

СДРУЖЕНИЕ „ПРОГРАМНИ ПРОДУКТИ И СИСТЕМИ — АВАНГАРД“



ГРАФОРТ

ПРОГРАМЕН ПРОДУКТ

ИЗДАВА СК "НППФ"

КОРПОРАЦИЯ "ПРОГРАМНИ ПРОДУКТИ И СИСТЕМИ"
ИНЖЕНЕРНО-ВНЕДРИТЕЛСКО СТОПАНСКО ДРУЖЕСТВО "АВАНГАРД"

ОДРУЖЕНИЕ "ПРОГРАМНИ ПРОДУКТИ И СИСТЕМИ - АВАНГАРД"

ПРОГРАМЕН ПРОДУКТ

Г р а ф о р т

Рег. номер 1.А097.00826-01

Разработил:
Орлин Вълчев

ИЗДАВА СТОПАНСКИ КОМБИНАТ "НАЦИОНАЛЕН ПРОГРАМЕН И ПРОЕКТЕН ФОНД"

С Ъ Д Ъ Р Ж А Н И Е

УВОД	5
I ПРЕДВАРИТЕЛНИ СВЕДЕНИЯ	6
I.1 ХАРДУЕРНИ ИЗИСКВАНИЯ	6
I.2 ОПЕРАЦИОННА СИСТЕМА	6
I.3 ДИСТРИБУТИВЕН НОСИТЕЛ	6
I.4 ПЪРВИ СВЕДЕНИЯ ЗА ЕЗИКА	7
A. СТЕК ЗА ДАННИ	7
B. ЧИСЛА	7
C. НЯКОЛКО СПИТА	7
II ОПИСАНИЕ НА КОМАНДИТЕ	9
II.1 ДУМИ ЗА РАБОТА СЪС СТЕКА	9
II.2 АРИТМЕТИЧНИ ДУМИ	11
II.3 ДЕФИНИРАНЕ НА НОВИ ДУМИ	14
A. ПРЕМАХВАНЕ НА ДУМИ ОТ БИБЛИОТЕКАТА	15
II.4 СЪЗДАВАНЕ НА ЦИКЛИ	16
A. СТРУКТУРА DO - LOOP	16
B. ВЪЗВРАТЕН СТЕК	18
II.5 СРАВНЯВАНЕ НА ЧИСЛА	19
II.6 ВЗИМАНЕ НА РЕШЕНИЯ И ДУМИ ЗА ПРЕХОД	21
A. IF - THEN	21
B. IF - ELSE - THEN	22
C. BEGIN - UNTIL	25
D. BEGIN - WHILE - REPEAT	26
E. CASE: - THEN	27
II.7 ПРОГРАМНА СТРУКТУРА	30
A. ИЗПОЛЗВАНЕ НА ДУМИ	30
B. ТЕКСТ ИЛИ ГРАФИКА	30
C. ПАМЕТ (PROTOR И \$LIST)	31
II.8 РАБОТА С КЛАВИАТУРАТА	32
A. КИРИЛИЦА И ЛАТИНИЦА	32
B. ПРИДВИЖВАНЕ НА КУРСОРА	32
II.9 ТЕКСТОВ РЕДАКТОР	32
A. LIST (L)	33
B. AUTONUM (A)	34
C. DELETE (D)	34
D. ERASE (E)	34
E. АВТОМАТИЧНО ВМЪКВАНЕ	34
F. INSERT (I)	35
G. SAVE (S)	36
H. GET (G)	36
I. DOS КОМАНДИ (ID)	36
J. ОТПЕЧАТВАНЕ НА ФАЙЛОВЕ	37
K. СЪОБРАЖЕНИЯ ЗА ПАМЕТТА	37
L. BYE (B)	38
II.10 КОМПИЛИРАНЕ НА ПРОГРАМИ	38
A. КОМЕНТАРИ	38
II.11 ДУМИ ФОРМАТИРАЩИ ТЕКСТ	39
II.12 ДУМИ ЗА ЗАПАЗВАНЕ И ИЗВЛИЧАНЕ ОТ ПАМЕТТА	40
A. РЕЕК И РОКЕ .A.	41
B. ПРОМЕНЛИВИ	42
II.13 НИЗОВЕ	43
A. ДЕФИНИРАНЕ НА НИЗОВЕ	43
B. ИЗПОЛЗВАНЕ НА НИЗОВЕ	44
C. ПРЕОБРАЗУВАНЕ НА НИЗОВЕ	45
D. СИСТЕМЕН НИЗ (PAD)	46

E.	ДУМИ ЗА РАБОТА С НИЗОВЕ НА ДИСКЕТАТА	46
II.14	ДУМИ ЗА РАБОТА НАД ЕДИНИЧНИ СИМВОЛИ	48
II.15	ЧИСЛА В ДРУГИ БРОЙНИ СИСТЕМИ	49
II.16	ИЗПОЛЗВАНЕ НА DOS ОТ ГРАФОРТ	49
II.17	ДУМИ ЗА УПРАВЛЕНИЕ НА ПРОГРАМИ	50
II.18	ЗАПАЗВАНЕ НА СИСТЕМАТА	50
II.19	ПРЕПОКРИВАЩИ СЕ СТРУКТУРИ (ОВЪРЛЕИ)	51
II.20	ПРЕМЕСТВАНЕ В ПАМЕТА	51
II.21	ОТКРИВАНЕ АДРЕСИТЕ НА ДУМИ	51
II.22	ИЗПЪЛНЕНИЕ НА ПОДПРОГРАМИ НА МАШИЕН ЕЗИК	52
II.23	КОМПИЛИРАНЕ НА ЧИСЛОВИ ТАБЛИЦИ	52
II.24	ПРИКЛЮЧВАНЕ НА РАБОТАТА С ГРАФОРТ	52
II.25	ГРАФИКИ С ГРАФОРТ	52
II.26	ДВУМЕРНИ ГРАФИЧНИ ФУНКЦИИ	53
A.	PLOT, LINE И FILL	53
B.	COLOR	55
C.	UNPLOT, UNLINE И EMPTY	55
D.	INVERSE И NORMAL	56
E.	ORMODE И EXMODE	57
F.	GREEK	58
II.27	ГРАФИКИ ТИП "КОСТЕНУРКА"	59
A.	PENUP	59
B.	PENDOWN	59
C.	MOVE	59
D.	TURNTD	59
E.	TURN	59
F.	MOVE TO	60
II.28	СПЕЦИАЛНИ СИМВОЛИ НА ИЗХОД	62
II.29	СМЯНА НА РАЗМЕРА И ЦВЕТА НА СИМВОЛИТЕ	62
II.30	ИЗБОР НА ШРИФТ	63
II.31	СИМВОЛЕН РЕДАКТОР (С.РЕДАКТОР)	64
A.	ИЗБОР И ИЗВЕЖДАНЕ НА НАБОР СИМВОЛИ	65
B.	ИЗВЕЖДАНЕ НА СИМВОЛЕН БЛОК	65
C.	ДЕФИНИРАНЕ НА ФИГУРИ	67
D.	ЗАПАЗВАНЕ НА СИМВОЛЕН НАБОР	67
II.32	БЛОКОВО ИЗВЕЖДАНЕ ОТ ГРАФОРТ	67
A.	РАЗМЕР НА БЛОК (BLKSIZE)	68
B.	ИЗВЕЖДАНЕ НА БЛОК (PUTBLK)	68
C.	РЕЖИМ "ИЗКЛЮЧВАЩО ИЛИ" (EXMODE)	69
II.33	ОБЕМНИ ГРАФИКИ	70
A.	ФОРМАТ НА ОБЕМНА ФИГУРА	72
II.34	ПАРАМЕТРИ НА ФИГУРАТА	72
A.	ВЪРТЕНЕ	72
B.	СКАЛА	73
C.	ОБЕМНА ПЕРСПЕКТИВА	74
D.	ПОЗИЦИЯ	74
E.	ТРАНСМИРАНЕ	75
F.	ЦВЯТ НА ОБЕКТА	76
G.	МАНИПУЛАТОР ЗА ФИГУРИ "ОГЛЕД"	76
II.35	РЕДАКТОР ЗА ФИГУРИ (Ф.РЕДАКТОР)	79
A.	ИЗБОР НА АДРЕС И ФИГУРА	79
B.	ИЗБОР НА ГЛЕДНА ТОЧКА	80
C.	ДАНИИ ЗА ФАЙЛА С ФИГУРАТА	80
D.	СЪЗДАВАНЕ НА НОВИ ФИГУРИ	81
E.	ЗАПАЗВАНЕ НА ФАЙЛА С ФИГУРАТА	82
II.36	НАЧНИ ЗА ОБЕМНО ИЗВЕЖДАНЕ	82
II.37	ПРОСЛА	83

A. ПАРАМЕТРИ	83
B. ВЪВЕЖДАНЕ НА ДАННИ ОТ КЛАВИАТУРАТА	84
C. ВЪВЕЖДАНЕ НА ДАННИ ОТ ДИСКЕТАТА	84
D. СЪОБРАЖЕНИЯ ЗА ПАМЕТТА	85
II.38 МУЗИКА	85
A. VOICE	85
B. NOTE	85
C. СПРЕДЕЛЯНЕ ПРОДЪЛЖИТЕЛНОСТТА И ВИСОЧИНАТА	86
D. ПОЛЕЗНИ МУЗИКАЛНИ ДУМИ	86
ПРИЛОЖЕНИЕ А - СПИСЪК НА ДУМИТЕ ОТ БИБЛИОТЕКАТА ПО РЕДА НА ПОЯВЯВАНЕТО ИМ В РЪКОВОДСТВОТО	88
ПРИЛОЖЕНИЕ В - СПИСЪК НА ДУМИТЕ ОТ БИБЛИОТЕКАТА ПО АЗБУЧЕН РЕД	97
ПРИЛОЖЕНИЕ С - СПРАВОЧНИК НА ДИСКЕТИТЕ.....	106

У В О Д

ГРАФОРТ Е РАЗНОВИДНОСТ НА ИЗВЕСТНИЯ ЕЗИК ФОРТ. ТАЗИ ВЕРСИЯ НА ФОРТ Е НАПИСАНА СПЕЦИАЛНО ЗА КОМПЮТРИ ТИП "ПРАВЕЦ 82" И НЕ ПОДДЪРЖА ПРЕНΟΣИМ КОД - ПРОГРАМА НАПИСАНА НА ГРАФОРТ МОЖЕ ДА БЪДЕ ИЗПЪЛНЕНА САМО НА КОМПЮТРИ ОТ СЪЩИЯ ТИП. ПО НАТАТЪК ЩЕ БЪДЕ ОПИСАНА ПРОГРАМНАТА СРЕДА, В КОЯТО ГРАФОРТ РАБОТИ, КАКТО И САМИЯ ЕЗИК ЗА ПРОГРАМИРАНЕ. ТОВА Е КОМПЛЕКС ОТ ПРОГРАМИ, КОИТО ОСИГУРЯВАТ СЪЗДАВАНЕТО И ИЗПЪЛНЕНИЕТО НА ПРОГРАМИ НАПИСАНИ НА ГРАФОРТ. ЕТО И НЯКОИ ОТ ВЪЗМОЖНОСТИТЕ НА ГРАФОРТ:

- ЧЕРТАНЕ НА ЦВЕТНИ ОБЕМНИ ГРАФИКИ С ДОСТАТЪЧНА ЗА ЦЕЛИТЕ НА АНИМАЦИЯТА СКОРОСТ;
- ВГРАДЕН МУЗИКАЛЕН СИНТЕЗАТОР;
- ИЗВЕЖДАНЕ НА ТЕКСТОВЕ В РАЗЛИЧНА ГОЛЕМИНА, ЦВЯТ ИЛИ ШРИФТ, СМЕСЕНИ БЕЗ ОГРАНИЧЕНИЯ С ГРАФИКИ;
- ВЪЗМОЖНОСТ ЗА СЪЗДАВАНЕ НА СОБСТВЕНИ НАБОРИ СИМВОЛИ;
- БЛОКОВО ИЗВЕЖДАНЕ НА СИМВОЛИ С ЦЕЛ ДВУМЕРНА АНИМИРАНА ГРАФИКА.

