

АЗ ПРОГРАМИРАМ НА 9 ГОДИНИ

Петър
Станчев



НАРОДНА ПРОСВЕТА

ПЕТЪР СТАНЧЕВ

АЗ ПРОГРАМИРАМ
НА 9 ГОДИНИ

ДЪРЖАВНО ИЗДАТЕЛСТВО
„НАРОДНА ПРОСВЕТА“
СОФИЯ, 1987

Книгата съдържа начален, но същевременно и задълбочен курс по програмиране, за усвояването на който е необходимо желание и... завършен трети клас.

Под формата на разговори, проведени между баща и деветгодишен син пред компютър Правец—82, се обясняват основните понятия в програмирането и се показва тяхното използване в езика **БЕЙСИК**.

Освен за първоначално запознаване с програмирането и с основните структури на езика **БЕЙСИК** книгата може да служи и като самоучител при работа с компютъра Правец—82.

Възприетата форма на излагане на съвсем сериозна материя — теорията и практиката на програмирането, може да заинтересува широк кръг читатели — от ученици и студенти до работници, служители и други.

681.326

© Петър Любомиров Станчев, 1987 г.

© Любомир Михайлов Йовев — художник, 1987 г.
c/o Jusautor, Sofia

От автора за книгата

В книгата се предлага един своеобразен курс по програмиране, възникнал като резултат от проведен експеримент да се запознае едно деветгодишно дете с микрокомпютъра Правец—82.

Материалът е изложен популярно и използва познанията по математика, получени в училище до завършване на трети клас. Същевременно се предлага немалък обем информация, като е използвана съответната професионална терминология. Не са избягвани понятия с чуждестранен произход, като например компютър, монитор, транслатор и др., понеже средствата за масова информация боравят с тях.

Материалът е разпределен в петнадесет урока. Това съответства повече на вътрешната структура на материала, отколкото на реалното разпределение на учебното време. Всеки урок представлява диалог между баща и син, работещи с компютъра. За нагледност задачите, определенията, указанията във всеки урок са специално означени. Уроците завършват с по десет задачи за домашно. В края на книгата са приложени всички решения на тези задачи, но се препоръчва на читателя, ако е използвал помощта на отговорите, да напише и сам исканите програми. На разположение на читателя са и азбучен списък на изучаваните инструкции и тълковен речник с използваните термини и понятия, също приложени в края на книгата.

Изложението започва с въвеждане на основни понятия, като компютър, програма, език за програмиране, представят се компонентите на

компютъра, описват се основни операторски дейности. Последователно се отделя внимание на използването на компютъра за редактиране на текстове и за извършване на пресмятания, и двете на ниво съответни инструкции. Дават се първи представи за алгоритмите и езиците за програмиране. Конкретно се описва въвеждането, изпълняването и съхраняването на програми на БЕЙСИК. Разглеждат се въпроси като: как се въвеждат числа и текст по време на изпълнението на програма, как компютърът взема решения (използване на условна инструкция), коя е сърцевината на програмирането (цикли и начини за организирането им), как се съставя автоматизиран „бележник“ (използване на цикли при съхраняване и търсене на данни). Един разговор е посветен на използването на случайните числа в програмирането. Други разговори представят компютъра като художник (графични възможности на компютрите — и чертане на геометрични обекти, букви, рисуване на картини), като музикант (начини за получаване на звук, програми за изпълняване на мелодии). Отделено е внимание на средствата и начините за съставяне на програми за игри.

Изборът на компютъра Правец—82 и на езика за програмиране БЕЙСИК се дължи единствено на широкото им разпространение сред евентуалните бъдещи програмисти. В книгата са разгледани само тези изразни средства на езика БЕЙСИК, които според автора дават възможност както да се съставят достатъчно смислени програми, така и да се разберат основните конструкции, характерни за програмирането.

Изложението курс е действително проведен с деца на посочената възраст, отделно от това и с възрастни, което дава увереност на автора да

твърди, че при желание от страна на читателя, незнаещ да програмира, и при изпълняване на включените в книгата нетрудни упражнения може да се навлезе успешно в материала на програмирането.

Книгата написах по молба на моя син, на когото не мога нищо да откажа, още повече че той едва ли не я написа сам и вложи труд, равносителен на моя.

За подобряването на книгата непосредствено влияние оказаха и М. Дондова, А. Звездинов, Н. Лазарова, Р. Павлова, Е. Сендова и Б. Христов.



УВОД

Описва се разговор между деветгодишно момче и неговия баща, който е донесъл въкъщи микроботър *Правец—82*. Въвеждат се понятията компютър, програма, машинен език, език за програмиране, транслатор. Дават се кратки исторически сведения за създаването на компютрите.



Казвам се Любомир. Ученик съм в III клас.
Днес се връщам по-рано от училище. Бързам да
оставя чантата и да изляза да играя футбол.
Влизам вкъщи и виждам баща ми да работи с
някаква машина.

Татко, каква е тази машина?

Това е микрокомпютър.

Какво значи микрокомпютър?

Микрокомпютър означава малък компютър,
а компютър означава изчислител. Това е устройство,
с помощта на което могат да се извършват
действия с данни, т. е. с числа или текстове,
или, както още се казва, да се обработва информация.

Елката микрокомпютър ли е?

Не. Калкулаторите, какъвто е ЕЛКАТА, за да извършват някакво действие, трябва да бъдат ръководени ръчно, стъпка по стъпка. А компютрите могат сами да направляват изпълнението на своите действия.

А как компютрите могат да направляват своите действия?

С помощта на програми. Програмата е описание на работа, която компютърът трябва да изпълни. Представлява група инструкции за действия, зададени на език, който компютърът разбира.

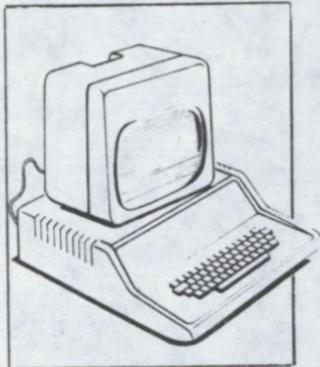
Език ли? Какъв е този език, който компютърът разбира?

Компютрите от един и същи вид имат свой собствен език, който разбират. Той се нарича машинен език. Такъв език обикновено не е удобен за писане на програми. Освен такива машинни езици има и други, които се разбират от всички видове компютри, стига да са снабдени с преводачи от съответния език на машинния им език. Тези по-общи езици се наричат езици за програмиране и са създадени специално за тази цел.

Преводачът какво е?

Преводачът от даден език за програмиране е програма, която превежда от един език за програмиране на машинен език. Програмите преводачи се наричат транслатори.





Татко, аз мога ли да науча някой език, който компютърът разбира?

Разбира се. Ако искаш, ще те науча да програмираш на БЕЙСИК.

БЕЙСИК? Добре. А като го науча, какво ще мога да правя с компютъра?

Ще можеш да си играеш с компютъра, като сам съставяш програми за игри с него, ще можеш да решаваш задачи от учебника по математика, като съставяш програми за тях.

А какви игри ще мога да съставям?

Помниш ли електронните автомати за игри на Златните пясъци? Ако си старателен ученик, и ти ще можеш да правиш подобни игри за този компютър, като създаваш нужните програми.

И ще ми даваш после да играя с тях?

Защо не, стига да си научил другите уроци. Но ще видиш, че удоволствието от това, че си създал програма за игра, е по-голямо от това, да играеш после играта.

Татко, този компютър има ли име?

Да. Това е българският микрокомпютър Правец—82.

А кой го е създад

Създаден е от Института по техническа кибернетика и роботика при Българската академия на науките и се произвежда в Приборостроителния завод в гр. Правец.

Татко, това ли е първият компютър?

Първият компютър в света е създаден през годините от 1936 до 1941 г. в САЩ от Джон Атанасов. Чу ли добре името? Дядото на Джон Атанасов е бил българин. А първият български компютър се казва Витоша и е създаден в Института по математика с Изчислителен център при Българската академия на науките под ръководството на И. Юлзари и Бл. Сендов през 1962 г.

Вие, големите, за какво използвате компютрите?

Компютрите се използват вече в почти всички области на нашия живот. Те участват в управлението на предприятия, на химически процеси, на космически полети, помагат да се тър-



сят данни в големи справочници; композират и изпълняват музика; чертаят схеми и рисуват картини; проектират машини; извършват и други всевъзможни дейности, някои от които по-бързо от самите хора, а други не само по-бързо, но и по-добре от тях. Не е далеч денят, в който, за да разберем например кой е най-високият връх в света, ще включим домашния компютър в мрежата от компютри, по която ще получим отговора. Както сега върши имаме контакти за електричество и телефон, ще имаме и контакт, от който чрез нашия компютър ще можем да получаваме каква ли не информация, която е записана в огромни хранилища от знания.

Като слушаш всичко това, ти нали не забравиш, че компютрите само изпълняват програми, написани от хора? „Компютрите са само инструмент и като всички инструменти, ако не се използват добре, няма да има сполука.“ Запомни тези думи на създателя на първия компютър Джон Атанасов.

Татко, хайде да започваме да учим. Искам бързо да се науча да работя с компютър.

Добре. Само че вече е късно, трябва да вечеряш и да си лягаш. Ще започнем от утре.

Как да е късно? Аз щях да излизам да играя навън...

Ще започнем утре. Всяка вечер ще работим двамата с компютъра. Ще имаш и домашно, което трябва да правиш сутринта, след като си готов с домашните от училище. Ще ти давам задачи и от твоите учебници за трети клас.

ОТ КАКВО СЕ СЪСТОИ МИКРОКОМПЮТЪРЪТ

Разговорът е за съставните части на микрокомпютрите. Използва се микрокомпютърът Правец—82. Дават се популярни сведения за клавиатурата, изчислителния блок, монитора, флотидисковото устройство. Разглеждат се касетофон, печатащо устройство, чертаещо устройство, устройство за въвеждане на чертежи и ръчки за игра.



Започваме нашия първи урок. Той е за частите, от които се състои всеки микрокомпютър. Да разгледаме една задача:

Автомобил се движжи със средна скорост 80 км/час и изминава пътя между два града A и B за 4 часа. Колко е разстоянието между градовете A и B?

Аз мисля, че е 320.

Какво е това 320?

Разстоянието между A и B е 320 км.

Как реши задачата?

Умножих скоростта 80 км/час и времето 4 часа.

Какви числа трябва да въведем в компютъра, за да реши той тази задача?

Числата 80 и 4.

Какво още е необходимо на компютъра, за да реши задачата?

Програма.

Как ще въведем в компютъра числата 80 и 4 и програмата, която ще съдържа инструкциите за действия с тях?

С копчетата.

Точно така. Тези копчета се казват клавиши, а всички заедно образуват клавиатура. Има ли нещо общо между тях и пищещата машина?

Има нещо общо, само че тук буквите са по две на един клавиш.

Да. Отгоре са буквите на кирилица, а отдолу — на латиница. Има и клавиши с цифри. Има и още клавиши с думи и знаци върху тях, има и един голям клавиш, на който не пише нищо, както на пищещата машина. Но как се работи с клавиатурата, ще започнем да учим утре.

За какво са латинските букви?

Инструкциите за действия на езика БЕЙСИК се задават чрез английски думи. За да ги изписваме тях, ще ни трябват латински букви.

А защо тогава има букви на кирилица?

За да може да се използва компютърът за отпечатване на текстове на български език. Например можем да поискаме компютърът да отпечата отговора на разглежданата задача не като 320, а като „Разстоянието между градовете A и B е 320 км“.



Разбрах. И какво става по-нататък?

Както човек запомня числата, с които трябва да реши една задача и начина, по който да я реши, така и компютърът запомня данните (в случая това са числата 80 и 4) и програмата за решаване на задачата в своя вътрешна памет. Тази вътрешна памет се състои от отделни клетки. В тях се записват и от тях се четат данните и програмите. Големината на клетките на паметта на различните видове компютри е различна. При определена големина (каквато е в Правец—82) клетките се наричат байтове. Обикновено в 1 байт се записва 1 символ, който е буква, цифра или знак. Големината на вътрешната памет на Правец—82 е 65 536 байта и тъй като 1024 байта наричаме 1 килобайт, то-ва число може да се представи като 64 килобайта (пише се 64 Кбайта).

Татко, а кой ще умножи числата от задачата?

Микропроцесорът. Той е сърцето на микрокомпютъра. Микропроцесорът извършва аритметични действия, прави сравнения, проверява условия, управлява въвеждането и извеждането на данни във и от вътрешната памет.

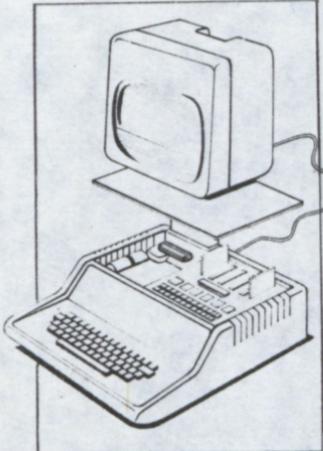
А от какво е направено това чудо?

Наистина микропроцесорът е чудо. Той представлява малко кварцово кристалче в пластмасова обвивка, но какво съдържа то, ще разбереш след няколко години, когато учиш в училище какво е транзистор.

Микропроцесорът и вътрешната памет се намират в т. нар. изчислителен блок. Там се намират и някои допълнителни устройства, като например високоговорител, с помощта на който Правец—82 може да възпроизвежда звуци. Пред изчислителния блок се намира клавиатура, за която говорихме, а върху него какво е поставено?

Този телевизор.

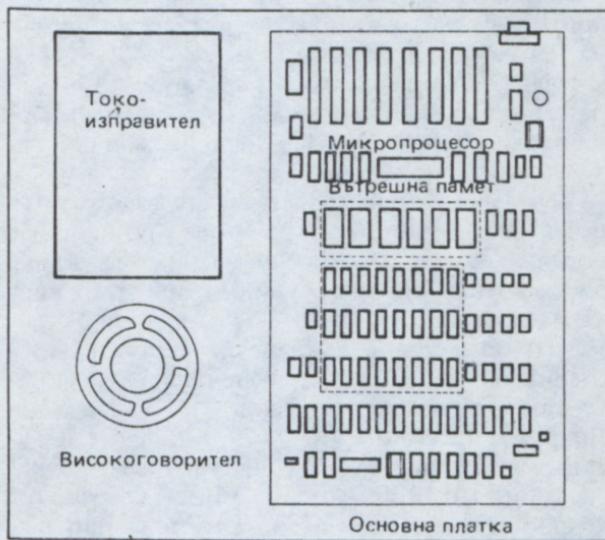
Да. Този телевизор можеш да го наричаш монитор. За какво служи, сигурно се досещаш. Върху него се появява това, което се въвежда от клавиатурата, а също и резултатите, които се получават от работата на микропроцесора.

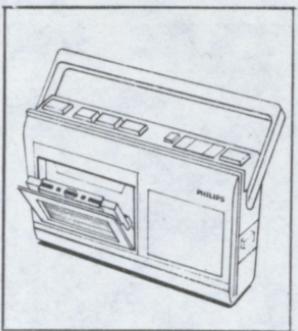


Върху монитора ще се появи и отговорът на нашата задача: „Разстоянието между градовете A и B е 320 км.“ Когато се извежда само текст върху монитора, се казва, че той работи в текстов режим и тогава върху экрана могат да се побират 24 реда с по 40 символа на ред. Когато се чертае върху монитора, се казва, че той работи в графичен режим и тогава върху экрана могат да се изчертават до 40 реда с по 40 „големи“ точки на ред или 160 реда с по 280 „малки“ точки на ред. Ако мониторът е цветен, цветовете са 16 при работа с „големи“ точки и 8 при работа с „малки“ точки.

А тази кутия за какво е?

Това е флопидисково устройство. То е от тези устройства, които се наричат външни запомнящи устройства на компютрите. Както човек записва на лист това, което не иска да помни непрекъснато, така с помощта на такива устройства се запомня информацията, която не може да стои непрекъснато във вътрешната памет на компютъра. Флопидисковото устройство запомня информацията, като я записва върху така наречените магнитни дискети. Данните и програмите се записват върху магнитните ди-





скети както във вътрешната памет. Всеки път, когато са необходими, те се прехвърлят във вътрешната памет и не се налага повторно да се въвеждат символ по символ от клавиатурата. Разбра ли всичко, което ти казах дотук?

Мисли, че съм разбрал.

За да разбереш по-добре, ще ти кажа, че към този компютър може да се включи твоят касетофон, който ще играе ролята на външно запомнящо устройство, макар и много по-бавно от флопидисковото устройство. Касетофонът запомня информацията, като я записва върху магнитните касети. И тук данните и програмите, записани на касети, могат лесно да се прехвърлят във вътрешната памет и да се ползват след дни и месеци, с което се избягва набирането им всеки път чрез клавиатурата.

Искам да ти обърна внимание на факта, че микропроцесорът работи с данни, намиращи се във вътрешната памет много по-бързо, отколкото, ако трябва първо да ги прехвърли от външната във вътрешната памет, но затова пък на външна памет може да се съхранява много повече и за по-дълго време информация.

Съвршихме ли? Не виждам други устройства.

Искам да спомена за още някои устройства, които могат да се включат към микрокомпютъра, но ги нямаме тук.

Печатащо устройство. Свързано с компютъра, то действа като автоматична пишеща машина, която печата на хартия програми или резултати от пресмятания.

Чертаещо устройство. Чрез него на хартия се изчертават чертежи. Устройството разполага с писалка, която автоматично се движи върху лист хартия и изчертава запомнени от компютъра чертежи.

Устройство за въвеждане на чертежи. Чрез специална писалка върху специална плоча се изчертават ръчно чертежи, които се въвеждат в микрокомпютъра.

Ръчки за игра. Това са една или повече ръчки, с помощта на които играчи могат да играят с компютъра различни игри, стига в компютъра



да се въведат програми за тези игри. Чрез тях играчите имат възможност да задават желаната от тях посока на движение на даден обект, изобразен на екрана на микрокомпютъра.

Какви игри например могат да се играят с тях?

Например футбол. Чрез ръчките се движат играчите на двата отбора.

Жалко. Излиза, че липсват най-важните части на този компютър.

За днес приключваме. Ето и домашното.

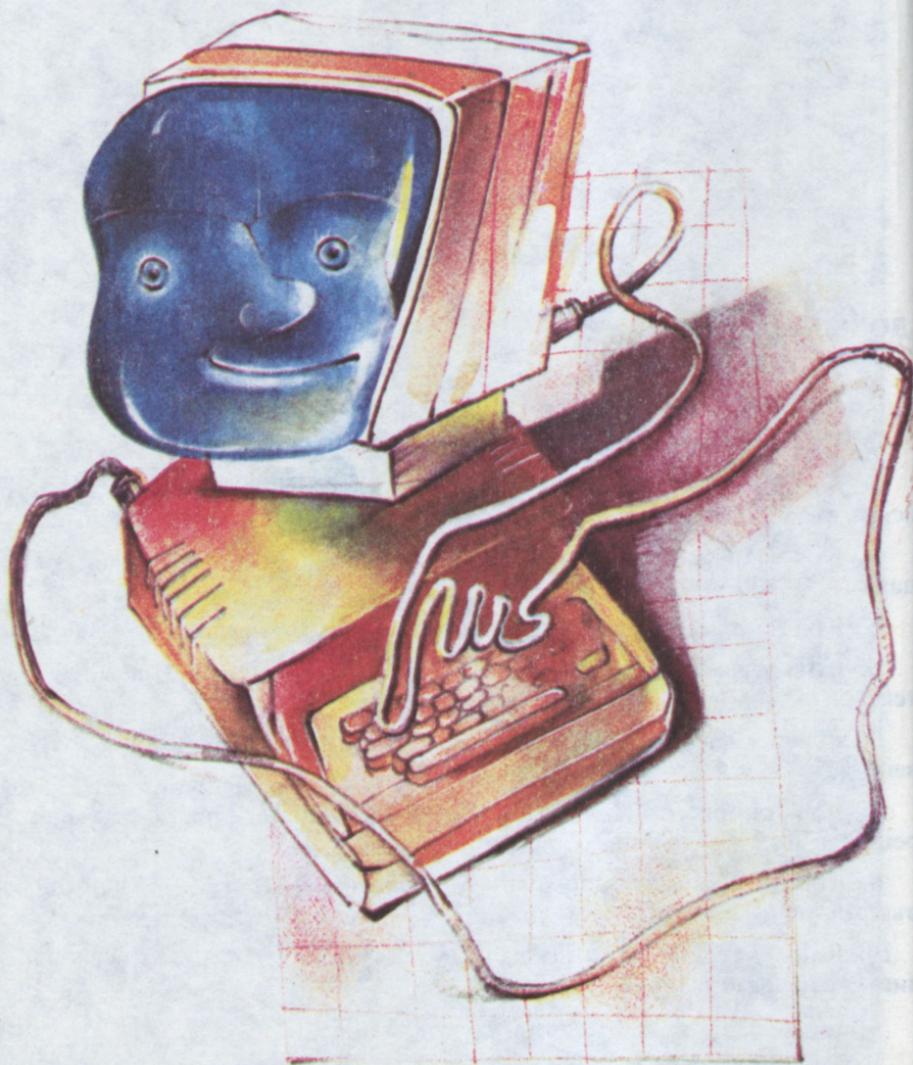
ДОМАШНА РАБОТА

1. Избройте основните части на микрокомпютъра Правец—82.
2. Какво включва изчислителният блок на Правец—82?
3. Ако считаме вътрешната памет на Правец—82 за книга, в която всяка страница има по 256 байта, колко страници има тази книга?
4. Коя част на Правец—82 управлява въвеждането и извеждането на данните?
5. Коя част на Правец—82 извършва пресмятанията?
6. Колко символа могат да се съберат на екрана на Правец—82 в текстов режим?
7. Върху каква външна памет може да се пази информация в Правец—82?
8. Микропроцесорът работи по-бързо с данни, намиращи се във вътрешната или във външната памет?
9. Кое от външните запомнящи устройства на Правец—82 най-бързо въвежда и извежда информация?
10. Какви устройства могат да се включат към Правец—82 освен клавиатурата и монитора?

II урок

КАК СЕ РАБОТИ С КОМПЮТЪРА

Разговорът е за въвеждане на информация в компютъра Правец—82. Започва се с включването на компютъра. Обяснява се набирането и редактирането на текст. Разглежда се инструкцията PRINT. Дава се обяснение на съобщението на компютъра "SYNTAX ERROR".



Днес ще се научим да въвеждаме информация в Правец—82. За целта трябва първо да се научим да го включваме.

Включването се извършва в следната последователност:

1. Поставя се подгответа дискета във флопидисково устройство.
2. Включва се мониторът (ключето се намира под екрана).
3. Изчаква се 10-ина секунди.
4. Включва се изчислителният блок (ключето се намира на задната страна на кутията му).

Да го включва ли сам?

Добре. Ще чуеш кратък звук. Ето какво се появява на екрана:



мигаш
правоъгълник

Знакът  се нарича напомнящ знак и напомня, че работим с езика БЕЙСИК, а мигащият правоъгълник се нарича указател и показва мястото върху екрана, където ще се появи съответният символ, написан чрез клавиатурата.

Правец—82 е готов за работа. Светят две лампички — едната на монитора, а другата на изчислителния блок, които показват, че те са включени.

А сега да разгледаме клавиатурата. Правец—82 се произвежда с няколко разновидности на клавиатурата. Ще разгледаме клавиатурата на този компютър, който съм донесъл. Тя е и най-разпространената.



Ще се опитам да напиша нещо:

Напиши думата МИР.

Трябва последователно да се натиснат клавишиите **M** **I** **R** и на екрана ще се появи



Опитай.

Наистина се появи думата МИР.

А сега да видим всеки клавиш за какво е и как ще можем да го използваме.

Започваме с червения клавиши — най-десният в най-горния ред, на който пише **RST**. При натискането му се чува кратък звук, указателят се премества на следващия ред, а това, което сме въвели от клавиатурата, не се възприема от компютъра. Натискането на клавиша предизвиква прекъсване на текущата работа на компютъра. Натисни го.



Както вече забеляза, върху повечето от клавищите има два символа — главна буква на кирилица и главна латинска буква или други два символа. Кой от двата символа се въвежда при натискане на клавиша, зависи от трите клавиша **LAT** на клавиатурата. В момент, в който е натиснат кой и да е от белите клавиши **LAT**, се казва, че компютърът е в режим латиница и се въвежда долният от двата символа, означени върху натиснатия клавиши. Еднократно натискане на жълтия клавиши **LAT** сменя режим латиница с кирилица или обратно, до следващото му натискане. Ако с жълтия клавиши **LAT** е въведен режим латиница, на клавиатурата свети лампичка, указваща, че режимът е латиница.

Нали така е и при пишещата машина.

Да. Само че при нея при натискането на съответния клавищ се сменят малки с големи букви.

Сега да направим няколко упражнения.

Да се въведе английската дума PRINT.

С една ръка държим постоянно натиснат белия клавищ **ЛАТ**, а с другата набирате по-следователно:

П **Р** **И** **Н** **Т**



Да се въведе текстът PRINT МИР.

Преди да започнеш, натисни клавиша **RST**, за да се откажеш от преди това набрания текст, понеже сега само се учим. Указателят преминава на нов ред. Първата дума въведи с натиснат бял клавищ **ЛАТ**.

А как да разделям думите?

Място между думите се оставя с големия клавищ най-отдолу, върху който не пише нищо. Ще го наричаме интервал. Интервал може да се въвежда независимо дали режимът е кирилица или латиница.

Ако събъркаш, натисни червения клавищ **RST** и започни отново.

Да се опитаме да въведем същия текст, като използваме жълтия клавищ **ЛАТ**.



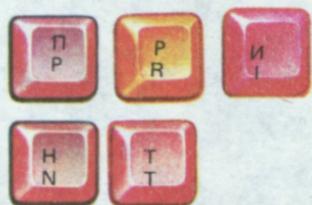


Използваме клавиша **RST**, за да започнем отново. Натискаме жълтия клавиш **ЛАТ**. Светва лампичката за режим латиница. Натискаме последователно **П Р И Н Т**, натискаме отново жълтия клавиш **ЛАТ**. Лампичката угасва. Натискаме големия клавиш интервал и **М М И И Р Р**.



А сега да напишем и няколко числа. Цифрите са на най-горния ред. Цифрата нула е отбелязана като **Ø**, за да се различава от буквата **O**. Цифрите се въвеждат, когато сме в режим кирилица, а знаците под тях, когато сме в режим латиница.

Да се въведе PRINT "123".



За последен път започваме с червения бутон **RST**. Натискаме жълтия клавиш **ЛАТ**, а след това **П Р И Н Т**, поставяме интервал, натискаме **2**. Натискаме отново жълтия клавиш **ЛАТ** и минаваме в режим кирилица, натискаме последователно **1 2 3 #**.



Натискаме белия клавиш **ЛАТ** и едновременно с него натискаме **2**. Досега се отказвахме от набрания текст, като натискахме **RST**. Сега ще ти кажа как може текст да се въведе в компютъра. Това става

чрез клавиша **RETURN**. Да го натиснем.
На екрана се появява:



PRINT "123"

PRINT "123"

123

] ■

Компютърът ни отговори.

Отговори ни, понеже въведохме с клавиши **RETURN** текст. Но тук има нещо по-интересно. Той разпозна думата PRINT като инструкция на езика БЕЙСИК за отпечатване на записания в кавичките текст, т. е. ние муказахме да отпечата 123 и той го направи. Използвахме инструкцията PRINT, която има вида:

PRINT "123"

думата PRINT _____
знакът кавички _____
текст за отпечатване _____
знакът кавички _____

Да се отпечата думата МИР.

Каква инструкция трябва да напишем?

PRINT "МИР", накрая с **RETURN**.

Въведи я.

PRINT "МИР"

PRINT "МИР"

МИР

] ■

А сега още малко за инструкцията PRINT. Вместо думата PRINT може да се използва въпросителен знак, като видът на инструкцията се запазва.

Да се отпечата ДНЕС Е ПОНЕДЕЛНИК.

За целта ще използваме ? "ДНЕС Е ПОНЕДЕЛНИК". Набери и да видим какво ще се получи. Не забравяй клавиша [RETURN]

]? "ДНЕС Е ПОНЕДЕЛНИК"

]? "ДНЕС Е ПОНЕДЕЛНИК"
ДНЕС Е ПОНЕДЕЛНИК

]? ■

Добре. А сега за още една възможност на инструкцията PRINT. Освен чрез написване един до друг и чрез знака плюс можем да съединяваме няколко текста.

Какво отпечатва инструкцията:

? "ДНЕС Е ПОНЕДЕЛНИК" + "12 МАЙ"

Да я наберем и да видим резултата:

]? "ДНЕС Е ПОНЕДЕЛНИК" + "12 МАЙ"

]? "ДНЕС Е ПОНЕДЕЛНИК" + "12 МАЙ"
ДНЕС Е ПОНЕДЕЛНИК 12 МАЙ

]? ■

Видяхме как се въвежда и отпечатва текст. А сега да научим нещо и за останалите клавиши.

Ако се натисне клавишът **RPT** едновременно с даден клавиш, той предизвиква много-кратното появяване на символа от този клавищ върху экрана.



Камо се използва клавиша **RPT**, да се въведе инструкцията ? "1111111111111111"

Набирам въпросителен знак, последван от кавички, двадесет единици набирам чрез едновременно натискане на **1** и **RPT**.

Наистина е по-лесно, отколкото двадесет пъти да се натисне клавишът 1.



А сега да видим как се поправя неправилно въведен символ. Сигурно ти омръзна все да започваме отново с клавиша **RST**, като сбъркame.

Видях те да връщаш указателя назад.

Да. Това става чрез натискане на клавиша **X** при натиснат клавиш **MK**. Указателят се връща един символ назад и може да се набере верният символ.



*Да се поправи инструкцията:
PRINT "12#" на PRINT "123".*

Натиснали сме **ЛАТ** по-рано. Натискаме **МК** и без да го освобождаваме, натискаме два пъти **XH**. Указателят се връща две позиции назад. Набираме 3" и въвеждаме набрания текст с натискане на **RETURN**.

Какво ще се получи, ако допуснем грешка?
Да въведем PRIND "A" и **RETURN**.

На экрана се появява нещо.

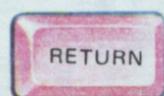


PRINT "A"

PRINT "A"

? SYNTAX ERROR

] ■



? SYNTAX ERROR означава „сънтактична грешка“. Това е съобщение, че текстът, който сме въвели, не представлява инструкция на езика БЕЙСИК или ако има такава инструкция, не е спазен точният ѝ вид и затова ни се казва, че сме допуснали грешка. Ние сами трябва да открием грешката и да напишем реда отново. Указателят е готов за възприемане на нов ред.

