, т.е. 124 Кбайта - $126976 : 1024$ бита = 124 Кбайта/. Върху една писта може да се нанесе информация от 4096 байта / $16 \times 256 = 4096$ байта/.

При работа с ПМК свързан с магнитнодисково /флопидисково/

устройство трябва да се спазват някои условия, свързани със започване на работа, по време на работа и при завършване на работа. Тези условия трябва да се спазват както при работа с ПМК ПРАВЕЦ-82, така и с ИМКО-2 и ЕИПЪЛ.

При започване на работа с ПМК трябва да се спазват следните условия:

1. Да се включи мониторът /да се натисне бутон 3 на фиг. 6/.
2. Да се постави системната дискета /SISM/ или която и да е друга инициализирана дискета в дисковото устройство /при две устройства, в устройство №1/.
3. Да се включи превключвателят на изчислителният блок /този ключ се намира от задната лява страна на блока/.

По време на работа на флопидисковото устройство /когато свети червената контролна лампа, намираща се от предната лява страна/ се забранява смяна на дискета или отваряне предпазната ключалка на устройството.

При свършване на работа е необходимо:

1. Изваждане от устройството на дискетата.
2. Изключване на превключвателя на изчислителния блок.
3. Изключване на монитора.

Основните процедури, които се извършват с флопидисковото устройство при работа с ПМК са инициализация на нова дискета, копиране, записване, четене и др.

5.2.2. Инициализация на дискети

За да се използва нова дискета, същата трябва да бъде инициализирана. Инициализацията на новите дискети изисква най-напред да се осъществи стартиране на ДОС. За целта е необходимо:

а/ да се включи монитора;

б/ да се постави системната дискета /SIS-MASTER/ в дисковото устройство /при две дискови устройства, в устройство №1/.

След извършването на горните процедури в оперативната памет на ПМК се записва необходимата за работа от системната част ДОС. На дисковото устройство светва червената контролна лампа, което показва, че същата работи. След завършване на прехвърлянето на тази част от ДОС, тя стартира и на екрана се появява съобщение, което има следния вид:

П Р А В Е И - 82

DOS VERSION 3,3 SYSTEM MASTER

JANUARY 1,1983

Всичко това показва, че могат да бъдат използвани командите на ДОС.

За да се извършат всички операции, свързани със записване на информация върху нови дискети последните трябва да бъдат инициализирани. За тази цел се използва системна програма, при стартирането на която, се извършва необходимата инициализация.

Инициализацията на която и да е нова дискета, се извършва в следната последователност:

1. Включване на монитора.

2. Поставяне дисковото устройство на SIS-MASTER и върху

екрана се появява посоченото по-горе съобщение /т. а и б/.

3. Изваждане от дисковото устройство на системната дискета.

4. Поставяне в дисковото устройство на новата /неинициализираната/дискета.

5. Написване на кратка поздравяваща програма на БЕЙСИК /например: 10 PRINT "HELLO" /- името се нанася в кавички .

6. Проверява се верността на написаната програма чрез нейното стартиране:

7. Нанася се ДОС - команда `INIT` и името на програмата /в нашия случай `HELLO`/ и се натиска клавишът `RTN`.

В резултат на всички тези процедури новата дискета е инициализирана и върху нея може да се нанасят програми и данни. Едновременно с това върху новата дискета е записана поздравяващата програма като масив със съответното име `HELLO`.

Ако по-нататък новата дискета се извади от дисковото устройство и отново се постави в него /или без да се изважда, след инициализацията/ и се нанесе ДОС командата `CATALOG` и `RTN`, то на монитора ще се появи текста на тази програма /на програмата със име `HELLO`/. По този начин може чрез името да се извика всяка програма, записана върху дискетата.

По нататък, когато се използват понятията стартиране на ДОС трябва да се разбира поставяне на системната дискета `SIS - MASTER` /или за краткост `SISM`/ във дисковото устройство и появата върху монитора на скобата и маркера.

5.2.3. Записване на масиви върху дискета

Съставените и нанесени чрез клавиатурата на ПМК програми могат да бъдат записани върху дискета. В зависимост от големината дадена програма може да се нанесе върху монитора или върху монитора и оперативната памет. Това разбира се няма никакво значение при извършване на посочените по-долу процедури, необходими за запис на масиви върху дискетата.

За да се въведе даден масив върху дискетата, последната трябва да бъде инициализирана по посочения по-горе начин.