НАСТОЯЩЕТО РЪКОВОДСТВО СЪДЪРЖА:

- ОПИСАНИЕ НА КОМАНДИТЕ НА ГРАФОРТ;
- ОПИСАНИЕ НА ПОМОЩНИТЕ ПРОГРАМИ ОТ ОБКРЪЖЕНИЕТО НА ГРАФОРТ;
- КРАТЪК ПРЕГЛЕД НА КОМАНДИТЕ НА ЕЗИКА С ЦЕЛ БЪРЗИ СПРАВКИ.

I ПРЕДВАРИТЕЛНИ СВЕДЕНИЯ

I.1 ХАРДУЕРНИ ИЗИСКВАНИЯ

- КОМПЮТЪР "ПРАВЕЦ 82" ИЛИ СЪВМЕСТИМ С 48К "РАМ"-ПАМЕТ.
- ПОНЕ ЕДНО ФЛОПИДИСКОВО УСТРОЙСТВО.
- ЧЕРНОБЯЛ ИЛИ ЦВЕТЕН МОНИТОР.
- ПО ВЪЗМОЖНОСТ 16К "РАМ"-КАРТА.

I.2 ОПЕРАЦИОННА СИСТЕМА

ГРАФОРТ ИЗПОЛЗВА СТАНДАРТНАТА ЗА ТОЗИ ТИП КОМПЮТРИ СИСТЕМА DOS 3.3. С ЦЕЛ, ОБАЧЕ, ОПТИМАЛНО ИЗПОЛЗВАНЕ НА ПАМЕТА ПРЕДОСТАВЕНА ОТ ЕВЕНТУАЛНО НАЛИЧНИ "ЕЗИКОВА" ИЛИ "РАМ" КАРТИ, ИЗПОЛЗВАНИЯТ ТУК DOS Е ЛЕКО МОДИФИЦИРАН - ТОЙ СЕ ЗАРЕЖДА В ЕЗИКОВАТА КАРТА, КАТО ПО ТОЗИ НАЧИН ОСТАВЯ ПОВЕЧЕ МЯСТО В ОПЕРАТИВНАТА ПАМЕТ ЗА ПОТРЕБИТЕЛСКА РАБОТА.

I.3 ДИСТРИБУТИВЕН НОСИТЕЛ

СИСТЕМАТА ГРАФОРТ СЕ РАЗПРОСТРАНЯВА ВЪРХУ ДВЕ ДИСКЕТИ. НА ПЪРВАТА ОТ ТЯХ СЕ НАМИРАТ ОСВЕН ОСНОВНИТЕ ПРОГРАМИ И ДЕМОНСТРАЦИОННИТЕ ПРИМЕРИ, А НА ВТОРАТА ВМЕСТО ДЕМОНСТРАЦИОННИТЕ ПРИМЕРИ СЕ НАМИРАТ ПОМОЩНИ ФАЙЛОВЕ СЪДЪРЖАЩИ ОПИСАНИЕТО НА ВСЯКА ЕДНА ОТ 147-ТЕ КОМАНДИ НА ЕЗИКА ГРАФОРТ. И В ДВАТА ВАРИАНТА НА "ГРАФОРТ"-ДИСКЕТАТА НЯМА ДОСТАТЪЧНО МЯСТО ЗА ПОТРЕБИТЕЛСКИ ПРОГРАМИ. ТОВА ОБАЧЕ НЕ ТРЯБВА ДА ВИ БЕЗПОКОИ. ДИСКЕТАТА МОЖЕ (И ДОРИ Е ЗА ПРЕПОРЪЧВАНЕ) ДА БЪДЕ ПРЕЗАПИСАНА С КОЯ ДА Е ПОМОЩНА ПРОГРАМА - НАПРИМЕР "FID", "COPY", "LOCKSMITH", "SUPER COPY" И Т.Н. ОТ КОПИЕТО МОГАТ ДА БЪДАТ ИЗТРИТИ ПО-ГОЛЯМАТА ЧАСТ ОТ ФАЙЛОВЕТЕ С ПРОГРАМИ, ТЪИ КАТО САМО НЯКОЛКО СА ЗАДЪЛЖИТЕЛНИ. В КРАЯ НА РЪКОВОДСТВОТО ЩЕ НАМЕРИТЕ СПИСК НА ФАЙЛОВЕТЕ ВЪРХУ ДИСТРИБУТИВНИТЕ ДИСКЕТИ, КАКТО И УКАЗАНИЯ КОИ СА НЕОБХОДИМИ. НЕУЖНИТЕ ФАЙЛОВЕ МОГАТ ДА БЪДАТ ИЗТРИТИ ЧРЕЗ DOS-КОМАНДАТА DELETE. ЗА ДА ИЗДАДЕТЕ DOS-КОМАНДИ ОТ ГРАФОРТ, ТРЯБВА ДА СПАЗИТЕ СЛЕДНАТА ПОСЛЕДОВАТЕЛНОСТ ОТ ДЕЙСТВИЯ:

- СЛЕД ЗАРЕЖДАНЕ (ОТГОВАРЯЙКИ С НЕ НА ВЪПРОСА "ДЕМОНСТРАЦИЯ (Д/Н)?") НАПИШЕТЕ "EDIT"
- НАТИСНЕТЕ КЛАВИША "МК" И ЗАЕДНО С НЕГО "Д/Д" (ПО-НАТАТЪК ЩЕ ГО БЕЛЕЖИМ С [D])

- В ОТГОВОР НА "ВЪВЕДИ DOS КОМАНДА:" МОГАТ ДА БЪДАТ ДАДЕНИ ВСИЧКИ DOS-КОМАНДИ - CATALOG, DELETE, RENAME И Т.Н. ЗАРЕЖДАНЕ НА СИСТЕМАТА.

ГРАФОРТ СЕ РАЗПРОСТРАНЯВА В ДВА ВАРИАНТА - ВЪРХУ ПЪРВАТА ДИСКЕТА ОСВЕН СИСТЕМАТА СА ЗАПИСАНИ И ДЕМОНСТРАЦИОННИ ПРИМЕРИ, А ВЪРХУ ВТОРАТА - ВМЕСТО ДЕМОНСТРАЦИЯТА ДИСКОВОТО МЯСТО Е ИЗПОЛЗВАНО ЗА ДА БЪДЕ ЗАПИСАНА ПОМОЩНА ИНФОРМАЦИЯ. ТАКА СЛЕД ЗАРЕЖДАНЕ НА ПЪРВАТА ДИСКЕТА ЩЕ ВИДИТЕ СЪОБЩЕНИЕТО "ДЕМОНСТРАЦИЯ (Д/Н) ?". АКО ОТГОВОРИТЕ С Д(А), ЩЕ ВИДИТЕ КАКВО МОЖЕ ГРАФОРТ. СЛЕД ЗАРЕЖДАНЕ НА ВТОРАТА ДИСКЕТА, ТАКОВА СЪОБЩЕНИЕ НЯМА ДА СЕ ПОЯВИ, НО ЗА СМЕТКА НА ТОВА МОЖЕТЕ ДА НАПИШЕТЕ "?", ПОСЛЕДВАНО ОТ RETURN: ТАКА ЩЕ ИМАТЕ ДОСТЪП ДО ПОМОЩНИЯ ФАЙЛ, КАТО ПОСРЕДСТВОМ МЕНЮ ЩЕ МОЖЕТЕ ДА ВИДИТЕ КРАТКО ОБЯСНЕНИЕ НА ВСЯКА ЕДНА ОТ 147-ТЕ КОМАНДИ НА ГРАФОРТ. РАЗБИРА СЕ, САМО ТОВА ОПИСАНИЕ НЯМА ДА Е ДОСТАТЪЧНО ЗА ДА РАБОТИТЕ С ГРАФОРТ - ПРОЧЕТЕТЕ ВНИМАТЕЛНО ТОВА РЪКОВОДСТВО, А СЛЕД ТОВА ПОМОЩНАТА ИНФОРМАЦИЯ СЕ ВИ БЪДЕ ПОЛЕЗНА ПРИ РАБОТА.

1.4 ПЪРВИ СВЕДЕНИЯ ЗА ЕЗИКА

МОЖЕМ ДА РАЗДЕЛИМ ЕЗИКЪТ НА ДВЕ ОСНОВНИ ЧАСТИ:

- ПОДПРОГРАМИ НА КОМПИЛАТОРА;
- БИБЛИОТЕКА ОТ ДУМИ.

БИБЛИОТЕКАТА ОТ ДУМИ Е ВИДИМАТА ЧАСТ ОТ СИСТЕМАТА. ТЯ Е СЪСТАВЕНА ОТ ГОЛЯМ НАБОР "ДУМИ" НА ГРАФОРТ. LIST Е ДУМАТА, КОЯТО ПОЗВОЛЯВА ИЗВЕЖДАНЕТО ВЪРХУ ЕКРАНА НА ВСИЧКИ ДУМИ НА ГРАФОРТ. ВСЯКА ГРАФОРТ-ДУМА ИЗПЪЛНЯВА ОПРЕДЕЛЕНА ЗАДАЧА. ДУМИТЕ МОГАТ ДА БЪДАТ ПРОГРАМИ, ПОДПРОГРАМИ, ПРОМЕНЛИВИ ИЛИ НИЗОВЕ. ЕДНА ПРОГРАМА НА ГРАФОРТ ПРЕДСТАВЛЯВА НЕ ПРОГРАМНИ РЕДОВЕ, А НИЗ ОТ ДУМИ.

ИМЕТО НА ДУМА МОЖЕ ДА БЪДЕ КАКЪВ ДА Е НАБОР ОТ ASCII-СИМБОЛИ С ИЗКЛЮЧЕНИЕ НА ИНТЕРВАЛ И RETURN. ИНТЕРВАЛЪТ РАЗДЕЛЯ ДУМИТЕ ЕДНА ОТ ДРУГА, А RETURN ПРЕДИЗВИКВА КОМПИЛИРАНЕ НА ВЪВЕДЕНИЯ РЕД И ЕВЕНТУАЛНО НЕГОВОТО НЕЗАБАВНО ИЗПЪЛНЕНИЕ.