Действа като учителка. Подчертава само реда, а ние трябва да намерим грешката и да си я поправим.

Разгледахме всички клавиши, с изключение на **[OCB]**. За него ще говорим утре вечер. А сега домашното. Днес то ще е малко по-дълго. Всеки ден то ще става все по-дълго, т. е. ще ти е нужно повече време, но сам ще се увериш, че бъде и по-интересно.

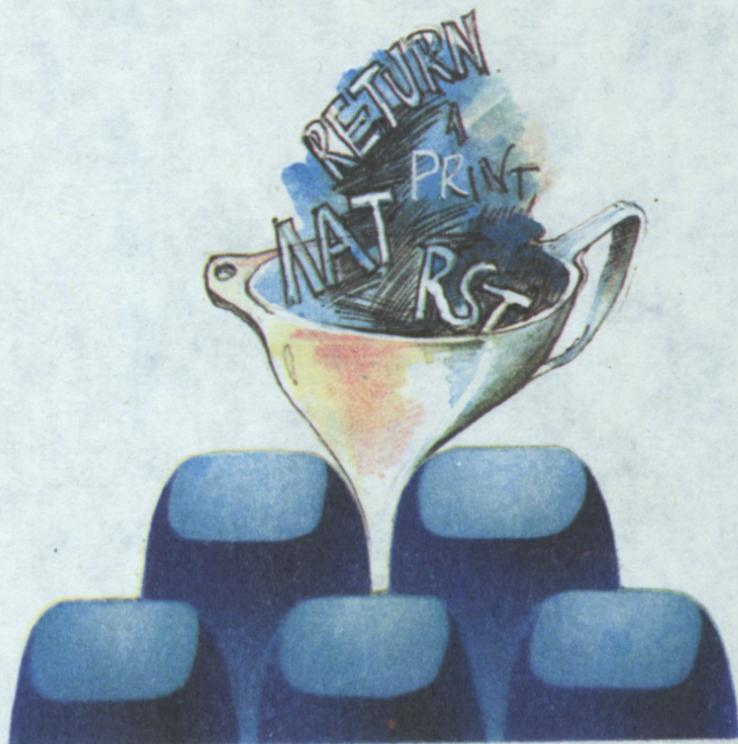
ДОМАШНА РАБОТА

1. Да се включи Правец—82.
2. Какво извършва червеният клавиш **[RST]** ?
3. С кой клавищ завършва въвеждането на текст?
4. Да се накара Правец—82 да отпечата следните текстове:
 - а) ОБИЧАЙ ГОРЕЩО И БЕЗЗАВЕТНО СВОЯ НАРОД И СВОЯТА РОДИНА!
 - б) КОЙТО НИКОГА НЕ ЗАПОЧВА, ТОЙ НИКОГА НЕ ЗАВЪРШВА!
 - в) ВАРТБУРГ 353W
5. Каква е разликата при използването на въпросителен знак вместо "PRINT"?
6. Да се въведе PRINT "ГРАТ СОФИЯ" и да се поправи буквата Т с Д.
7. Какво ще се отпечата с инструкциите:

]? "+ "+"

]? "+" "+"

8. Като се използва клавишът **RPT**, да се открие максималният брой символи, от които може да се състои една инструкция на БЕЙСИК.
9. Какво се прави при съобщението "?SYNTAX ERROR"?
10. Да се въведат няколко реда текст и да се разпечатат.



III урок

КОМПЮТЪРЪТ ИЗЧИСЛЯВА

Разговорът е за използването на компютъра Правец-82 като калкулатор. Обяснява се начинът за запис на основните аритметични действия и прости аритметични изрази. Разглеждат се нови възможности на инструкцията PRINT и други възможности за редактиране на текст.



Днес ще накараме Правец—82 да пресмята. Ще му задаваме да пресмята аритметични изрази и така ще го използваме като калкулатор.

Ще работим с него като с ЕЛКА?

Да. Нека първо да ти припомня, че аритметичните изрази включват числа, знаци за аритметични действия и скоби. Ето с кои знаци се записват аритметичните действия в езика БЕЙСИК:

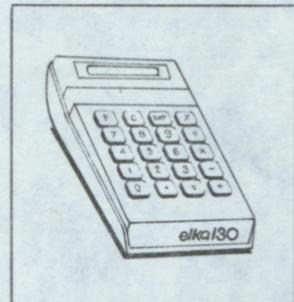
за събиране се използва знакът + за умножение се използва знакът *
за изваждане се използва знакът — за деление се използва знакът /

Значи умножението и делението означаваме по нов начин.

Правилно. Още трябва да знаеш, че в аритметичен израз без скоби действията се изпълняват от компютъра подред отляво надясно, като действията умножение и деление се изпълняват преди събиране и изваждане. Ако има скоби, действията в тях се изпълняват преди останалите. Скобите се пишат по познатия ти от училище начин. Въобще всичко е същото, както си учили в училище, с изключение на споменатите означения на действията умножение и деление.

Да ти кажа сега нещо за числата в аритметичните изрази. Компютърът извършва аритметични действия не само с цели числа, каквито познаваш до момента. В четвърти клас ще учиш за дробни числа и за представянето на числата като десетични дроби. Например числото две и половина като десетична дроб се записва така: 2,5, числото 3 е цяло, но може да се запише и като десетична дроб: 3,0. Записването на числата в езика БЕЙСИК е: за целите числа — по обичайния начин, а за десетичните дроби вместо запетая се използва точка за отделяне на цялата и дробната част. Например числото 2,5 в БЕЙСИК се записва 2.5, числото 3 — като 3 или като 3.0.

За да изчисляваме аритметични изрази и за



да отпечатваме техния резултат, ще използваме инструкцията PRINT в следния наши вид:

PRINT 5 * 4

думата PRINT —

аритметичен израз —

Добре. Разбрах всичко. Давай ми сега задачи, аз ще ги решавам.

Да се пресметне сумата на числата 216 и 123.

Вместо PRINT може да се употреби и въпросителен знак, нали?

Да. А сега набери:

] ?216+123

RETURN

Провери дали смята вярно.

Вярно е.

Четем "колко е 216 плюс 123". Накрая натисни RETURN

] ?216+123

] ?216+123

339

Да се пресметне разликата на числата 489 и 126.

] ?489-126

] ?489-126

363

Да се пресметне произведението на числата 6 и 5.

□ ? $6 \cdot 5$

□ ? $6 \cdot 5$
6 · 5

Тук се отпечата десетичната дроб шест и половина вместо тридесет. Защо? Какъв беше знакът за умножение?

”*”.

□ ? $6 * 5$

□ ? $6 * 5$
30

Да се пресметне частното на числата 81 и 9.

□ ? $81 / 9$

□ ? $81 / 9$
9

Вярно смята. Хайде да напишем трудна задача и да видим колко бързо ще я реши?

Добре. Да вземем изрази със скоби.

Да се пресметне $(18 + 54) : 6$.

□ ? $(18 + 54) : 6$

□ ? $(18 + 54) : 6$
12

Да се пресметне $(38184:43):37$.

] $?(38184/43)/37$

] $?(38184/43)/37$

24

Да се пресметне $(3000 + 5000 \cdot 4):2$.

] $?(3000+5000*4)/2$

] $?(3000+5000*4)/2$

11500

Много е бърз. А как става със задачите, в които се иска да се пресмята някакво неизвестно x ?

Имаш предвид уравнения или вие ги наричахте все още равенства с неизвестно? Да разгледаме задачата:

Да се намери стойността на x в уравнението $x:24 = 1768$.

Както сам разбра, досега използвахме инструкцията PRINT за отпечатване стойността на някакъв аритметичен израз. Можем да използваме същата инструкция и за решаването на тази задача, само че трябва сами да решим уравнението, т. е. да намерим неизвестното x , като го изразим като аритметичен израз и чак тогава Правец—82 ще може да го пресметне.

Значи не всичко става с компютъра?

Нали ти казах, компютърът е само инструмент и човекът е този, който го ръководи. Що се отнася до решаване на уравнения, ще ти покажа какво е необходимо на компютъра, за да прави това, но след няколко урока.

А сега кажи на какво е равно x ?

$x = 1768.24$

Задай на компютъра да го изчисли.

? $1768 * 24$

? $1768 * 24$

42432

Да решим още няколко задачи.

Да се намери стойността на x в уравнението $x + 4250 = 18000 : 2$.

? $18000 / 2 - 4250$

? $18000 / 2 - 4250$

4750

Да се намери стойността на x в уравнението $x \cdot 4 = 1265 + 3583$.

? $(1265 + 3583) / 4$

? $(1265 + 3583) / 4$

1212

А сега да се върнем към задачата, която разгледахме първата вечер. Да се опитаме да я решим с помощта на Правец—82.

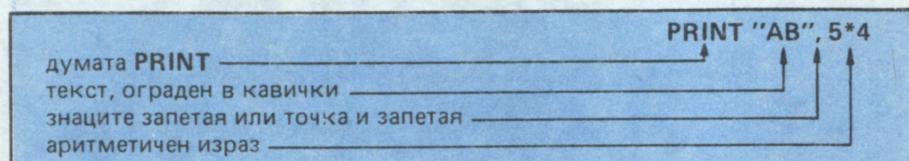
Автомобил се движжи със средна скорост 80 км/час и изминава пътя между двуа града A и B за 4 часа. Колко е разстоянието между градовете A и B?

Хайде да решиш задачата.

☐ ? $80 * 4$

☐ ? $80 * 4$
320

Това е добре, но ние искахме да се отпечатва "Разстоянието между гр. А и В е 320 км". Това ще извършим, като използваме инструкцията PRINT едновременно по двата познати ни машина, за отпечатване на текст и за отпечатване на стойност на аритметичен израз. Видът ѝ е:



С инструкцията PRINT могат да се отпечатват стойности на няколко аритметични израза и няколко текста. Ако разделителят е точка и запетая, стойността на аритметичния израз и текстът се появяват едно до друго на екрана; ако е запетая — компютърът автоматично ги отделя едно от друго с интервали.

А сега ето едно решение на нашата задача.
? "РАЗСТОЯНИЕТО МЕЖДУ ГР. А И В Е"
"; 80*4; " КМ "

Набери го и на екрана трябва да се получи решението на задачата. Обърни внимание, че след първия текст и пред км съм оставил по един интервал. Това е, за да бъде отговорът в по-добър вид, за да има по един интервал преди и след числото 320.

Д?"РАЗСТОЯНИЕТО МЕЖДУ ГР. А И В Е " ;80*4;" КМ"

Д?"РАЗСТОЯНИЕТО МЕЖДУ ГР. А И В Е " ;80*4;" КМ"
РАЗСТОЯНИЕТО МЕЖДУ ГР. А И В Е 320 КМ

Ето още няколко примера.

Да се пресметне разликата на 55 000 кг и 12 830 кг.

Как ще решим тази задача?

Ще извадим 12 830 от 55 000.

И какво ще поставим след разликата?

Килограми.

Какъв разделител ще използваме, запетая или точка и запетая?

Точка и запетая.

За да е по-добре оформлен резултатът, какво ще поставим пред текста кг?

Интервал.

Добре. А сега въведи самата инструкция.

Д?55000 – 12830;" КГ"

Д?55000 – 12830;" КГ"
42170 КГ

Да се пресметне разликата на 12 год. 8 мес. и 10

Как се решава тази задача?

Първо ще извадим годините, а след това месеците.

А помежду им какъв знак ще поставим?

Точка и запетая.

Добре. Въведи инструкцията.

] ?12-10;" ГОД ";8-6;" МЕС"

] ?12-10;" ГОД ";8-6;" МЕС"
2 ГОД 2 МЕС

Татко, на компютъра позволено ли му е да дели на нула?

Дали му е позволено? Той също като нас не знае какво значи да се дели на 0. Опитай, раздели едно число на нула.

] ?5/0

] ?5/0
? DIVISION BY ZERO ERROR

Компютърът ни казва, че делим на нула и това е грешка.

Дотук се запознахме с възможностите за пресмятане и отпечатване стойности на аритметични изрази, а сега ще ти покажа още някои възможности за поправка на грешки.

Да въведем:

] PRINT 12:3+3

и нека все още не сме натиснали **[RETURN]**.
Трябва да се върнем четири символа и да по-

правим ":" с "/". Връщането ставаше с натиска-
не на **[X]** при натиснат предварително кла-
виш **[MK]**. Направи го. Така.

А сега да поправим ":" с "/". Така. При въ-
веждане на останалите символи досега ги наби-
рахме отново. Но те са на екрана. Чрез нати-
скане на клавиша **[Y]** при натиснат клавиш
[MK] указателят се предвижва надясно и сим-
волите, през които минава, се въвеждат автома-
тично. Въведи така символите до края на реда
и натисни **[RETURN]**



Изпълни сега тази задача:

Чрез **[MK]** **[X]** и **[MK]** **[Y]** да се поправи инструкцията **PRINT**

5.3 + 1235 на **PRINT 5*3 + 1235** и да се изпълни.

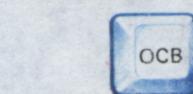
Набираме



7x



6x



и преди да е натиснат клавишът **[RETURN]**, 7
пъти натискаме **[MK]** **[X]**, набираме "*",
6 пъти натискаме **[MK]** **[Y]** и накрая
[RETURN].

] PRINT 5*3+1235

] PRINT 5*3+1235

1250

А сега ще разгледаме още едно средство за редактиране. Спомняш ли си кой клавиш от клавиатурата остана неразгледан?

Клавиша [OSB] .

Да. При натискането му компютърът влиза в режим на еcranно редактиране и реагира по особен начин на някои символи. Чрез тях може да се управлява придвижването на указателя не само наляво и надясно, но и нагоре и надолу по экрана.

Да наберем на екрана "РАЗСТОЯНИЕТО Е " ; 50+150; " М" и да натиснем **[RETURN]** .

] "РАЗСТОЯНИЕТО Е ";50+150;" М"

] "РАЗСТОЯНИЕТО Е "; 50+150; " М"

? SYNTAX ERROR

Появява се

? SYNTAX ERROR.

Какво събрахме? А, пропуснали сме въпросителния знак.

Да се опитаме да го поправим.

Не е ли късно да го поправяме? Вече има съобщение за грешка.

Не. Това става чрез влизане в режим еcranно редактиране. При него може да се използва всяка към текст, който е на экрана.

Работата в режим еcranно редактиране се извършва в следната последователност (т. 1 и т. 2 могат да са и в обратен ред):

1. Натиска се жълтият клавищ **ЛАТ**. Светва жълтата лампичка и клавиатурата е в режим латиница.

2. Натиска се еднократно **ОСВ**. Компютърът е в режим екранно редактиране.

3. При натискане на клавиша **И** указателят се движи наляво, при **К** — надясно, при **И** — нагоре, и при **М** — надолу. Тези четири клавиша са разположени така, че натискането на горния от тях води до движение на указателя нагоре, на долния — надолу, на левия — наляво, на десния — надясно.

4. Чрез еднократно натискане на **ОСВ** се излиза от режим екранно редактиране, при което указателят се пренася от едно на друго място на экрана.

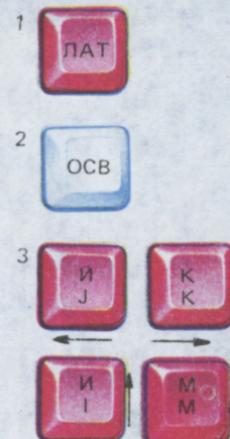
А сега сто какво трябва да направим в нашия пример, за да поправим грешката:

1. Натискаме жълтия клавищ **ЛАТ** — влизаме в режим латиница.

2. Натискаме клавиша **ОСВ** — влизаме в режим екранно редактиране.

3. Натискаме един път **И**, три пъти **И** и указателят е в началото на искания ред.

4. Натискаме **ОСВ** и излизаме от режим екранно редактиране.



А сега да поправим реда:

1. Въвеждаме въпросителен знак.

2. При натиснат клавиш **МК** натискаме





[Y] и [RPT] , с което въвеждаме реда.
3. Натискаме [RETURN] .
Иeto, получаваме:

]? "РАЗСТОЯНИЕТО Е ";50+150;" M"

]? "РАЗСТОЯНИЕТО Е ",50+150," M"

РАЗСТОЯНИЕТО Е 200 M

Искам да ти обърна внимание, че ако поправяме нещо в средата на даден ред, трябва да поставим указателя първо в началото на реда, за да може с [MK] и [Y] да въведем целия ред. При въвеждането поправяме и грешния символ.

И така, скрannото редактиране спестява писане, като се използва вече написано нещо на екрана. Разбира се, вместо всички тези неща можехме отново правилно да напишем сгрешната инструкция, което в някои случаи може да е и по-просто, и по-бързо.

С това за тази вечер свършихме. Вече можеш да въвеждаш всякакъв текст от клавиатурата, да местиш указателя, където пожелаеш по екрана, да отпечатаваш стойността на аритметични изрази и текстове.

ДОМАШНА РАБОТА

1. Да се пресметнат следните аритметични изрази и да се отпечатат резултатите:

- а) $(16 + 28):4;$
- б) $(418 - 173).4 + 32.0;$
- в) $(70:10).6 + 15;$
- г) $38 \cdot 542:7 - 12 \cdot 498;$
- д) $45 \cdot 720:9 + 32 \cdot 640:8.$

2. Да се пресметне и отпечатана стойността на x в уравненията:

- a) $x \cdot 3 = 10\ 000 - 361$;
- б) $3 \cdot x = 120\ 693$;
- в) $134325 + x = 400\ 027$;
- г) $x : 23 = 84\ 235 : 5$;
- д) $4 \cdot x = 1536 - 1492$.

3. Какво отпечатват инструкциите:

- а) ?1;2,3,,4,,5
- б) ?1;2;3
- в) ?1,2;3
- г) ?1;2,3

4. Разстоянието между два града е 400 км. Един автомобил го изминава за 8 часа. Да се пресметне скоростта на автомобила.

5. Да се пресметне колко години са минали от създаването на българската държава.

6. Дълчините на страните на един правоъгълник са 50 см и 20 см.

Да се намери лицето на правоъгълника.

7. Да се пресметне:

- а) $127 \text{ м } 34 \text{ см} + 53 \text{ м } 57 \text{ см}$;
- б) $27 \text{ кг } 850 \text{ г} - 12 \text{ кг } 390 \text{ г}$;
- в) $4 \text{ мин } 30 \text{ сек} - 2 \text{ мин } 15 \text{ сек}$.

8. Да се въведат следните текстове, след което да се заменят надебелените символи съответно с: Ц. *, /, и да се изпълнят инструкциите:

- а) ? Лицето е равно на "5 * 4;" кв.см"
- б) ?37.24 + 173
- в) ?49:7 + 124

9. Да се въведе грешна инструкция и след това да се поправи, като се използва режимът екранно редактиране.

10. Да се решат 5 задачи, аналогични на горните, с помощта на компютъра.

IV урок

КАК СЕ СЪХРАНЯВАТ ДАННИ

Разговорът е за понятието променлива и нейното използване в БЕЙСИК. Въвеждат се чрез примери числови и символни променливи, масиви. Разглежда се инструкцията за присвояване.



Къде компютърът запомня данните, които въеждаме чрез клавиатурата?

В паметта.

Запомнил си. Както говорихме, паметта на компютрите се състои от множество клетки. Те имат свои номера. Транслаторът от БЕЙСИК разполага данните по клетки, запомня номерата им и при необходимост много бързо ги нарина.

Да си представим паметта на Правец-82 като един шкаф, състоящ се от чекмеджета, върху всяко от които е написано какво се съдържа в него, например — "Температура на въздуха в 12 часа" или "Име на носителката на световния рекорд на скок височина".

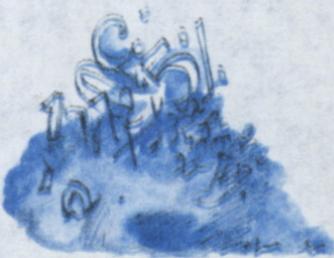
Едно такова чекмедже, в което си мислим, че се съдържа или число, или текст, ще ни служи за представа за това, какво е променлива. Променливата се характеризира с име (представяй си името, което е написано върху чекмеджето) и стойност (числото или текстът, който се намира вътре). В даден момент в едно чекмедже може да има само една стойност, която може да бъде сменявана с друга, когато пожелаем.

Да разгледаме променливите: "Температура на въздуха в 12 часа" и "Име на носителката на световния рекорд на скок височина". Какво трябва да правим, за да има в тях винаги вярна информация?

Всеки ден ще поставяме в чекмеджето с надпис "Температура на въздуха в 12 часа" картонче с числото, показващо измерената в 12 часа температура. В чекмедже с надпис "Име на носителката на световния рекорд на скок височина" ще сложим картонче с името Стефка Костадинова. Ако някоя друга състезателка подобри рекорда, там ще поставим нейното име.

Правилно. А сега да се върнем към компютъра.

Имената на променливите в БЕЙСИК трябва да започват с латинска буква, последвана от латинска буква или цифра. Те могат да бъдат много дълги, но компютърът различава само първите два символа, например имената ABXC



и A_{B1} се разбират от транслатора като едно и също име.

Тогава няма смисъл да използваме имена, по-дълги от два символа.

Прав си, но понякога е удобно да използваме и по-дълги имена на променливи. Така например, ако искаме в една променлива да пазим имена, може да я означим с IME, въпреки че транслаторът ще я помни като IM. Друго нещо, на което искам да ти обърна внимание, е, че първите две букви, с които започват инструкциите, не могат да се използват в същия ред за имена на променливи. Помисли например може ли да има променлива PR?

Не, защото така започва инструкцията PRINT.

Браво. Продължавам. В зависимост от това, какво ще записваме в една променлива, тя бива числови или символни. За да различава транслаторът символните променливи, след името си те имат знака "¤" (който на екрана излиза като \$).

Стойностите на числовите променливи са числа — десетични дроби, а на символните променливи — последователности от символи с дължина от 0 до 255 символа.

А сега няколко упражнения върху това, което говорихме.



Име на каква променлива е A¤, числови или символна?

Символна.

Име на каква променлива е A?

Чисрова.

Име на каква променлива е PR¤?

PR¤ не е име на променлива, защото започва с PR както и инструкцията PRINT.

Име на каква променлива е 1P?

1P не е име на променлива, защото не започва с латинска буква.

Име на каква променлива е FD?

FD не е име на променлива, защото в него има буква от кирилица.

Да видим за какво са нужни променливите. Казахме, че ще ги използваме за въвеждане и извеждане на данни.

Първо ще видим как могат да се въведат данни в една променлива, т. е. как тя може да получи стойност. Това става чрез инструкцията за присвояване, чийто вид е:

The diagram illustrates the structure of the LET statement. It shows the word "LET" at the top, followed by a variable name "A", an assignment operator "=", and the value "5". Arrows point from the text labels below to the corresponding parts of the statement: an arrow points from "думата LET (незадължителна част)" to "LET"; another arrow points from "име на променлива" to "A"; a third arrow points from "знакът за равенство" to "="; and a fourth arrow points from "стойност на променлива" to "5".

думата **LET** (незадължителна част) _____
име на променлива _____
знакът за равенство _____
стойност на променлива _____
(може да е аритметичен израз)

LET A = 5

Знакът за равенство показва, че променливиата A приема стойност 5 или още се казва, че на променливата A се присвоява стойност 5. Това не е математическо равенство и не определя никакво уравнение. При тази инструкция транслаторът извършва следното: ако не е имало до момента чекмедже с име A, на едно празно чекмедже се поставя име A, а в него число 5. Ако вече има чекмедже с име A, вместо числото, което е било в него, се поставя число 5. След изпълнение на тази инструкция вече има създадена променлива с име A и стойност 5.

Стойностите, които се присвояват на символите променливи, се задават оградени с кавички.

Думата LET не е задължителна и обикновено се пропуска.

Да разгледаме няколко примера:

Да се зададе на променливата A стойност 5.

3 A=5

Да се зададе на символната променлива А¤ стойност МИР.

```
] A=5  
] A¤="МИР"
```

Да се провери какво има в променливите А и А¤ от предишните две задачи, като се разпечатат стойностите им.

Нали се сещаш, че ще използваме инструкцията PRINT.

```
] A=5  
] A¤="МИР"  
] ?A
```

```
] A=5  
] A¤="МИР"  
] ?A  
5
```

```
] A=5  
] A¤="МИР"  
] ?A  
5  
] ?A¤
```

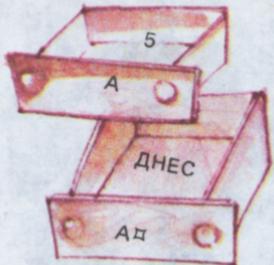
```
] A=5  
] A¤="МИР"  
] ?A  
5  
] ?A¤  
МИР
```

Когато пишеш имена на променливи, не забравяй, че те съдържат латински букви и въпреки че някои кирилски и латински букви излизат на екрана по един и същ начин, те за транслатора са различни.

Да се зададе стойност ДНЕС на променливата А¤.

```
] A¤="ДНЕС"
```

Да се прочете стойността на променливата A.



] A=“ДНЕС”

] ?A

] A=“ДНЕС”

] ?A

ДНЕС

Мисля, че ти стана ясно, че като се зададе нова стойност, съдържанието на променливата се сменя и старата стойност се изгубва.

Да се зададе стойност 5 на променливата X и да се отпечата произведението на числото 2 и стойността на променливата X.

] X=5

] ?2*X

] X=5

] ?2*X

10

В тази задача използваме за първи път аритметичен израз, в който освен числа и знаци за операции има и променливи. При пресмятане на израза се вземат стойностите на променливите.

Какво отпечатва следната поредица от инструкции:

] X=5

] Y=3

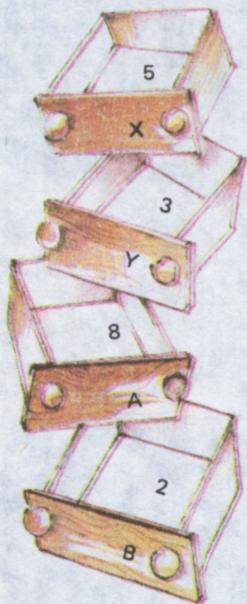
] A=X+Y

] B=X-Y

] ?X,Y,A,B

Набери и виж.

Ето какво се получи:



] X=5
] Y=3
] A=X+Y
] B=X-Y
] ?X,Y,A,B

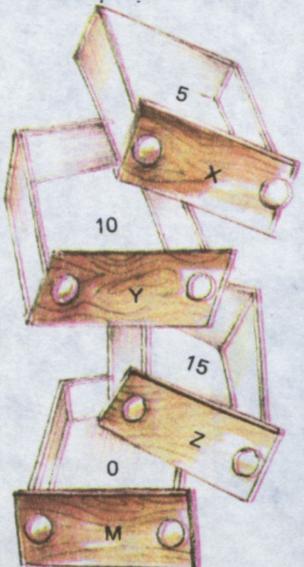
] X=5
] Y=3
] A=X+Y
] B=X-Y
] ?X,Y,A,B
5 3 8
2

Нали разбра какво извършихме. На променливата X дадохме стойност 5, на променливата Y — стойност 3, на променливата A — стойност сумата от стойностите в променливите X и Y ($5 + 3 = 8$), на променливата B — разликата от стойностите в променливите X и Y ($5 - 3 = 2$). Накрая отпечатахме стойностите на четирите променливи ($X = 5$, $Y = 3$, $A = 8$, $B = 2$).

Обърни внимание на следното: ако на дадена променлива не сме присвоили стойност чрез оператора за присвояване и я използваме, транслаторът отделя за нея ново място, като ѝ се дава стойност нула, ако е числова, и текст с дължина нула, ако е символна.

Ето една такава задача:

Какъв е резултатът от изпълнението на следните инструкции:



] X=5
] Y=10
] Z=X+Y+M
] ?X,Y,Z,M

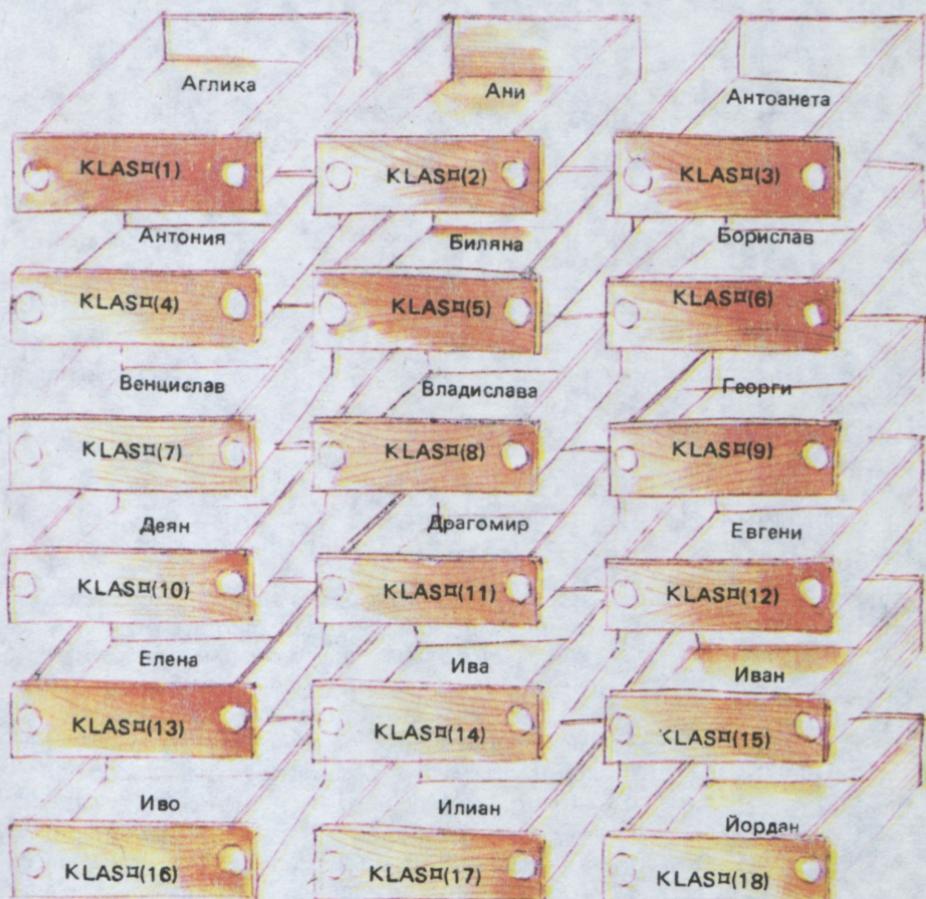
] X=5
] Y=10
] Z=X+Y+M
] ?X,Y,Z,M

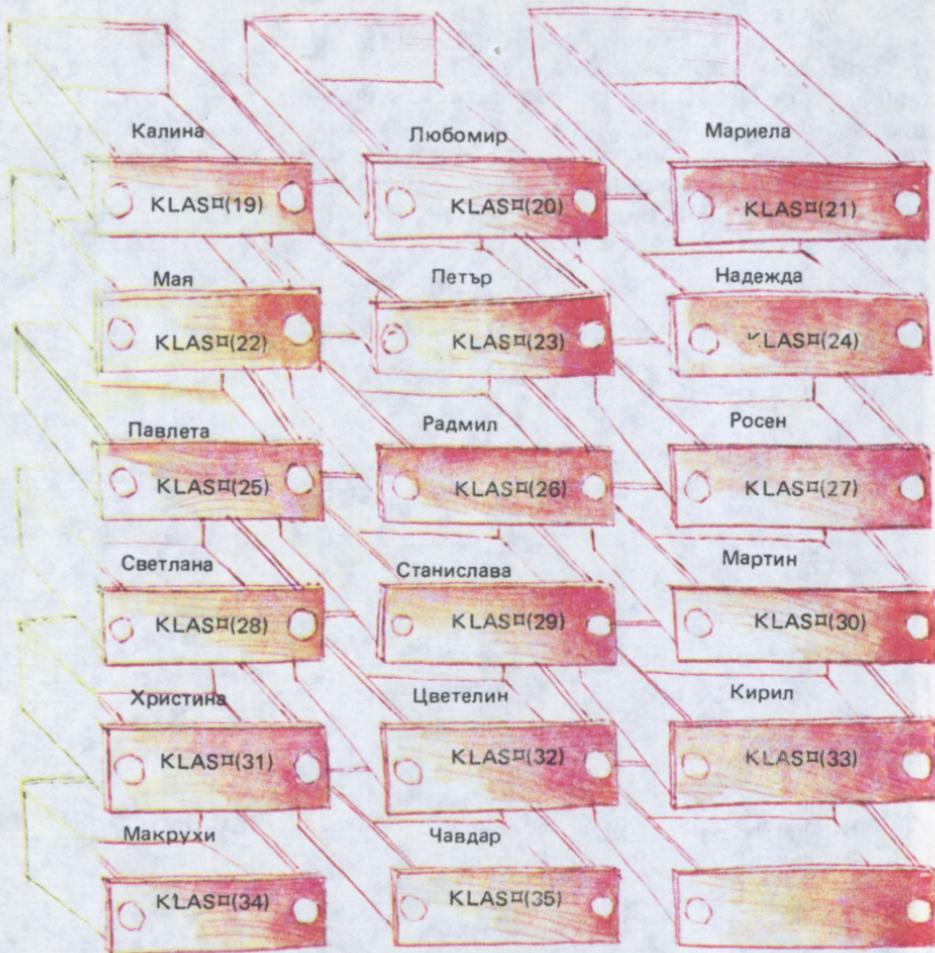
Ще проверя.