Необходимо е пак да се отбележи, че преди нанасянето в паметта на всяка нова програма трябва предварително да се нанесат командите HOME и NEW за изчистване на оперативната памет и екрана.

Нанасянето на която и да е програма или масив върху дискетата трябва да се извършва със съответно име. Самото записване на програмата върху дискета се извършва с помощта на DOS-командата SAVE.

Записването на програми и масиви върху дискетата може да се извърши при спазване на следната последователност:

1. Включване на монитора.
2. Да се постави дисковото устройство /№1/, новата инициализирана дискета.
3. Да се включи изчислителния блок.
4. Да се нанесе програмата, която ще се прехвърли върху дискетата. Нека за случая програмата да има следния вид:

```
1Ø A=2Ø  
2Ø B=1Ø  
3Ø C=A+B  
4Ø PRINT C  
5Ø END
```


Тази програма трябва да бъде проверена чрез стартиране и откритите грешки коригирани.

5. Да се нанесе ДОС-командата `SAVE ADDITION /име на програмата/` и `RTN`.

По този начин нанесената върху екрана програма е записана на дискетата като масив с име `ADDITION`.

Възможно е след всичко посочено до тук да се провери дали действително горната програма е записана върху дискета. Това се постига чрез нанасяне върху екрана на ДОС-командата `CATALOG` и `RTN`. След тази команда на екрана се появява името на програмата. А чрез командата `RUN ADDITION` и `RTN` се появява текстът на програмата.

5.2.4. Четене на масиви от дискета

Твърде често в практиката при работа с дискетни устройства се налага програмите и масивите да се четат от дискетата и да се въвеждат в оперативната памет на ПМК.

Четенето на програми и масиви от дискета и прехвърлянето им в оперативната памет се извършва в следната последователност:

1. Да се постави дискетата, от която ще се чете програмата или масива, в дисковото устройство;

2. Нанася се командата `CATALOG` и на екрана се появяват имената на всички програми и масиви, които са съхранени в дискетата;

3. Нанася се ДОС-командата `LOAD` и име на програмата, която трябва да бъде стартирана, т.е. програмата, която ще използва потребителят и `RTN`.

В каталога се съдържат всички имена на програмите или масивите. В дадена дискета може да се съдържа една или повече програми.

Ако от потребителя се знаят точният набор и особено най-

менованието на програмите, ДОС-командата CATALOG може да не се нанася, а от т.1 да се премине към т.3.

След командата LOAD програмата се прехвърля от дискетата върху оперативната памет. Изпълнението на програмата може да се осъществи чрез командата RUN.

Съдържанието на прочетената от дискетата и въведена в оперативната памет програма може да се провери чрез командата LIST.

Изпълнението на една програма, нанесена върху дискета, без да се използва командата LOAD за четене и зареждане на оперативната памет може да се използва командата RUN име на програмата /в нашия случай ADDITION и RTN/. По този начин програмата се чете от дискетата и се изпълнява. За дадената по-горе програма върху екрана ще се появи резултатът от събирането на А и В, а именно 30.

Когато се търси даден масив, който не съществува /не е записан/ върху дискетата след ДОС-командата LOAD - име на масива и RTN, на монитора се появява съобщение FILE NOT FOUND.

До този момент в дискетата има две програми: поздравляваща /HELLO/ и програма за събиране /ADDITION/.

При нанасяне на ДОС-командата CATALOG и RTN, на монитора се появява следният текст:

A 002 HELLO

A 002 ADDITION

Символите на екрана имат следния смисъл:

Буквата А означава, че програмата е написана на БЕЙСИК.

Цифрите 002 показват колко сектора от дискетата са необходими, за да бъдат записани програмите. Най-малкият масив може да се нанесе върху 002 сектора. Както бе посочено по-горе върху една дискета могат да бъдат използвани 496 сектора.

Например програмата HELLO е записана върху 002 сектора.

Върху екрана се появяват имената само на 18 масива, съдържащи се в дискетата. Ако този брой е повече и трябва те да се появят на екрана, то е необходимо да се натисне даден бутон от клавиатурата.

Когато името на дадена програма или масив трябва да се измени, за целта се използва ДОС- командата `RENAME` /преименование/.

За промяна на името на програмата `ADDITION` в `ADD` се нанася командата:

<code>RENAME</code>	<code>ADDITION,</code> старо име	<code>ADD</code> ново име
---------------------	-------------------------------------	------------------------------

След стартиране се извършва промяна на името. За да се провери дали действително е направена промяна може отново да се нанесе ДОС-командата `CATALOG` и `RTN`.