А. СТЕК ЗА ДАННИ

МОЖЕМ ДА СИ ПРЕДСТАВИМ СТЕКА (STACK) КАТО КУП ЧИНИИ - МОЖЕМ ДА СЛАГАМЕ ЧИНИИ ОТГОРЕ И ДА ВЗИМАМЕ ПАК ОТТАМ. ПЪРВАТА ЧИНИЯ КОЯТО МОЖЕМ ДА ВЗЕМЕМ ОТ КУПА Е ПОСЛЕДНАТА СЛОЖЕНА ПРЕДИ ТОВА. ТОЗИ ПРИНЦИП НА СТЕКОВА ОРГАНИЗАЦИЯ СЕ НАРИЧА LIFO (LAST IN FIRST OUT) - ПОСЛЕДЕН ВЛЯЗЪЛ, ПЪРВИ ИЗЛЯЗЪЛ. КОГАТО СЕ ИЗПЪЛНЯВА ЕДНА ГРАФОРТ-ДУМА, НАПРИМЕР "+", АРГУМЕНТИТЕ И ТРЯБВА ПРЕДВАРИТЕЛНО ДА БЪДАТ ДОСТЪПНИ (В СЛУЧАЯ ДВЕТЕ СЪБИРАЕМИ). ТЕ СЕ НАМИРАТ В СТЕКА ЗА ДАННИ. ОТ ДРУГА СТРАНА РЕЗУЛТАТЪТ ТРЯБВА ДА БЪДЕ ИЗВЕДЕН НЯКЪДЕ - ТОВА МЯСТО Е СЪЩО СТЕКА ЗА ДАННИ. ПОСНАТАТЪК СТЕКА ЗА ДАННИ ЩЕ НАРИЧАМЕ САМО "СТЕК".

В. ЧИСЛА

ГРАФОРТ Е ЦЕЛОЧИСЛЕН ЕЗИК. ИЗПОЛЗВАТ СЕ ЧИСЛА МЕЖДУ -32768 И 32767, Т.Е. ЦЕЛИ ЧИСЛА СЪС ЗНАК ЗАПИСАНИ В ДВА БАЙТА. АКО СЕ ВЪВЕДЕ ЧИСЛО ИЗВЪН ТОЗИ ИНТЕРВАЛ, ТО ТО ЩЕ БЪДЕ ПРЕОБРАЗУВАНО В НАИ-БЛИЗКОТО ЧИСЛО ВЪТРЕ В ИНТЕРВАЛА. НЯКОИ ОПЕРАЦИИ (НАПРИМЕР "ДЕЛЕНЕ") ЩЕ ОТРЕЖАТ ДРОБНАТА ЧАСТ ОТ РЕЗУЛТАТА - 5/2 ЩЕ БЪДЕ 2.

С. НЯКОЛКО ОПИТА

ЗАРЕДЕТЕ ГРАФОРТ ОТ ДИСКЕТАТА. НАПИШЕТЕ:

```
#> 3 4 5
```

ЧИСЛАТА 3, 4 И 5 СА ЗАПИСАНИ В СТЕКА. ДА ПРОВЕРИМ:

```
#> STACK
```

```
[3]
```

```
[4]
```

```
[5]
```

```
#> 6 7
```

```
[3]
```

```
[4]
```

```
[5]
```

```
[6]
```

```
[7]
```

```
#>
```

ВТОРИЯТ ПЪТ ЧИСЛАТА 6 И 7 БЯХА ПРИБАВЕНИ НА ВЪРХА НА СТЕКА (НАИ-ГОРНОТО Е ПОСЛЕДНОТО ВЪВЕДЕНО Т.Е. 7). ДУМАТА "." ИЗВЛИЧА ЧИСЛО

ОТ СТЕКА И ГО ИЗВЕЖДА НА ЕКРАНА.

#> .
7
[3]
[4]
[5]
[6]

#> +

[3]
[4]
[11]

ДУМАТА "+" ИЗВАДИ ОТ СТЕКА НАЙ-ГОРНИТЕ ДВЕ ЧИСЛА (5 И 6), А НА ТЯХНО МЯСТО ЗАПИСА СУМАТА ИМ (11). НЕКА СЕГА РАЗПЕЧАТАМЕ СТЕКА:

#>
1143

#>

ТРИТЕ СТОИНОСТИ ОТ СТЕКА СЕ ИЗВЕЖДАТ БЕЗ ИНТЕРВАЛИ ПОМЕЖДУ ИМ. В ТОВА РЪКОВОДСТВО, А И В ПОМОЩНИЯ ФАЙЛ (ИЗВИКВАН ЧРЕЗ ДУМАТА "?"), СТЕКА ЩЕ БЪДЕ МАРКИРАН ЧРЕЗ ТРИ ТИРЕТА. ЗА ВСЯКА ДУМА НА ГРАФОРТ ЩЕ БЪДЕ ДАВАНО СЪСТОЯНИЕТО НА СТЕКА ПРЕДИ И СЛЕД ИЗПЪЛНЕНИЕ НА ДУМАТА. ВЪРХЪТ НА СТЕКА Е В ДЯСНО - ТАКА НОТАЦИЯТА "M N -" ОЗНАЧАВА, ЧЕ Е ВЪВЕДЕНО M, А СЛЕД НЕГО N. А ЕТО И ТОКУ ЩО РАЗГЛЕДАНИТЕ ДУМИ:

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====		
+	M N - N - -	N=M+N
STACK	- - - - -	ДИСПЛЕИ ИЛИ НЕ НА СТЕКА
.	N - - - -	ИЗВЕЖДА ЧИСЛОТО N
LIST	- - - - -	ИЗВЕЖДА ДУМИТЕ ОТ БИБЛИОТЕКАТА

РАЗБИРА СЕ В СТЕКА МОЖЕ ДА ИМА МНОГО ДРУГИ ЧИСЛА ПРЕДИ ТЕЗИ МАРКИРАНИ НА ДИАГРАМИТЕ, НО ТЕ НЕ КАСАЯТ СЪОТВЕТНАТА ДУМА.

II ОПИСАНИЕ НА КОМАНДИТЕ

II.1 ДУМИ ЗА РАБОТА СЪС СТЕКА

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
----------	-----------	-----------

=====ПРЕДИ=====СЛЕД=====

SWAP	M N - N M -	
	РАЗМЕСТВА ГОРНИТЕ 2 ЧИСЛА	

DROP	N - - - -	
	МАХА ЧИСЛО ОТ СТЕКА	

DUP	N - - N N -	
	ДУБЛИРА СТОИНОСТТА	

OVER	M N - M N M	
	КОПИРА 2-РАТА СТОИНОСТ	

PICK	..M N - ..M N -	
	КОПИРА N-ТАТА СТОИНОСТ	

А ЕТО И НЯКОЛКО ПРИМЕРА:

#> STACK 1 2 3

[1]
[2]
[3]

#> SWAP

[1]
[3]
[2] РАЗМЕНЯ ГОРНИТЕ 2 ЧИСЛА

#> DUP

[1]
[3]
[2]
[2] ДУБЛИРА НАИ-ГОРНОТО ЧИСЛО

#> DROP

[1]
[3]
[2] МАХА НАИ-ГОРНОТО ЧИСЛО

#> OVER

[1]

[3]

[2]

[3]

ДУБЛИРА ПРЕДПОСЛЕДНОТО ЧИСЛО

#> 4 PICK

[1]

[3]

[2]

[3]

[1] ДУБЛИРА 4-ТОТО В СТЕКА ЧИСЛО
ПОСЛЕДНОТО МОЖЕШЕ ДА БЪДЕ НАПИСАНО НА ДВА ЕТАПА:
#> 4

[1] <-----+ 4

[3] ! 3

[2] ! 2

[3] -----+ 1

[4]

#> PICK

+----- [1]

! [3]

! [2]

! [3]

+-----> [1]

PICK ВЗЕ СТОИНОСТ ЗА АРГУМЕНТА СИ ОТ СТЕКА (4), А ВЪРНА В СТЕКА РЕЗУЛТАТА ОТ ОПЕРАЦИЯТА (ЧЕТВЪРТТО ЧИСЛО НАЗАД В СТЕКА).

#> DROP DROP . . .

231

СЛЕД ОТСТРАНЯВАНЕТО НА ВРЪХНИТЕ ДВЕ ЧИСЛА, СЕ ОТПЕЧАТВАТ ЕДНО СЛЕД ДРУГО ОСТАНАЛИТЕ ВЪТРЕ ТРИ ЧИСЛА. СТЕКЪТ ВЕЧЕ Е ПРАЗЕН.

II.2 АРИТМЕТИЧНИ ДУМИ

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
 =====ПРЕДИ=====СЛЕД=====

CHS M - - N - - CHANGE SIGN
 ПРОМЯНА НА ЗНАКА N = -M

ABS N - - M - - ABSOLUTE
 АБСОЛЮТНА СТОЙНОСТ M=!N!

SGN M - - N - - SIGN
 ЗНАК НА ЧИСЛОТО N=-1,0,1

RND - - - N - - RANDOM
 СЛУЧАЙНО ЧИСЛО -32K ДО 32K

RNDB - - - N - - RANDOM BYTE
 СЛУЧАЙНО ЧИСЛО 0 ДО 255

MIN M N - N - - N=MIN(M,N)
 МИНИМУМ НА ДВЕ ЧИСЛА

MAX M N - N - - N=MAX(M,N)
 МАКСИМУМ НА ДВЕ ЧИСЛА

MOD M N - N - -
 ОСТАТЪК СЛЕД M/N

/ M N - N - - N=M/N

* M N - N - - N=M*N

+ M N - N - - N=M+N

- M N - N - - N=M-N

SIN M - - N - - N=SIN(M)
 СКАЛАРЕН СИНУС (-128,127)

ЕТО И НЯКОЛКО ПРИМЕРА:

#) 23 5 / .
 4

#> 23 5 MOD .
3

23 ДЕЛЕНО НА 5 Е 4 С ОСТАТЪК 3.

#> STACK 6 CHS

[-6]
#> ABS .
6

#> 7 11 MIN .
7

#> -7 SGN .
-1

#> RND .
19352

#> RNDV .
132

ПОСЛЕДНИТЕ ДВЕ ДУМИ ГЕНЕРИРАТ ПСЕВДОСЛУЧАЙНИ ЧИСЛА, ТАКА ЧЕ КОГАТО ГИ ТЕСТВАТЕ ЩЕ ПОЛУЧИТЕ, НАИ-ВЕРОЯТНО, ДРУГИ ЧИСЛА.

ДОТУК ПРЕДПОЛАГАХМЕ, ЧЕ СТЕКА Е ПРАЗЕН ПРЕДИ ПРИМЕРИТЕ. ЗА ДА ГО ИЗЧИСТИМ, ИМА ДВА НАЧИНА:

МНОГОКРАТНО ИЗДАВАНЕ НА DROP ИЛИ ".";

ИЗДАВАНЕ НА ДУМАТА ABORT. УДОБНА Е В ДИАЛОГОВ (ОТ КЛАВИАТУРАТА), НО НЕ И В ПРОГРАМЕН РЕЖИМ. ABORT РЕСТАРТИРА СИСТЕМАТА, КАТО ВРЪЩА ВСИЧКО В НАЧАЛНО СЪСТОЯНИЕ.

ОПЕРАТОР	С Т А С К	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

ABORT - - - - -
 РЕСТАРТИРА ГРАФОРТ

ВИДЯХМЕ, ЧЕ СЪБИРАНЕТО НА ДВЕ ЧИСЛА СТАВА ПО СЛЕДНИЯ НАЧИН:

#> 3 4 + .
7

ЧИСЛАТА (АРГУМЕНТИТЕ) ПРЕДШЕСТВАТ ДУМАТА (ФУНКЦИЯТА). ТОВА СЕ НАРИЧА "ОБРАТНО ПОЛСКО ПРЕДСТАВЯНЕ" И СЕ ИЗПОЛЗВА ВЪВ ВСИЧКИ ВЕРСИИ НА ЕЗИКА ФОРТ.

НЕКА РАЗГЛЕДАМЕ АРИТМЕТИЧЕН ИЗРАЗ ОТ БЕЙСИК.