] X=5
] Y=10
] Z=X+Y+M
] ?X,Y,Z,M
5 10 15
0

Добре. Сега си почини една минута, след кое-
то искам да внимаваш много.

Представи си, че имаме 35 чекмеджета едно
до друго. В тях са поставени имената на учени-
ците от твоя клас: Аглика, Ани, ... Чавдар.
Нека всички чекмеджета заедно носят името
KLAS \square . За име на всяко едно чекмедже поста-
вяме: KLAS $\square(1)$, KLAS $\square(2)$, ... KLAS $\square(35)$.





Съвкупността от еднотипни променливи (само числови или само символни), означени с едно и също име, ще наричаме масив, а отделните променливи, от които той се състои — негови елементи. Елементите се означават с името на масива, последвано от поредния номер на элемента в масива, поставен в скоби.

Само за сведение ще ти спомена един факт, който не ти е нужен, за да разбереш същността на понятието масив, и поради това не искам да го вземаш под внимание. Това е фактът, че масивите в БЕЙСИК имат нулев елемент, т. е. първата променлива носи номер 0, а не 1.

Преминаваме нататък. Въвеждането на стойности на елементите на един масив става чрез инструкция за присвояване.

Да видим на практика как става това.

На елементите на масив A с пет елемента да се присвоят съответно стойностите от 1 до 5 и да се провери правилно ли е извършено присвояването чрез отпечатване на стойностите на елементите на масива.

```
] A(1)=1  
] A(2)=2  
] A(3)=3  
] A(4)=4  
] A(5)=5  
] ?A(1),A(2),A(3),A(4),A(5)
```

```
] A(1)=1  
] A(2)=2  
] A(3)=3  
] A(4)=4  
] A(5)=5  
] ?A(1),A(2),A(3),A(4),A(5)  
1           2           3  
4           5
```

Добре. И какво представляват тези числа 1, 2, 3, 4, 5?

Стойности на елементите на масива.

С това за днес завършваме. Разгледахме понятията променлива и масив — две основни понятия в програмирането. В програмирането ще си служим вместо с конкретни числа с имената на променливите, а компютърът ще извършва указаните в програмите действия с техните стойности. Създаваните по този начин програми могат да се изпълняват всеки път с различни данни.

А сега домашното.

ДОМАШНА РАБОТА

1. Кога една променлива получава стойност?
2. Инструкция за присвояване ли е:

| LET A=5КАМИОНА

3. Защо следните имена не са имена на променливи:
 - a) LE \sqcap
 - б) ЗA
 - в) A1 + 2
 - г) АГ \sqcap
 - д) ГАА
4. Един и същ резултат ли ще се получи при:

a)

] A=B+C/D-E

] A=(B+C)/(D-E)

б)

] A=(B*C)*D

] A=B*C*D

Да се сверят резултатите, като се зададат стойности на B, C, D и E.

5. Какво се отпечатва с инструкциите:

] LET D=15*4

] ?15,4,D

] A=5

] B=8

] ?A+B+K,K

6. Какво се отпечатва с инструкциите:

]A=5
]B=8
]?A+53)*2-B*2

]A=5
]B="ДНИ"
]?A;B

7. Инструкция за присвояване ли е

] A+B=C

и защо?

8. Да се попълни масив с десет елемента KLAS и с десет имена и да се разпечатва.

Разговорът е за същността на писането на програми. Изяснява се понятието алгоритъм. Дават се кратки сведения за езиците за програмиране. Описват се етапите при създаване на една програма.



Днес ще говорим за това, как се пишат програми и ще оставим Правец-82 да си почине.

Досега използвахме компютъра, като въвеждахме инструкции на БЕЙСИК в директен режим.

Какво означава директен режим?

При директния режим всяка инструкция, която въвеждаме с натискането на **[RETURN]**, се изпълнява веднага. Този режим се използва за несложни пресмятания. Освен в такъв режим може да се работи и в програмен режим.

В такъв режим сигурно се пишат истински програми.

Точно така.

Татко, а как се пишат истински програми?

За това ще говорим тази вечер. Програмата представлява описание на алгоритъм, който искаеме компютърът да изпълни, зададено на език, който той разбира, в случая — БЕЙСИК.

Има ли нещо въкъщи, което работи по зададена програма?

Пералнята.

Правилно. Например програмата за пране на цветно пране представлява описание на алгоритъм, който последователно води до изпълняване на: първо пране при температура на перилния разтвор 30°C , второ пране при температура 60°C , 4 изплаквания с кратки центрофугирания, последно 6-минутно центрофугиране. За да разбере пералната машина, че искаем да извърши точно тези процедури и в този ред, ние ѝ задаваме описание на алгоритъма, по който да действа на език, който тя разбира, а именно завъртваме кръга с букви до буквата А и натискаме клавиша **[2]**. Кръгът се завърта след изпълнението на всяка процедура до следващата означена върху него буква и по този начин се задава следната последователност от действия: А, Б, В, Г, Д.

Не съм предполагал, че има нещо общо между пералнята и компютъра.

А защо не питаш какво е алгоритъм?

Зашото съм чувал тази дума.

А какво означава?



Не знам.

Алгоритъмът представлява последователност от елементарни за изпълнителя действия, осигуряващи решаването на дадена задача.

Кои действия са елементарни за изпълнителя?

Това зависи от възможностите на изпълнителя, ако е човек — възможностите му са едни, ако е компютър — други. Зависи и от самия човек или от самия компютър.

Ще разгледаме няколко примера за съставяне на алгоритми.

Да се опише алгоритъм за пържене на три яйца.

Ето един алгоритъм:

- 1 действие — слагаме масло в тигана.
- 2 действие — слагаме тигана на печката.
- 3 действие — включваме печката.
- 4 действие — чакаме 2 минути.
- 5 действие — счупваме 3 яйца.
- 6 действие — изсипваме яйцата в тигана.
- 7 действие — чакаме 5 минути.
- 8 действие — изключваме печката.
- 9 действие — изсипваме яйцата в чиния.

Добър ли е алгоритъмът?

Как ще е добър, като не си сложил сол?

Кажи кога се слага сол и ще добавя едно действие.

Е, не е важно. И така става.

Добре. Дай да видим дали няма да открием алгоритми в някой от твоите учебници.

Мисля, че в учебника по "Труд и творчество" може нещо да се намери. Ето ти един алгоритъм за направата на косичник.

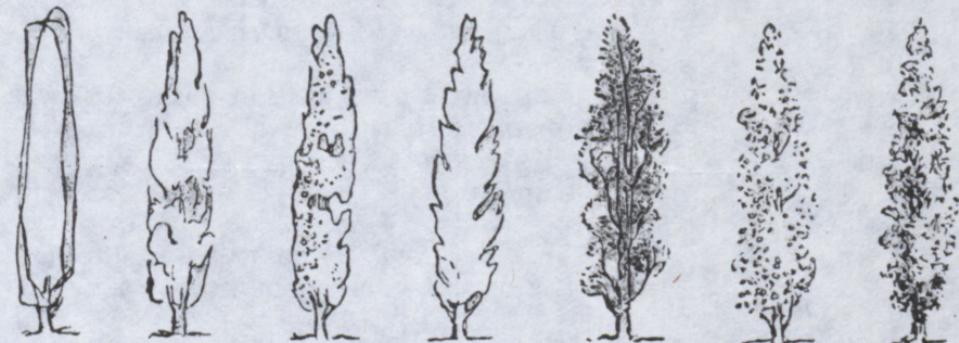
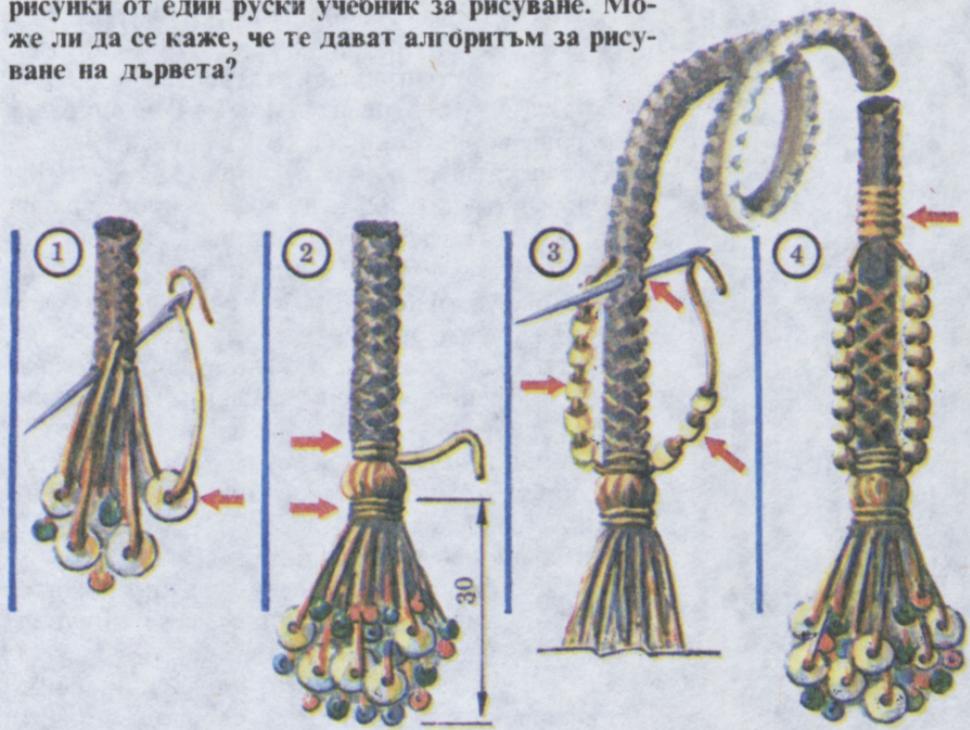
Чудесно. Наистина в този учебник има дадени доста алгоритми. А спомняш ли си някакъв алгоритъм, който съм ти давал аз преди няколко дни?

Не си спомням.

А последователността от действия за пускане на компютъра какво е?



Да, наистина. Сещам се нещо друго. Я виж тези рисунки от един руски учебник за рисуване. Може ли да се каже, че те дават алгоритъм за рисуване на дървета?



Да. Дори много ми харесва примерът.

А сега да съставим алгоритъм за решаване на една математическа задача.

Да се състави алгоритъм за сравняване стойностите на две променливи A и B и отпечатване на $A > B$, ако стойността на A е по-голяма от B , $A = B$, ако стойностите на A и B са равни и $A < B$, ако стойността на A е по-малка от тази на B .

Ето и алгоритъмът:

1. Въвеждат се стойностите на А и В.
2. Ако $A > B$, отпечатваме $A > B$ и спираме.
3. Ако $A = B$, отпечатваме $A = B$ и спираме.
4. Ако това не е изпълнено, изпълняваме 4.

Ако това не е изпълнено, изпълняваме 4.

При действие 4 няма нужда да проверяваме дали $A < B$, защото при действие 2 сме видели, че условието $A > B$ не е изпълнено, а при действие 3 — че условието $A = B$ не е изпълнено, т. е. останала е само възможността $A < B$.

А сега да се върнем към компютрите. Искаме те да ни решат някаква задача. Първо, ние трябва да имаме алгоритъм за решаването ѝ. След това трябва да можем да го опишем на език, който компютърът разбира, т. е. за който той има транслатор.

А има ли много такива езици?

Има създадени много езици за програмиране. За кой език за програмиране вече си чувал?

Нали ме учиш на БЕЙСИК?

Да, извинявай, че те питам очевидни неща. Проверявам дали добре ме разбираш.

БЕЙСИК е създаден през 1964 г. и е твърде прост за изучаване и използване. С транслатор от него са снабдени почти всички микрокомпютри.

Други разпространени езици, които могат да се разбират от Правец-82, са:

ФОРТРАН — създаден през 1956 г. и предназначен предимно за научни и технически задачи;

ПАСКАЛ — създаден през 1971 г. и предназначен за описание на широк клас от алгоритми;

ЛОГО — създаден през 1968 г. и подходящ за обучение на деца.

Това означава ли, че ние можем да пишем програми и на ФОРТРАН, и на ПАСКАЛ, и на ЛОГО?

Да. Но за тях — додатък. А сега ще се опитам да ти дам алгоритъм за писане на програми.

А като го науча, ще мога ли да програмирам?

Не. Не е така просто. Сега ще видим какви действия трябва да се извършат, а за тяхната същност ще говорим в следващите уроци.

Ето и самият алгоритъм за писане на програми.

1. Изясняване условията на задачата, която решаваме — кои данни са известни и кои търсим.
2. Съставяне на алгоритъм, с който се решава задачата.
3. Описване на алгоритъма (програмиране) на конкретен език за програмиране.
4. Въвеждане на създадената програма в компютъра.
5. Отстраняване на евентуални синтактични грешки. (Транслаторът ни помага, като ги показва.)
6. Изprobване на програмата с данни от прости примери, за които знаем предварително резултата. Ако отговорът не се получи, се връщаме на действие 3 — ако грешката е при написване на програмата, или на действие 2 — ако грешката е в самия алгоритъм.
7. Изпълняване на програмата с истински данни.

С това завършваме.

Тази вечер разглеждахме много важни неща от програмирането.

А сега да ти дам домашното.

ДОМАШНА РАБОТА

1. Да се повторят действията, които трябва да се извършват при писане на програми.
2. Да се състави алгоритъм за миене на зъби.
3. Да се състави алгоритъм за лъскане на обувки.
4. Да се състави алгоритъм за намиране скоростта на автомобил по зададени път и време.
5. Да се състави алгоритъм за намиране на пътя, който изминава автомобил по дадени скорост и време.
6. Да се състави алгоритъм за намиране периметъра на един триъгълник по дълчините на три страни.
7. Да се състави алгоритъм за намиране периметъра на квадрат по дължината на страна.
8. Да се състави алгоритъм за намиране лицето на един квадрат по дължината на страна.
9. Да се състави алгоритъм за сравняване стойността на променливиата A с числото 100 и отпечатване $A > 100$, ако стойността на A е по-голяма от 100, $A < 100$, ако стойността на A е по-малка от 100, и $A = 100$, ако стойността на A е равна на 100.
10. Да се формулира една задача и да се състави алгоритъм за решаването ѝ.

VI урок

КАК СЕ ВЪВЕЖДАТ, ИЗПЪЛНЯВАТ И СЪХРАНЯВАТ ПРОГРАМИ

Разговорът е за начините за въвеждане, изпълняване и съхраняване на програми. Разглежда се понятието номер на ред в езика БЕЙСИК. Въвеждат се инструкциите HOME, END, LIST, RUN, DEL, NEW. Представя се начин за съхраняване на програми върху дискети и касети, като се разглежда програмата СТРЕЛБА. Усвояват се инструкциите SAVE, RUN и CATALOG.



Днес ще започнем да въвеждаме и изпълняваме инструкции в програмен режим.

VI вечер

Ще пишем програми?

Да. Програмата на БЕЙСИК за Правец-82 се представя в редове. Всеки ред започва с номер. Номерът е цяло число между 0 и 63999 и се записва непосредствено след напомняния знак. Транслаторът автоматично подрежда редовете по нарастващ ред на номерата.

Ще започнем с инструкцията HOME, която има вида:

думата HOME

HOME
↓

С тази инструкция се изчиства еcranът. Опитай.

Добре.

Сега ще ти покажа програма за отпечатване на първите четири реда от следния куплет от стихотворението „Хаджи Димитър“ на Христо Ботев.

Настане вечер,
месец изгрее,
звезди обсипят
свода небесен;
гора зашуми,
вятър повее —
Балканът пее
хайдушка песен!

```
10 PRINT "НАСТАНЕ ВЕЧЕР,"  
20 PRINT "МЕСЕЦ ИЗГРЕЕ,"  
30 PRINT "ЗВЕЗДИ ОБСИПЯТ"  
40 PRINT "СВОДА НЕБЕСЕН;"  
50 END
```

Да я разгледаме внимателно. Първо кои са номерата на редовете?

Мисля, че са 10, 20, 30, 40, 50. Защо са през 10?

При първоначално писане на програми за номера на редовете се избират непоследователни числа, например нарастващи с 10, за да има

място, ако сме забравили нещо, да го вмъкнем след това.

А сега да видим каква програма се е въвела във вътрешната памет: Това става чрез инструкцията LIST, чийто вид е:

думата LIST —————
два номера, разделени със запетая —————
(един номер или нищо)

LIST 10,50

Инструкцията LIST отпечатва програмата, която е във вътрешната памет. При задаване на два номера на редове (вторият по-голям от първия) тя отпечатва частта на програмата между тях (включително и тях), при задаване на един номер отпечатва само този ред, а ако не се зададе номер, отпечатва цялата програма.

Хайде да се уверим, че е така:

Какво отпечатва LIST?

] LIST

] LIST

10 PRINT "НАСТАНЕ ВЕЧЕР,"
20 PRINT "МЕСЕЦ ИЗГРЕЕ,"
30 PRINT "ЗВЕЗДИ ОБСИПЛЯТ"
40 PRINT "СВОДА НЕБЕСЕН;"
50 END

Какво отпечатва LIST 10, 40?

] LIST 10,40

] LIST 10,40

```
10 PRINT "НАСТАНЕ ВЕЧЕР,"  
20 PRINT "МЕСЕЦ ИЗГРЕЕ,"  
30 PRINT "ЗВЕЗДИ ОБСИПЯТ"  
40 PRINT "СВОДА НЕБЕСЕН;"
```

Последният ред не се отпечатва.
Какво отпечатва LIST 10?

] LIST 10

] LIST 10

```
10 PRINT "НАСТАНЕ ВЕЧЕР,"
```

Отпечатата се съдържанието само на първия ред.
Нещо неясно да виждаш в програмата?
Какво прави това END?

Това е инструкцията END, чийто вид е:

думата END —

END
↑

Тя служи за означаване края на програмата.
Може и да липсва в програмите.

Въвеждането на редовете може да става в
произволен ред. Транслаторът има грижа да ги
подреди по номерата им.

А как се изпълнява една програма?

Това става с инструкцията RUN, чийто вид
е:

RUN

думата RUN —

Инструкцията задава изпълняването на програмата, която се намира във вътрешната памет. Да изпълним програмата, която въведохме:

] RUN

] RUN

НАСТАНЕ ВЕЧЕР,
МЕСЕЦ ИЗГРЕЕ,
ЗВЕЗДИ ОБСИПЯТ
СВОДА НЕБЕСЕН;

А сега добави и останалата част на куплета.
Въвеждането на нов ред със съществуващ номер предизвиква изтриване на стария ред. Започни с ред, чийто номер е 50.

50 PRINT "ГОРА ЗАШУМИ,"
60 PRINT "ВЯТЪР ПОВЕЕ—"
70 PRINT "БАЛКАНЪТ ПЕЕ"
80 PRINT "ХАЙДУШКА ПЕСЕН!"
90 END

Изпълни програмата.

] RUN

] RUN
НАСТАНЕ ВЕЧЕР,
МЕСЕЦ ИЗГРЕЕ,
ЗВЕЗДИ ОБСИПЯТ
СВОДА НЕБЕСЕН;
ГОРА ЗАШУМИ,
ВЯТЪР ПОВЕЕ –
БАЛКАНЪТ ПЕЕ
ХАЙДУШКА ПЕСЕН!

А сега да изтрием първите пет реда. Изтриване на един ред става чрез въвеждане на номера му, т. е. със същия номер записваме празен ред.
Изтрий първия ред на стихотворението и виж какво е останало.

] 10
] LIST

] 10
] LIST
20 PRINT "МЕСЕЦ ИЗГРЕЕ,"
30 PRINT "ЗВЕЗДИ ОБСИПЯТ"
40 PRINT "СВОДА НЕБЕСЕН;"
50 PRINT "ГОРА ЗАШУМИ;"
60 PRINT "ВЯТЪР ПОВЕЕ –"
70 PRINT "БАЛКАНЪТ ПЕЕ"
80 PRINT "ХАЙДУШКА ПЕСЕН!"
90 END

Изтриването на един или няколко реда може да се извърши и с инструкцията DEL, чийто вид е:

думата **DEL**

номера на редове, разделени със запетая

DEL 20,50



Инструкцията е за премахване във вътрешната памет на всички редове, чиито номера сме задали, и редовете между тях. А сега да изтрием от втория до петия ред и да видим какво е останало от програмата.

] DEL 20,50

] LIST

] DEL 20,50

] LIST

60 PRINT "ВЯТЪР ПОВЕЕ -"

70 PRINT "БАЛКАНЪТ ПЕЕ"

80 PRINT "ХАЙДУШКА ПЕСЕН!"

90 END

Да разгледаме още една инструкция — NEW. Нейният вид е:

думата **NEW**

NEW



Инструкцията е за изтриване на цялата програма, която се намира във въгрешната памет. Провери действието на инструкцията NEW.

] NEW

] LIST

] NEW

] LIST



А сега да въведем програмата на една игра.

Най-после игра.

Ще ти дам програмата наготово. Ти ще я въведеш сам, като не забравяш, че нула и буквата О не трябва да се бъркат и че променливите се състоят от латински букви.

Да се въведе програмата на играта СТРЕЛБА

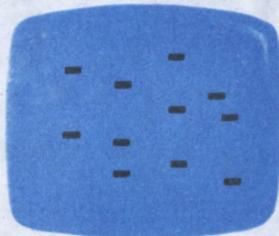
```
10 DIM A(10),B(10)
20 GR
30 COLOR=15
40 FOR X=0 TO 39 STEP 2
50 PLOT X,0
60 NEXT X
70 FOR Y=0 TO 39 STEP 2
80 PLOT 0,Y
90 NEXT Y
100 FOR I=1 TO 10
110 K=INT (39 * RND (1))+1
120 T=INT (39 * RND (1))+1
130 A(I)=K
140 B(I)=T
150 PLOT K,T
160 NEXT I
170 S=0
180 FOR I=1 TO 10
190 INPUT X,Y
200 FOR J=1 TO 10
210 IF A(J)=X AND B(J)=Y THEN GOTO 270
220 NEXT J
230 NEXT I
240 HOME
250 PRINT S;" ПОПАДЕНИЯ"
260 END
270 S=S+1
280 COLOR=0
290 PLOT X,Y
300 COLOR=15
310 FOR K=1 TO 50
320 W=PEEK (-16336)
330 NEXT K
340 GOTO 220
```

Въведох програмата. А сега какво?

Изпълни я.

**С RUN, нали? На екрана се появиха точки.
Какво да правя с тях?**

Трябва да стреляш по тях. Екранът на монитора е разделен на 40×40 точки, които се посочват чрез две числа, първото — номера на точката в реда, а второто — номера на реда.



0,0	1,0	2,0	3,0	4,0		39,0
0,1	1,1	2,1	3,1	4,1		39,1
0,2	1,2	2,2	3,2	4,2		39,2
0,3	1,3	2,3	3,3	4,3		
0,4	1,4	2,4	3,4	4,4		
0,39	1,39	2,39	3,39	4,39		39,39

Имаш право на 10 удара, като при всеки удар задаваш точката чрез двете числа след въпросителния знак и натискаш **RETURN**.

Броят на точките, по които стреляш, е 10. Може понякога на екрана да се появят и по-малко точки, ако някои съвпадат. При улучване на дадена точка тя се маха от екрана. След завършване на играта се появява резултатът.

А сега мога ли да поиграя?

По-добре утре. Сега нека запазим играта на една дискета, така че утре да не е необходимо отново да я въвеждаш. Ще трябва само да я прехвърлиш от дискетата във вътрешната памет и там ще я изпълниш.

Това много ми харесва, да пазя програмите върху дискети.

Да вземем една дискета. Тя трябва предварително да е подгответа, за да може върху нея да се записва.

А как става това?

Засега това ще го правя аз. От теб се иска да прехвърлиш програмата върху дискетата, като използваш инструкцията **SAVE**:

SAVE СТРЕЛБА

думата **SAVE** —

име, под което ще бъде записана
програмата върху дискетата —

С инструкцията се запазва програмата върху дискетата под зададеното име.

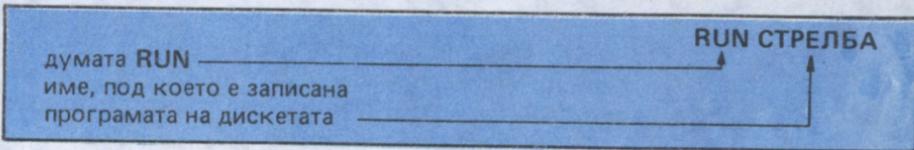
Да се запази програмата, която се намира във вътрешната памет под името СТРЕЛБА.

Поставяме дискетата във флопидисковото устройство.

Задаваме инструкцията: SAVE СТРЕЛБА
Но СТРЕЛБА се състои от кирилски букви.
Името, под което ще се запише програмата, може да включва произволни символи.

Светва лампичката на флопидисковото устройство, което означава, че то работи. Появява се указателят на следващия ред. Това показва, че записът е готов.

Да разгледаме и в един друг вид инструкцията RUN.



В този вид инструкцията служи за въвеждане на програмата от флопидисковото устройство във вътрешната памет и за изпълняването ѝ.

Да я използваме, за да проверим дали сме успели да запазим програмата СТРЕЛБА върху дискетата.

Започваме с изчистване на вътрешната памет и екрана.

] NEW
] HOME
] LIST

] ■
Нищо не излиза на екрана.
Така. Програмата е изтрита от вътрешната памет. Сега я прочети от дискетата.

] RUN СТРЕЛБА

На екрана са отново точките. Да си поиграя ли?

Не. Прекъсни играта. Това ставаше как?
Не помня.



С червения клавиши

RST

Вярно.

А сега да видим какво има във вътрешната памет.

LIST. Появи се програмата.

Да, защото отново е в паметта.

Сигурно трябва да запомня името на програмата, за да мога да я пусна утре.

Няма нужда. Може всеки момент да се направи справка какво има записано върху дискетата. Това става с инструкцията CATALOG, чийто вид е:

CATALOG

думата CATALOG

CATALOG

CATALOG
DISK VOLUME 254

A 002 ЗДРАВЕЙ
A 003 СТРЕЛБА

А какво е това ЗДРАВЕЙ?

Това е името на програмата, която записах при подготовката на дискетата за работа. Буквата А означава, че програмите са на БЕЙСИК, а 002 и 003 — показват дълчините им върху дискетата.

Татко, а с касетофон как се записват програми?

Поставяш касетата в касетофона и я нагласяваш на мястото, където искаш да записваш. Набираш инструкцията **SAVE**, без да указаваш име. Проверяваш силата на звука да е около или малко над средната. Включваш касетофона на запис. Натискаш **[RETURN]**. Указателят изчезва. След 10—15 секунди се чува кратък звук, който е указание за започване на записването. Когато записът свърши, се чува отново кратък звук и указателят се появява отново на екрана.

Много е сложно. Добре, че си имаме флотидско устройство.

Толкова за днес. Вече знаеш как да въвеждаш програми в компютъра, да ги редактираш, изпълняваш и съхраняваш върху лискиети.

ДОМАШНА РАБОТА

1. Проверете на практика има ли значение в каква последователност се употребяват инструкциите **NEW** и **HOME**.

2. Да се въведе първият куплет на стихотворението Асамблея "Знаме на мира" от Цветан Ангелов като програма, която го отпечатва.

3. Да се добави към програмата от задача 2 вторият куплет на стихотворението.

4. Да се изтрие тази част от програмата от задача 2 и 3, която отпечатва първия куплет на стихотворението.

5. Да се въведе отново отпечатване на първия куплет на стихотворението към програмата от задача 2, 3 и 4 и да се изпълни програмата.

6. Да се състави програма за пресмятане и отпечатване на сумата на числата 10 и 15 и да се запише на дискета под име **A**.

7. Да се изпълни програмата **A** от задача 6.

8. Да се прочетат имената на програмите, записани върху дискетата.