В случаите, когато една или повече програми трябва да бъдат изтрети от дискетата може да се използва ДОС-командата `DELETE`.

За изтриване на нашата втора програма `ADD` с клавиатурата върху екрана се нанася:

`DELETE ADD и RTN`

С тази ДОС-команда се изтрива програмата `ADD`. Ако сега се нанесе ДОС-командата `CATALOG` и `RTN` ще се види, че на екрана се появява само поздравляващата програма `HELLO` т.е. програмата `ADD` е изтрита.

Както се вижда твърде често се използват понятията ДОС-команда и номер на дисковото устройство. Всички ДОС-команди позволяват обръщане към конкретно дисково устройство. Към един ПМК могат да се свържат по две дискови устройства /ДУ 1 и ДУ 2/. Дисковото устройство се управлява от дисков контролер, който се свързва с ПМК към даден СЛОТ. Ако след включване на ПМК ПРАВЕЦ-82 не се укаже, че чрез ДОС-команда, ще се работи с ДУ 2, то се подразбира обръщане към ДУ 1. Когато към ПРАВЕЦ-82 са включени две дискови устройства и ще се ра-

боти с ДУ 2, към ДОС-команда трябва да се добави ДУ 2. Например ако дадена дискета е поставена в ДУ 2 и искаме да получим имената на масивите ДОС-командата е следната - CATALOG D2 и RTN.

Когато след обръщане към ДУ 2 не се окаже в следващата ДОС-команда необходимост от работа с ДУ 1, то се подразбира адресация на ДУ 2.

Следователно, ако трябва да се работи едновременно и с двете дискови устройства, то с ДОС-командите трябва да се укаже с кое устройство ще се работи.

Например, ако до този момент потребителят е работил с информация от дискета, поставена на ДУ 1, а след това трябва да се чете и работи с масив от дискета на ДУ 2, командата има следния вид: LOAD DIV D2 и RTN.

С тази команда програмата DIV от ДУ 2 се зарежда в оперативната памет.

В ПМК ПРАВЕЦ-82 дисковият контролер е свързан към СЛОТ 6.

Стартирането на ДОС винаги се извършва от ДУ 1 /разбира се, когато устройствата са две/.

5.2.5. Копиране на масиви, програми и дискети

Копирането е една от най-често употребяваните процедури при работа с ПМК. Процедурата е необходима за постигане сигурност при работа с оригинални дискети /програми и масиви/, а също и при размножаване на определени програми или масиви. Една от основните задачи при започване на работа с ДОС е да се усвои и тази процедура, за да може да се копират всички оригинални дискети. Разбира се, веднага трябва да се отбележи, че има случаи, когато дадени програми, масиви или цялата дискета са защитени, което не позволява тяхното копиране.

Чрез копирането се размножават програми на икономически задачи, делови игри и др. и се създават повече дискети, които се използват от преподавателите.

Необходимо е да се прави разлика между копиране на програми и масиви, от една страна, и копиране на дискети - от друга. Това са две различни процедури.

По долу ще бъдат разглеждани тези две процедури, като се има предвид обстоятелството, че се използва само едно дисково устройство.

КОПИРАНЕ НА ПРОГРАМИ

За копиране на програми и масиви не се използват специални команди. Прехвърлянето на програма и масив от една дискета върху друга може да се извършва с помощта на DOS-командите SAVE и LOAD. Последователността на това копиране е следната:

1. Стартиране на DOS.

2. Поставяне на оригиналната дискета /дискетата, от която ще се извършва прекопирването/. Нека тази дискета бъде означена като А.

3. Нанасяне на командата LOAD DIV и RTN /приема се, че трябва да се копира програма DIV от дискета А/.

Когато ще се използват две ДУ, в ДУ 1 се поставя оригиналната дискета, а в ДУ 2 - новата дискета, върху която ще се извърши копирането, т.е. дискета В.

Разбира се при две ДУ DOS-командите ще бъдат:

```
LOAD DIV, D1 и RTN
```

```
SAVE DIV, D2 и RTN
```

По този начин програмата DIV се пренася чрез ДУ 2 върху дискета В.

4. Изваждане на дискета А от дисковото устройство /при работа с едно ДУ/ и поставяне в ДУ на дискета В.

5. Нанасяне на ДОС-командите SAVE DIV и RTN.

За проверка на резултата от горните процедури може да се нанесат ДОС-командите CATALOG и RTN.