$X = (3+5) * (7-3)$

ЕТО КАК ЩЕ ИЗГЛЕЖДА ТОИ НА ГРАФОРТ:

```
#> 3 5
```

```
[3]
```

```
[5]
```

```
#> +
```

```
[8]
```

```
#> 7 3
```

```
[8]
```

```
[7]
```

```
[3]
```

```
#> -
```

```
[8]
```

```
[4]
```

```
#> *
```

```
[32]
```

```
#> .
```

```
32
```

```
#>
```

РАЗБИРА СЕ, ТОВА МОЖЕ ДА БЪДЕ НАПИСАНО НА ЕДИН РЕД:

```
#> 3 5 + 7 3 - * .
```

```
32
```

А ЕТО И ПОВДИГАНЕ НА 3-ТА СТЕПЕН:

```
#> 4 DUP DUP * * .
```

```
64
```

ОТПЕЧАТВАНЕ (ИЗВЕЖДАНЕ НА ЕКРАНА) НА ТЕКСТ В ГРАФОРТ СТАВА ПО СЛЕДНИЯ НАЧИН:

```
#> PRINT " ТЕКСТ ЗА ИЗВЕЖДАНЕ "  
ТЕКСТ ЗА ИЗВЕЖДАНЕ
```

КАВИЧКИТЕ (") СА ГРАФОРТ ДУМА И ТРЯБВА ДА БЪДАТ ПРЕДШЕСТВАНИ И ПОСЛЕДВАНИ ОТ ИНТЕРВАЛ. ТЕ МОГАТ, ВСЕ ПАК, ДА СЕ ИЗПОЛЗВАТ В ТЕКСТ, АКО НЕ СА ОГРАДЕНИ ОТ ДВЕТЕ СТРАНИ ОТ ИНТЕРВАЛИ:

```
#> PRINT " "ГРАФОРТ" ТЕКСТ "  
"ГРАФОРТ" ТЕКСТ
```

ТЪМ КАТО PRINT НЕ ИЗВЕЖДА АВТОМАТИЧНО ИНТЕРВАЛ ИЛИ RETURN В КРАЯ НА ТЕКСТА ИМА ОЩЕ ДВА ОПЕРАТОРА ПОЛЕЗНИ В ТОЗИ СЛУЧАЙ: SPCE (ИНТЕРВАЛ) И CR (CARRIAGE RETURN).

```
#> PRINT " ГРА " PRINT " ФОРТ "  
ГРАФОРТ
```

#> PRINT " GRA " SPCE PRINT " OPT "
GRA OPT

#> PRINT " GRA " CR PRINT " OPT "
GRA
OPT

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ====СЛЕД=====

PRINT - - - - -
 ИЗВЕЖДА СЛЕДВАЩИЯ ТЕКСТ

" - - - - -
 ОБГРАЖДА ТЕКСТ В ПРОГРАМАТА

SPCE - - - - - SPACE
 ИЗВЕЖДА ПРАЗНО (КОД 160)

CR - - - - - CARRIAGE RET.
 ВРЪЩАНЕ НА КУРСОРА (КОД 141)

II.3 ДЕФИНИРАНЕ НА НОВИ ДУМИ

ПРОГРАМИРАНЕТО НА ГРАФОРТ СЕ СВЕЖДА ДО ДЕФИНИРАНЕ НА НОВИ ДУМИ НА БАЗА НА ВЕЧЕ СЪЩЕСТВУВАЩИТЕ. ТЕЗИ НОВИ ДУМИ СЕ ДОБАВЯТ КЪМ БИБЛИОТЕКАТА ОТ ДУМИ. ТАКА ЕЗИКЪТ ГРАФОРТ СЕ РАЗШИРЯВА ЗА ДА СТАНЕ ВАШАТА ПРОГРАМА.

ДУМИТЕ ":" И ";" СЛУЖАТ ЗА ДЕФИНИРАНЕ НА НОВИ ДУМИ:

: <ДУМА> <НИЗ ДЕФИНИРАНИ ВЕЧЕ ДУМИ> ;

С ":" СЕ ЗАПОЧВА НОВА ДЕФИНИЦИЯ. <ДУМА> ДАВА ИМЕТО НА НОВАТА ДУМА - ВИНАГИ, КОГАТО НАПИШЕТЕ <ДУМА>, ЩЕ БЪДАТ ИЗПЪЛНЕНИ ПОСЛЕДОВАТЕЛНО ДУМИТЕ ОТ СПИСЪКА <НИЗ ДЕФИНИРАНИ ВЕЧЕ ДУМИ>. КРАЯТ НА ТОЗИ СПИСЪК СЕ ОПРЕДЕЛЯ ОТ СПЕЦИАЛНАТА ДУМА ";" - КРАИ НА ДЕФИНИЦИЯТА.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ====СЛЕД=====

;
 - - - - -
 КРАИ НА ДЕФИНИЦИЯ НА ДУМА

;
 - - - - -
 НАЧАЛО НА ДЕФИНИЦИЯ НА ДУМА

ЕДИН ПРИМЕР:

#> : СУМА PRINT " СУМАТА Е " + . ;

```
#> 13 8 СУМА
СУМАТА Е 21
#> LIST
```

```
СУМА
?
CHS
ABS
SGN
CALL
.
.
.
```

НЕКА ДА ДЕФИНИРАМЕ НОВА ДУМА, КОЯТО ДА ОТПЕЧАТВА И СЪБИРАЕМИТЕ И СУМАТА:

```
#> : СУМА1
#> OVER OVER PRINT " СУМАТА НА " .
#> PRINT " И " .
#> PRINT " Е " + . ;
```

НЕКА ВИДИМ СЕГА БИБЛИОТЕКАТА:

```
#> LIST

СУМА1
СУМА
?
CHS
ABS
SGN
CALL
.
.
```

```
#> 24 7 СУМА1
СУМАТА НА 7 И 24 Е 31
```

СУМА1 МОЖЕ ДА БЪДЕ ВИКАНА ПО ВСЯКО ВРЕМЕ, КАКТО ДИРЕКТНО, ТАКА И ОТ ДРУГА ДЕФИНИЦИЯ.

А. ПРЕМАХВАНЕ НА ДУМИ ОТ БИБЛИОТЕКАТА.

ЗА ИЗТРИВАНЕ НА ДУМИ ОТ БИБЛИОТЕКАТА СЕ ИЗПОЛЗВА ДУМАТА FORGET ПОСЛЕДВАНА ОТ ДЕФИНИРАНА ВЕЧЕ ДУМА. FORGET (ЗАБРАВЯМ) НЕ МОЖЕ ДА ИЗВАДИ САМО ЕДНА ДУМА ОТ БИБЛИОТЕКАТА - ТЯ ИЗТРИВА ВСИЧКИ ДУМИ ОТ НАЧАЛОТО ДО СПЕЦИФИЦИРАНАТА ДУМА:

```
#> FORGET СУМА
#> LIST
```


?
 CHS
 ABS
 SGN
 .
 .

ВИЖДАТЕ, ЧЕ И СУМА И СУМА1 БЯХА ИЗТРИТИ ОТ БИБЛИОТЕКАТА.

ОПЕРАТОР С Т А С К ОБЯСНЕНИЯ
 =====ПРЕДИ=====СЛЕД=====

FORGET - - - - -
 ИЗТРИВА БИБЛИОТЕКАТА ДО
 ДУМАТА СЛЕД FORGET

II.4 СЪЗДАВАНЕ НА ЦИКЛИ

A. СТРУКТУРА DO - LOOP.

<КРАИ> <НАЧАЛО> DO <ДУМИ ЗА ПОВТАРЯНЕ> LOOP

ДУМАТА DO ИЗВАЖДА ДВЕ СТОИНОСТИ ОТ СТЕКА - НАИ-ГОРНОТО ЧИСЛО СЕ ИЗПОЛЗВА КАТО НАЧАЛНА СТОИНОСТ, А СЛЕДВАЩОТО ЧИСЛО - КАТО КРАИНА. ИЗПЪЛНЯВАТ СЕ ДУМИТЕ МЕЖДУ DO И LOOP СЛЕД КОЕТО НАЧАЛНАТА СТОИНОСТ СЕ УВЕЛИЧАВА С 1. АКО НОВАТА СТОИНОСТ (СТОИНОСТ НА ЦИКЪЛА) Е ПО-МАЛКА ОТ КРАИНАТА СЕ ИЗПЪЛНЯВАТ ОТНОВО ДУМИТЕ МЕЖДУ DO И LOOP. ВЪТРЕ В ЦИКЪЛА ДУМАТА "I" ИЗВЕЖДА ТЕКУЩАТА СТОИНОСТ НА ЦИКЪЛА И Я ПОСТАВЯ В СТЕКА.

```
#> 3 0 DO PRINT " ЦИКЪЛ " I . CR LOOP
ЦИКЪЛ 0
ЦИКЪЛ 1
ЦИКЪЛ 2
```

ЗА ДА СЕ РАБОТИ С НАРАСТВАНЕ РАЗЛИЧНО ОТ 1, СЕ ИЗПОЛЗВА СТРУКТУРАТА DO - +LOOP. ДУМАТА +LOOP ВЗИМА СТОИНОСТ ОТ СТЕКА И Я ИЗПОЛЗВА КАТО НАРАСТВАНЕ.

```
#> 6 0 DO I . CR 2 +LOOP
0
2
4
```

+LOOP ИЗПОЛЗВАШЕ 2 КАТО НАРАСТВАНЕ.

```
#> 10 20 DO I . CR -2 +LOOP
20
18
16
14
12
```

ТУК НАРАСТВАНИЕТО БЕШЕ -2.

МОГАТ ДА БЪДАТ ВМЪКВАНИ ЕДИН В ДРУГ ЦИКЛИ. СТОИНОСТТА НА
ВЪНШНИЯ ЦИКЪЛ МОЖЕ ДА БЪДЕ ПРЕХВЪРЛЕНА В СТЕКА С ДУМАТА "J", А ОТ
НАИ-ВЪНШНИЯ (ТРЕТИ) - ЧРЕЗ "K".

```
#> : ЦИКЪЛ

#> 4 0 DO

#>     PRINT " ВЪНШЕН ЦИКЪЛ: " I . CR

#>     3 0 DO

#>         J . SPCE I . CR

#>     LOOP

#> LOOP ;

#> ЦИКЪЛ
ВЪНШЕН ЦИКЪЛ: 0
0 0
0 1
0 2
ВЪНШЕН ЦИКЪЛ: 1
1 0
1 1
1 2
ВЪНШЕН ЦИКЪЛ: 2
2 0
2 1
2 2
ВЪНШЕН ЦИКЪЛ: 3
3 0
3 1
3 2
```

ОПЕРАТОР S T A C K ОБЪАСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

DO M N - - - - DO... LOOP
 N - НАЧАЛО, M - КРАИ

LOOP - - - - DO... LOOP
 АКО Е < ОТ КРАИ -> ОБРАТНО

+LOOP N - - - - DO...+LOOP
 ИЗПОЛЗВА N КАТО НАРАСТВАНЕ

I - - - N - -
 ВРЪЩА СТОИНОСТ НА ВЪТР.ЦИКЪЛ

J - - - N - -
 СТОИНОСТ НА ВЪНШЕН ЦИКЪЛ

K - - - N - -
 ЗА ВТОРИ ВЪНШЕН ЦИКЪЛ

В. ВЪЗВРАТЕН СТЕК

DO - LOOP ИЗПОЛЗВА СТЕК ПОДОБЕН НА СТЕКА ЗА ДАННИ НАРЕЧЕН "ВЪЗВРАТЕН СТЕК" (RETURN STACK). ПРИ ИЗПЪЛНЕНИЕ НА DO, ГОРНИТЕ ДВЕ СТОИНОСТИ ОТ СТЕКА ЗА ДАННИ СЕ ПРЕХВЪРЛЯТ ВЪВ ВЪЗВРАТНИЯ СТЕК. НАИ-ГОРЕ СТОИ СТОИНОСТТА НА ЦИКЪЛА, А НЕПОСРЕДСТВЕНО СЛЕД НЕЯ - КРАЙНАТА СТОИНОСТ. LOOP УВЕЛИЧАВА ГОРНАТА СТОИНОСТ И Я СРАВНЯВА С ТАЗИ СЛЕД НЕЯ. ПРИ СРЕЩАНЕ НА "I" ГОРНАТА СТОИНОСТ ОТ ВЪЗВРАТНИЯ СТЕК СЕ ПРЕХВЪРЛЯ В СТЕКА ЗА ДАННИ. СЛЕД ЗАВЪРШВАНЕ НА ЦИКЪЛА, ДВЕТЕ ГОРНИ СТОИНОСТИ ОТ ВЪЗВРАТНИЯ СТЕК СЕ ОТСТРАНЯВАТ.