9. Да се изпробва какво се получава, ако върху дискетата се запише нова програма с вече използвано име.

10. Да се упражни играта **СТРЕЛБА**.



АСАМБЛЕЯ
„ЗНАМЕ НА МИРА“

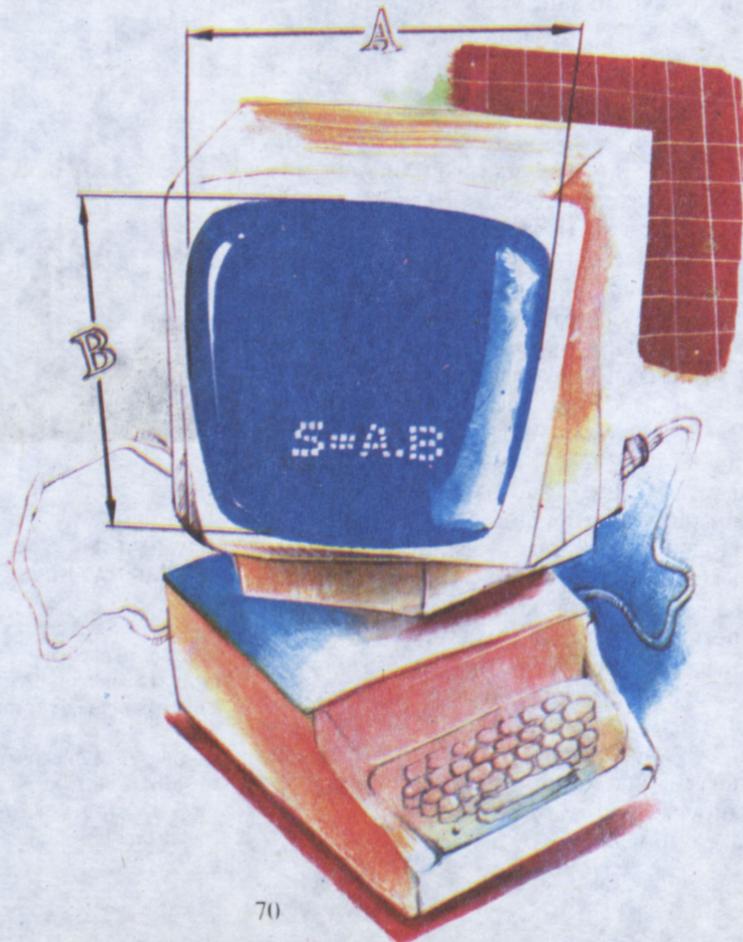
Коя е светлата страна,
събрала пъстри знамена
от двете земни полуния?
Димитровска България!

И кой е този дивен кът,
където песните летят
с чавдарчето и комсомолеца?
Любимата ни столица!

VII урок

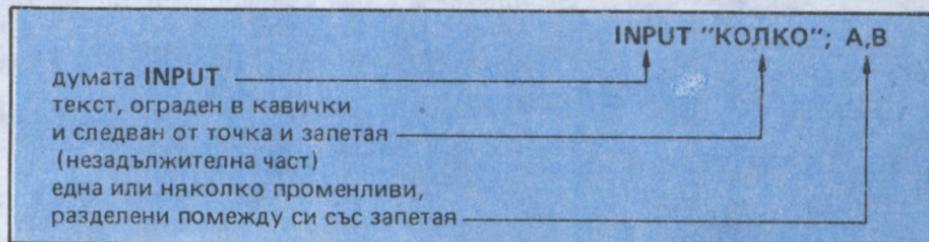
КАК СЕ ВЪВЕЖДАТ ЧИСЛА И ТЕКСТ ПО ВРЕМЕ НА ИЗПЪЛНЕНИЕ НА ПРОГРАМАТА

Разговорът е за въвеждане на числови и текстови стойности на променливи по време на изпълнение на програмата. Описва се инструкцията INPUT. Разглеждат се множество задачи за съставяне на програми с помощта на въведените досега инструкции, като например задача за намиране лице на правоъгълник, задача за преобразуване от една мерна единица в друга, задача за решаване на уравнението $X:A = B$.



За всички програми, които написахме досега, данните се указваха направо в програмите. Сега ще видим как програмата може да работи с данни, задавани по време на изпълнението ѝ от клавиатурата.

За целта се използва инструкцията INPUT, която има вида:



Инструкцията се използва за въвеждане на стойности на променливи по време на изпълнение на програмата.

При изпълнение на тази инструкция програмата спира работата си и на екрана се появява текстът, който сме задали. Ако не сме задали текст, появява се въпросителен знак. Ние трябва да въведем стойностите на променливите, разделени помежду си със запетая.

Да се състави програма, която въвежда едно число и го отпечатва.

Къде ще въведем числото?

В никакво чекмедже.

Използвай занапред думата променлива. Явно вече си разбрал за какво се използват променливите. Сега да решим задачата:

```
10 INPUT A  
20 PRINT A  
30 END
```

И да я изпълним:

```
10 INPUT A  
20 PRINT A  
30 END  
] RUN
```

```
10 INPUT A  
20 PRINT A  
30 END  
] RUN
```

?

Какво става? Появи се въпросителен знак.
Задай някаква стойност, например 10, и виж какво отпечатва програмата.

```
] RUN
```

```
] RUN
```

?

```
] RUN
```

? 10

```
] RUN  
? 10  
10
```

А сега изпълни програмата с друга стойност, например 20.

```
] RUN
```

```
] RUN
```

?

```
] RUN  
? 20
```

```
] RUN  
? 20  
20
```

Да се състави програма, която въвежда едно число и отпечатва числото минус 1, самото число и числото плюс 1.

```
10 INPUT A  
20 PRINT A-1,A,A+1  
30 END
```

А сега да проверим дали правилно работи програмата с примери. Да опитаме последователно например с числата 5000 и 5397.

] RUN

] RUN

?

] RUN

? 5000

] RUN

] RUN

?

] RUN

? 5397

]RUN

?5000

4999

5000

5001

5396

5397

5398

Добре. Започваме да пишем пълноценнни програми.

Да се състави програма, с която по дадени две страни на правоъгълник се намира лицето му.

Когато съставим тази програма, ще можеш да решаваш различни задачи от учебника по математика със същото това условие и с различни данни.

Преди да започнем, да си спомним за етапите на писане на програми и да ги следваме.

Първо — какви данни са ни известни и какво търсим?

Известни са ни дълчините на страните на правоъгълника. Търсим лицето му.

Нека дълчините на страните означим с A и B, а лицето с S. Дадени са A и B. Търсим S.

Да преминем към съставяне на алгоритъм за решаване на задачата:

1. Въвеждат се стойности на A и B.
2. Пресмята се $S = A \cdot B$.
3. Отпечатва се S.
4. Край.

Като знаем алгоритъма, написването на програмата е много лесно:

```
10 INPUT A,B  
20 S=A*B  
30 PRINT S  
40 END
```

Да наберем програмата и да видим има ли синтактични грешки.

Няма.

Тогава да я изprobваме с прости примери. Нека дълчините на страните са 2 см и 3 см.

] RUN

] RUN

] RUN

] RUN

? 2,3

6

Смята правилно.

Още един пример — със 7 см и 3 см.

] RUN

] RUN

] RUN

?

?

?

] RUN

? 7,3

21

Добре. А сега да разширим възможностите на програмата. Нека тя да пита за дълчините на страните на правоъгълника, а и когато отпечатва, да казва какво отпечатва.

Не е трудно.

Това ще направим така: първо, в инструкцията INPUT ще зададем текста "ДЪЛЖИНИ НА СТРАНИТЕ?" и, второ, в инструкцията PRINT ще добавим и текста "ЛИЦЕТО Е РАВНО НА "

```
10 INPUT "ДЪЛЖИНИ НА СТРАНИТЕ ";A,B  
20 S=A*B  
30 PRINT "ЛИЦЕТО Е РАВНО НА ";S  
40 END
```

Изпълни програмата с A = 2, B = 5.

] RUN

] RUN

ДЪЛЖИНИ НА СТРАНИТЕ

] RUN

ДЪЛЖИНИ НА СТРАНИТЕ 2,5

] RUN

ДЪЛЖИНИ НА СТРАНИТЕ 2,5

ЛИЦЕТО Е РАВНО НА 10

Въведи в началото на програмата още една инструкция, която да отпечатва каква задача се решава, например "ЛИЦЕ НА ПРАВОЪГЪЛНИК", и разпечатай цялата програма.

] 5 PRINT "ЛИЦЕ НА ПРАВОЪГЪЛНИК"

] LIST

] LIST

5 PRINT "ЛИЦЕ НА ПРАВОЪГЪЛНИК"

10 INPUT "ДЪЛЖИНИ НА СТРАНИТЕ ";A,B

20 S=A*B

30 PRINT "ЛИЦЕТО Е РАВНО НА ";S

40 END

Изпълни програмата за правоъгълник с дължина на страните 50 см и 70 см.

] RUN

] RUN

ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ

] RUN

ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 50,70

] RUN

ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 50,70
ЛИЦЕТО Е РАВНО НА 3500

Още малко да допълним програмата. Нека да чете в какви мерни единици са страните на правоъгълника и съответно да отпечатва в какви единици се получават резултатите.

```
15 INPUT "МЕРНА ЕДИНИЦА "; X□  
30 PRINT "ЛИЦЕТО Е РАВНО НА "; S;  
" KB. "; X□
```

Разпечатай програмата и я разгледай добре.

] LIST

] LIST

```
5 PRINT "ЛИЦЕ НА ПРАВОЪГЪЛНИК"  
10 INPUT "ДЪЛЖИНИ НА СТРАНИТЕ "; A,B  
15 INPUT "МЕРНА ЕДИНИЦА "; X□  
20 S=A*B  
30 PRINT "ЛИЦЕТО Е РАВНО НА "; S;  
" KB. "; X□  
40 END
```

Ясно ли ти е всичко в нея?

Да. Дай да я изпълни с 2 см и 3 см.

3 RUN

3 RUN
ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ

3 RUN
ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 2,3

3 RUN
ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 2,3
МЕРНА ЕДИНИЦА

3 RUN
ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 2,3
МЕРНА ЕДИНИЦА СМ

3 RUN
ЛИЦЕ НА ПРАВОЪГЪЛНИК
ДЪЛЖИНИ НА СТРАНИТЕ 2,3
МЕРНА ЕДИНИЦА СМ
ЛИЦЕТО Е РАВНО НА 6 КВ.СМ

Тази програма вече ми харесва.

А сега следващата задача. Преди това запиши върху дискетата последната програма, щом ти харесва, под име ЛИЦЕ.

SAVE ЛИЦЕ

Да се състави програма, която превръща от килограми в грамове.

Откъде да започнем? Какво е дадено и какво търсим?

Дадено е число А в кг. Търсим число В в г.

Добре. Ще се опитам да дам алгоритъм.

1. Въвежда се стойността на А.
2. $B = A \cdot 1000$.
3. Отпечатва се В.
4. Край.

А сега програмата:

```
10 PRINT "ПРЕВРЪЩАНЕ ОТ КГ В Г"  
20 INPUT "ЗАДАЙ В КГ ";A  
30 B=A*1000  
40 PRINT B;" Г"  
50 END
```

Провери дали смята вярно с 10 кг.

] RUN

] RUN
ПРЕВРЪЩАНЕ ОТ КГ В Г
ЗАДАЙ В КГ

] RUN
ПРЕВРЪЩАНЕ ОТ КГ В Г
ЗАДАЙ В КГ 10

] RUN
ПРЕВРЪЩАНЕ ОТ КГ В Г
ЗАДАЙ В КГ 10
10000 Г

Виждам, че започваш да пишеш програми.

Това ли е всичко?

Това е най-важното. Останалото лесно ще го научиш. Сега смятам да си изпълня и обещанието.

Не си спомням вече.

Сега ще се сетиш. Да разгледаме такава задача:

Да се състави програма, която при дадени стойности на A и B, $A \neq 0$, решава уравнението $X:A = B$

Мисля, че мога да я напиша сам.

```
5 PRINT "РЕШЕНИЕ НА X:A=B"
10 INPUT "ЗАДАЙ А И В ";A,B
20 X=B*A
30 PRINT "X= ";X
40 END
```

Правилно. А сега изпробвай програмата за
A = 24 и B = 1768.

] RUN

] RUN
РЕШЕНИЕ НА X:A=B
ЗАДАЙ А И В

] RUN
РЕШЕНИЕ НА X:A=B
ЗАДАЙ А И В 24,1768

] RUN
РЕШЕНИЕ НА X:A=B
ЗАДАЙ А И В 24,1768
X= 42432

Чак сега се сетих за какво става дума. Това беше уравнението, което исках автоматично да се реши.

Да, сега вече имаш програма, която решава всяко уравнение от вида $X:A=B$.

С това за днес завършихме.

Днешният ден ще считам за деня, в който си започнал да пишеш програми.

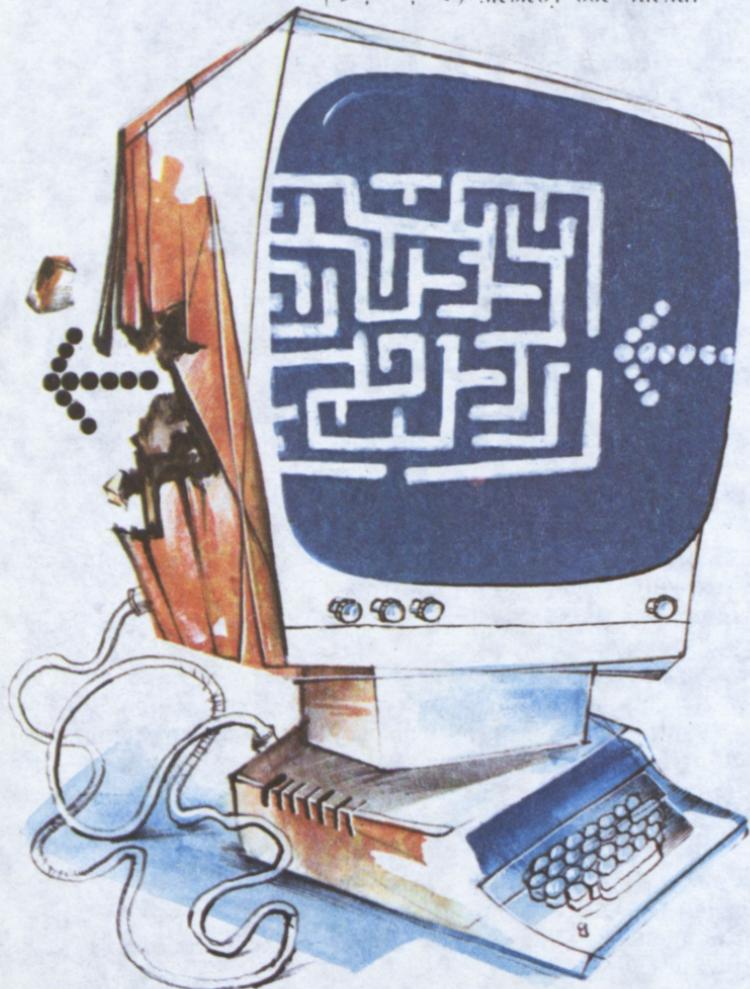
А сега домашното: то е по-интересно от досегашните, защото ще ти позволи да създадеш собствени програми, които могат да ти бъдат полезни при решаване на задачите от домашните по математика. При съставянето на програмите направи така, че те да питат за данните, които са им необходими, и да отпечатват с думи какво се получава.

ДОМАШНА РАБОТА

1. Да се състави програма за въвеждане на едно число и отпечатване на числото плюс 1, самото число и числото минус 1.
2. Да се състави програма за пресмятане периметъра на триъгълник по дадени дължини на трите страни.
3. Да се състави програма за пресмятане периметъра на правоъгълник по дадени дължини на две от страните му.
4. Да се състави програма за пресмятане лицето на квадрат по дадена дължина на страната му.
5. Да се състави програма за пресмятане периметъра на квадрат по дадена дължина на страната му.
6. Да се състави програма за намиране скоростта на автомобил, ако се знае пътят, който той изминава, и времето, за което го изминава.
7. Да се състави програма за намиране времето, за което автомобил изминава даден път, като се движи с дадена скорост.
8. Да се състави програма за намиране пътя, който изминава автомобил при дадени скорост и време.
9. Да се състави програма за превръщане от метри в сантиметри.
10. Да се формулира задача, подобна на горните, и се състави програма за решаването ѝ.

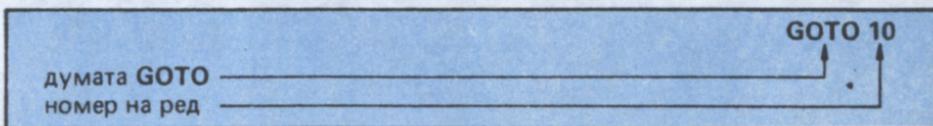
КОМПЮТЪРЪТ ВЗЕМА РЕШЕНИЯ

Разговорът е за програмиране на преходи и логически условия. Описват се инструкциите *GOTO* и *IF*, както ѝ видът, и начините на записване на условие. Съставят се програми за откриване дали две числа са последователни в естествения ред на числата, за проверка дали правилно се отговаря на загадка и за откриване на верния знак ($>$, $=$, $<$) между две числа.



Днес ще се занимаваме с още по-интересни неща. Ще накараме компютъра да взема решения, като в зависимост от стойността на дадена променлива изпълнява една или друга поредица от инструкции.

Досега компютърът изпълняващ инструкция след инструкция по реда на номерата на редовете, в които бяха записани. Но често се налага да се променя този ред и тогава ще използваме инструкцията GOTO. Чрез нея може да се наруши редът на изпълнение на инструкциите. Видът на инструкцията GOTO е следният:



Инструкцията служи за изпълняване вместо на следващата инструкция на инструкцията, чийто номер на ред е указан в нея.

Да видим какво извършва следната програма:

```
10 PRINT "A"  
20 GOTO 10
```

Отпечатва се един път буквата A, след това се преминава на ред 20, който ни връща на ред 10, отново се отпечатва един път буквата A. Колко пъти ще се отпечата буквата A?

Непрекъснато ще се печата.

А как ще може да се спре програмата?

С червения бутон.

Добре. Запомни добре тази програма. Това беше първата ни среща с цикъла — най-същественото нещо в програмирането. Но за него ще говорим утре вечер.

А сега да разгледаме още една инструкция, която нарушава последователността на изпъл-

нение на редовете. Това е инструкцията IF. Тя най-често се среща във вида:

думата IF _____
условие _____
думата GOTO _____
номер на ред _____

IF A > B GOTO 10

Смисълът на инструкцията е следният: Ако условието е изпълнено (в случая, ако A е по-голямо от B), преминава се към ред, чийто номер е посочен в инструкцията. Ако условието не е изпълнено (A не е по-голямо от B, т. е. A е равно на B или B е по-голямо от A), се преминава към изпълнение на инструкцията в следващия ред.

Един друг вид на инструкцията IF е:

думата IF _____
условие _____
думата THEN _____
произволна инструкция _____

IF A > B THEN PRINT "ЗДРАВЕЙ"

Ако условието е изпълнено, изпълнява се инструкцията, която следва THEN. Ако условието не е изпълнено, се преминава към инструкцията в следващия ред. В случая, ако A е по-голямо от B, отпечатва се "ЗДРАВЕЙ", а ако A не е по-голямо от B, преминава се направо на следващия ред.

Операциите за сравняване стойностите на две променливи или числа се записват по следния начин:

равенство като =
по-голямо като >
по-малко като <

не равно като >< или <>
по-малко или равно като <=
по-голямо или равно като >=

Освен тези операции можем да използваме и други операции. Например операцията AND, означаваща "и". С използването ѝ се проверява единовременното изпълнение на две условия. Можем да използваме и операцията OR, озна-

чаваща "или". Чрез нея проверяваме дали поне едно от две условия е изпълнено:

А сега да започнем с примери:

Да се запише А е по-голямо от В.

A > B

Да се запише А е равно на В.

A = B

Да се запише А е различно от В.

A < > B

Да се запише А е по-голямо от В и същевременно В е по-голямо от С.

A > B AND B > C

Да започнем и с писане на програми.

Да се състави програма за проверяване дали две числа са последователни в естествения ред на числата.

Какви са данните за тази задача и какъв е резултатът?

Данните са две числа А и В. Резултатът е "следва" или "не следва".

Хайде да съставим алгоритъм:

1. Въвеждат се стойности на А и В.
2. Проверява се дали $B = A + 1$. Ако е равно, преминава се към 3, ако не — към 4.
3. Отпечатва се "следва".
4. Отпечатва се "не следва".
5. Край.

Общо взето е добре. Но има една грешка. Нека да проследим алгоритъма с два примера и да видим какво ще се получи.

Да започнем с 1 и 3:

1. $A = 1, B = 3$.
2. $3 \neq 2$, преминава се на 4.
4. Отпечатва се "не следва".
5. Край.

Добре, а сега с 1 и 2:

1. $A = 1, B = 2$.
2. $2 = 2$, преминава се на 3.
3. Отпечатва се "следва".
4. Отпечатва се "не следва".

Алгоритъмът продължава:

5. Край.

Виждаш ли? Отпечатата "следва", но след това и "не следва". Какво трябва да се направи?
След ред 3 да се вмъкне още един ред с край.
Или ред, който води до преминаване на 5.
А сега да съставим програмата:

```
10 INPUT A,B  
20 IF B=A+1 THEN GOTO 50  
30 PRINT B;" НЕ СЛЕДВА ";A  
40 GOTO 60  
50 PRINT B;" СЛЕДВА ";A  
60 END
```

Да я изprobваме с числата 1,3 и 1, 2:

] RUN

] RUN

] RUN

?

?

] RUN

? 1,3

3 НЕ СЛЕДВА 1

Получи се верен отговор. 1 и 3 не са последователни.

] RUN

] RUN

] RUN

?

?

] RUN

? 1,2

2 СЛЕДВА 1

2 следва 1. Вярно е.

Добре. Да видим друга задача.

Да се състави програма за проверяване правилно ли се отговаря на гатанката:

В ЛЕСУ РОДИЛАСЬ В ВОДЕ ЖИВЕТ

Ще направим програма да отпечатва текста на гатанката и да чака правилния отговор. При правилен отговор да отпечатва ПОЗНА, а при неправилен да задава въпросителен знак.

А сега да съставим алгоритъм:

1. Отпечатва се гатанката.
2. Пита се за отговора и се записва в А_п.
3. Ако А_п съвпада с верния отговор, изпълнява се 4, а в противен случай — 2.
4. Отпечатва се "ПОЗНА".
5. Край.

Какво ще правиш, ако не можеш да познаеш отговора?

Ще го видя от учебника по руски.

Не се шегувай, питам какво ще правиш с програмата?

Ще я прекъсна с червения бутона.

Какъв е верният отговор?

ЛОДКА.

Добре. А сега програмата?

```
10 PRINT "В ЛЕСУ РОДИЛАСЬ"  
20 PRINT "В ВОДЕ ЖИВЕТ"  
30 INPUT Aп  
40 IF Aп="ЛОДКА" THEN GOTO 60  
50 GOTO 30  
60 PRINT "ПОЗНА"  
70 END
```

Да проверим дали работи. Първо да зададем грешен отговор, а после ще я проверим и с върен отговор.

] RUN

] RUN

В ЛЕСУ РОДИЛАСЬ
В ВОДЕ ЖИВЕТ
?

] RUN

В ЛЕСУ РОДИЛАСЬ
В ВОДЕ ЖИВЕТ
? СЛОН

] RUN

В ЛЕСУ РОДИЛАСЬ
В ВОДЕ ЖИВЕТ
? СЛОН
?

] RUN

В ЛЕСУ РОДИЛАСЬ
В ВОДЕ ЖИВЕТ
? СЛОН
? ЛОДКА

] RUN

В ЛЕСУ РОДИЛАСЬ
В ВОДЕ ЖИВЕТ
? СЛОН
? ЛОДКА
ПОЗНА

Добре. Да видим следната задача:

Да се състави програма, която открива верния знак ($>$, $=$, $<$) между две числа.

Да започнем с алгоритъма.

Но такъв алгоритъм правихме преди няколко вечери.

Браво, че си спомни. Тогава остава да напишеш само програмата.

```
10 INPUT A,B  
20 IF A=B THEN GOTO 60  
30 IF A>B THEN GOTO 80  
40 PRINT "A < B"  
50 GOTO 90  
60 PRINT "A=B"  
70 GOTO 90  
80 PRINT "A > B"  
90 END
```

Да я проверим с различни числа.

] RUN

] RUN

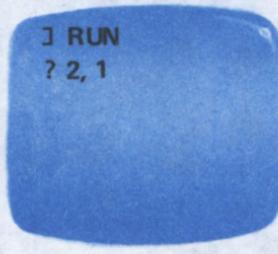
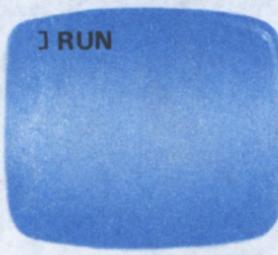
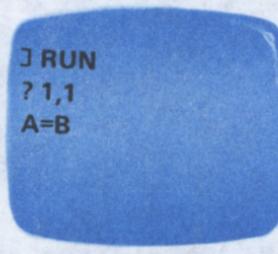
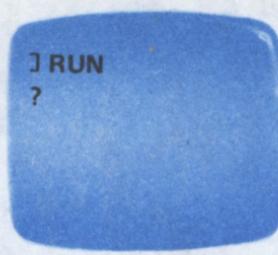
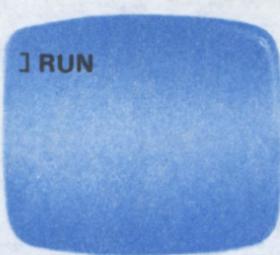
?

] RUN

? 1,2

] RUN

? 1,2
A < B



Днес съм много доволен от теб. Вече усвои
най-основните средства от езика БЕЙСИК и ще
можеш да се разбираш с компютъра.

Остана домашното.

ДОМАШНА РАБОТА

1. Да се състави програма за непрекъснато отпечатване на текста "Аз
съм добър ученик".
2. Какво отпечатва програмата

```
10 A=5  
20 GOTO 50  
30 A=6  
40 GOTO 70  
50 A=7  
60 GOTO 30  
70 PRINT A
```

3. Кой ред ще изпълни програма след реда:

```
100 IF K > N THEN 300
```

ако $K = 5$ и $N = 20$?

4. Кои инструкции могат да се посочват след THEN в инструкцията IF?

5. Да се състави програма за въвеждане на число и отпечатване на " $A > 100$ ", ако числото е по-голямо от 100 и " $A \leq 100$ ", ако числото е по-малко или равно на 100.

6. Да се състави програма за въвеждане на число и отпечатване на " $A > 100$ ", ако числото е по-голямо от 100, " $A < 100$ ", ако числото е по-малко от 100 и " $A = 100$ ", ако числото е равно на 100.

7. Да се състави програма за проверяване правилно ли се отговаря на гатанката:

НА СТУД СЕ БЕЛЧО ПЕРИ
ОТ СЛЪНЦЕТО ТРЕПЕРИ
ЩО Е ТО?

8. Да се състави програма за откриване на липсваща буква в изречението "Краси реже хляба с но?"

9. Да се състави програма за въвеждане на два текста — първият гатанка, а вторият — нейният отговор. След това да се изчиства екранът и да се отпечатва гатанката и програмата да чака верния отговор. При съвпадение да отпечатва — ПОЗНА. Да се провери програмата със следните гатанки:

ЗЕЛЕНА ПАНИЦА,
ЧЕРВЕНА ГОЗБИЦА.

ЩО Е ТО?

НЕ КРОИ, НЕ ШИЕ,
А МНОГО РИЗИ НОСИ.

ЩО Е ТО?



10. Да се формулира една задача, подобна на горните, и да се състави програма за решаването ѝ.

IX урок

СЪРЦЕВИНАТА НА ПРОГРАМИРАНЕТО

Разговорът е за начините за построяване на цикли. Описани са общите схеми за построяване на цикъл по брояч и цикъл по условие. Съставят се програми за намиране на: сумата на първите N естествени числа, сумата на елементите на един масив, средноаритметична стойност на елементите на един масив, сумата на елементите на един масив до срещане на нулев елемент. Разгледана е инструкцията DIM.



Лесно ли направи домашното?

Ами, този компютър само ме ядосва...

Да програмираш, изглежда ще се научиш, но какво става с българския?

Какво „с български“?

Като се върнах днес, компютърът беше зациклен на:

„Аз съм добър оченик“.

Какво значи „зациклен“.

Ами пишеше „Аз съм добър оченик, Аз съм добър оченик...“

За българския ще видим. А сега започваме да правим цикли, които сами завършват.

Построяването на цикли представлява сърцевината на програмирането. Трябва да го овладееш добре. Да си изясним първо какво ще разбираме под понятието цикъл.

Под цикъл в един алгоритъм се разбира последователността от действия, които се изпълняват многократно в един и същ ред, преди да се премине към изпълнението на действието, следващо тази последователност. В езика БЕЙСИК цикълът е последователност от инструкциите, които се изпълняват многократно преди преминаването към друга инструкция.

Всеки цикъл трябва да е организиран така, че да завърши сам или след преминаване през него определен брой пъти, или след нарушаване на някакво условие. В зависимост от това, как завършват, циклите се разделят на два вида:

- цикъл по брояч;
- цикъл по условие.

Какво е това цикъл по брояч?

Ще се опитам да ти дам най-обща схема за построяване на цикли.

Започвам с така наречения цикъл по брояч.

Най-общата схема за построяване на цикъла по брояч изглежда така:

- I. Подготовка на цикъла. Зареждане брояча на цикъла.
- II. Тяло на цикъла.

III. Проверки за край на цикъла. Ако през тялото на цикъла е преминато необходимия брой пъти — излиза се от цикъла, ако не — преминава се на II.

Тялото на цикъла е последователността от инструкциите, които се изпълняват многократно.

След тези обяснения ще решим следната задача:

Да се състави програма за пресмятане на сумата от първите N естествени числа, където стойността на N се въвежда по време на изпълнението на програмата.

Да се придържаме към общата схема за построяване на цикъл и да започнем да съставяме алгоритъма.

I. Подготовка на цикъла. Зареждане брояча на цикъла.

За нашата задача това ще бъде:

1. Въвеждане стойността на N .
2. Даване стойност нула на някаква променлива (например S). Това ще бъде променливата, към която ще се прибавят последователно числата 1, 2, 3... и в която ще се получи резултатът. Предварително записаната нула ще се прибави първоначално към сумата.

3. Даване стойност единица на някаква променлива (например — I). Това ще бъде броячът на цикъла.