КОПИРАНЕ НА ДИСКЕТИ

Под копиране на дискети в случая се разбира пренасянето на всички програми и масиви от дискета А върху дискета В.

За копиране на дискети може да се използват две програми, намиращи се в системната дискета SISM. С помощта на една от програмите се извършва копиране на цялото съдържание на дискетата. Съществуващата информация на дискетата, върху която ще се извършва копирането, се изтрива.

Програмите в SISM имат едно и също действие и се означават с COPY и COPYA. Обясненията за копиране на дискети ще се дават при използване на ДОС-командата COPYA /копиране/. И при копирането е възможно използването на ДУ 1 и ДУ 2, поради което и програмата COPYA работи в два режима.

Последователността на копирането на дискетата е следната:

1. Включване на монитора.
2. Поставяне в дисковото устройство на системната дискета SISM /ако се работи с две ДУ SISM се поставя в ДУ 1/.
3. Включване на изчислителния блок, след което започва зареждането на ПМК с ДОС - от SISM и свети червената контролна лампа.
4. Написва ДОС - командата RUN COPYA
/след изгасване на червената лампа/.

С помощта на тази команда програмата за копиране се прехвърля в оперативната памет и след стартиране върху екрана се появява съобщението:

DISK DUPLICATION PROGRAM

ORIGINAL SLOT : DEFAULT = 6

RTN

DRIVE : DEFAULT = 1

RTN

DUBPLICATE SLOT : DEFAULT = 6

RTN

DRIVE : DEFAULT = 2

Натиска се клавишът 1 /т.е. работи се с ДУ 1/ и на екрана се появява съобщението:

PRESS "RETURN" YEY

TO BEGIN COPY

Необходимо е да се натисне клавишът RTN, при което на екрана се появява съобщението:

INSERT ORIGINAL DISK AND

PRESS RETURN

Това съобщение показва, че трябва да се постави оригиналната дискета. За целта от ДУ се изважда дискета SISM и се поставя дискетата, от която ще се извършва копирането /оригиналната, която условно наричаме A/.

Натиска се клавишът RTN и от оригиналната дискета /A/ започва да се чете информацията READING. След прочитане на определената порция информация, четенето спира и на екрана се появява съобщението: INSEKT DUBPLICATE DISK AND PRESS RETURN.

Поставя се новата инициализирана дискета, върху която ще се извършва записването /дискета В/. Натиска се клавишът RTN и започва едновременно инициализирането /ако не е инициализирана/ и записването WRITING. След записването на екрана се появява съобщение за поставяне на оригиналната дискета /А/ - вж. съобщение 1. Поставя се оригиналната дискета и се натиска клавишът RTN.

С това започва отново четене. След свършване на четенето върху екрана се появява съобщение за поставяне на дублиращата дискета /В/ и се натиска клавишът RTN /съобщение 2/. Тези процедури в горната последователност продължават до презаписването на информацията на цялата дискета.

В зависимост от дължината на програмите, съдържащи се в дискетата за копиране - потребителят трябва внимателно да следи съобщенията върху екрана на ПК, за да може последователно да поставя и изважда дискета А и В /дискетата,, от която се копира и дискетата върху която се копира информацията/.

Когато се работи с две дискови устройства, в ДУ 1 се поставя оригиналната дискета /А/, а в ДУ 2 - дискетата за дублиране /В/. В този случай оригиналната дискета и дублираната до момента на завършване на копирането постоянно се намират в дисковите устройства.

След завършване на копирането на дискетата върху екрана се появява съобщението:

```
DO YOU WISH TO  
MAKE ANOTHER COPY?
```

/искате ли да направите още копия?/

Това съобщение дава възможност да се повторят всички процедури, свързани с копирането на дискетата, ако потребителят желае това.

При необходимост от копиране на още дискети непосредствено до въпроса се нанася YES или само Y. С потвърдителния отговор

на екрана се появява съобщение `PRESS "RETURN" KEY TO BEGIN COPY.`

След натискане на клавиша `RTN` на монитора се появява съобщение както при първоначално копиране на дискетата /вж.т. 1/.

Когато потребителят не желае да копира повече дискети, въпроса дали иска да продължи копирането, нанася отрицателен отговор `NO` или само `N`. В резултат на отрицателния отговор екранът се изчиства и в горния ляв ъгъл се появяват знакът и курсорът.

Защитените дискети не могат да бъдат копирани, поради което при опит за копиране на такива дискети на екрана се появява следното съобщение:

```
I/O ERROR
STOPPED AT
```

Подобни съобщения могат да се получат в случаите, когато дискетата е защитена от четене, когато в дисковото устройство не е поставена оригинална дискета или не е затворена веригата на устройството.