ЕТО И ДУМИТЕ УПРАВЛЯВАЩИ ВЪЗВРАТНИЯ СТЕК:

PUSH ПРЕХВЪРЛЯ ЧИСЛО ОТ СТЕКА ЗА ДАННИ ВЪВ ВЪЗВРАТНИЯ СТЕК.

PULL ПРЕХВЪРЛЯ ЧИСЛО ОТ ВЪЗВРАТНИЯ СТЕК В СТЕКА ЗА ДАННИ.

POP ПРЕМАХВА ГОРНИЯ ЕЛЕМЕНТ ОТ ВЪЗВРАТНИЯ СТЕК.

В ПРИМЕРА, КОИТО СЛЕДВА ВЪЗВРАТНИЯТ СТЕК СЕ ИЗПОЛЗВА ЗА ДА СЕ РАЗМЕСТЯТ ДОЛНИТЕ ДВЕ ОТ ТРИТЕ ЧИСЛА В СТЕКА ЗА ДАННИ.

#> STACK 3 2 1

[3]

[2]

[1]

#> PUSH

[3]

[2]

<1>

#> SWAP

[2] <1>

[3]

#> PULL

[2]

[3]

[1]

#>

ГОРНОТО МОЖЕ ДА БЪДЕ НАПИСАНО, РАЗБИРА СЕ И ТАКА:

#> 3 2 1 PUSH SWAP PULL . . .

132

ОПЕРАТОР S T A C K ОБЪСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

POP - - - - -
 МАХА ОТ ВЪЗВРАТНИЯ СТЕК

PULL - - - N - - -
 ВЪЗВРАТЕН СТЕК -> СТЕК

PUSH N - - - - -
 СТЕК -> ВЪЗВРАТЕН СТЕК

II.5 СРАВНЯВАНЕ НА ЧИСЛА

ЕТО ГРАФОРТ-ДУМИТЕ ИЗПОЛЗВАНИ ЗА СРАВНЯВАНЕ НА ЧИСЛА:

<> НЕРАВНО НА
= РАВНО НА
> ПО-ГОЛЯМО ОТ
< ПО-МАЛКО ОТ
>= ПО-ГОЛЯМО ОТ ИЛИ РАВНО НА
<= ПО-МАЛКО ОТ ИЛИ РАВНО НА

ВСЯКА ОТ ТЕЗИ ДУМИ ИЗВАНДА ДВЕ ЧИСЛА ОТ СТЕКА, СРАВНЯВА ПРЕДПОСЛЕДНОТО С ПОСЛЕДНОТО И ВРЪЩА В СТЕКА РЕЗУЛТАТ 1 АКО СРАВНЕНИЕТО Е ВЯРНО И 0 В ПРОТИВЕН СЛУЧАЙ.

#> 3 3 = .

1

#> 4 1 <> .

1

#> -2 3 > .

0

#> 21 3 >= .

1

ОПЕРАТОР S T A C K ОБЪАСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

<> N M - N - - N=0,1
 1, АКО ЧИСЛАТА НЕ СА РАВНИ

= N M - N - - N=0,1
 1, АКО СА РАВНИ

> N M - N - - N=0,1
 1, АКО N > M

< N M - N - - N=0,1
 1, АКО N < M

>= N M - N - - N=0,1
 1, АКО N >= M

<= N M - N - -
 1, АКО N <= M

СЪЩЕСТВУВАТ ОЩЕ ДВЕ ДУМИ - AND И OR, КОИТО ИЗВЪРШВАТ ПОБИТОВО
 "И" ИЛИ "ИЛИ" НАД ДВЕ ЧИСЛА ОТ СТЕКА. ВСЯКО ЧИСЛО В СТЕКА Е ЗАПИСАНО
 В 2 БАЙТА, СЛЕДОВАТЕЛНО AND И OR ДЕЙСТВАТ НАД 16 БИТА.

AND!	0	1	OR !	0	1
-----+			-----+		
	0	0		0	1
	!			!	
	1	0		1	1

ОПЕРАТОР S T A C K ОБЪАСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

OR M N - N - - N = M OR N
 ПОБИТОВО ИЛИ

AND M N - N - - N = M AND N
 ПОБИТОВО И

ПРИМЕР: ДАЛИ ЕДНО ЧИСЛО Е МЕЖДУ 5 И 10 ?

```
#> STACK 7
[7]
#> DUP 5 >
[7]
[1]
#> SWAP
[1]
[7]
#> 10 <
[1]
[1]
#> AND
[1]
```

ДА, 7 Е МЕЖДУ 5 И 10.

```
#> 13
[13]
#> DUP 5 >
[13]
[1]
#> SWAP
[1]
[13]
#> 10 <
[1]
[0]
#> AND
[0]
```

13 НЕ Е МЕЖДУ 5 И 10.

II.6 ВЗИМАНЕ НА РЕШЕНИЯ И ДУМИ ЗА ПРЕХОД

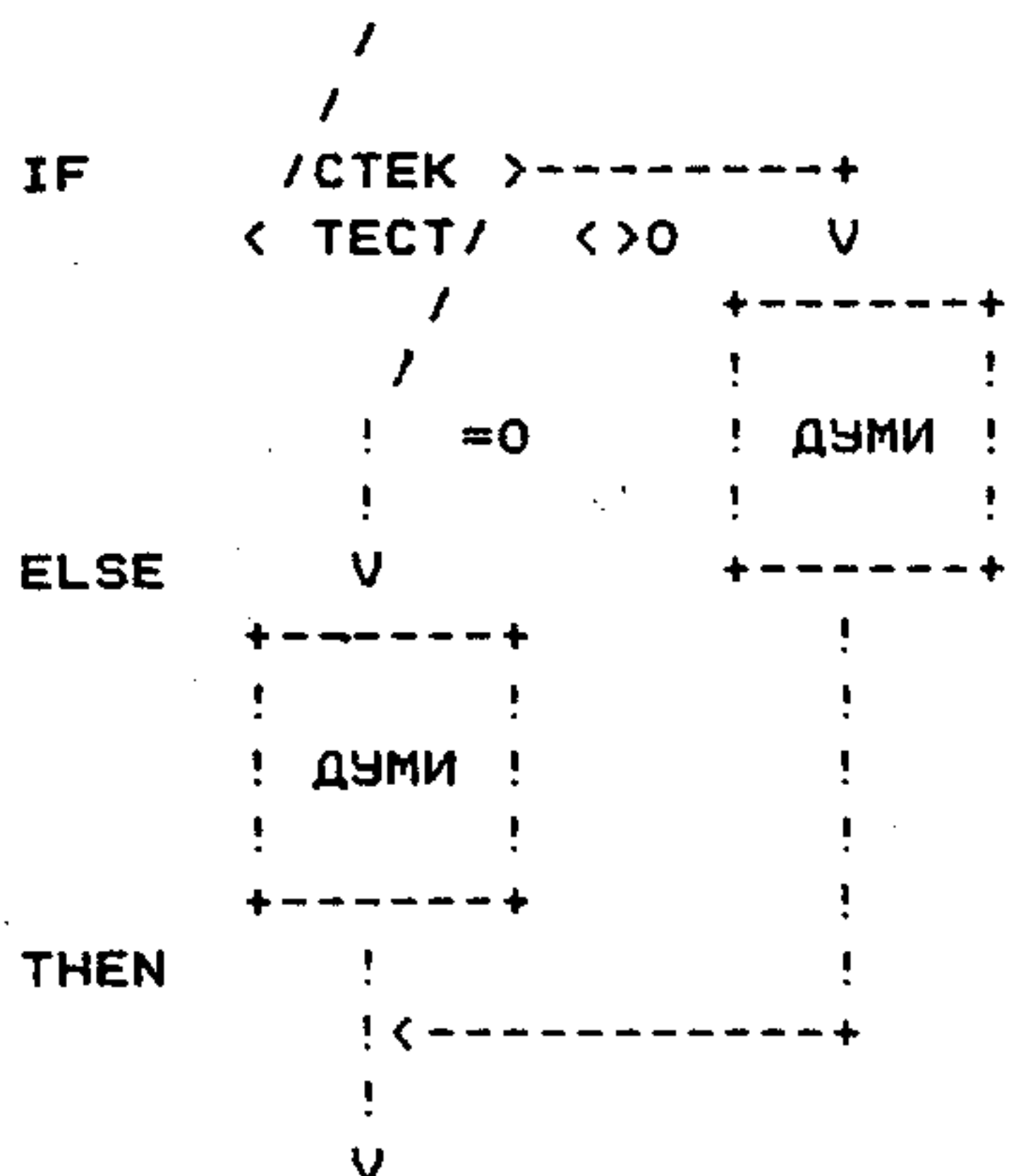
ИМА 5 КОНСТРУКЦИИ НА ГРАФОРТ, КОИТО СА ПРЕДНАЗНАЧЕНИ ЗА ТЕСТВАНЕ НА УСЛОВИЯ И ИЗВЪРШВАНЕ НА ПРЕХОДИ В ЗАВИСИМОСТ ОТ РЕЗУЛТАТИТЕ ОТ ТЕСТА. ТЕ ВЗИМАТ ЧИСЛО ОТ СТЕКА И И ПРОВЕРЯВАТ ДАЛИ ТО Е НУЛА ИЛИ Е РАЗЛИЧНО ОТ НУЛА.

A. IF - THEN

```
<СТЕК ТЕСТ>
      IF
      <ДУМИ ЗА ИЗПЪЛНЕНИЕ>
      THEN
```

ДУМАТА IF ВЗИМА ЧИСЛО ОТ СТЕКА. АКО ТО Е РАЗЛИЧНО ОТ НУЛА, ДУМИТЕ МЕЖДУ IF И THEN СЕ ИЗПЪЛНЯВАТ, В ПРОТИВЕН СЛУЧАИ ИЗПЪЛНЕНИЕТО ПРОДЪЛЖАВА СЛЕД THEN.

IF ОТНОВО ТЕСТВА ГОРНОТО ЧИСЛО ОТ СТЕКА. АКО ТО Е РАЗЛИЧНО ОТ
 НУЛА, СЕ ИЗПЪЛНЯВАТ ДУМИТЕ МЕЖДУ IF И ELSE; В ПРОТИВЕН СЛУЧАЙ СЕ
 ИЗПЪЛНЯВАТ ДУМИТЕ МЕЖДУ ELSE И THEN. И В ДВАТА СЛУЧАЯ ИЗПЪЛНЕНИЕТО
 ПРОДЪЛЖАВА СЛЕД THEN.