II. Тяло на цикъла.

Към стойността на променливата S се прибавя I и резултатът се записва отново в S .

III. Проверка за край на цикъла.

Ако I е равно на N , отпечатва се S и край, ако не — броячът I се увеличава с единица и се преминава там, където започва тялото на цикъла.

Разбра ли последователността от действия?

Мисля, че я разбрах.

Добре. Сега ще ти опиша още веднъж тази последователност в по-близък до програмата вид.

подготовка

{ 1. Въвежда се N .

2. $S = 0$.

3. $I = 1$.

{ 4. $S = S + I$.

5. Ако $I = N$, преминава се към 8, ако не — към 6.

6. $I = I + 1$.

7. Преминава се към 4.

8. Отпечатва се S .

9. Край.

тяло на цикъла

проверка за край на
цикъла

Да допуснем, че е въведена стойност на N —
3. Тогава получаваното решение трябва да е:

$$1 + 2 + 3 = 6.$$

Хайде да проверим алгоритъма.

1. Въвежда се $N = 3$.

2. $S = 0$.

3. $I = 1$.

4. $S = 0 + 1$.

5. $1 \neq 3$, преминава се към 6.

6. $I = I + 1 = 2$.

7. Преминава се към 4.

4. $S = 0 + 1 + 2$.

5. $2 \neq 3$, преминава се към 6.

6. $I = 2 + 1 = 3$.

7. Преминава се към 4.

4. $S = 0 + 1 + 2 + 3$.

5. $3 = 3$, преминава се към 8.

8. Отпечатва се $S = 0 + 1 + 2 + 3 = 6$.

9. Край.

Изглежда, че смята вярно.

Разгледай още веднъж алгоритъма и премини
към написване на програмата.

подготовка {
10 INPUT N
20 S=0
30 I=1

тело на цикъла {
40 S=S+I

проверка за край {
50 IF I=N THEN GOTO 80
60 I=I+1
70 GOTO 40
80 PRINT S
90 END

Провери я с числата 3 и 10.

] RUN

] RUN

?

] RUN
?3

] RUN
?3
6

] RUN

] RUN
?

] RUN
? 10

] RUN
? 10
55

Искам да ти обърна внимание на две инструкции $S = S + I$ и $I = I + 1$.

Какво означават те?

Те означават, че към това, което има в променливите, ще се добави още нещо: за първата I , а за втората — I . Именно затова поставихме първоначално в S числото нула, а в I единица.

Обърни внимание и на ред 70. От него се преминава към ред 40, там, където започва тялото на цикъла.

А сега изчисти вътрешната памет и екрана с инструкциите NEW и HOME и опитай сам да напишеш същата програма.

Ще помагаш ли?

Само ако има нужда.

Добре. Сега искам да си припомним какво беше масив.

Група от променливи.

Да. Съвкупност от променливи, означени с общо име. За да знае трансляторът колко място да отдели за всеки отделен масив във вътрешната памет, трябва да му зададем от колко елемента се състои вски масив. Това се прави чрез инструкцията DIM, която има вида:

думата **DIM** _____
името на масива _____
символа отваряща скоба _____
брой на елементите в масива _____
символа затваряща скоба _____

DIM A(5)

Тази инструкция е описателна. За разлика от изучените досега тя само запазва място за елементите на масива.

Да видим какво означават инструкциите:

]DIM A(5)
]A(5)=5

Първата означава, че ще използваме масив с 5 елемента, а втората, че на елемента с име A(5), т. е. на петия елемент на масива A се присвоява стойност 5.

Добре. А сега да решим следващата задача:

Да се състави програма за намиране сумата на елементите на масив с 5 елемента.

Отново аз ще ти опиша алгоритъма за решаването ѝ, въпреки че е много подобен на предишния.

Нека масивът да означим с A, тогава елементите му ще са: A(1), A(2), A(3), A(4), A(5). Търсим $S = A(1) + A(2) + A(3) + A(4) + A(5)$. Това е и решението на задачата. Но ако масивът е с 10 000 елемента, тогава събирането без цикъл е почти невъзможно.

Да напишем последователността от действия:

I. Подготовка на цикъла. Даване на начална стойност на брояча на цикъла.

1. Въвеждане на елементите на масива A.

2. Даване на стойност нула на променливата S, където ще се натрупва сумата.

3. Даване на стойност единица на брояча на цикъла.

II. Тяло на цикъла.

Към стойността на S се прибавят последовательно елементите на масива.

III. Проверка за край на цикъла.

Ако I е равно на 5, отпечатва се S и край, ако I е различно от 5, на I се присвоява I плюс 1 и се преминава към началото на тялото на цикъла.

А сега да опишем всичко по-подробно и поблизко до програмата.

подготовка

- { 1. Въвеждат се A(1), A(2), A(3), A(4), A(5).
- 2. S = 0.
- 3. I = 1.
- 4. S = S + A(I).
- 5. Ако I = 5, преминава се към 8, ако не към 6.
- 6. I = I + 1.
- 7. Преминава се към 4.
- 8. Отпечатва се S.
- 9. Край.

А сега да проверим алгоритъма:

тяло на цикъла

- 1. Въвеждат се A(1), A(2), A(3), A(4), A(5).

проверка за край

- 2. S = 0.

- 3. I = 1.

- 4. S = 0 + A(1).

- 5. $1 \neq 5$, преминава се към 6.

- 6. I = I + 1 = 2.

- 7. Преминава се към 4.

- 4. S = 0 + A(1) + A(2).

- 5. $2 \neq 5$, преминава се към 6.

- 6. I = 2 + I = 3.

- 7. Преминава се към 4.

- 4. S = 0 + A(1) + A(2) + A(3).

- 5. $3 \neq 5$, преминава се към 6.

- 6. I = 3 + I = 4.

- 7. Преминава се към 4.

- 4. S = 0 + A(1) + A(2) + A(3) + A(4).

изпълняват се по
време на изпълне-
ние на цикъла

5. $4 \neq 5$, преминава се към 6.
 6. $I = 4 + 1 = 5$.
 7. Преминава се към 4.
 4. $S = 0 + A(1) + A(2) + A(3) + A(4) + A(5)$.
 5. $5 = 5$, преминава се към 8.
 8. Отпечатва се.
- $S = 0 + A(1) + A(2) + A(3) + A(4) + A(5)$.
9. Край.

изпълняват се по време на изпълнение на цикъла

Дойде време да се заемем и с програмата:

След като имаме алгоритъма, програмата ще напишем много лесно. В случая всяка инструкция от програмата ще съответства на едно действие от алгоритъма. Само ще добавим инструкция, която запазва място за елементите на масива.

подготовка	<pre>5 DIM A(5) 10 INPUT A(1),A(2),A(3),A(4),A(5) 20 S=0 30 I=1</pre>
тело на цикъла	<pre>40 S=S+A(I)</pre>
проверка за край	<pre>50 IF A(I)=0 THEN GOTO 80 60 I=I+1 70 GOTO 40 80 PRINT S 90 END</pre>

Сега ще я изпробваме ли?

Да. Нека $A(1)=1$, $A(2)=2$, $A(3)=3$, $A(4)=4$ и $A(5)=5$.

] RUN

] RUN

?

] RUN

? 1,2,3,4,5

] RUN

? 1,2,3,4,5

15

Смята вярно.

Обърни внимание на ред 40. Използването на елемент на масива A — A(I), където променливата I мени стойностите си от 1 до 5, е начин за последователно работене с елементите на масива. Сега въведи инструкциите NEW, HOME и ти сам набери програмата.

Добре. А сега следващата задача.

Да се състави програма за намиране на средноаритметичното от елементите на масив с 5 елемента.

А какво значи средноаритметично?

Представи си, че в масива са записани пет оценки и ти искаш да знаеш средния успех. Тогава сумата им се дели на 5.

Няма ли да стане, ако променим ред 80 от предишната програма с PRINT S/5.

Много добре. Промени програмата.

```
5 DIM A(5)
10 INPUT A(1),A(2),A(3),A(4),A(5)
20 S=0
30 I=1
40 S=S+A(I)
50 IF I=5 THEN GOTO 80
60 I = I+1
70 GOTO 40
80 PRINT S/5
90 END
```

Провери я с две шестици, две четворки и една петица.

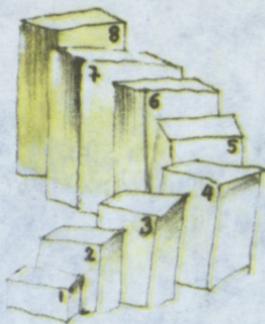
] RUN

] RUN

?

RUN
?6,6,4,4,5

RUN
?6,6,4,4,5
5



Добре. А сега ще преминем към другия вид цикъл — цикъл по условие. И той чака да бъде изучен тази вечер.

Общата схема за построяването на цикъла по условие е подобна на тази при цикъл по брояч.

I. Подготовка на цикъла. Задаване на начални стойности на условието.

II. Тяло на цикъла.

III. Проверка за изпълнение на условието. Ако е изпълнено, излиза се от цикъла, ако не — преминава се към II.

Сега да решим следната задача:

Да се състави програма за пресмятане сумата от първите няколко елемента на даден масив с 5 елемента до срещане на първия елемент със стойност нула, като се знае, че масивът има елемент със стойност нула.

Откъде ще започнем?

От алгоритъма.

подготовка

- { 1. Въвежда се $A(1), A(2), A(3), A(4), A(5)$.
- 2. $S=0$ — тук ще се натрупва сумата.
- 3. $I=1$, чрез I ще се вземат последователно елементите на масива, без да се използва за брояч.

тяло на цикъла

- { 4. $S=S+A(I)$.
- 5. Ако $A(I)$ е равно на нула, преминава се към 8, ако не — към 6.
- 6. $I=I+1$.
- 7. Прекинава се към 4.
- 8. Отпечатва се S .
- 9. Край.

Към сумата, ако забеляза, се прибавя и елемент със стойност нула.

Понякога при решаване на такива задачи се разменят местата на т. II и т. III от общата схема, т. е. първо се проверява дали условието е

изпълнено и в зависимост от това се изпълнява или не тялото на цикъла.

Остана да напишем програмата.

```
5 DIM A(5)
10 INPUT A(1),A(2),A(3),A(4),A(5)
20 S=0
30 I=1
40 S=S+A(I)
50 IF A(I)=0 THEN GOTO 80
60 I=I+1
70 GOTO 40
80 PRINT S
90 END
```

} подготвка

} тяло на цикъла

} проверка на условие

По какво тази програма се различава от тази в предната задача?

По ред 50.

Тук излизането от цикъла става не след събиране на петте елемента, а при първото срещане на елемент със стойност нула.

Да изprobваме програмата с $A(1)=1$, $A(2)=2$, $A(3)=0$, $A(4)=5$, $A(5)=10$.

] RUN

] RUN

] RUN
? 1,2,0,5,10

] RUN

? 1,2,0,5,10
3

Добре. Днес урокът беше много труден. Но усилията ни не са напразни. Ако можеш да пишеш програми с цикли, можеш да пишеш добри програми.

И за да завършим за днес, ето ти домашното. И то е по-трудно.

ДОМАШНА РАБОТА

1. Да се състави програма, която да отпечатва едно име пет пъти.
2. Какво отпечатва следната програма:

```
10 A=5
20 IF A*A > 250 THEN GOTO 60
30 PRINT A,A*A
40 A=A+5
50 GOTO 20
60 END
```

3. Колко числа въвежда и на колко пъти следната програма:

```
10 DIM A(5)
20 INPUT A(1),A(2)
30 INPUT A(3),A(4)
40 INPUT A(5)
```

4. Какво извършва следната програма? След като се разбере смисълът, програмата да се напише самостоятелно.

```
10 DIM A(10)
20 I=1
30 INPUT A(I)
40 IF I=10 THEN GOTO 70
50 I=I+1
60 GOTO 30
70 END
```

5. Да се състави програма за пресмятане на произведението на първите N естествени числа (N се въвежда по време на изпълнение на програмата и не надминава 10). Каква стойност трябва да се присвои предварително на променливата, в която ще се получи произведенietо?

6. Да се състави програма за пресмятане на произведенietо от елементите на масив с 10 елемента.

7. Да се състави програма, която въвежда две цели числа и ги отпечатва заедно с целите числа между тях. Знае се, че първото число е по-малко от второто.

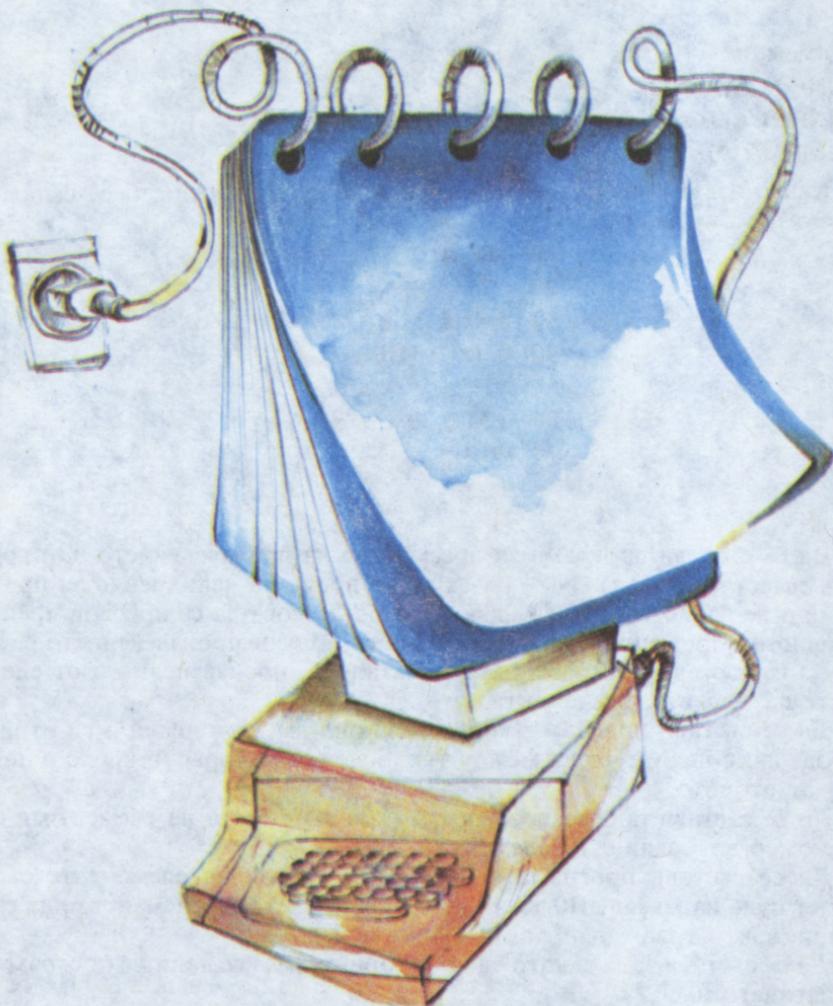
8. Да се състави програма за пресмятане на сумата на елементите с четен номер на масив с 10 елемента.

9. Да се състави програма за пресмятане броя на елементите със стойност нула на масив с 10 елемента. Въвеждането на елементите да се извърши, както това е направено в задача 4.

10. Да се състави задача като една от горните и да се напише програма за решаването ѝ.

ОТНОВО ЦИКЪЛ, НО С ПО-МАЛКО ТРУД

Разговорът е за използването на инструкциите за цикъл *FOR* и *NEXT*. Съставят се програми за решаване на задачите от предния урок чрез използването на тези инструкции. Съставя се програма, наречена **АВТОМАТИЗИРАН БЕЛЕЖНИК** за съхраняване и търсене на данни.



Днес отново ще пишем програми, съдържащи цикъл, но това ще става с по-малко труд. За целта ще използваме инструкциите от езика БЕЙСИК FOR и NEXT. Подготвката на боячка на цикъла може да се извърши чрез инструкцията FOR, чийто вид е:

FOR I = 1 TO 5 STEP 2

думата FOR _____

име на променлива _____

знакът за равенство _____

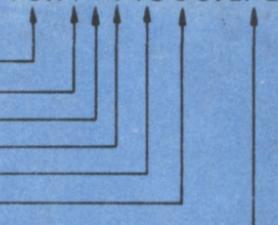
число (аритметичен израз) _____

думата TO _____

число (аритметичен израз) _____

думата STEP последвана от число _____

(аритметичен израз) (незадължителна част)



Променливата представлява бояч на цикъла. На нея се присвоява начална стойност, дадена с израза след знака за равенство. Изразът след думата TO дава крайната стойност на боячка на цикъла. Последната част от инструкцията включва думата STEP, последвана от стойност, наречена „стъпка“ на цикъла. Тази стойност показва с колко нараства боячът на цикъла. Ако тази част липсва, боячът нараства с единица.

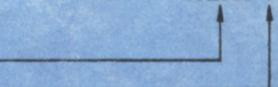
След инструкция FOR следва тялото на цикъла, а за означаване края на цикъла се използва инструкцията NEXT.

Инструкцията NEXT има следния вид:

думата NEXT _____

име на променлива _____

NEXT I



Инструкцията показва къде завършва цикълът. За име на променлива се поставя името на боячка на цикъла. Така се указва край на кой точно цикъл задава инструкцията.

Да се опитаме чрез инструкциите FOR и NEXT да напишем програмите от вчера.

Всичките ли?

Да, но сега ще бъде много лесно.

Да се състави програма за пресмятане на сумата от първите N естествени числа, където стойността на N се въвежда по време на изпълнение на програмата.

```
10 INPUT N  
20 S=0  
30 FOR I=1 TO N  
40 S=S+I  
50 NEXT I  
60 PRINT S  
70 END
```

Така е по-просто, забелязваш ли? Няма нужда да изписваме проверката за края на цикъла. Но да разгледаме внимателно старата и новата програма и да видим кои инструкции с кои сме заменили.

Вместо

I=1 е поставено FOR I=1 TO N

и вместо редовете

```
IF I=N THEN GOTO 80  
I=I+1  
GOTO 40
```

е поставено NEXT I

Изпълни, ако обичаш програмата с N=5.

] RUN

] RUN

?

] RUN

? 5

] RUN

? 5

15

А сега, както сигурно се досещаш, искам да зададеш инструкциите NEW и HOME и сам да напишеш програмата.

Трябва ли?

Добре. А сега — следващата задача:

Да се състави програма за намиране сумата от елементите на масив с 20 елемента.

```
10 DIM A(20)
20 S=0
30 FOR I=1 TO 20
40 INPUT A(I)
50 S=S+A(I)
60 NEXT I
70 PRINT S
80 END
```

Разгледай я добре. Забелязваш ли как се въвеждат елементите на масива?

Хитро си го измислил.

Нищо особено, просто в цикъла, в който правя сумата, въвеждам и елементите. В тази програма редовете 40 и 50 са тялото на цикъла.

А сега задай отново инструкциите NEW, HOME и ти напиши програмата.

Ето друга задача:

Да се състави програма за намиране на средноаритметичното от елементите на масив с 20 елемента.

Как ще стане? Сещаш ли се?

Ще изменим само ред 70 от предишната програма.

```
70 PRINT S/20
```

А как да проверим дали тази програма работи правилно.

Наум да съберем 20 числа не е трудно, но трябва време.

Добре. Нека да поправим малко програмата, така че да намира средноаритметичното на еле-

ментите на масив с произволен брой елементи до 20.

```
10 DIM A(20)
15 INPUT "БРОЙ"; N
20 S = 0
30 FOR I = 1 TO N
40 INPUT A(I)
50 S = S + A(I)
60 NEXT I
70 PRINT S/N
80 END
```

По този начин създадохме програма, която чете и намира средноаритметичното на масив, чиято дължина може да бъде от 1 до 20, а конкретната дължина въвеждаме по време на изпълнението на програмата.

Но нали е сложено DIM A(20)?

Да. Но транслаторът отделя място за 20 елемента, а колко от тях използваме, зависи от числото N, което четем.

А сега задай на N стойност 5 и провери програмата с две шестцици, две четворки и една петица.

] RUN

] RUN

] RUN

БРОЙ

БРОЙ

БРОЙ 5

] RUN

БРОЙ 5

?

] RUN

БРОЙ 5

? 6

? 6

? 4

? 4

? 5

5

Да разгледаме и една задача, подобна на последната задача от вчера.

Да се състави програма, която пресмята сумата от елементите на един масив с 1000 елемента, до срещане на елемент със стойност нула, ако има такъв.

За тази задача ние използваме цикъл по условие, а инструкциите FOR и NEXT ни помагат да организираме цикъл по брояч. Но както сега е изменено условието, ще ни трябва и брояч, който да следи да не прочетем повече от 1000 элемента. Това може да се случи, ако в масива няма нито един елемент със стойност нула.

Да напишем програмата.

```
10 DIM A(1000)
20 S=0
30 FOR I= 1 TO 1000
40 INPUT A(I)
50 IF A(I)=0 THEN GOTO 80
60 S=S+A(I)
70 NEXT I
80 PRINT S
90 END
```

Тази програма е по-универсална от вчерашната. Инструкциите FOR и NEXT сме използвали за две неща — първо, чрез променливата I въвеждаме последователните стойности на масива и, второ, чрез нея следим да не се прочетат повече от 1000 числа. В ред 50 проверяваме дали прочетеният елемент има стойност нула и ако е така, излизаме от цикъла, преди да сме прочели всичките елементи на масива.

А сега напиши програмата сам.

Добре. Ако искаш, почини малко, защото ще те занимая с една интересна задача, съвсем различна от досегашните.

Да се състави програма за въвеждане на информация за определен брой градове и за получаване на справки по името на града.

Ще наречем програмата АВТОМАТИЗИРАН БЕЛЕЖНИК. Нека бележникът се състои от 1000 реда. Всеки ред ще има две полета. В първото ще записваме името на града, а във второто — информация за него.

Ще създадем програмата в две части. Първата ще бъде за писане в бележника, а втората — за търсене от него.

Да помислим върху алгоритъма. Нека използваме два масива с по 1000 елемента. В първия масив ще записваме имената на градовете, а в съответните елементи на втория масив — данните за тях.

Изброй ми няколко града.

София, Пловдив, Бургас.

А сега какви ми някакви данни за тях.

София, столица, 1 000 000 жители.

Пловдив, в Южна България.

Бургас, на Черно море.

Да напишем частта на програмата, която е за писане в бележника.

```
10 DIM A□(1000),B□(1000)
20 INPUT "БРОЙ ";N
30 FOR I=1 TO N
40 INPUT A□(I),B□(I)
50 NEXT I
60 END
```

Отделните данни трябва да са разделени със запетая и да представляват текст с дължина до 255 символа. Предварително ще въведем броя на градовете. Той е цяло число, не по-голямо от 1000.

А сега да допълним програмата с частта за търсене в бележника. Тя ще въвежда дума и ще я търси в масива А□, ако я открие, ще отпечатва информацията за нея, която се съхранява в масива B□.

```
10 DIM A□(1000),B□(1000)
20 INPUT "БРОЙ ";N
30 FOR I=1 TO N
40 INPUT A□(I),B□(I)
50 NEXT I
60 INPUT "ЗАДАЙ КАКВО ТЪРСИШ ";S□
70 FOR I=1 TO N
80 IF A□(I)=S□ THEN GOTO 120
90 NEXT I
100 PRINT "НЯМА ГО В БЕЛЕЖНИКА"
110 GOTO 60
120 PRINT B□(I)
130 GOTO 60
```

Да изprobваме действието ѝ:

] RUN

] RUN
БРОЙ

] RUN
БРОЙ 3

] RUN

БРОЙ 3

? СОФИЯ, СТОЛИЦА 1 000 000 ЖИТЕЛИ

? ПЛОВДИВ, В ЮЖНА БЪЛГАРИЯ

? БУРГАС, НА ЧЕРНО МОРЕ

] RUN

БРОЙ 3

? СОФИЯ, СТОЛИЦА 1 000 000 ЖИТЕЛИ

? ПЛОВДИВ, В ЮЖНА БЪЛГАРИЯ

? БУРГАС, НА ЧЕРНО МОРЕ

ЗАДАЙ КАКВО ТЪРСИШ

] RUN

БРОЙ 3

? СОФИЯ, СТОЛИЦА 1 000 000 ЖИТЕЛИ

? ПЛОВДИВ, В ЮЖНА БЪЛГАРИЯ

? БУРГАС, НА ЧЕРНО МОРЕ

? ЗАДАЙ КАКВО ТЪРСИШ СОФИЯ

] RUN

БРОЙ 3

? СОФИЯ, СТОЛИЦА 1 000 000 ЖИТЕЛИ

? ПЛОВДИВ, В ЮЖНА БЪЛГАРИЯ

? БУРГАС, НА ЧЕРНО МОРЕ

ЗАДАЙ КАКВО ТЪРСИШ СОФИЯ

СТОЛИЦА 1 000 000 ЖИТЕЛИ

ЗАДАЙ КАКВО ТЪРСИШ

] RUN

БРОЙ 3

? СОФИЯ, СТОЛИЦА 1 000 000 ЖИТЕЛИ

? ПЛОВДИВ, В ЮЖНА БЪЛГАРИЯ

? БУРГАС, НА ЧЕРНО МОРЕ

ЗАДАЙ КАКВО ТЪРСИШ СОФИЯ

СТОЛИЦА 1 000 000 ЖИТЕЛИ

ЗАДАЙ КАКВО ТЪРСИШ ВИДИН

НЯМА ГО В БЕЛЕЖНИКА

Кога ще спре програмата според теб?

Така, както я написахме, само червеният клавиши **RST може да я спре.**

Нека да остане така.

Мисля, че вече те запознах с най-основните неща в програмирането: променливи, масиви, програмиране на условие и програмиране на цикъл. Може да спрем и дотук.

Но нали обеща 15 дена?

Добре. Имаме още 5 дни. В тях ще видим Правец-82 да рисува, да пее и да върши още много неща.

Ето и домашното. То се решава по-лесно и вероятно по-бързо от вчерашното.

ДОМАШНА РАБОТА

1. Колко пъти следната програма отпечатва "ЗДРАВЕЙ":

```
10 FOR I=1 TO 20 STEP 3  
20 PRINT "ЗДРАВЕЙ"  
30 NEXT I  
40 END
```

2. Да се състави програма, която да помага на наказан ученик, като отпечатва 20 пъти "ЩЕ ВНИМАВАМ В КЛАС"

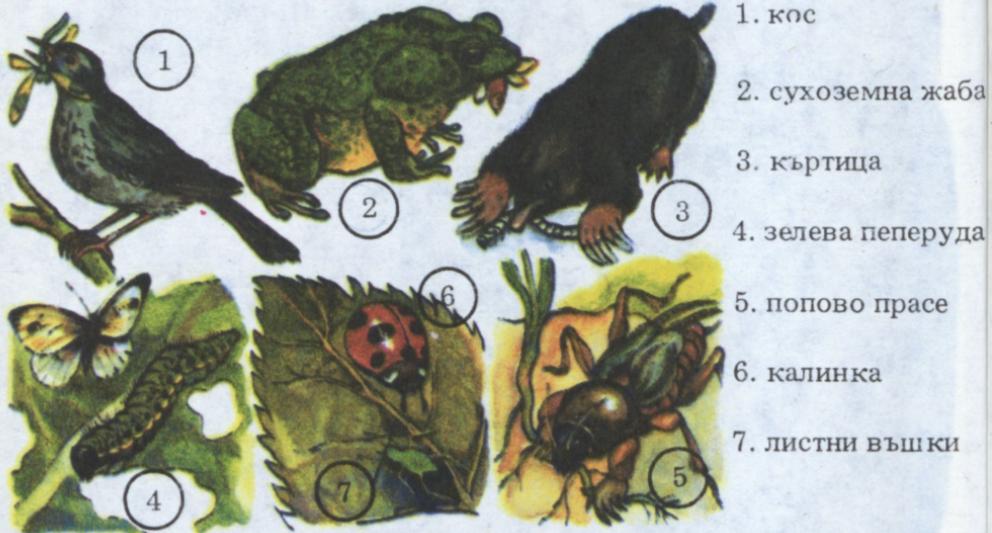
3. Да се състави програма за пресмятане на произведението на първите N естествени числа, където N се въвежда по време на изпълнение на програмата и не надминава 10.

4. Да се състави програма, която да въвежда 10 числа в масив и да пресмята произведените им.

5. Да се състави програма, която да въвежда 10 числа в масив и да пресмята сумата от елементите му, които са по-големи от 100.

6. Да се състави програма, която да въвежда 10 числа в масив и да пресмята сумата от елементите, които имат четен номер, т. е. $A(2) + A(4) + \dots + A(8)$ (спомнете си, че в инструкцията FOR може да има стъпка).

7. Да се състави програма на играта ПОЛЕЗНО ЖИВОТНО. Първоначално да се въвеждат имена на животни и данни за това, дали са полезни, или не. След това еcranът да се изчиства, да се отпечатват едно след друго имената на животните и да се очаква отговор от играча. Отговорът се сверява с първоначално въведения и се отговаря дали е правилен. При съставяне на програмата използвайте идеи от програмата АВТОМАТИЗИРАН БЕЛЕЖНИК. Въведете и изprobвайте програмата с данни за кос, сухоземна жаба, къртица, зелева пеперуда, попово прасе и калинка.



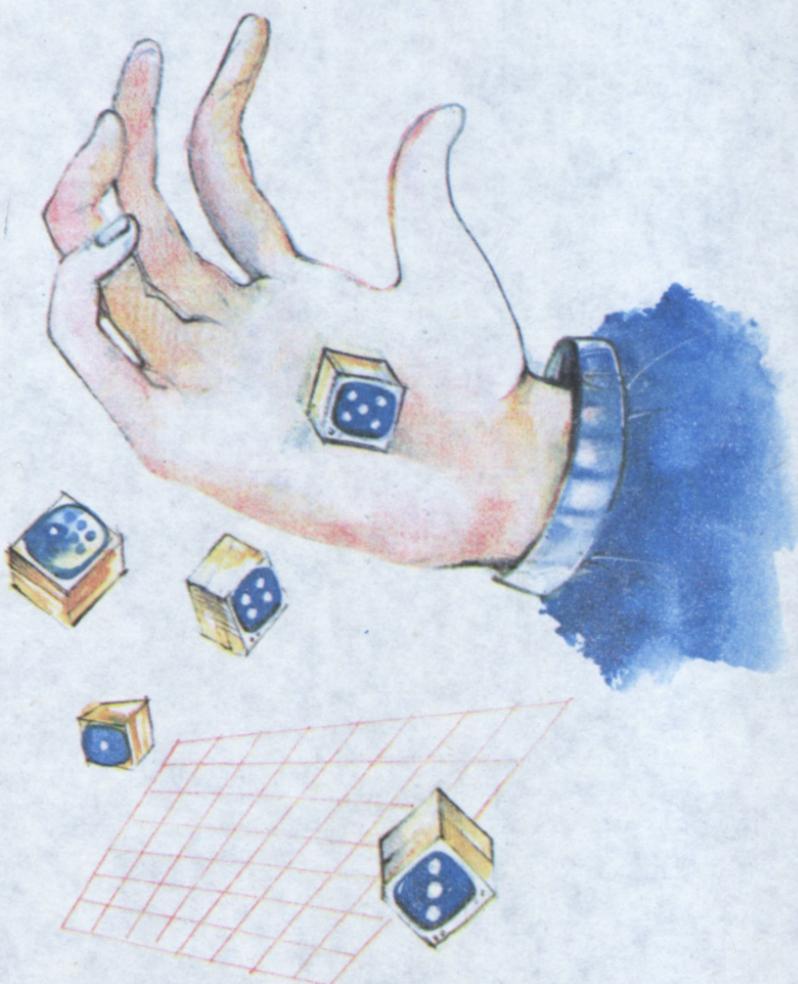
8. Да се състави програма, която да въвежда елементите на масив с 10 числа и да пресмята произведението им до срещане на елемент със стойност нула (първият елемент на масива е различен от нула).