Когато се включи мониторът, постави се в дискетното устройство инициализирана дискета, и се включи изчистителният блок, четящите и записващите глави се включват и започва четенето на информацията. На екрана долу вляво се появяват знаците на БЕЙСИК. По-нататък може да започне нанасянето или четенето на програмата.

В случаите, когато ПМК не се изключва и се постави нова инициализирана дискета, е необходимо да се нанесе ДОС-команда `CATALOG`, за да се провери съдържанието на дискетата.

Например ако от дадена дискета, след горната команда на екрана се появяват имената на командите: `HELLO, ADD, SUB, MUL` и `ADD`, по желание на потребителя, може да се извика и работи с която и да е от тях.

При извикването на програмите от дискетата трябва да се имат предвид особености, на някои от които е обърнато внимание по-долу.

Първата програма `ADD` осигурява извършването на действието умножение. Програмата може да реализира решаване на задачите при команда `RUN` и `RTN`. Възможно е чрез ДОС-командата `LOAD ADD` тази команда да се прехвърли в оперативната памет на ПМК и по-нататък да се листва и стартира или само да се стартира. По същия начин могат да се извикват всички останали програми, съхраняващи се в дискетата. За препоръчване е винаги преди ДОС-командата за четене на дискетата и прехвърляне на информацията в оперативната памет да се нанесат командите: `HOME` : `NEW`.

5.2.6. Защита на масиви и дискети

ЗАЩИТА НА МАСИВИ

При работа с дискетата често ще бъде необходимо един или друг масив или програма, съхраняващи се в дискетата, да бъдат защитени от случайно изтриване. За защита се използва ДОС-командата `LOCK` и съответното име на масива.

Защитата на масив може да се извърши в следната последователност:

1. Включване на монитора.
2. Поставяне на дискетата в дисковото устройство /при две ДУ в ДУ 2/.
3. Включване на изчислителния блок.

Преди защитата на даден масив може да се нанесе ДОС-команда-

та CATALOG с цел да се провери какъв е списъкът на наименованията на масивите и кой от тях да се защити.

4. Нанасяне на ДОС-командата LOCK DIV и RTN /при две устройства командата е LOCK DIV, D2/.

По този начин горният масив е защитен. Ако сега отново се изпълни ДОС-командата CATALOG ще се види, че пред програма DIV има символ ж. Този символ по желание на потребителя чрез горната команда може да се постави на съответните масиви и програми за тяхната защита.

При опити за изтриване на масив DIV или други масиви с използване на командата DELETE преименуване с командата RENAME ще се появи съобщението FILE LOCKED/масивът е защитен/.

Записването на масив с името на защитен такъв е невъзможно.

Например:

1Ø A = 2Ø	5Ø ENP
2Ø B = 1Ø	SAVEDIV
2Ø C = A/B	
4Ø PRINT C	

В подобни случаи на екрана пак се появява съобщение:

FILE LOCKED

За изтриването на защитени масиви е необходимо предварително да се премахне тази защита. Елиминирането на защитата се извършва с ДОС-командата UNLOCK.

Премахване на защитата от масив DIV става така:

UNLOCK DIV и RTN

/При работа с две ДУ командата е UNLOCK DIV, D2 и RTN/. По-нататък при изпълнение на командата CATALOG ще се види, че пред името на ма-

сив с име DIV не съществува символ * . С премахване на защитата върху масива могат да се правят различни промени.

ЗАЩИТА НА ДИСКЕТИ

В горната дясна страна на всяка дискета има процеп. Ако този процеп се покрие /например с помощта на скоч/ дискетата ще бъде защитена от записи върху нея. При опити за запис върху защитена дискета се появява съобщението:

```
WRITE PROTECTED
```

Докато в по-горния случай с командата LOCK можеше да се защитят отделни масиви, то сега с покритието на процепа се защитават всички масиви, съдържащи се в нея.

Необходимо е да се знае, че при защитени дискети може само да се чете информация.

ПРОВЕРКА НА МАСИВИ

При по-продължително използване на даден масив или програма последните могат да се повредят. За качеството на съхраняваната информация на дискетата оказват влияние още замърсяването, неправилното съхранение и др.

Необходимо е преди работа с дискетата да се провери нейното състояние. За тази операция може да се използва DOS-командата VERIFY.