ПРИМЕРИ:

#> : ТЕСТ2

#> DUP 100 >

#> IF

#> . PRINT " Е ПО-ГОЛЯМО ОТ 100 "

#> ELSE

#> . PRINT " Е ПО-МАЛКО ОТ ИЛИ РАВНО НА 100 "

#> THEN ;

#> 127 ТЕСТ2

127 Е ПО-ГОЛЯМО ОТ 100

#> 32 ТЕСТ2

32 Е ПО-МАЛКО ОТ ИЛИ РАВНО НА 100


```

#> : ОГРАНИЧЕНИЕ
#>   DUP 20 >
#>   IF
#>     PRINT " ПО-ГОЛЯМО ОТ 20 "
#>     DROP
#>   ELSE
#>     10 <
#>     IF
#>       PRINT " ПО-МАЛКО ОТ 10 "
#>     ELSE
#>       PRINT " МЕЖДУ 10 И 20 "
#>     THEN
#>   THEN ;

```

```

#> -5 ОГРАНИЧЕНИЕ
ПО-МАЛКО ОТ 10
#> 13 ОГРАНИЧЕНИЕ
МЕЖДУ 10 И 20
#> 32 ОГРАНИЧЕНИЕ
ПО-ГОЛЯМО ОТ 20

```

```

ОПЕРАТОР      S T A C K      ОБЪЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

```

```

IF           N - - - - - IF...THEN
              IF...ELSE...THEN
              N#0 СЛЕД IF, N#0-THEN (ELSE)

```

```

THEN         - - - - - IF...THEN
              КРАИ НА IF...THEN ФАЗА

```

```

ELSE         - - - - - IF..ELSE..THEN
              РАЗДЕЛЯ IF...THEN ФАЗАТА

```

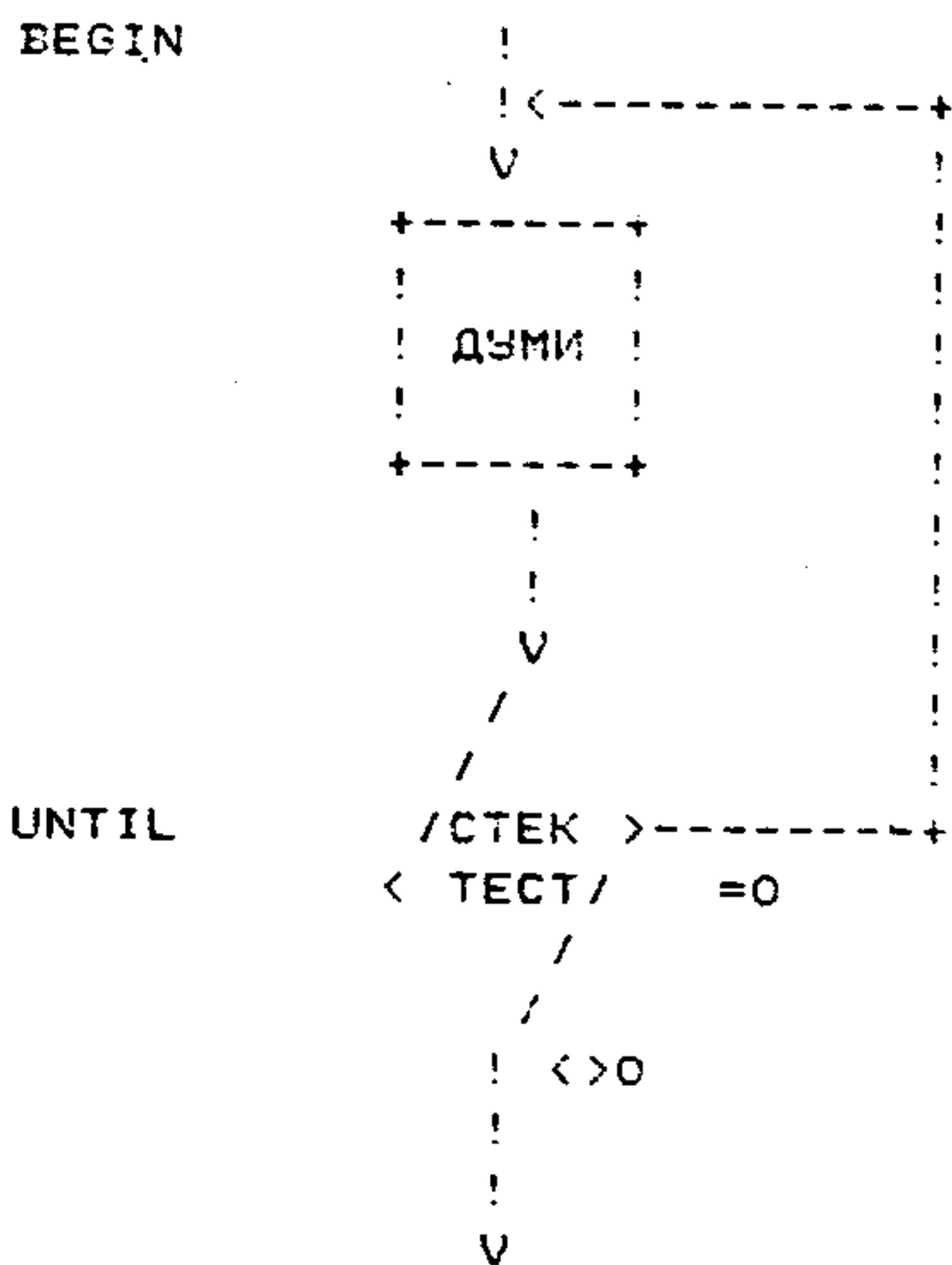
C. BEGIN - UNTIL

ФОРМАТ:

```

    BEGIN
        <ДУМИ ЗА ПОВТАРЯНЕ>
    <СТЕК ТЕСТ>
    UNTIL
  
```

ДУМИТЕ МЕЖДУ BEGIN И UNTIL СЕ ИЗПЪЛНЯВАТ. UNTIL ВЗИМА ЧИСЛО ОТ СТЕКА И АКО ТО Е НУЛА ВРЪЩА УПРАВЛЕНИЕТО СЛЕД BEGIN, В ПРОТИВЕН СЛУЧАЙ ПРОДЪЛЖАВА СЛЕД UNTIL.



ПРИМЕР:

```

#> 0 BEGIN DUP . CR 1 + DUP 5 = UNTIL
0
1
2
3
4
  
```

ОПЕРАТОР	С Т А С К	ОБЪСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====
BEGIN	- - - - -	ВЪЗВРАТ ОТ REPEAT, UNTIL
UNTIL	N - - - - -	BEGIN...UNTIL ВРЪЩА КОГАТО N = 0

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

BEGIN - - - - - -
 ВЪЗВРАТ ОТ REPEAT, UNTIL

WHILE N - - - - -
 N=0 СЛЕД REPEAT, N#0 - WHILE

REPEAT - - - - - -
 BEGIN ... WHILE ... REPEAT

E. CASE: - THEN

ФОРМАТ:

<СТЕК ТЕСТ>

 CASE:
 <ДУМА 0>
 <ДУМА 1>
 <ДУМА 2>
 . . .
 <ДУМА N>
 THEN

ДУМАТА CASE: ВЗИМА ЧИСЛО ОТ СТЕКА И ГО ИЗПОЛЗВА ЗА ДА ИЗПЪЛНИ ЕДНА ДУМА ИЗМЕЖДУ СПИСЪК ОТ ДУМИ. НУЛА ИЗБИРА ДУМА 0, ЕДИНИЦА - ДУМА 1 И Т.Н. АКО ЧИСЛОТО Е ПО-ГОЛЯМО ОТ БРОЯ НА ДУМИТЕ В СПИСЪКА ИЛИ Е ОТРИЦАТЕЛНО МОГАТ ДА СЕ СЛУЧАТ НЕПРЕДВИДЕНИ НЕЩА. ГРИЖА НА ПРОГРАМИСТА Е ДА НАПРАВИ ПРОВЕРКА ЗА КОРЕКТНОСТ ПРЕДИ ИЗПЪЛНЕНИЕТО НА CASE: - THEN.

#> : ТЕСТ3

#> CASE:

#> H

#> E

#> Д

#> BELL

#> THEN ;

#> 0 ТЕСТ3

ЧИСЛОТО Е НУЛА

#> 2 ТЕСТ3

ЧИСЛОТО Е ДВЕ

#> 3 ТЕСТ3

...<ЗВУК>...

ТРЯБВА ДА ПРЕДВИДИМ, ОБАЧЕ, ЧЕ МОЖЕ ДА БЪДЕ ЗАДАДЕНО И ЧИСЛО
РАЗЛИЧНО ОТ 0,1,2 ИЛИ 3.

#> : ПРОВЕРКА

#> DUP DUP 3 <= SWAP 0 >= AND

#> IF

#> ТЕСТ3

#> ELSE

#> PRINT " ЧИСЛОТО НЕ Е МЕЖДУ 0 И 3 "

#> DROP

#> THEN ;

#> 2 ПРОВЕРКА

ЧИСЛОТО Е ДВЕ

#> -34 ПРОВЕРКА

ЧИСЛОТО НЕ Е МЕЖДУ 0 И 3

#> 11 ПРОВЕРКА

ЧИСЛОТО НЕ Е МЕЖДУ 0 И 3

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

CASE: N - - - - - CASE: THEN
 ИЗПЪЛНЯВА N-ТАТА ДУМА (ОТ 0)

THEN - - - - - CASE: THEN
 КРАИ НА CASE:...THEN ПРАЗА

II.7 ПРОГРАМНА СТРУКТУРА

В ПОСЛЕДНИЯ ПРИМЕР (ЗА CASE:) ЗАПОЧНАХМЕ С ДЕФИНИРАНЕ НА ТРИ ДУМИ "Н", "Е" И "Д". СЛЕД ТОВА ДЕФИНИРАХМЕ "ТЕСТ3", КОЯТО ВИКА ЕДНА ОТ ТЕЗИ ТРИ ДУМИ. И НАКРАЯ ДЕФИНИРАХМЕ "ПРОВЕРКА", КОЯТО ВИКА "ТЕСТ3".

ТАЗИ ВЕРИГА ОТ ДЕФИНИЦИИ Е ПЪТЯТ, ПО КОИТО СЕ СЪЗДАВАТ ДЪЛГИ ПРОГРАМИ НА ГРАФОРТ. В КРАЙНА СМЕТКА СЕ ДЕФИНИРА ЕДНА ПОСЛЕДНА ДУМА, КОЯТО СЕ ДОБАВЯ НА ВЪРХА НА БИБЛИОТЕКАТА И НЕЙНОТО ИЗПЪЛНЕНИЕ ВОДИ ДО ИЗПЪЛНЕНИЕТО НА ПРОГРАМАТА.