9. Да се състави програма, която да въвежда 10 числа в един масив и да преброява елементите, имащи стойност нула.

10. Да се изпробва програмата АВТОМАТИЗИРАН БЕЛЕЖНИК, като се въвелят имена на познати и техните телефонни номера.

В СВЕТА НА СЛУЧАЙНОСТИТЕ

Разговорът е за използването на случаини числа в програмирането. Обяснява се използването на стандартни функции в езика БЕЙСИК. Използват се стандартни функции при задаване на случаино число и при задаване на случаайна буква. Съставят се програми за игри, използващи случаини числа, като "хвърляне на зарове", учене на чужди думи и др.



Спомняш ли си играта СТРЕЛБА, която наготово използвахме преди няколко вечери.

Да. Ти започваше с появяването на десет точки върху екрана, всеки път на различно място.

Бих искал да ти обясня как става това. Представи си екрана като игра-мозайка с точки и че можеш да сложиш всички точки в една кутия. Разбъркваш точките в кутията и вадиш една точка. Тя има място върху екрана.

Нещо като това?

Разликата е, че точката, която си извадил, връщаш в кутията, преди да извадиш друга точка. Затова може в играта СТРЕЛБА на данено място да има две и повече точки.

Описаният начин на избор ще наричаме случаен.

Ако чрез случаен избор получаваме числа, например между числата от 1 до 100, получаваните числа ще наричаме случаини числа.

И как се бърка в тази кутия от компютъра?

Това се извършва чрез стандартни функции, които представляват готови програми. Те са включени в транслятора и ние ги използваме наготово за извършване на различни, често срещани действия. Използваме ги, като се обръщаме към тях чрез техните имена, а в скоби задаваме с каква стойност искаме те да извършат харacterното за тях действие. Резултатът е друга стойност. Обръщения към стандартни функции могат да участват в аритметични изрази. Ние ще използваме някои инструкции за присвояване, в лясната част на които са включени обръщения към стандартни функции, но по-скоро заради интересните възможности, които те ни дават, отколкото за усвояването на употребата им.

Една такава инструкция за присвояване, в която се използват стандартни функции и която ще ни е полезна за намиране на цяло случаино число между нула и $N - 1$, е:



$$A = \text{INT}(N * \text{RND}(1))$$

име на променлива _____

знакът за равенство _____

точно този текст _____

числото N _____

точно този текст _____

Да разгледаме следната задача:

Да се състави програма-игра за „хвърляне на зар”, в която играчът се опитва да познае какво се е паднало.

Първата част от задачата е равносилна със задачата за получаване на едно цяло случайно число между 1 и 6.

Това ще стане чрез използване на

$$A = \text{INT}(6 * \text{RND}(1)) + 1$$

За да получим случайно цяло между 1 и 6, а не между 0 и 5, сме прибавили единица.

Цялата програма ще изглежда така:

```
10 A=INT(6*RND(1))+1
20 INPUT "КОЛКО СЕ ПАДНА ";B
30 IF A=B THEN PRINT "ПОЗНА"
40 GOTO 10
```

Разгледай ред 30. Не забравяй, че след THEN може да стои произволна инструкция. Ако си познал числото, отпечатва се "ПОЗНА" и се отива на следващия ред, който връща към ред 10 и играта започва отново. Ако условието не е изпълнено — не си познал, направо се отива на следващия ред, а оттам към началото на играта.

Разбра ли програмата?

Да.

А сега още една задача.

Да се състави програма, която отпечатва две цели случаечни числа, по-малки от 100, очаква да бъде въведено тяхното произведение и съобщава дали то е вярно.

Тази програма ще наречем МАТЕМАТИК.

Да я съставим. Напиши в ред с номер 10

инструкция, която в променлива A получава едно цяло случаино число между 0 и 99.

10 A=INT(100*RND(1))

Добре, а сега нека в променливата B се получава още едно цяло случаино число между 0 и 99.

20 B=INT(100*RND(1))

Изчисти экрана.

30 HOME

Напиши на экрана условието на задачата, която трябва да се решава, A.B=?

40 PRINT A," * ";B," = "

Прочети отговора.

50 INPUT S

Провери дали е даден правилен отговор, ако е даден — напиши "ПОЗНА" и се върни в началото за нови числа, ако не е — отново питай за правилния отговор.

```
60 IF S=A*B THEN GOTO 90  
70 PRINT "НЕ ПОЗНА"  
80 GOTO 50  
90 PRINT "ПОЗНА"  
100 GOTO 10
```

Добре. А сега можеш да поиграеш. Трябват ти и лист, и молив, за да можеш да умножаваш числата, които ти се задават.

Сега ще ти дам наготово друга инструкция за присвояване, която използва стандартна функция за получаване на една латинска буква по случаен начин.

име на променлива —————
знакът за равенство —————
точно този текст —————

A[□]=CHR[□](INT(26*RND(1)+65))

Да разгледаме такава задача:

Да се състави програма, която избира случаино една латинска буква и очаква да се познае какво е избрала.

Кое е по-трудно да се познае — каква латинска буква се е паднала или какво число на зара се е паднало?

Сигурно буквата е по-трудно да се познае.

Да. Защото латинските букви са 26, а цифрите при зара са само 6.

Да напишем програмата. Сега трябва да използваме символни променливи.

```
10 A□=CHR□(INT(26*RND(1)+65))
20 INPUT "ЗАДАЙ ЛАТИНСКА БУКВА";B□
30 IF A□=B□ THEN PRINT "ПОЗНА"
40 GOTO 10
```

Искаш ли да поиграеш?

Много е трудно да се познае. Друг път ще играя.

Тогава ще напишем една много полезна програма. Тя е предназначена за учене на чужди думи.

Да се състави програма, която въвежда 10 български и 10 чужди думи (на руски или английски език). След това в случаен ред да се появява българска дума и да се очаква въвеждане на съответната чужда дума. При грешка да се появява върната дума и обучението да продължава.

Тази програма прилича много на играта АВТОМАТИЗИРАН БЕЛЕЖНИК. Първата част е същата като в нейната програма. Напиши я.

```
10 DIM A□(10),B□(10)
20 FOR I=1 TO 10
30 INPUT A□(I),B□(I)
40 NEXT I
```

Изчисти экрана.

50 HOME

А сега получи едно случаено число между 0 и 9 в променливата S.

60 S=INT(10*RND(1))

Отпечатай S-тия елемент на масива A.

70 PRINT Aⁿ(S)

Въведи отговора.

80 INPUT Cⁿ

Ако прочетената дума е правилна, отпечатай "ПОЗНА" и премини към получаване на ново случаено число, ако прочетената дума е грешна, отпечатай "ГРЕШКА" и върната дума и премини към ново случаено число.

```
90 IF Cn=Bn(S) THEN GOTO 120
100 PRINT "ГРЕШКА";Bn(S)
110 GOTO 60
120 PRINT "ПОЗНА"
130 GOTO 60
```

А сега въведи английски думи от последния урок по английски.

- ? RUN
- ? PRINT, ОТПЕЧАТАЙ
- ? LET, НЕКА
- ? FOR, ЗА
- ? IF, АКО
- ? THEN, ТОГАВА
- ? GO TO, ПРЕМИНИ КЪМ
- ? INPUT, ВЪВЕДИ
- ? NEXT, СЛЕДВАЩ
- ? STEP, СТЪПКА
- ? NEW, НОВ

LET
? НЕКА
ПОЗНА
PRINT
? ОТПЕЧАТАЙ
ПОЗНА
GO TO
? ПРЕМИНИ КЪМ
ПОЗНА
PRINT
?

Научи ли ги?

Аз си ги знам.

Добре. За днес свършваме. Видяхме как може да се използват наготово програми, вложени в транслатора, и как може да се получава число или буква по случаен начин.

ДОМАШНА РАБОТА

1. Какво представлява резултатът от инструкцията?

A=INT(5*RND(1))

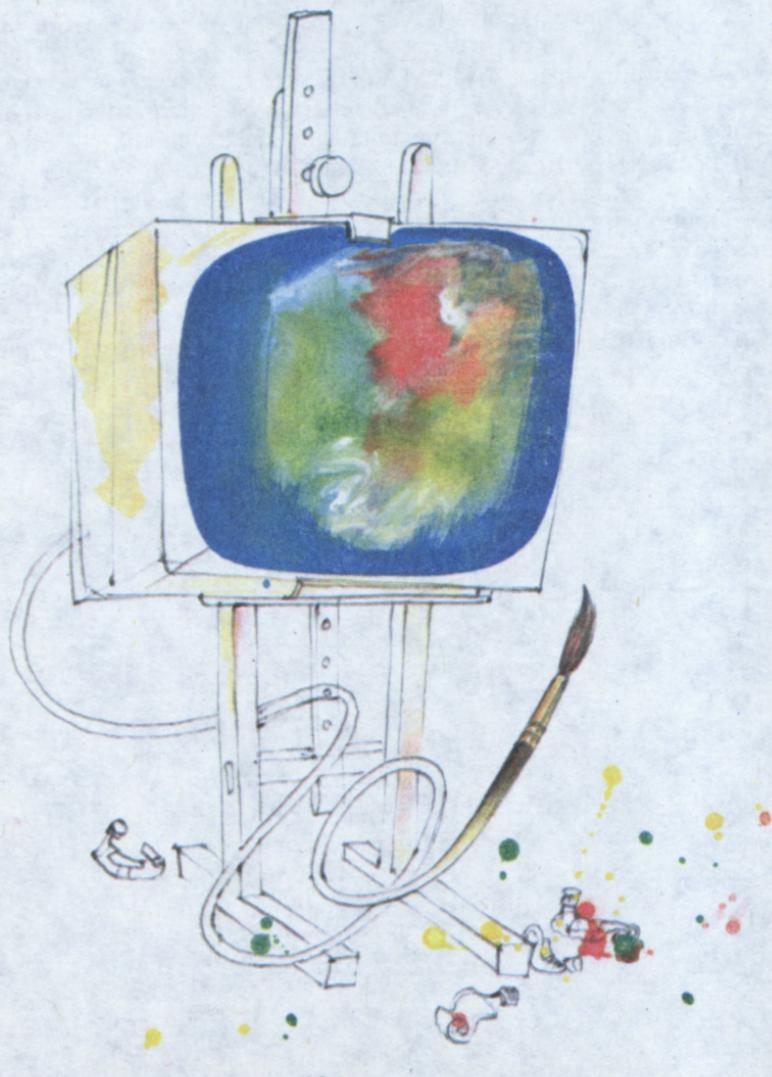
2. Какво представлява резултатът от инструкцията:

A#=CHR#(INT(26*RND(1)+65))

3. Да се състави програма за "хвърляне на два зара", като при познаване на сумата от резултатите да се отпечатва "ПОЗНА".
4. Да се състави програма за "хвърляне на два зара", като, ако сумата им е по-голяма от 6, да се отпечатва "БРАВО".
5. Да се състави програма за "изтегляне" на две случаини латински букви и отпечатване на "БРАВО", ако те съвпадат.
6. Да се състави програма МАТЕМАТИК, която да проверява усвоено ли е действието събиране на двуцифриeni числа.
7. Да се състави програма МАТЕМАТИК, която да проверява усвоено ли е действието изваждане на две двуцифриeni числа. (Да се проверява кое от двете случаини числа е по-голямо и от него да се изважда по-малкото.)
8. Да се състави програма МАТЕМАТИК, която проверява усвоено ли е действието деление, като се задава деление на четирицифрени на двуцифриeni числа. (Препоръчва се получаване на две случаини двуцифренi числа A и B, след което да се решава следната задача: $(A \cdot B) : B = ?$)
9. Да се състави програма за играта БЪРЗО ЧЕТЕНЕ, която да действува по следния начин: появяват се на екрана пет букви, задържат се малко (прави се с цикъл, който нищо не прави) и след това изчезват (изчтиства се еcranът). Думите трябва да се запомнят от играчите и правилно да се възпроизведат.
10. Да се състави самостоятелно задача, която използва случаини числа или букви, и да се състави програма за решаването ѝ.

КОМПЮТЪРЪТ КАТО ХУДОЖНИК

Разговорът е за графичните възможности на Правец-82. Разглеждат се инструкциите *GR*, *TEXT*, *COLOR*, *PLOT*, *HLIN*, *VLIN*. Съставят се програми за чертане на букви, случаини точки, "завеса" от линии и за рисуване на картини.



XII вечер

Как мислиш, дали можем да накараме Правец-82 да рисува?

След като хвърля зарове, нищо чудно и да може да рисува.

Така е... Трябва само да му кажем. Но напред необходимо е еcranът да бъде в графичен режим.

Досега в текстов режим ли беше?

Да. Преминаването в графичен режим става с инструкцията GR.

думата **GR** —

GR



При изпълнение на тази инструкция еcranът преминава в графичен режим и върху него могат да се чертаят до 40 реда по 40 точки на ред. Най-отдолу остават и 4 текстови реда, върху които могат да се пишат програми и да се получават или задават числови и текстови данни.

Изprobвай тази инструкция.

Указателят е върху един от 4-те текстови реда.

А как ще се върнем в текстов режим?

Можем да използваме инструкцията TEXT. Тя има вида

думата **TEXT** —

TEXT

След тази инструкция еcranът си възстановява предишното състояние.

А какво ще можем да правим в графичен режим?

Да започнем с това, че ще можем да чертаем в 16 цвята.

Но нашият телевизор е черно-бял.

Казахме, че се нарича монитор. Та при нашия монитор, който наистина е черно-бял, цветовете се заменят с различна яркост на экрана.

И така, както всеки художник, за да нарисува нещо, трябва да потопи четката си в боя, така и ние първо трябва да изберем цвет. Това става с инструкцията COLOR.

COLOR = 5

думата COLOR —————
знакът за равенство —————
число (аритметичен израз) —————

Инструкцията служи за задаване на цвета, в който ще се появяват точките.

Стойността, която задаваме, трябва да е цяло число между 0 и 15. Цят 0 се използва за изтриване.

А сега да видим как може да се изчертава една точка. Екранът е разделен на 40 реда с по 40 точки на ред:

0,0 1,0, 2,0 ... 39,0
0,1 1,1 2,1 ... 39,1
0,2 1,2 2,2 ... 39,2
0,39 1,39 2,39 ... 39,39

За да означим мястото на всяка една точка върху экрана, ще използваме двойка числа, разделени със запетая, от които второто число означава номера на реда, в който е точката (число между 0 и 39), а първото — поредния номер на точката в реда (число между 0 и 39).

Това вече го видях при играта СТРЕЛБА.

Изчертаването на една точка се задава чрез инструкцията PLOT.

PLOT 0,5

думата PLOT —————
номер на точката в реда —————
знакът запетая —————
номер на реда —————

Да се състави програма за изчертаване на най-горната лява точка.

Първо да преминем в графичен режим

10 GR

Да изберем цвета, нека е 15-и.

20 COLOR=15

Да отпечатаме точката

30 PLOT 0,0

Да изпълним програмата, задаваме инструкцията...

RUN



Да се състави програма за изчертаване на най-горната лява точка и за изтриването ѝ.

Първите три реда ги знаеш. А след това трябва да се използва боята, която изтрива, това беше цвят номер нула, и с него да начертаем повторно точката.

10 GR
20 COLOR= 15
30 PLOT 0,0
40 COLOR= 0
50 PLOT 0,0

Изпълни програмата.

Много бързо се изтрива точката. Почти не се вижда, като се появява.

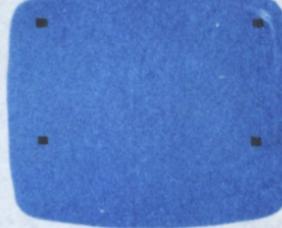
Ако помислиш, че се сетиш как да я задържаш за по-дълго, но сега това не е важно.

Да се състави програма за изчертаване по една точка в четирите края на екрана.

Първо премини в текстов режим, за да можеш да въведеш програмата, след това въведи програмата и накрая я изпълни.

10 GR
20 COLOR=3
30 PLOT 0,0
40 PLOT 0,39
50 PLOT 39,0
60 PLOT 39,39

JRUN



Появиха се четирите точки.

Браво, вече ще можеш да се научиш да чертаеш и отсечки.

Сещам се как се чертаят отсечки. Ще направим цикъл и в него ще чертаем точка по точка.

Правилно. А за изчертаването на хоризонтални и вертикални линии има специални инструкции: HLIN и VLIN. Ето как изглеждат те:

думата **HLIN** _____
номер в реда на началната точка
на отсечката _____
знакът запетая _____
номер в реда на крайната точка
на отсечката _____
думата **AT** _____
номер на реда, в който се чертае _____

HLIN 5,7 AT 10

думата **VLIN** _____
номер на реда на началната
точка на отсечката _____
знакът запетая _____
номер на реда на крайната
точка на отсечката _____
думата **AT** _____
пореден номер в редовете на
точките от отсечката _____

VLIN 5,7 AT 10

HLIN служи за изчертаване на хоризонтални
линии, а VLIN на вертикални.

Да се състави програма, която изчертава най-горния хоризонтален ред.

От кой до кой пореден номер в реда ще бъде отсечката?

От 0 до 39.

На кой ред ще чертаем?

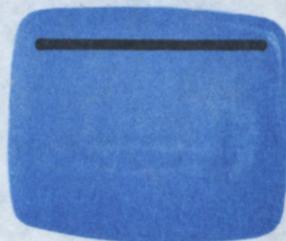
0.

А сега състави и набери програмата:

10 GR
20 COLOR= 15
30 HLIN 0,39 AT 0

JRUN

Да я изпълним.



Да се състави програма, която изчертава най-горната хоризонтална и най-долната вертикална линия.

Първата част за чертане на хоризонтален ред имаме. Да добавим за вертикалната линия.

От кой до кой ред ще бъде?

От 0 до 39.

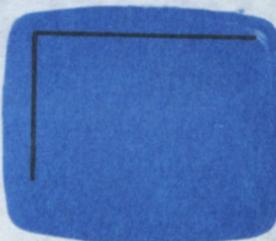
Кой е поредният номер на точките от линията?

0.

Напиши програмата и я изпълни.

10 GR
20 COLOR= 15
30 HLIN 0,39 AT 0
40 VLIN 0,39 AT 0

JRUN



Да се състави програма за изчертаване на буквата Т.

Първо трябва да решим откъде ще чертаем хоризонталната отсечка и на кой ред, след това да определим мястото и на вертикалната отсечка.

Ще напиша тази програма.

Добре. После я изпълни.

```
10 GR  
20 COLOR= 15  
30 HLINE 0,30 AT 5  
40 VLINE 5,30 AT 15
```

RUN



Да се състави програма, която изчертава на екрана 10 случаини точки.

Знаеш ли какво измислих? Ще разпечатам играта СТРЕЛБА и ще видя как става.

Не, сам я направи. Първо е нужен цикъл от 1 до 10. В тялото получавай две случаини числа между 0 и 39 и нека това са номерът на реда и номерът на точката в реда, в който се изчертава. Преди цикъла трябва да се влезе в графичен режим и да се зададе цвят.

Ще се опитам.

```
10 GR  
20 COLOR= 15  
30 FOR I= 1 TO 10  
40 A= INT(40*RND(1))  
50 B= INT(40*RND(1))  
60 PLOT A,B  
70 NEXT I  
80 END
```

Изпълни програмата.

Сега я изпълни още веднъж.

Наистина сега точките излязоха на друго място.

Да се състави програма, която чертае "завеса" от линии с 16-те възможни цвята.

"Завесата" се състои от 16 отсечки от горе до долу и всяка в различен цвят. Какъв алгоритъм ще предложиш?

Трябва ни цикъл. В него ще сменяме цветовете на вертикалните линии, които чертаем.

Да напишем програмата.

```
10 GR  
20 FOR I = 0 TO 15  
30 COLOR= I  
40 VLIN 0,39 AT I  
50 NEXT I  
60 END
```

Да я изprobваме.

|RUN



Линиите не се различават много добре една от друга на черно-белия монитор, но на цветен завесата ще изглежда много хубаво.

А сега още една програма. С нейна помощ ще можем да рисуваме картини.

Да се състави програма, която изчертава една желана точка, след това в зависимост от това, коя от буквите I, J, K, M се въвежда, предизвика изчертаване на точка, която е нагоре при въведено I, наляво — при J, надясно — при K, и надолу — при M:

А сега да създадем програмата. Първо да прочетем номера на една точка в A, B и да я начертаем.

```
10 GR  
20 COLOR= 15  
30 INPUT A,B  
40 PLOT A,B
```

Да прочетем една буква.

50 INPUT C

Ако прочетената буква е "K", стойността на А трябва да увеличим с единица:

60 IF C = "K" THEN A=A+1

Ако е J — стойността на А трябва да намалим с единица.

70 IF C = "J" THEN A=A-1

Ако е I — стойността на В трябва да намалим с единица.

80 IF C = "I" THEN B=B-1

Ако е M — стойността на В трябва да увеличим с единица.

90 IF C = "M" THEN B=B+1

А сега една проверка дали получената точка е в екрана, т. е. А и В не трябва да са по-малки от 0 и по-големи от 39. Ако това се случи — връщат им се предишните стойности и отново се чертае същата точка.

```
100 IF A<0 THEN A=0  
110 IF A>39 THEN A=39  
120 IF B<0 THEN B=0  
130 IF B>39 THEN B=39
```

Сега сме готови да начертаем новата точка и да се върнем да въвеждаме следващата. Какво следва?

140 GOTO 40



Да изprobваме програмата.

Добре. Но защо не чертае?

Натисна ли жълтия клавиши **ЛАТ**?

Бях го забравил.

Да запазя ли програмата на дискета?

Запази я под името ХУДОЖНИК.

SAVE ХУДОЖНИК

Добре. Днес започнахме да чертаем. Правенето на чертежи от компютрите е изключително полезно нещо, защото те могат да правят това по-точно от хората. Утре ще видим как ще можем да чертаем „по-фини“ рисунки.

А сега ето домашното. Много е лесно.

ДОМАШНА РАБОТА

1. Колко точки могат да се чертаят на екрана в графичен режим (след инструкцията GR)?

2. Да се състави програма за изчертаване на буквата П.

3. Да се състави програма за изчертаване на вертикални линии в различни цветове.

4. Да се допълни следната програма, така че да чертае правоъгълник:

```
10 GR  
20 COLOR=15  
30 HLIN 0,20 AT 30
```

5. Да се допълни следната програма, така че да чертае квадрат (в смисъл да има един и същ брой точки във всяка страна):

```
10 GR  
20 COLOR=15  
30 HLIN 0,20 AT 30
```

6. Да се състави програма за изчертаване на движеща се "завеса" (изчертава се една отсечка, след това се изтрива, а след това — съседната до нея).

7. Да се състави програма, която да изчертава случайна точка и веднага след това да я изтрива.

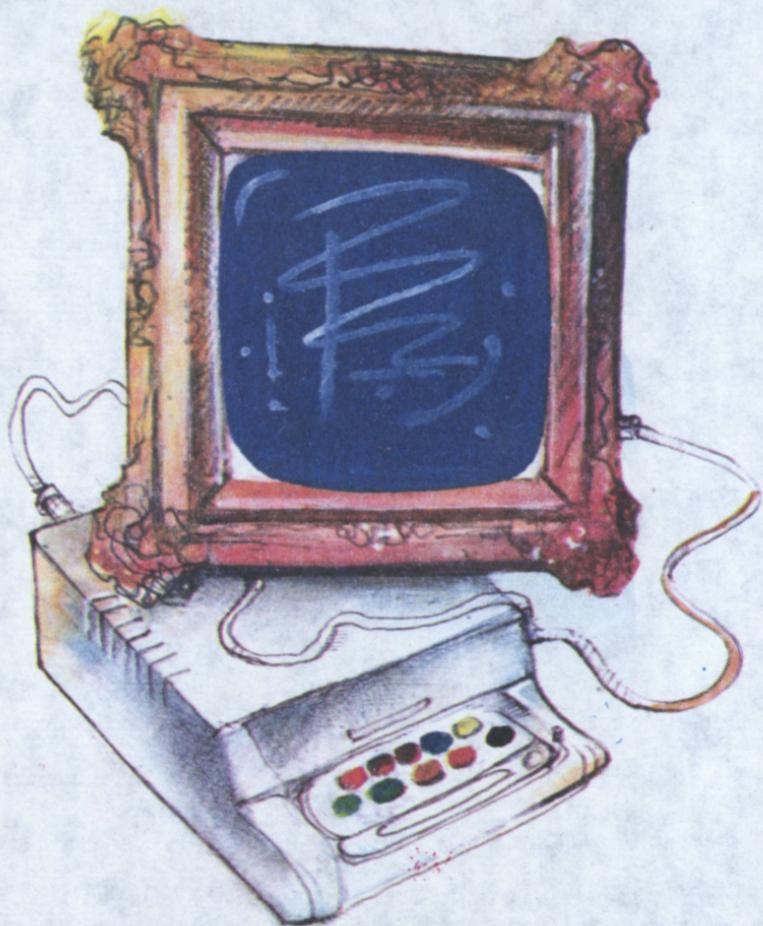
8. Да се начертате триъгълник с помощта на програмата ХУДОЖНИК.

9. Да се начертате дърво с помощта на програмата ХУДОЖНИК.

10. Да се състави програма за изчертаване на произволно избрана буква.

КОМПЮТЪРЪТ МОЖЕ ДА РИСУВА И ПО-ДОБРЕ

Разговорът е за възможностите на Правец-82 за "фина" графика. Разглеждат се инструкции HGR, HCOLOR и HPLÖT. Съставят се програми за изчертаване на геометрични обекти, букви, движеща се "завеса" от линии и за рисуване на картини.



Днес ще накараме компютъра да чертае по-добре. Вчера върху екрана можехме да разполагаме само 1600 точки и те бяха доста големи. Днес ще видим как еcranът може да се раздели на 160 реда, като във всеки ред има по 280 точки. Общо колко точки се получават?

Сега ще пресметна. Получават се 44 800 точки.

И когато чертаем по този начин, чертежите стават значително по-добре оформени. За да премине еcranът в този режим на работа, се използва инструкцията HGR, чийто вид е:

думата HGR

HGR

И тук за означаване на мястото на всяка точка върху екрана ще използваме двойка числа, разделени със запетая, от които второто число означава номер на реда, а първото — пореден номер на точката в реда. Номерата на редовете са от 0 до 159, а номерата на точките в редовете са от 0 до 279.

(0,0) (1,0) (2,0) (3,0) ... (279,0)
(0,1) (1,1) (2,1) (3,1) ... (279,1)
(0,2) (1,2) (2,2) (3,3) ... (279,2)

... (0,159) (1,159) (2,159) (3,159) ... (279,159)

Преминаването на екрана в текстов режим ще извършваме и тук с инструкция...

TEXT

Правилно. А сега за цветовете. При този режим на работа те са само 8 и се означават с числата от 0 до 7. Цвят нула и тук служи за изтриване. Инструкцията, която работи с цвета, действа като инструкцията COLOR. Ето вида ѝ.

думата HCOLOR

HCOLOR = 5

знак за равенство

номер на цвят

Да видим как може да се изчертае една точка.

Инструкцията, която се използва в този режим, е HPLOT.

Ето вида и:

H PLOT 5,6

думата H PLOT

номер на точката в реда

знакът запетая

номер на реда

Да се изчертаят с цвят 3 четирите крайни точки на экрана.

Това е лесно.

Влизаме в режим на "фина" графика и поставяме цвят номер 3.

10 HGR
20 HCOLOR=3

Изчертаваме последователно четирите точки.
И после изпълни програмата.

30 H PLOT 0, 0
40 H PLOT 0, 159
50 H PLOT 279, 0
60 H PLOT 279, 159

1 RUN

А сега да видим още една форма на инструкцията с двойката числа.

H PLOT TO 5,7

думата H PLOT

думата TO

номер на точката в реда

знакът запетая

номер на реда

Тази форма е предназначена за изчертаване на отсечка с начало последната изчертана на екрана точка и край точката, зададена в инструкцията с двойката числа.

Да се състави програма за изчертаване на най-лявата вертикална линия.

Първо трябва да изчертаем началната точка — в случая тя е ...

0,0

Да видим програмата

```
]TEXT  
10 HGR  
20 HCOLOR=3  
30 HPLOT 0,0
```

А сега ще начертаем отсечка от тази точка до крайната в реда, двойката числа за нея са...

0,159

```
40 HPLOT TO 0,159
```

Изprobай програмата

```
]RUN
```

Да се състави програма за изчертаване на най-лявата вертикална линия и най-горната хоризонтална линия.

Нека се опитам сам.

Добре.

```
10 HGR  
20 HCOLOR=3  
30 HPLOT 0,0  
40 HPLOT TO 0,159  
50 HPLOT TO 279,0
```

Да я изprobваме.

Да.

RUN



Остава да разгледаме и третата форма на инструкцията HPLOT, в която инструкцията има най-големи възможности:

HPLOT 0,0 TO 0,20 TO 20,20 TO 5,0

думата HPLOT

номер в реда, знакът запетая и
номер на реда на точка

думата TO

номер в реда, знакът запетая и
номер на реда на точка

думата TO

номер в реда, знакът запетая и
номер на реда на точка

думата TO

номер в реда, знакът запетая и
номер на реда на точка

Инструкцията е предназначена за изчертаване на последователност от отсечки. Най-напред се изчертава първата точка с даден номер в ред и номер на ред, зададени след HPLOT, след това отсечката до следващата дадена точка, после следващата отсечка и т. н.

Да се състави програма за изчертаване на най-лявата вертикална линия и най-горната хоризонтална линия.