Проверката за това дали правилно е записана програмата DIV-командата има следния вид:

```
VERIFY DIV и RTN
```


Когато няма грешка, след изпълнение на горната програма на екрана се показва символът за работа в езика БЕЙСИК, а в случаите, когато има грешка се появява съобщението: I/O ERROR.

Контролни въпроси и задачи

1. За какви цели се използва флопидисковото устройство?
2. Колко писти се съдържат в една дискета?
3. Колко сектора се съдържат в една писта и общо в дискетата?
4. Колко байта информация се съдържа в един сектор?
5. До кои писти потребителите нямат достъп?
6. Каква е последователността на включване на ПК при работа с дисково устройство?
7. Какво разбирате под стартиране на ДОС?
8. Как се инициализира нова дискета и коя е ДОС-командата, която ще се използва за тази цел?
9. Коя е ДОС-командата за запис на информация върху дискета?
10. Запишете програмата:

```
200 DATA 2, 7, 9, 4, 6, 10
```

```
210 READ A, B
```

```
220 ? A, B, A * B
```

```
230 GO TO 210
```

```
240 END
```

и запишете същата върху дискети с име MUL2.

11. Коя е ДОС-командата за четене на информация от дискета?
12. Прехвърлете програмата от въпрос 10 от дискетата в оперативната памет на ПК.
13. Как същата програма може да се появи на монитора?

14. Какво се получава с ДОС-командата CATALOG?
15. Как може да се изпълни от ПМК една програма без същата да се чете от дискетата и прехвърля в оперативната памет?
16. С каква ДОС-команда може да се измени името на масива /респ. на програмата/?
17. Изменете старото име на програмата МУЛТ 2 /въпрос 10/ в НСВО име МУЛТ 3.
18. С помощта на коя ДОС-команда се извършва изтриване на програмата?
19. Как се извършва копиране на програми?
20. Коя програма от системната дискета се използва за копиране на дискети?
21. Как се извършва копирането на дискети?
22. Коя е ДОС-командата за защита на програми?
23. Направете защита на програмата МУЛТ 3.
24. Какво се постига със защита на програми?
25. С коя ДОС-команда се премахва защита на програми?
26. Елиминирайте защитата на програмата МУЛТ 3.
27. Как се извършва защита на дискети и какво се постига с това?
28. С коя ДОС-команда се извършва проверка на масиви и програми?
29. Проверете дали правилно е записана програмата МУЛТ 3.
30. Колко информация се съхранява в една дискета модел ИЗОТ-5253 Е?

С Ъ Д Ъ Р Ж А Н И Е

1. Въведение в персоналните микрокомпютри	3
2. Техникоексплоатационни възможности на ПМК ПРАВЕЦ-82	8
3. Основи на програмирането на ПМК ПРАВЕЦ-82	22
3.1. Директен режим на работа	22
3.2. Програмен режим на работа	27
3.2.1. Основни въпроси, Оператори за присвояване и печат	27
3.2.2. Начини за коригиране на праграми	45
3.2.3. Отпечатване на български текст	50
3.2.4. Програмиране при използване на оператори IF, THEN, GOTO, FOR и NEXT	53
3.2.5. Програмиране при използване на оператори за данни	69
4. Графичен режим на работа	77
4.1. Режим на груба графика	77
4.2. Режим на финна графика	81
5. Запаметяващи устройства	89
5.1. Работа с магнитолентови запаметяващи устройства	90
5.2. Работа с магнитнодискови запаметяващи устройства	93
5.2.1. Общи сведения	93

5.2.2. Инициализация на дискети	96
5.2.3. Записване на масиви върху дискета	98
5.2.4. Чертане на масиви от дискета	99
5.2.5. Копиране на масиви, програми и дискети	102
5.2.6. Защита на масиви и дискети	108

ВЪВЕДЕНИЕ В ПРОГРАМИРАНЕТО НА ПЕРСОНАЛНИТЕ
МИКРОКОМПЮТРИ ПРАВЕЦ-82

Първа част

Проф. д.ик.н. Стоян Ангелов Айков

Езиков редактор Н е в я н а Х р и с т о в а

Художник И с т и л и я н Б о ж и л о в

Техн. редактор Т е о д о р а М а р к о в а

Коректор Т о д о р к а Т о н ч е в а

Формат 16/70/100

Печ. коли 7,25

Изд. коли 6,74

Тираж 300

Цена 0,60 лв.

Печатна база на ВИИ "Карл Маркс" - София

Цена 0,60 лв.