А. ИЗПОЛЗВАНЕ НА ДУМИ

```
#> 5 0 DO I . CR НИЩО LOOP
```

```
НИЩО НЕНАМЕРЕНО (RETURN)
```

```
0  
1  
2  
3  
4  
#>
```

ТОВА БЕШЕ ПРИМЕР С НЕДЕФИНИРАНА ДУМА - ТЯ СЕ ИГНОРИРА И ИЗПЪЛНЕНИЕТО ПРОДЪЛЖАВА.

```
#> : OVER PRINT " ГРЕШКА " ;
```

```
OVER ДУБЛИРАНО (RETURN)
```

ТУК ДЕФИНИРАМЕ ДУМА С ИМЕ, КОЕТО СЪЩЕСТВУВА ВЕЧЕ В БИБЛИОТЕКАТА. СЪОБЩЕНИЕТО Е ИНФОРМАЦИОННО, ДУМАТА СЕ ДОБАВЯ НА ВЪРХА НА БИБЛИОТЕКАТА, НО ПОВЕЧЕ НЯМАМЕ ДОСТЪП ДО ПРЕДИШНАТА ДУМА OVER. FORGET OVER НЕ ОПРАВИ НЕЩАТА.

В. ТЕКСТ ИЛИ ГРАФИКА

ГРАФИЧНИЯТ ЕКРАН СЕ ИЗПОЛЗВА ОТ ГРАФОРТ ЗА ИЗВЕЖДАНЕ НА ТЕКСТОВЕ, ГРАФИКИ И СПЕЦИАЛНИ СИМВОЛНИ НАБОРИ. ПРИДВИЖВАНЕТО НА ТЕКСТА ПО ЕКРАНА НЕ Е ТАКА БЪРЗО КАКТО ПРИ НОРМАЛНИЯ ТЕКСТОВ РЕЖИМ. ГРАФОРТ ПРИТЕЖАВА ДВЕ ДУМИ, КОИТО УПРАВЛЯВАТ РЕЖИМА НА ИЗВЕЖДАНЕ:

```
GR - ГРАФИЧЕН (НОРМАЛЕН ЗА ГРАФОРТ);  
TEXT - ТЕКСТОВ.
```

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ  
=====ПРЕДИ=====СЛЕД=====
```

```
GR            - - - - - GRAPHICS  
              ГРАФИЧЕН РЕЖИМ
```

```
TEXT         - - - - -  
              ТЕКСТОВ РЕЖИМ
```

С. ПАМЕТ (PRGTOP И \$LIST)

В ГРАФОРТ СА ВЛОЖЕНИ МНОГО БЪЗМОЖНОСТИ, А И ДВЕТЕ ГРАФИЧНИ СТРАНИЦИ СЕ ИЗПОЛЗВАТ. СВОБОДНОТО МЯСТО ЗА СЪЗДАВАНЕ НА ПРОГРАМИ НЕ Е МНОГО ГОЛЯМО. ПРИСЪСТВИЕТО НА ЕЗИКОВА КАРТА УЛЕСНЯВА РАБОТАТА С ГРАФОРТ.

ЕДИН НАЧИН ДА СЕ ПОДДЪРЖА ПАМЕТТА СВОБОДНА Е ДА СЕ ИЗДАВА FORGET ЗА ДУМИТЕ, КОИТО НЕ СА ПОВЕЧЕ НЕОБХОДИМИ.

ЕТО ДВЕ ДУМИ, КОИТО ПОЗВОЛЯВАТ ОЦЕНКА НА ПАМЕТТА ЗАЕТА ОТ БИБЛИОТЕКАТА.

```
#> PRGTOP
-32040
```

PRGTOP ПРЕХВЪРЛЯ В СТЕКА АДРЕСА НА ВЪРХА НА БИБЛИОТЕКАТА.

\$LIST ИЗВЕЖДА ПОДОБНО НА LIST ДУМИТЕ ОТ БИБЛИОТЕКАТА, НО ЗАЕДНО С ШЕСТНАДЕСЕТИЧНИТЕ ИМ АДРЕСИ:

```
#> $LIST
```

```
$82CB ?
$82B1 CHS
$82A3 ABS
$8281 SGN
. . .
. . .
```

ТОБА СА НАЧАЛНИТЕ АДРЕСИ НА ДУМИТЕ. НЕКА ДЕФИНИРАМЕ ЕДНА ФИКТИВНА ДУМА, КОЯТО ЩЕ НИ ПОМОГНЕ ДА УЗНАЕМ ДЕЙСТВИТЕЛНИЯ КРАИ НА БИБЛИОТЕКАТА.

```
#> : * ;
```

```
#> $LIST
```

```
$82D8 *
$82CB ?
$82B1 CHS
$82A3 ABS
. . .
```

СЛЕД ИЗТРИВАНЕ НА ФИКТИВНАТА ДУМА (FORGET *), НЕИНИЯТ НАЧАЛЕН АДРЕС (\$82D8) ЩЕ БЪДЕ ДЕЙСТВИТЕЛНИЯ КРАИ НА БИБЛИОТЕКАТА.

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
PRGTOP      - - -      A - -      PROGRAM TOP
                АДРЕС НА ВЪРХА НА БИБЛИОТ.
```

```
$LIST      - - -      - - -
                ИЗВЕЖДА ДУМИТЕ С АДРЕСИТЕ ИМ
```


II.8 РАБОТА С КЛАВИАТУРАТА

А. КИРИЛИЦА И ЛАТИНИЦА

НОРМАЛНО МОЖЕТЕ ДА ИЗПОЛЗВАТЕ КЛАВИШИТЕ "ЛАТ" ЗА ПРЕМИНАВАНЕ В РЕЖИМ ЛАТИНИЦА, НО ГРАФОРТ ВИ ДАВА НЯКОИ ДОПЪЛНИТЕЛНИ УДОБСТВА. ЕДНОВРЕМЕННОТО НАТИСКАНЕ НА "МК" И "О" (ЩЕ ГО БЕЛЕЖИМ [O]) ПОСЛЕДВАНО ОТ "Е", "У" ИЛИ "Л" ВИ ПОЗВОЛЯВА ДА РАБОТИТЕ С КЛАВИАТУРАТА ПО РАЗЛИЧЕН НАЧИН.

[O]E - САМО В РЕЖИМ С НАТИСНАТ ЖЪЛТ КЛАВИШ "ЛАТ". ВСИЧКИ ВЪВЕЖДАНИ БУКВИ СА НА КИРИЛИЦА. ПРИ НАТИСКАНЕ НА "ОСВ" (ESC) СЛЕДВАЩИЯТ СИМВОЛ Е ОТ ЛАТИНИЦАТА.

[O]У - В РЕЖИМ КИРИЛИЦА, ВСИЧКИ ВЪВЕЖДАНИ БУКВИ СА ОТ ЛАТИНИЦАТА.

[O]Л - ВСИЧКИ ВЪВЕЖДАНИ БУКВИ СА ОТ КИРИЛИЦАТА.

[O] - [O] ПОСЛЕДВАНО ОТ ИНТЕРВАЛ - НОРМАЛЕН РЕЖИМ.

В. ПРИДВИЖВАНЕ НА КУРСОРА

ПОДОБНО НА СТАНДАРТНАТА РАБОТА С "ПРАВЕЦ", [U] И [H] СА ДВЕТЕ СТРЕЛКИ "-->" И "<--". СЛЕД НАТИСКАНЕ НА "ОСВ" (ESC) "I", "J", "K" И "M" ОТ ЛАТИНИЦАТА ПРИДВИЖВАТ КУРСОРА В ЧЕТИРИТЕ ПОСОКИ. НАТИСКАНЕТО НА [I] ПРЕДИЗВИКВА ОСВОБОЖДАВАНЕТО НА МЯСТО ЧРЕЗ ОТМЕСТВАНЕ НА ДЯСНО НА ЦЕЛИЯ РЕД. ПО ТОЗИ НАЧИН В РАМКИТЕ НА ЕДИН РЕД ОТ 40 ЗНАКА МОГАТ ДА СЕ ВМЪКВАТ ДРУГИ СИМВОЛИ.

II.9 ТЕКСТОВ РЕДАКТОР

НА ДИСКЕТАТА СЪЩЕСТВУВАТ ДВА ЕДНАКВИ ТЕКСТОВИ РЕДАКТОРА - Г*РЕДАКТОР1 И Г*РЕДАКТОР2. Г*РЕДАКТОР1 СЕ ЗАРЕЖДА КОГАТО ПАМЕТТА Е 48К И ПОЗВОЛЯВА РЕДАКТИРАНЕ НА ОКОЛО 2000 СИМВОЛА. Г*РЕДАКТОР2 Е РЕДАКТОРЪТ, КОИТО СЕ ИЗПОЛЗВА КОГАТО ИМА ЕЗИКОВА ИЛИ РАМ КАРТИ И ПОЗВОЛЯВА РЕДАКТИРАНЕ НА ОКОЛО 11500 СИМВОЛА.

КОМПИЛАТОРЪТ ГРАФОРТ И ТЕКСТОВИЯ РЕДАКТОР ИЗПОЛЗВАТ СТАНДАРТНИ ТЕКСТОВИ ФАЙЛОВЕ (ТИП "Т"). ЗА ПИСАНЕ НА ПРОГРАМИ МОЖЕТЕ ДА ИЗПОЛЗВАТЕ ВСЕКИ ТЕКСТОВ РЕДАКТОР ЗА ПРАВЕЦ, КОИТО СЪЗДАВА ФАЙЛОВЕ ОТ ТИП "Т".

ЗА ДА РАБОТИТЕ С РЕДАКТОРА, НАПИШЕТЕ "EDIT". СЪОТВЕТНИЯТ РЕДАКТОР ЩЕ БЪДЕ ЗАРЕДЕН В ПАМЕТТА.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

EDIT - - - - -
 ВХОД В ТЕКСТОВИЯ РЕДАКТОР

ПЪРВАТА КОМАНДА, КОЯТО ТРЯБВА ДА ЗНАЕТЕ Е "?".

?

SAVE
GET
INSERT
DELETE
PROGRAM
MEMORY
LIST
WRITE
ERASE
AUTONUM
BYE
CTRL-D=DOS

"?" ИЗВЕЖДА ВСИЧКИ КОМАНДИ НА РЕДАКТОРА. ЗА ДА ИЗДАДЕТЕ НЯКОЯ ОТ ТЕЗИ КОМАНДИ Е ДОСТАТЪЧНО ДА НАПИШЕТЕ САМО НЕИНАТА ПЪРВА БУКВА И ТО НЕЗАДЪЛЖИТЕЛНО В РЕЖИМ ЛАТИНИЦА.

ПРИ ИЗВЕЖДАНЕТО НА ЕКРАНА ВСИЧКИ РЕДОВЕ ОТ ЕДНА ПРОГРАМА СА НОМЕРИРАНИ ОТ 10 И ПРЕЗ 10. ТЕЗИ НОМЕРА НЕ СЕ ПАЗЯТ НА ДИСКЕТАТА И НЯМАТ ЗНАЧЕНИЕ ЗА КОМПИЛАТОРА - ТЯХНОТО ПРЕДНАЗНАЧЕНИЕ Е ЕДИНСТВЕНО ОБСЛУЖВАЩО. ПРИ ВМЪКВАНЕ ИЛИ ИЗТРИВАНЕ НА РЕДОВЕ ОТ ПРОГРАМАТА, ТЯ СЕ ПРЕНОМЕРИРА АВТОМАТИЧНО ПРЕЗ 10.

ЗА ВЪВЕЖДАНЕ НА РЕДОВЕ СЕ ПИШЕ НОМЕРА ПОСЛЕДВАН ОТ ТЕКСТА - СЪЩО КАТО ПРИ БЕЙСИК.

10 ПЪРВИ РЕД
20 ВТОРИ РЕД
30 ТРЕТИ РЕД

A. LIST (L)

L
10 ПЪРВИ РЕД
20 ВТОРИ РЕД
30 ТРЕТИ РЕД

КРАИ

"КРАИ" Е ИНДИКАЦИЯ ЗА ТОВА, ЧЕ РЕДАКТОРЪТ УСПЕШНО Е ИЗВЪРШИЛ ИСКАНАТА ФУНКЦИЯ.