Сега редовете 30, 40 можем да заменим с един.

Напиши програмата.

10 HGR

20 HCOLOR=3

30 HPLOT 0,159 TO 0,0 TO 279,0

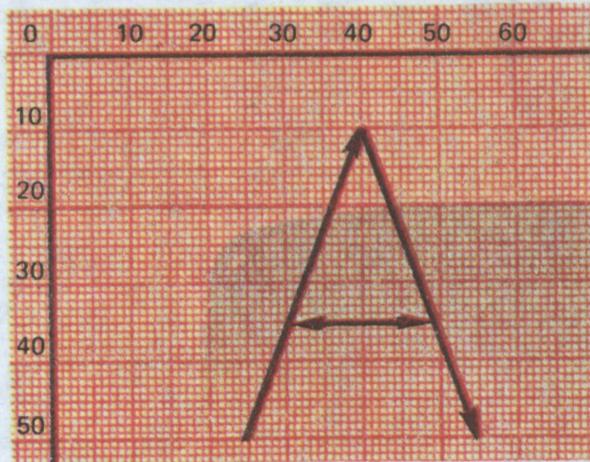
Изprobай я.

RUN



Да се състави програма за изчертаване на буквата А.

Да решим по какъв начин ще начертаем буквата А.



Изчертаването ще извършим, като начертаем последователно 5 отсечки — първата от (10,60) до (40,10), втората от (40,10) до (58,40), третата от (58,40) до (22,40), четвъртата — отново на същото място от (22,40) до (58,40) и петата — от (58,40) до (70,60).

А сега от теб се иска програмата. Тя е много лесна.

```
10 HGR  
20 HCOLOR = 3  
30 HPLOT 10,60 TO 40,10 TO 58,40  
TO 22,40 TO 58,40 TO 70,60  
40 END
```

И да я изпълним.

1 RUN

A

Да се изчертае движеща се "завеса", състояща се от вертикални линии.

Какво се иска?

Трябва една след друга да се начертаят вертикални линии.

За целта ни трябва цикъл от 0 до 279, в който ще чертаем линия след линия.

Ще се опитам да напиша програмата.

```
10 HGR  
20 HCOLOR=3  
30 FOR I=0 TO 279  
40 HPLOT I,0 TO I,159  
50 NEXT I  
60 END
```

Изпълни я.

Линиите се появяват една след друга, докато изпълнят экрана.

А сега още една задача:

Да се състави програма за рисуване чрез клавишите I, J, K, M

Разликата от програмата ХУДОЖНИК, създадена вчера, е, че ще използваме по-фина графика.

Трябва да сменим ред 10 с:

10 HGR

Ред 20 — с:

20 HCOLOR=3

и съответно редове 110 и 130 — с:

110 IF A>279 THEN A=279

130 IF B>159 THEN B=159

А сега направи цялата програма

```
10 HGR
20 HCOLOR=3
30 INPUT A,B
40 HPLOT A,B
50 INPUT C□
60 IF C□="K" THEN A=A+1
70 IF C□="J" THEN A=A-1
80 IF C□="I" THEN B=B-1
90 IF C□="M" THEN B=B+1
100 IF A<0 THEN A=0
110 IF A>279 THEN A=279
120 IF B<0 THEN B=0
130 IF B>159 THEN B=159
140 GOTO 40
```

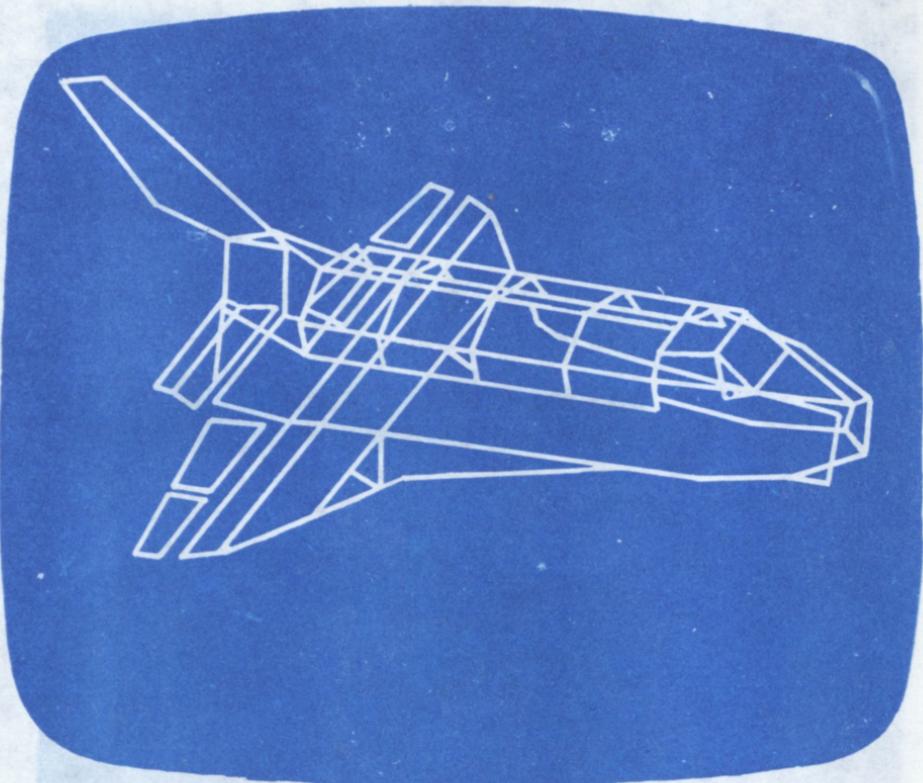
Начертай сега една буква с програмата.

Става много хубаво, но е бавно.

Ще направим програмата да работи по-бързо, но след две вечери.

А сега запиши програмата под името ХУ-ДОЖНИК—2.

С това завършваме. Вече знаеш как да караш компютъра да рисува каквото поискаш.

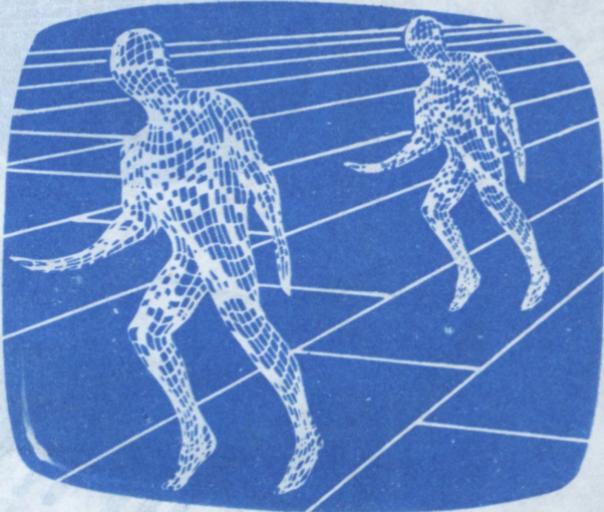


Изчертаването на графики, схеми, чертежи е твърде полезно за редица области. Както вече ти казах, към компютрите може да се включва устройство за чертане. По такъв начин се изчертават плановете на корабите и самолетите. Получават се по-прецизни, отколкото чертежите на човек-проектант.

Остава домашното.

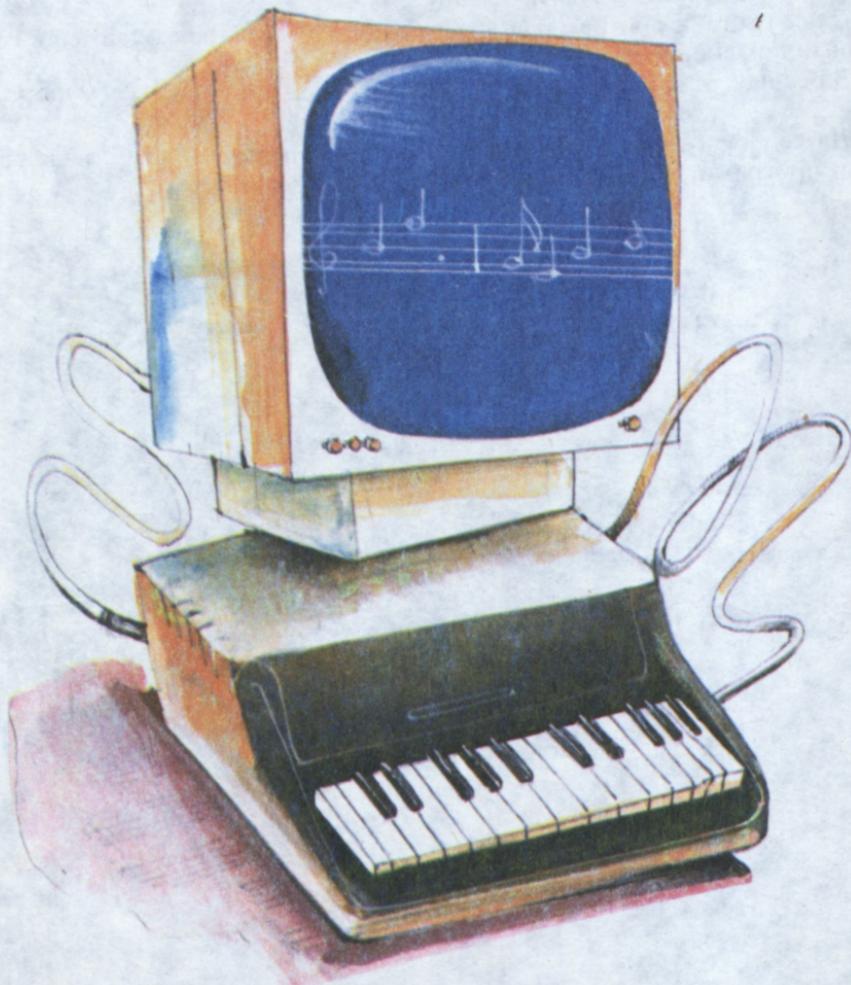
ДОМАШНА РАБОТА

1. Да се състави програма за изчертаване на буквата Б.
2. Да се състави програма за изчертаване на правоъгълник.
3. Да се състави програма за изчертаване на квадрат.
4. Да се състави програма за изчертаване на равнобедрен триъгълник.
5. Да се състави програма за изчертаване на прав ъгъл с връх в точката (50,50).
6. Да се състави програма за изчертаване на две перпендикулярни прости линии, пресичащи се в точка (50,50).
7. Да се състави програма за изчертаване на две неперпендикулярни прости, пресичащи се в точката (50,50).
8. Да се състави програма за изчертаване на завеса чрез движещи се хоризонтални линии.
9. Да се начертаят автомобил, като се използва програмата ХУДОЖНИК—2.
10. Да се формулира задача, използвща "фината графика", и да се състави програма за решаването ѝ.



КОМПЮТЪРЪТ ИЗПЪЛНЯВА МЕЛОДИИ

Разговорът е за възможностите на компютъра Правец-82 да възпроизвежда звукове. Описва се начинът за получаване на звук. Дава се програма, с помощта на която може да се изпълняват мелодии. Като пример е показан начин за изпълнение на песента "Писната гайда шарена" от Паращек Хаджиев.



След като накарахме компютъра да рисува, днес, да се надяваме, ще го чуем да изпълнява мелодии.

Изпълнението на звук е свързано с преминаване на мембранията на високоговорителя от едно в друго положение. Това става чрез многократно изпълняване на следната инструкция за присвояване:

име на променлива ————— → A = PEEK (-16336)
знакът за равенство —————
точно този текст —————

Многократно изпълнение на инструкцията може да получим, като я поставим в...

Цикъл.

Да напишем програма за издаване на звук, като се използва споменатата инструкция 100 пъти.

```
10 FOR I=1 TO 100  
20 A=PEEK(-16336)  
30 NEXT I  
40 END
```

В коя игра имаше тези инструкции?

Мисля, че в СТРЕЛБА.

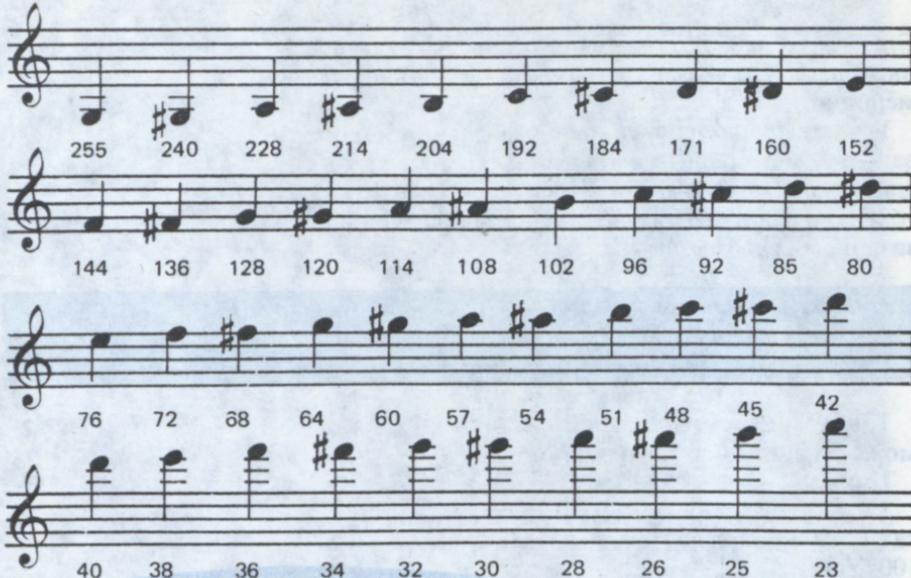
Правилно. А сега изпълни програмата.

Обади се.

За да могат да се изпълняват различни мелодии, в паметта има записана специална програма, чрез която се извършва контролиране както на тона, така и на неговата трайност. Нейното използване обаче е много сложно за теб.

Затова ще постъпим така. Ще ти дам готова програма (на която не всички инструкции знаеш), която използва тази специална програма. Ще ти дам още две таблици, с помощта на които ще изразяваш нотите и тяхната трайност с числа. *

Първата таблица е на най-употребяваните тонове (42 на брой), които могат да бъдат изпълнявани на този компютър, и числа, чрез които могат да се задават.



$\textcircled{o} = 512$

$\textcircled{d} = 384$

$\textcircled{\text{d}} = 256$

$\textcircled{\text{d}} = 192$

$\textcircled{\text{c}} = 128$

$\textcircled{\text{c}} = 96$

$\textcircled{\text{b}} = 64$

$\textcircled{\text{b}} = 48$

$\textcircled{\text{a}} = 32$

$\textcircled{\text{a}} = 43 + 43 + 43$

$\textcircled{\text{g}} = 16$

$\textcircled{\text{g}} = 21 + 21 + 21$

Таблицата е дадена за тонове с една и съща трайност (128). За теб ще е интересно да научиш, че при композиране се използват само 84 тона, а компютърът може да възпроизвежда 256 тона, задавани с числата от 0 до 255. При използване на тонове извън тези 84 компютърът свири фалшиво.

Във втората таблица са дадени номерата, съответстващи на трайностите на нотите, определящи се от вида им.

И тук има възможност за задаване на 256 вида трайности, но както и при тоновете е уместно да се използват само някои от тях.

Сега ще ти дам готовата програма, която изпълнява мелодии с не повече от 1000 ноти. Ти задаваш броя на нотите, а след това за всяка нота последователно задаваш номера на нотата заедно с номера на трайността, като използваш горните две таблици.

```
10 DIM W(1000),P(1000)
20 DATA 173,48,192,136,208,5,206,1,3,240,9
30 DATA 202,208,245,174,0,3,76,2,3,96,0,0
40 FOR X=770 TO 792
50 READ Y
60 POKE X,Y
70 NEXT X
80 HOME
90 INPUT "БРОЙ НОТИ /ОТ 1 ДО 1000/ "; N
100 PRINT "КОД, ТРАЙНОСТ"
110 FOR I=1 TO N
120 PRINT "НОТА";I;
130 INPUT W(I),P(I)
140 NEXT I
150 FOR I=1 TO N
160 POKE 768,W(I)
170 POKE 769,P(I)
180 CALL 770
190 NEXT I
200 END
```

А какво правят инструкциите, за които в тази програма не си ми казвал нищо?

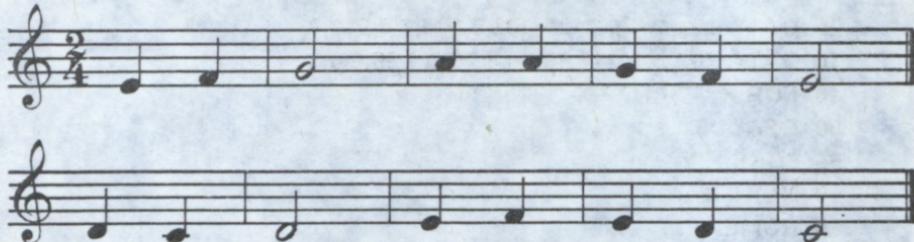
За тях, както вече казах, няма да говорим. И без тях можеш да пишеш достатъчно много програми. Ако все пак остане у теб това желание, след време може и да ги разгледаме.

Разгледай само редовете от 90 до 140. Те въвеждат броя на тоновете в N, поставят техните номера в масива W, а номерата за трайността им — в масива P.

Татко, хайде да изprobваме програмата.

Добре. Набери програмата и я запиши под името МУЗИКАНТ.

А сега да изпълним чрез програмата МУЗИКАНТ песента "Писната гайдашарена" от Парацеке Хаджисев.



Като гледаме в двете таблици, виждаме, че номерата на тоновете и техните трайности са:
(152,128); (144,128); (128,255); (114,128);
(114,128); (128,128); (144,128); (152,255);
(171,128); (192,128); (171,255); (152,128);
(144,128); (152,128); (171,128); (192,255).

Изпълни програмата с тези ноти и чуй мелодията.

Хареса ли я?

Общо взето свирит правилно.

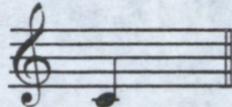
Урокът днес беше по-кратък. Що се отнася до домашното и то няма да е трудно.

ДОМАШНА РАБОТА

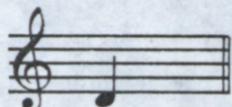
1. Колко различни тона може да възпроизвежда Правец - 82 и с какви числа се означават?

2. Колко различни трайности на един тон може да възпроизвежда Правец-82 и с какви числа се означават?

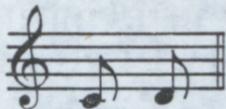
3. С помощта на програмата МУЗИКАНТ да се изпълни:



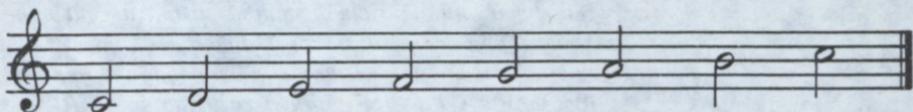
4. С помощта на програмата МУЗИКАНТ да се изпълни:



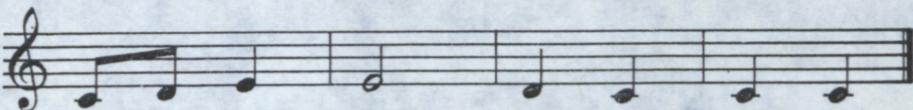
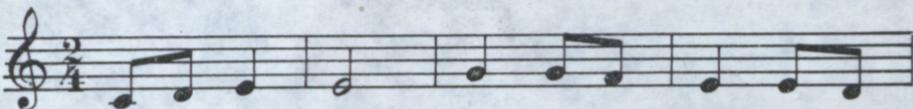
5. С помощта на програмата МУЗИКАНТ да се изпълни:



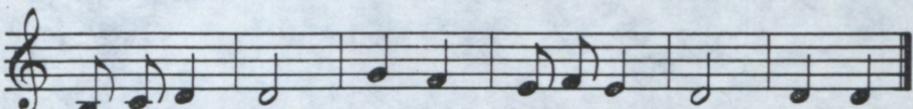
6. С помощта на програмата МУЗИКАНТ да се изпълни:



7. С помощта на програмата МУЗИКАНТ да се изпълни следната народна песен от Луковитско:



8. С помощта на програмата МУЗИКАНТ да се изпълни следната народна песен от Тракия:



9. Да се състави програма КЛАКСОН, която свири с прекъсване.

10. Да се избере мелодия и да се изпълни на Правец – 82 с помощта на програмата МУЗИКАНТ.

XV урок

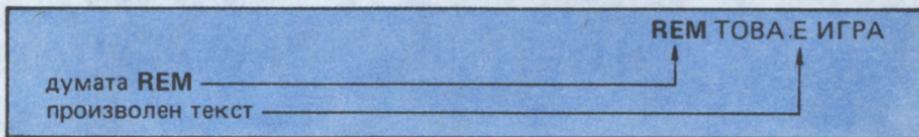
СЪСТАВЯНЕ НА ИГРИ

Разговорът е за някои препоръки, които е добре да се имат предвид при писане на програми. Споменава се инструкцията REM. Дава се алгоритъм на играта СТРЕЛБА. Изброени са няколко съвета и относно средствата за писане на програми, реализиращи игри. Описва се начин за непосредствено въвеждане на символ чрез инструкцията GET. Подобрява се играта ХУДОЖНИК.



Неусетно стигнахме до последния урок. От утре ти оставаш сам с компютъра. Сега искам да ти кажа някои неща, които, ако ги спазваш, ще ти бъде по-лесно да пишеш програми.

1-во. Програмите трябва да са ясни, така че, ако ги прочетеш след време или ги прочете друг, да могат да бъдат разбрани. За целта се употребява инструкцията REM. Тя има вида:



С тази инструкция се записват обяснителни текстове вътре в самата програма. Те подсещат какво извършват следващите инструкции. Транслаторът не ги взема предвид и тази инструкция с нищо не променя изпълнението на програмата.

2-ро. Програмите трябва да съдържат проверки на данните, които се въвеждат, дали са правилни и ако не са, да издават съответни съобщения. Досега това не сме го правили и ако на входа бяхме задали неправилни данни, щяхме да си имаме главоболия с компютъра.

3-то. При писане на по-дълги програми почесто да се използва инструкцията PRINT, като се отпечатват не само крайните резултати. Ако има грешки в програмата с получаването на междинни резултати, те по-лесно могат да се открият. Когато програмата заработи, излишните инструкции PRINT могат да се махнат.

А сега да изясним най-после наготово използваната програма на играта СТРЕЛБА.

Да я извикам ли от дискетата?

Да, ако обичаш.

Първо, да я разгледаме на части, защото е по-дълга от един екран. За целта, ако си спомняш, след LIST трябва да укажем началния и крайния номер на групите редове за разпечатване.

А сега ще ти обясня как е съставена програмата:

в ред 10 се запазва място за елементите на два масива, в които ще се пазят числата, за-

даващи точките, които ще се изчертават на екрана;

в редове 20—30 се преминава в графичен режим и се задава цвят;

в редове 40—90 се изчертават през една точка най-горната хоризонтална и най-лявата вертикална линия;

в редове 100—160 се "хвърлят два зара" (всеки с по 40 страни), записват се стойностите им в масивите и се изчертават съответните точки;

в ред 170 се дава стойност на брояча, в който ще се броят сполучливите удари;

в редове 180—230 се въвеждат точките, чрез които се стреля, и ако някоя от тях съвпада с някоя от точките от екрана, се преминава на ред 270;

в ред 250 се отпечатва броят на попаденията;

редове 270—290 са за случай на сполучка. Броячът се увеличава с 1 и се изтрива точката;

в редове 300—340 се връща цветът, чува се звук. Преминава се на ред 220.

В тази програма се използва цикъл в цикъл. Това е едно от най-сложните неща в програмирането.

Ето как може да изглежда същата програма с обяснителни редове:

```
5 REM ИГРА СТРЕЛБА
10 DIM A(10),B(10)
15 REM ГРАФИЧЕН РЕЖИМ
20 GR
30 COLOR=15
35 REM ИЗЧЕРТАВАНЕ НА ЛИНИИТЕ
40 FOR X=0 TO 39 STEP 2
50 PLOT X,0
60 NEXT X
70 FOR Y=0 TO 39 STEP 2
80 PLOT 0,Y
90 NEXT Y
95 REM ПОЛУЧАВАНЕ И ИЗЧЕРТАВАНЕ НА ТОЧКИТЕ
100 FOR I=0 TO 9
```

```
110 K=INT(39*RND(1))+1
120 T=INT(39*RND(1))+1
130 A(I)=K
140 B(I)=T
150 PLOT K,T
160 NEXT I
165 REM S—БРОЙ ПОПАДЕНИЯ
170 S=0
175 REM СТРЕЛБА
180 FOR I=0 TO 9
190 INPUT X,Y
200 FOR J=0 TO 9
205 REM ПРОВЕРКА ЗА ПОПАДЕНИЕ
210 IF A(J)=X AND B(J)=Y THEN GOTO 270
220 NEXT J
230 NEXT I
240 HOME
250 PRINT S;“ ПОПАДЕНИЯ”
260 END
265 REM ПОПАДЕНИЕ
270 S=S+1
280 COLOR=0
290 PLOT X,Y
300 COLOR=15
305 REM ЗВУК
310 FOR K=1 TO 50
320 W=PEEK(-16336)
330 NEXT K
340 GOTO 220
```

А сега искам да ти дам два съвета за писане на програми за игри.

1-во. Програмата е добре да започва с'въпрос, дали играчът знае правилата на играта, и ако отговорът е отрицателен, компютърът да му разпечатва инструкция за игра, при това в приятен за играча вид.

Ето как например може да стане това:

```
10 INPUT "ЗНАЕТЕ ЛИ ПРАВИЛАТА НА ИГРАТА? "; X□  
20 IF X□="ДА" THEN GOTO 40  
30 PRINT "ИГРАТА... СЕ ИГРАЕ ПО СЛЕДНИЯ НАЧИН..."  
40 REM ЗАПОЧВА ПРОГРАМАТА
```

2-ро. Повечето игри изискват от играта бърза реакция. Такива са например игрите с топка. При тях е необходимо въвеждане на символа от клавиатурата да се извършва, без да се натиска **RETURN**.

Въвеждане непосредствено след натискане на даден символ от клавиатурата в променлива става чрез инструкцията GET, имаща вида:

думата **GET** —————
име на символна променлива —————

GET X□

Да изprobваме действието на инструкцията GET, като разгледаме задачата:

Да се състави програма, която отпечатва текста: "НАТИСНАХТЕ СИМВОЛА" и самия символ непосредствено след натискането му.

Първо да въведем в променливата X□ символът, който се натиска от клавиатурата.

10 GET X□

Да го разпечатаме заедно с текста "НАТИСНАХТЕ СИМВОЛА".

20 PRINT "НАТИСНАХТЕ СИМВОЛА "; X□

Да преминем в началото на програмата, за да сме готови да въвеждаме нов символ непосредствено след натискането му.

30 GOTO 10

А сега да изprobваме програмата.
Работи правилно. А как ще я спрем? С червения клавиши ли?

Сега я спри с него, но ще допълним програмата, така че тя да спира действието си при даване на определен символ — например "Q".

15 IF C[□]="X" THEN END

Провери дали можеш да прекъсваш работата на програмата с въвеждането на Q.

Сега да направим нещо друго. Да ускорим работата на програмата ХУДОЖНИК—2.

Да се създаде програма за чертане, която променя посоката на чертане непосредствено след натискане на някой от клавишите I, J, K, M.

За целта ща извикаме програмата ХУДОЖНИК—2.

А сега сигурно трябва да заменим инструкцията INPUT с GET.

50 GET C[□]

Трябва да заменим и C със символната променлива CS.

Ще нанеса измененията.

```
10 HGR
20 HCOLOR=3
30 INPUT A,B
40 HPLOT A,B
50 GET C□
60 IF C□="K" THEN A=A+1
70 IF C□="J" THEN A=A-1
80 IF C□="I" THEN B=B-1
90 IF C□="M" THEN B=B+1
100 IF A<0 THEN A=0
110 IF A>279 THEN A=279
120 IF B<0 THEN B=0
130 IF B>159 THEN B=159
140 GOTO 40
```

А сега да изprobваме променената програма. Ако се натисне друг символ, различен от I, J, K, M, движение няма да има. Ако въобще не започне да чертае, виж дали клавиатурата не е в режим кирилица.

С това завършвам. Има и домашно. Вече знаеш да пишеш почти всякакви програми. Раз-

бира се, аз не те запознах с всички инструкции, но ти говориш и на български, без да знаеш значението на всички думи. На твоята възраст не можеш да знаеш всичко. Засега ти можеш да съставяш алгоритми за задачи, които разбириш, можеш да описваш тези алгоритми на БЕЙСИК и да изпълняваш създаваните програми на Правец – 82.

ДОМАШНА РАБОТА

1. Кои полезни съвети за писане на програми си спомняш?
2. Кои съвети за писане на програми за игри си спомняш?
3. Да се поставят инструкции REM в програмата ХУДОЖНИК—2.
4. Да се поставят инструкции за проверка на верността на въвежданието данни в програмата МУЗИКАНТ.
5. Да се поставят инструкции, задаващи условията на играта в програмата ХУДОЖНИК—2.
6. Да се измени програмата ХУДОЖНИК, така че да се въвеждат непосредствено данните от клавиатурата.
7. Да се въведе програмата:

```
10 REM НАЛЕЙ ЧАША
20 REM ПРАВИЛА НА ИГРАТА
30 INPUT "ЗНАЕШ ЛИ ПРАВИЛАТА НА ИГРАТА ";X□
40 IF X□="ДА" THEN GOTO 90
50 PRINT "С КЛАВИША J ЧАШАТА СЕ ДВИЖИ НАЛЯВО,
А С К НАДЯСНО. ПРИ ДРУГ СИМВОЛ СПИРА.
ЦЕЛТА Е ДА СЕ ХВАНЕ КАПКАТА, КОЯТО ПАДА"
60 REM ПРОЧИТАНЕ НА ПРАВИЛАТА
70 FOR I=1 TO 5000
80 NEXT I
90 REM ПОЯВА НА ЧАШАТА (X,39)
100 GR
110 COLOR=15
120 X=INT(40*RND(1))
130 PLOT X,39
140 REM ПОЯВА НА КАПКАТА(Y,S)
150 Y=INT(40*RND(1))
160 S=0
170 PLOT Y,S
```

```
180 REM ПОСОКА НА КАПКАТА
190 Q=1
200 REM ДВИЖЕНИЕ НА ЧАШАТА
210 Z=X
220 GET A□
230 IF A□="K" THEN X=X+1
240 IF A□="J" THEN X=X-1
250 IF X<0 THEN X=0
260 IF X>39 THEN X=39
270 COLOR=0
280 PLOT Z,39
290 COLOR=15
300 PLOT X,39
310 REM ДВИЖЕНИЕ НА КАПКАТА
320 IF S=39 AND Y=X THEN GOTO440
330 IF S=39 THEN GOTO100
340 IF Y=0 THEN Q=1
350 IF Y=39 THEN Q=-1
360 COLOR=0
370 PLOT Y,S
380 Y=Y+Q
390 S=S+1
400 COLOR=15
410 PLOT Y,S
420 GOTO 200
430 REM ПОПАДЕНИЕ
440 FOR I=1 TO 100
450 W=PEEK(-16336)
460 NEXT I
470 PRINT "БРАВО"
480 END
```

8. Да се разбере действието на програмата от задача 7.
9. Да се изиграят няколко игри с програмата от задача 7
10. Състави програма, която използва инструкцията GET. (

РЕШЕНИЯ НА ДОМАШНИТЕ РАБОТИ

1-ви урок

I вечер

- Изчислителен блок, клавиатура, монитор, флопидисково устройство.
 - Вътрешна памет, микропроцесор, високоговорител.
 - $65 \cdot 536 : 256 = 256$ страници.
 - Микропроцесорът.
 - Микропроцесорът.
 - $40,24 = 960$ символа.
 - Дискети, касети.
 - Във вътрешната.
 - Флопидисково устройство.
 - Флопидисково устройство, касетофон, печатащо устройство, чертаещо устройство, устройство за въвеждане на чертежи, ръчки за игра.

2-ри урок

II вечер

1. Поставяме дистигта, включваме монитора, включваме изчислителния блок.
 2. Прекъсва текущата работа на компютъра.

3. RETURN

4. а) и б)

]?"ОБИЧАЙ ГОРЕЩО И БЕЗЗАВЕТНО СВОЯ НАРОД И СВОЯТА РОДИНА!"

?"КОЙТО НИКОГА НЕ ЗАПОЧВА, ТОЙ НИКОГА НЕ ЗАВЪРШВА!"

12

]?"ВАРТБУРГ 353 W"

5. Няма разлика.

]?"ГРАД СОФИЯ"

7 a) $\left(\begin{array}{c} ++ \\ -+ \end{array} \right)$

6) ++

8 239 символа

9. Въвежда се отново инструкцията, като се поправя грешката в нея.

3-и урок

III вечер

1. а) 11 в) 57 д) 9160

- a) **2010** b) **205700** c) **11**

- 6) **40231** 7) **387481**

3. a) 1 2 | 3 6) b) c)
 | 4 123 1 23 12 3
 | 5

4.

]?"СКОРОСТТА Е "; 400/8; " КМ/Ч"
СКОРОСТТА Е 50 КМ/Ч

5.

]?"МИНАЛИ СА "; 1987 – 681; " ГОДИНИ"
МИНАЛИ СА 1306 ГОДИНИ

6.

]?"ЛИЦЕТО Е "; 50*20; " КВ.СМ"
ЛИЦЕТО Е 1000 КВ.СМ

7. а)

]?127+53; " М "; 34+57+" СМ"
180 М 91 СМ

б)

в)

]?27 – 12; " КГ "; 850–390; " Г"
15 КГ 460 Г

]?4 – 2; " МИН "; 30 – 15; " СЕК"
2 МИН 15 СЕК

4-ти урок

IV вечер

1. При появяването ѝ в програмата.

2. Не.

3. а) Започва както инструкцията LET. б) Започва с цифра.

в) Съдържа знака плюс. г) Съдържа кирилска буква. д) Започва с кирилска буква.

4. а) Не б) Да

5.

15

4

60

6.

13

0

1000

5ДНИ

7. Не, отляво на знака за равенство стои аритметичен израз, а не променлива.

5-ти урок

V вечер

4. 1. Въвежда се скоростта в S и времето в T.

2. Пресмята се $V=S/T$.

3. Отпечатва се V.

5. 1. Въвежда се скоростта във V и времето в T.

2. Пресмята се $S=V \cdot T$.

3. Отпечатва се S.

6. 1. Въвеждат се дълчините в A, B, C.

2. Пресмята се $P=A+B+C$.

3. Отпечатва се P.

8. 1. Въвежда се дължината в A.

2. Пресмята се $S=A \cdot A$.

3. Отпечатва се S.

9. 1. Въвежда се стойността на A.

2. Ако $A > 100$, отпечатва се $A > 100$ и се спира, ако не — преминава се в 3.

3. Ако $A = 100$, отпечатва се $A = 100$ и се спира, ако не — преминава се в 4.

4. Отпечатва се $A < 100$ и се спира.

11. Аз програмирам на 9 години

6-ти урок

1. Няма значение.

2.

```
10 PRINT "КОЯ Е СВЕТЛАТА СТРАНА,"  

20 PRINT "СЪБРАЛА ПЪСТРИ ЗНАМЕНА"  

30 PRINT "ОТ ДВЕТЕ ЗЕМНИ ПОЛУШАРИЯ?"  

40 PRINT "ДИМИТРОВСКА БЪЛГАРИЯ!"
```

3.

```
10 PRINT "КОЯ Е СВЕТЛАТА СТРАНА,"  

20 PRINT "СЪБРАЛА ПЪСТРИ ЗНАМЕНА"  

30 PRINT "ОТ ДВЕТЕ ЗЕМНИ ПОЛУШАРИЯ?"  

40 PRINT "ДИМИТРОВСКА БЪЛГАРИЯ!"  

50 PRINT "И КОЙ Е ТОЗИ ДИВЕН КЪТ,"  

60 PRINT "КЪДЕТО ПЕСНИТЕ ЛЕТЯТ"  

70 PRINT "С ЧАВДАРЧЕТО И КОМСОМОЛЕЦА?"  

80 PRINT "ЛЮБИМАТА НИ СТОЛИЦА!"  

90 END
```

4.

DEL 10,40

6.

 10 ?10+15
 SAVE A

7.

 RUN A
25

5. Текстът на програмата е същият както в зад. 3.

КОЯ Е СВЕТЛАТА СТРАНА,
СЪБРАЛА ПЪСТРИ ЗНАМЕНА
ОТ ДВЕТЕ ЗЕМНИ ПОЛУШАРИЯ?
ДИМИТРОВСКА БЪЛГАРИЯ!
И КОЙ Е ТОЗИ ДИВЕН КЪТ,
КЪДЕТО ПЕСНИТЕ ЛЕТЯТ
С ЧАВДАРЧЕТО И КОМСОМОЛЕЦА?
ЛЮБИМАТА НИ СТОЛИЦА!

8.

CATALOG
DISK VOLUME 254
A 002 ЗДРАВЕЙ
A 003 СТРЕЛБА
A 002 A

9. Старата се изтрива и остава само новата.

7-ми урок

1.

```
10 INPUT "ЗАДАЙ ЕДНО ЧИСЛО ";A  

20 PRINT A+1,A,A-1  

30 END
```

2.

```
10 INPUT "ДЪЛЖИНИ НА СТРАНИТЕ А,В,С "; А,В,С  
20 PRINT "ПЕРИМЕТЪРЪТ Е "; А+В+С  
30 END
```

3.

```
10 INPUT "ДЪЛЖИНИ НА СТРАНИТЕ А,В "; А,В  
20 PRINT "ПЕРИМЕТЪРЪТ Е "; 2*(А+В)  
30 END
```

4.

```
10 INPUT "ДЪЛЖИНА НА СТРАНАТА "; А  
20 PRINT "ЛИЦЕТО Е "; А*А  
30 END
```

5.

```
10 INPUT "ДЪЛЖИНА НА СТРАНАТА "; А  
20 PRINT "ПЕРИМЕТЪРЪТ Е "; 4*А  
30 END
```

6.

```
10 INPUT "ЗАДАЙ ПЪТЯ S, ВРЕМЕТО Т "; $,Т  
20 PRINT "СКОРОСТТА Е "; $/Т  
30 END
```

7.

```
10 INPUT "ЗАДАЙ ПЪТЯ S, СКОРОСТТА V "; $,V  
20 PRINT "ВРЕМЕТО Е "; $/V  
30 END
```

8.

```
10 INPUT "ЗАДАЙ ВРЕМЕТО Т, СКОРОСТТА V "; Т,В  
20 PRINT "ПЪТЯТ Е "; В*T  
30 END
```

9.

```
10 PRINT "ПРЕВРЪЩАНЕ ОТ М В СМ"  
20 INPUT "ЗАДАЙ В М "; А  
30 PRINT А*100; " СМ"  
40 END
```

8-ми урок

VIII вечер

1.

```
10 PRINT "АЗ СЪМ ДОБЪР УЧЕНИК"  
20 GOTO 10
```

2. 6. 3. Следващият ред. 4. Произволна инструкция.

5.

```

10 INPUT A
20 IF A>100 THEN GOTO 50
30 PRINT "A<=100"
40 GOTO 60
50 PRINT "A>100"
60 END

```

6.

```

10 INPUT A
20 IF A>100 THEN GOTO 60
30 IF A<100 THEN GOTO 80
40 PRINT "A=100"
50 GOTO 90
60 PRINT "A>100"
70 GOTO 90
80 PRINT "A<100"
90 END

```

7.

```

10 PRINT "НА СТУД СЕ БЕЛЧО ПЕРИ"
20 PRINT "ОТ СЛЪНЦЕТО ТРЕПЕРИ"
30 PRINT "ЩО Е ТО"
40 INPUT A□
50 IF A□="СНЯГ" THEN GOTO 70
60 GOTO 40
70 PRINT "ПОЗНА"
80 END

```

8.

```

10 INPUT "КРАСИ РЕЖЕ ХЛЯБА С НО? ";A□
20 IF A□="Ж" THEN GOTO 40
30 GOTO 10
40 PRINT "ПОЗНА"
50 END

```

9.

```

10 INPUT A□
20 INPUT B□
30 HOME
40 PRINT A□
50 INPUT C□
60 IF B□=C□ THEN GOTO 80
70 GOTO 50
80 PRINT "ПОЗНА"
90 END

```

9-ти урок

IX вечер

1.

```

10 I=1
20 PRINT "ЛЮБОМИР"
30 IF I=5 THEN GOTO 60
40 I=I+1
50 GOTO 20
60 END

```

2.

JRUN	
5	25
10	100
15	225

3. Пет числа на три пъти.

4. Въвежда стойностите на масива A.

5.

```
10 INPUT N  
20 P=1  
30 I=1  
40 P=P*I  
50 IF I=N THEN GOTO 80  
60 I=I+1  
70 GOTO 40  
80 PRINT P  
90 END
```

6.

```
5 DIM A(10)  
10 P=1  
20 I=1  
30 INPUT A(I)  
40 P=P*A(I)  
50 IF I=10 THEN GOTO 80  
60 I=I+1  
70 GOTO 30  
80 PRINT P  
90 END
```

7.

```
10 INPUT A,B  
20 S=A  
30 PRINT A  
40 S=S+1  
50 PRINT S  
60 IF S<B THEN GOTO 40  
70 END
```

9.

```
5 DIM A(10)  
10 S=0  
20 I=1  
30 INPUT A(I)  
40 IF A(I)=0 THEN GOTO 80  
50 IF I=10 THEN GOTO 100  
60 I=I+1  
70 GOTO 30  
80 S=S+1  
90 GOTO 50  
100 PRINT S  
110 END
```

8.

```
5 DIM A(10)  
10 INPUT A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8),A(9),A(10)  
20 S=0  
30 I=2  
40 S=S+A(I)  
50 IF I=10 THEN GOTO 80  
60 I=I+2  
70 GOTO 40  
80 PRINT S  
90 END
```

10-ти урок

X вечер

1. 7 пъти.

2.

```
10 FOR I=1 TO 20  
20 PRINT "ЩЕ ВНИМАВАМ В КЛАС"  
30 NEXT I  
40 END
```

3.

```
10 INPUT N  
20 S=1  
30 FOR I=1 TO N  
40 S=S*I  
50 NEXT I  
60 PRINT S  
70 END
```

4.

```
10 DIM A(10)  
20 P=1  
30 FOR I=1 TO 10  
40 INPUT A(I)  
50 P=P*A(I)  
60 NEXT I  
70 PRINT P  
80 END
```

5.

```
10 DIM A(10)  
20 S=0  
30 FOR I=1 TO 10  
40 INPUT A(I)  
50 IF A(I)>100 THEN GOTO 90  
60 NEXT I  
70 PRINT S  
80 END  
90 S=S+A(I)  
100 GOTO 60
```

6.

```
10 DIM A(10)  
20 FOR I=1 TO 10  
30 INPUT A(I)  
40 NEXT I  
50 S=0  
60 FOR I=2 TO 10 STEP 2  
70 S=S+A(I)  
80 NEXT I  
90 PRINT S  
100 END
```

7.

```
10 DIM A(10)  
20 FOR I=1 TO 10  
30 INPUT A(I)  
40 NEXT I  
50 P=1  
60 FOR I=1 TO 10  
70 IF A(I)=0 THEN GOTO 100  
80 P=P*A(I)  
90 NEXT I  
100 PRINT P  
110 END
```

8.

```
10 DIM A(10)  
20 S=0  
30 FOR I=1 TO 10  
40 INPUT A(I)  
50 IF A(I)=0 THEN S=S+1  
60 NEXT I  
70 PRINT S  
80 END
```

```

10 DIM A□(100),B□(100)
20 INPUT "БРОЙ ДУМИ ",N
30 FOR I=1 TO N
40 INPUT A□(I),B□(I)
50 NEXT I
60 HOME
70 FOR I=1 TO N
80 PRINT "ПОЛЕЗНО ЛИ Е? ";A□(I)
90 INPUT C□
100 IF B□(I)=C□ THEN GOTO 120
110 GOTO 80
120 PRINT "ПОЗНА"
130 NEXT I
140 END

```

```

]RUN
БРОЙ ДУМИ 5
? КОС,ДА
? СУХОЗЕМНА ЖАБА,ДА
? ЗЕЛЕВА ПЕПЕРУДА,НЕ
? ПОПОВО ПРАСЕ,НЕ
? КАЛИНКА,ДА
ПОЛЕЗНО ЛИ Е? КОС
? ДА
ПОЗНА
ПОЛЕЗНО ЛИ Е? СУХОЗЕМНА ЖАБА
? НЕ
ПОЛЕЗНО ЛИ Е? СУХОЗЕМНА ЖАБА
? ДА
ПОЗНА
ПОЛЕЗНО ЛИ Е? ЗЕЛЕВА ПЕПЕРУДА
? НЕ
ПОЗНА
ПОЛЕЗНО ЛИ Е? ПОПОВО ПРАСЕ
? НЕ
ПОЗНА
ПОЛЕЗНО ЛИ Е? КАЛИНКА
? ДА
ПОЗНА

```

11-ти урок

XI вечер

- Цялo число между 0 и 4.
- Буква от латинската азбука.

3.

```

10 A=INT(6*RND(1))+1
20 B=INT(6*RND(1))+1
30 INPUT "СУМАТА Е ";C
40 IF C=A+B THEN PRINT "ПОЗНА"
50 GOTO 10

```

5.

```

10 A□=CHR□(INT(26*RND(1)+65))
20 B□=CHR□(INT(26*RND(1)+65))
30 PRINT A□,B□
40 IF A□=B□ THEN PRINT "БРАВО"
50 GOTO 10

```

4.

```

10 A=INT(6*RND(1))+1
20 B=INT(6*RND(1))+1
30 PRINT A,B
40 IF A+B>6 THEN PRINT "БРАВО"
50 GOTO 10

```

6.

```

10 A=INT(100*RND(1))
20 B=INT(100*RND(1))
30 HOME
40 PRINT A;" + ";B;" =? "
50 INPUT C
60 IF C=A+B THEN GOTO 90
70 PRINT "НЕ ПОЗНА"
80 GOTO 50
90 PRINT "ПОЗНА"
100 GOTO 10

```

7.

```

10 A=INT(100*RND(1))
20 B=INT(100*RND(1))
30 HOME
35 IF B>A GOTO 110
40 PRINT A;" - ";B;" =? "
50 INPUT C
60 IF C=A-B THEN GOTO 90
70 PRINT "НЕ ПОЗНА"
80 GOTO 50
90 PRINT "ПОЗНА"
100 GOTO 10
110 C=A
120 A=B
130 B=C
140 GOTO 40

```

8.

```

10 A=INT(100*RND(1))
20 B=INT(100*RND(1))
30 IF B=0 THEN GOTO 20
40 HOME
50 PRINT A*B;" : ";B;" =? "
60 INPUT C
70 IF C=A GOTO 100
80 PRINT "НЕ ПОЗНА"
90 GOTO 60
100 PRINT "ПОЗНА"
110 GOTO 10

```

9.

```

10 DIM A□(5),B□(5)
20 FOR I=1 TO 5
30 A□(I)=CHR□(INT(26*RND(1))+65)
40 NEXT I
50 PRINT A□(1);A□(2);A□(3);A□(4);A□(5)
60 FOR I=1 TO 3500
70 NEXT I
80 HOME
90 INPUT "ЗАДАЙ БУКВИТЕ, РАЗДЕЛЕНИ С , ";B□(1),B□(2),B□(3),B□(4),B□(5)
100 FOR I=1 TO 5
110 IF A□(I)<>B□(I) GOTO 20
120 NEXT I
130 PRINT "ВЯРНО"
140 GOTO 20

```

12-ти урок

XII вечер

1. 1600 точки.

2.

```
10 GR  
20 COLOR=15  
30 HLIN 10,20 AT 1  
40 VLIN 1,20 AT 10  
50 VLIN 1,20 AT 20  
60 END
```

5.

```
40 HLIN 0,20 AT 10  
50 VLIN 10,30 AT 0  
60 VLIN 10,30 AT 20  
70 END
```

3.

```
10 GR  
20 FOR I=0 TO 15  
30 COLOR=I  
40 VLIN 0,39 AT I  
50 NEXT I  
60 END
```

4.

```
40 HLIN 0,20 AT 0  
50 VLIN 0,30 AT 0  
60 VLIN 0,30 AT 20  
70 END
```

6.

```
10 GR  
20 FOR I=0 TO 39  
30 COLOR=15  
40 VLIN 0,39 AT I  
50 COLOR=0  
60 VLIN 0,39 AT I  
70 NEXT I  
80 END
```

7.

```
10 GR  
20 X=INT(40*RND(1))  
30 Y=INT(40*RND(1))  
40 COLOR=15  
50 PLOT X,Y  
60 COLOR=0  
70 PLOT X,Y  
80 GOTO 20
```

13-ти урок

XIII вечер

1.

```
10 HGR  
20 HCOLOR=3  
30 HPLOT 70,0 TO 0,0 TO 0,100 TO 70,100 TO 70,40 TO 0,40  
40 END
```

2.

```
10 HGR  
20 HCOLOR=3  
30 HPLOT 0,0 TO 100,0 TO 100,50 TO 0,50 TO 0,0  
40 END
```

3.

```
10 HGR  
20 HCOLOR=3  
30 HPLOT 100,0 TO 100,100 TO 0,100 TO 0,0 TO 100,0  
40 END
```

4.

```
10 HGR  
20 HCOLOR=3  
30 HPLOT 100,0 TO 50,100 TO 150,100 TO 100,0  
40 END
```

5.

```
10 HGR
20 HCOLOR=3
30 HPLOT 0,50 TO 50,50 TO 50,159
40 END
```

```
6. 10 HGR
20 HCOLOR=3
30 HPLOT 50,0 TO 50,159
40 HPLOT 0,50 TO 279,50
50 END
```

7.

```
10 HGR
20 HCOLOR=3
30 HPLOT 50,0 TO 50,159
40 HPLOT 0,0 TO 179,179
50 END
```

8.

```
10 HGR
20 FOR I=0 TO 159
30 HCOLOR=3
40 HPLOT 0,I TO 279,I
50 NEXT I
60 END
```

14-ти урок**XIV вечер**

1. 256, с числата от 0 до 255.
2. 256, с числата от 0 до 255.
3. (192,255).
4. (171,128).
5. (192,64); (171,64).
6. (192,255); (171,255); (152,255); (144,255); (128,255); (114,255); (102,255); (96,255).
7. (192,64); (171,64); (152,128); (152,255); (128,128); (128,128); (128,64); (144,64); (152,128); (152,64); (171,64); (192,64); (171,64); (152,128); (152,255); (171,128); (192,128); (192,64).
8. (204,64); (192,64); (171,128); (171,255); (128,128); (144,128); (152,64); (144,64); (152,128); (171,255); (152,64); (171,64); (192,128); (204,64); (192,64); (171,128); (171,255); (128,125); (144,128); (152,64); (144,64); (152,128); (171,255); (171,128); (171,128).
- 9.

```
10 FOR I=1 TO 100
20 A=PEEK(-16336)
30 NEXT I
40 FOR I=1 TO 2000
50 NEXT I
60 GOTO 10
```

15-ти урок**XV вечер**

1. Яснота при писане, наличие на инструкция REM, проверка на въвежданите данни, наличие на инструкции PRINT.
2. Да има обяснителен текст, да се използва инструкцията GET за игри, изискващи бърза реакция.

3.

```

5 REM РИСУВАНЕ НА КАРТИНИ
7 REM ВЛИЗАНЕ ВЪВ "ФИНА" ГРАФИКА
10 HGR
20 HCOLOR=3
25 REM НАЧАЛНА ТОЧКА
30 INPUT A,B
40 HPLOT A,B
45 REM ВЪВЕЖДАНЕ НА ПОСОКА НА ДВИЖЕНИЕ
50 GET C□
55 REM ПОЛУЧАВАНЕ НА СЛЕДВАЩА ТОЧКА
60 IF C□="K" THEN A=A+1
70 IF C□="J" THEN A=A-1
80 IF C□="I" THEN B=B-1
90 IF C□="M" THEN B=B+1
95 REM ПРОВЕРКА ДАЛИ ТОЧКАТА Е В ЕКРАНА
100 IF A<0 THEN A=0
110 IF A>279 THEN A=279
120 IF B<0 THEN B=0
130 IF B>159 THEN B=159
135 REM СЛЕДВАЩА ТОЧКА
140 GOTO 40

```

4.

```

10 DIM W(1000),P(1000)
20 DATA 173,48,192,136,208,5,206,1,3,240,9
30 DATA 202,208,245,174,0,3,76,2,3,96,0,0
40 FOR X=770 TO 792
50 READ Y
60 POKE X,Y
70 NEXT X
80 HOME
90 INPUT "БРОЙ НОТИ/ОТ1 ДО 1000/ ";N
95 IF N>1000 THEN GOTO 90
100 PRINT "КОД, ТРАЙНОСТ"
110 FOR I=1 TO N
120 PRINT "НОТА ";;
130 INPUT W(I),P(I)
135 IF W(I)>255 OR W(I)<0 THEN GOTO 120
137 IF P(I)>255 OR P(I)<0 THEN GOTO 120
140 NEXT I
150 FOR I=1 TO N
160 POKE 768,W(I)
170 POKE 769,P(I)
180 CALL 770
190 NEXT I
200 END

```

5.

```
1 REM РИСУВАНЕ НА КАРТИНИ
2 INPUT "ЗНАЕШ ЛИ КАК СЕ ИГРАЕ ";X□
3 IF X□="ДА" THEN GOTO10
5 PRINT "ЗАДАЙ ТОЧКА, С І НАГОРЕ, С Ј НАЛЯВО, С К НАДЯСНО, С
   М НАДОЛУ, С ДРУГ КЛАВИШ СПИРА."
7 REM ВЛИЗАНЕ ВЪВ "ФИНА" ГРАФИКА
10 HGR
20 HCOLOR=3
25 REM НАЧАЛНА ТОЧКА
30 INPUT A,B
40 HPLOT A,B
45 REM ВЪВЕЖДАНЕ НА ПОСОКА НА ДВИЖЕНИЕ
50 GET C□
55 REM ПОЛУЧАВАНЕ НА СЛЕДВАЩА ТОЧКА
60 IF C□="K" THEN A=A+1
70 IF C□="J" THEN A=A-1
80 IF C□="I" THEN B=B-1
90 IF C□="M" THEN B=B+1
95 REM ПРОВЕРКА ДАЛИ ТОЧКАТА Е В ЕКРАНА
100 IF A<0 THEN A=0
110 IF A>279 THEN A=279
120 IF B<0 THEN B=0
130 IF B>159 THEN B=159
135 REM СЛЕДВАЩА ТОЧКА
140 GOTO 40
```

6.

```
10 GR
20 COLOR=15
30 INPUT A,B
40 PLOT A,B
50 GET C□
60 IF C□="K" THEN A=A+1
70 IF C□="J" THEN A=A-1
80 IF C□="I" THEN B=B-1
90 IF C□="M" THEN B=B+1
100 IF A<0 THEN A=0
110 IF A>39 THEN A=39
120 IF B<0 THEN B=0
130 IF B>39 THEN B=39
140 GOTO 40
```

Списък на изучаваните инструкции

CATALOG	отпечатване на имената на програмите, записани на дискета
COLOR	задаване на цвят в графичен режим
DEL	изтриване на редове от програмата
DIM	задаване размерност на масив
END	край на програмата
FOR... TO... STEP	за дадена начална стойност на брояча до определена крайна стойност на брояча с дадена стъпка да се изпълни тялото на цикъла
GET	въвеждане на символ, непосредствено след натискането му от клавиатурата, в променлива
GOTO	преминаване към ред с даден номер
GR	преминаване в графичен режим
HCOLOR	задаване на цвят при "фина графика"
HGR	преминаване във "фина" графика
HLIN	изчертаване на хоризонтална линия
HOME	изчистване на екрана
HPLOT	изчертаване на точка, отсечка или отсечки във "фина" графика
IF... GOTO...	преминаване към ред с даден номер при изпълнено условие
IF... THEN...	изпълнение на инструкцията след THEN при изпълнено условие
INPUT	въвеждане на стойност от клавиатурата по време на изпълнение на програмата
LET	инструкция за присвояване
LIST	разпечатване на редове от програмата
NEW	изчистване на вътрешната памет
NEXT	инструкция за край на цикъл
PLOT	изчертаване на точка
PRINT	отпечатване върху екрана
REM	поставяне на бележки
RUN	изпълнение на програма
SAVE	запазване на програма
TEXT	преминаване в текстов режим
VLIN	изчертаване на вертикална линия

Тълковен речник на използвани понятия и термини

Алгоритъм	Последователност от елементарни за изпълнителя действия, осигуряващи решението на дадена задача
Байт	Единица място в паметта, в което се записва един символ
БЕЙСИК	Език за програмиране
Графичен режим	Режим на монитора, в който се чертае
Данни	Числа или текст, които се въвеждат или извеждат от компютъра
Дискета	Магнитно покрит диск, служещ за външна памет при микрокомпютрите
Език за програмиране	Език за писане на програми, който обикновено се разбира от всички видове компютри, които разполагат с транслатор от този език на техния машинен език
1Кбайт	1024 байта
Инструкция	Изразно средство в езиците за програмиране, използвано обикновено за описание на действие
Клавиатура	Устройство, чрез което се въвежда информация в компютъра. Използва се по начин, подобен на пишещата машина
Компютър	Автоматично устройство, способно самостоятелно да обработва информация по зададен алгоритъм
Масив	Съвкупност от еднотипни променливи, означени с общо име
Машинен език	Език за общуване с конкретен вид компютри, чието възприемане се извършва чрез средства на самите компютри (т. е. чрез апаратни средства)
Микропроцесор	Централна част на микрокомпютъра, способна да извършва аритметични действия, да прави сравнения, да проверява условия, да управлява въвеждането и извеждането на данни
Монитор	Устройство, служещо за изобразяване на въвежданите данни и получаваните от компютъра резултати
Микрокомпютър	Компютър, реализиран чрез средства, позволящи значително намаляване на размерите му

Памет	Устройство за съхраняване на данни и програми. Бива вътрешна и външна
ПАСКАЛ	Език за програмиране
Правец-82	Български микрокомпютър
Програма	Описание на алгоритъм на машинен език или на език за програмиране
Променлива	Място в паметта, където се съхраняват данни. Характеризира се с име и стойност
Текстов режим	Режим на монитора, в който се извежда текст
Транслатор	Програма-преводач, която обикновено превежда от даден език за програмиране на машинен език
ФОРТРАН	Език за програмиране
Цикъл	Последователност от действия, които се изпълняват многократно в един и същ ред, преди да се премине към изпълнението на действието, следващо тази последователност

Използвана литература

1. Димитров, Е. BASIC за Правец-82. Корпорация "Програмни продукти и системи", СМБ. София, 1985.
2. Документация на Правец-82. Приборостроителен завод, гр. Правец.
3. Занев, Вл. Ръководство на БЕЙСИК, УСМТА. София, 1985.
4. Сп. „Млад конструктор“, изд. ЦК на ДКМС, 1984, 1985 г.

СЪДЪРЖАНИЕ

От автора за книгата	3
I. Увод	6
I урок. От какво се състоеят микрокомпютърът	10
II урок. Как се работи с компютъра	16
III урок. Компютърът изчислява	26
IV урок. Как се съхраняват данни	40
V урок. Как се пишат програми	52
VI урок. Как се въвеждат, изпълняват и съхраняват програми	58
VII урок. Как се въвеждат числа и текст по време на изпълнение на програмата	70
VIII урок. Компютърът взема решения	84
IX урок. Сърцевината на програмирането	94
X урок. Отново цикъл, но с по-малко труд	106
XI урок. В света на случайностите	118
XII урок. Компютърът като художник	126
XIII урок. Компютърът може да рисува и по-добре	136
XIV урок. Компютърът изпълнява мелодии	146
XV урок. Съставяне на игри	152
Решения на домашните работи	160
Списък на изучаваните инструкции	173
Тълковен речник на използвани понятия и термини	174
Използвана литература	175

ст. н. с. *Петър Любомиров Станчев*

АЗ ПРОГРАМИРАМ НА 9 ГОДИННИ

Рецензенти: *Евгения Сендова, Богдан Христов, Атанас Звездинов*

Редактор *Мария Дондова*

Художник на корицата, илюстрациите и оформлението *Любомир Михайлов*

Художник-редактор *Красимира Коцева*

Технически редактор *Мария Заешева*

Калиграф *Мариана Димитрова*

Коректор *Часка Алипиева*

Код: 01/9532125237/2012-3-87

Българска. Издание I. Дадена за набор на 12.V.1987 г. Подписана за печат на 3.XII.1987 г.

Излязла от печат на 28.XII.1987 г. Формат 60x84/16. Печ. коли 11. Изд. коли 10,26.

УИК 9,08. Тираж 40 000 + 103. Поръчка № 2564. Цена 1,59 лв.

Държавно издателство „Народна просвета“ – София

Набор ЕЦФН „Георги Димитров“ – София

Цветоотделяне ДП „Балкан“ – София

Печат ДП „Дунав“ – Русе