НЕКА ВМЪКНЕМ РЕД 15:

15 ВМЪКНАТ РЕД

L
10 ПЪРВИ РЕД
20 ВМЪКНАТ РЕД
30 ВТОРИ РЕД
40 ТРЕТИ РЕД

LIST (L ИЛИ Л) МОЖЕ ДА СЕ ИЗПОЛЗВА ЗА ИЗВЕЖДАНЕ НА ЕДИН ИЛИ ГРУПА РЕДОВЕ.

L 20 ИЗВЕЖДА САМО РЕД 20
L 20,40 ИЗВЕЖДА РЕДОВЕ ОТ 20 ДО 40
L 20, ИЗВЕЖДА ОТ РЕД 20 ДО КРАЯ
L ,40 ИЗВЕЖДА ОТ НАЧАЛОТО ДО РЕД 40

В. AUTONUM (A)

AUTONUM ПОЗВОЛЯВА АВТОМАТИЧНО НОМЕРИРАНЕ НА РЕДОВЕТЕ.

A
50 ТЕКСТ1
60 ТЕКСТ2
70

ЗА ДА СПРЕТЕ AUTONUM НАТИСНЕТЕ RETURN БЕДНАГА СЛЕД НОМЕРА.

ЗА ДА ПРОМЕНИТЕ ЕДИН ВЕЧЕ ВЪВЕДЕН РЕД ИЗПОЛЗВАЙТЕ "ОСВ" ПОСЛЕДВАНО ОТ I,J,K ИЛИ M И ЧРЕЗ СТРЕЛКИТЕ ([U] И [H]) ВЪВЕДЕТЕ КОРЕКТНИЯ РЕД ОТНОВО.

С. DELETE (D)

ВЪВЕЖДАНЕТО НА ПРАЗЕН РЕД (САМО НОМЕР) НЯМА ДА ГО ИЗТРИЕ ПОДОБНО НА БЕЙСИК. ИЗТРИВАНЕТО НА РЕДОВЕ СТАВА СЪС КОМАНДАТА DELETE (D). ПОДОБНО НА LIST, DELETE МОЖЕ ДА ИЗТРИЕ ЕДНА ИЛИ ГРУПА ЛИНИИ

D 30
D 30,80
D 30,
D ,80

Д. ERASE (E)

ТОВА Е КОМАНДАТА, КОЯТО СЕ ИЗПОЛЗВА ЗА ДА СЕ ИЗТРИЕ ФАЙЛА ЗА РЕДАКТИРАНЕ ОТ ПАМЕТТА.

E
ИЗТРИВАНЕ (Д/Н) :

ТРЯБВА ДА ПОТВЪРДИТЕ С Д)А УНИЩОЖАВАНЕТО НА ФАЙЛА В ПАМЕТТА.

Е. АВТОМАТИЧНО ВМЪКВАНЕ.

ИЗПОЛЗВАХМЕ ВЕЧЕ AUTONUM ЗА АВТОМАТИЧНО ДОБАВЯНЕ КЪМ ФАЙЛА. В СРЕДАТА НА ФАЙЛА AUTONUM МОЖЕ ДА СЕ ИЗПОЛЗВА СЪЩО ЗА ВМЪКВАНЕ НА РЕДОВЕ.

НЕКА ИЗТРИЕМ ВСИЧКО ОТ ПАМЕТТА:

E
ИЗТРИВАНЕ (Д/Н) : Д
КРАИ

10 ПЪРВИ РЕД
20 ПОСЛЕДЕН РЕД

L
10 ПЪРВИ РЕД
20 ПОСЛЕДЕН РЕД

ДА ДОБАВИМ ЕДИН РЕД:

15 РЕД 1

НЕКА СЕГА ВЪВЕДЕМ AUTONUM:

A
30 РЕД 2
40 РЕД 3
50 РЕД 4
60

А СЕГА ДА ВИДИМ ОТНОВО ФАЙЛА:

L
10 ПЪРВИ РЕД
20 РЕД 1
30 РЕД 2
40 РЕД 3
50 РЕД 4
60 ПОСЛЕДЕН РЕД

F. INSERT (I)

С КОМАНДАТА INSERT СЪЩО МОЖЕ ДА СЕ ЗАПОЧНЕ ВМЪКВАНЕ. ВМЕСТО ДА СЕ НАПИШЕ ПЪРВИЯ ВМЪКНАТ РЕД, ПРЕДИ AUTONUM СЕ ИЗДАВА "I". НЕКА ИЗТРИЕМ ПЪРВО РЕДОВЕ 20 ДО 50:

D 20,50

КРАИ

L
10 ПЪРВИ РЕД
20 ПОСЛЕДЕН РЕД

НЕКА ВМЪКНЕМ ОТНОВО РЕДОВЕТЕ ПО СРЕДАТА:

I 15

A
20 РЕД 1
30 РЕД 2
40 РЕД 3
50 РЕД 4
60

РЕДОВЕ ОТ 20 ДО 50 СЕ НАМИРАТ ОТНОВО ВЪВ ФАЙЛА.

G. SAVE (S)

ЗА ДА СЕ ЗАПАЗИ ФАЙЛА НА ДИСКЕТАТА, СЕ ИЗПОЛЗВА КОМАНДАТА S)AVE.

S

(ФАЙЛ-ИМЕ):

ЗАДАЙТЕ ИМЕТО, ПОД КОЕТО ИСКАТЕ ДА БЪДЕ ЗАПАЗЕН ФАЙЛА. СПОРЕД СТАНДАРТНИТЕ DOS-КОНВЕНЦИИ МОГАТ ДА БЪДАТ ЗАДАДЕНИ НОМЕР НА СЛОТ И НОМЕР НА УСТРОЙСТВО:

(ФАЙЛ-ИМЕ): ТЕСТ

(ФАЙЛ-ИМЕ): ТЕСТ,36,D1

МОЖЕТЕ ДА ЗАПАЗИТЕ И ЧАСТИ ОТ ФАЙЛА:

(ФАЙЛ-ИМЕ): ТЕСТ/20,50

(ФАЙЛ-ИМЕ): ТЕСТ/,50

(ФАЙЛ-ИМЕ): ТЕСТ/20,

H. GET (G)

ЗА ДА ЗАРЕДИТЕ ФАЙЛ В ПАМЕТТА СЕ ИЗПОЛЗВА КОМАНДАТА G)ET. ЗА РАЗЛИКА ОТ LOAD В БЕЙСИК, КОМАНДАТА GET НЕ ИЗЧИСТВА ПРЕДВАРИТЕЛНО ПАМЕТТА, А ВМЪКВА ФАЙЛА СЛЕД ТЕКУЩИЯ В ПАМЕТТА. АКО ИСКАТЕ САМО ТОЗИ ФАЙЛ ДА СЕ НАМИРА В ПАМЕТТА, ИЗДАЙТЕ ПЪРВО КОМАНДАТА E)RASE.

G

(ФАЙЛ-ИМЕ): ТЕСТ

В ИМЕТО НА ФАЙЛА МОЖЕ ДА СЕ УКАЖЕ МЕЖДУ КОИ РЕДОВЕ ДА БЪДЕ ВМЪКНАТ ЗАРЕЖДЕНИЯ ФАЙЛ:

G

(ФАЙЛ-ИМЕ): ТЕСТ/35

СЛЕД ГОРНАТА КОМАНДА ФАЙЛЪТ ТЕСТ ЩЕ БЪДЕ ВМЪКНАТ МЕЖДУ РЕДОВЕ 30 И 40 НА ТЕКУЩИЯ ФАЙЛ.

I. DOS КОМАНДИ ([D])

НАТИСНЕТЕ ЗАЕДНО "МК" И "D/D" (ОТБЕЛЯЗВАНО КАТО [D]), А СЛЕД ТОВА RETURN:

ВЪВЕДИ DOS КОМАНДА:

МОЖЕТЕ ВЕЧЕ ДА ИЗДАДЕТЕ ВСЯКА ЕДНА ОТ DOS-КОМАНДИТЕ. ЗА ДА СЕ ВЪРНЕТЕ КЪМ НОРМАЛНИЯ РЕЖИМ НА РЕДАКТОРА ПРОСТО ОТГОВОРЕТЕ ДВА ПЪТИ С RETURN.

Ж. ОТПЕЧАТВАНЕ НА ФАЙЛОВЕ

В РЕЖИМ ДОС-КОМАНДИ НА РЕДАКТОРА НАПИШЕТЕ PR#1. ПРИ ИЗДАВАНЕ СЛЕД ТОВА НА КОМАНДАТА LIST ИЛИ ОЩЕ ПО ДОБРЕ WRITE (ЗА ДА НЕ СПИРА ВСЕКИ 16 РЕДА) ВАШАТА ПРОГРАМА ЩЕ БЪДЕ ИЗВЕДЕНА НА ПЕЧАТ. ТЪИ КАТО ГРАФОРТ НЕ РАЗПОЗНАВА "PR#0", ЗА ДА СЕ ВЪРНЕТЕ КЪМ НОРМАЛЕН РЕЖИМ НАТИСНЕТЕ ЧЕРВЕНОТО КОПЧЕ "RST".

К. СЪОБРАЖЕНИЯ ЗА ПАМЕТА

С НАРАСТВАНЕТО НА БИБЛИОТЕКАТА ГРАФОРТ ЗАПОЧВА ДА ИЗПОЛЗВА И ПАМЕТА ОТДЕЛЕНА ЗА РЕДАКТОРА.

ЗА ДА ВИДИТЕ РАЗМЕРА НА СВОБОДНАТА ЗА РЕДАКТОРА ПАМЕТ, ИЗДАЙТЕ КОМАНДАТА MEMORY.

М

СВОБ. ПАМЕТ 11773
КРАИ

ЗА ДА ИЗБЕГНЕТЕ КОФЛИКТ С КОМПИЛАТОРА, МОЖЕ БИ ЩЕ ВИ СЕ НАЛОЖИ ДА ПРОМЕНИТЕ РАЗМЕРА НА ИЗПОЛЗВАНАТА ПАМЕТ. ТОВА МОЖЕ ДА СТАНЕ С ПОМОЩТА НА КОМАНДАТА POSITION:

Р

ДЪЛЖ. ПРОГРАМА ****
ПОЗИЦИЯ 34817
СВОБ. ПАМЕТ ****
ПРОМЯНА ПОЗИЦИЯ (Д/Н) : Д

ВЪВЕДИ НОВА ПОЗИЦИЯ : ****

АКО НЯМАТЕ ЕЗИКОВА КАРТА, А ТРЯБВА ДА РЕДАКТИРАТЕ ДОСТА ОБЕМИСТ ФАЙЛ, МЯСТОТО В ПАМЕТА НЯМА ДА ВИ Е ДОСТАТЪЧНО. ЗА ДА МОЖЕТЕ ДА РАБОТИТЕ ПРЕМИНЕТЕ В ТЕКСТОВ РЕЖИМ И СЛЕД ТОВА ИЗПОЛЗВАЙТЕ ГРАФИЧНИТЕ СТРАНИЦИ ЗА РАБОТНА ПАМЕТ НА РЕДАКТОРА. СЛЕД ПОДОБНО ИЗПОЛЗВАНЕ МОЖЕ ДА ВИ СЕ НАЛОЖИ ДА ЗАРЕДИТЕ ОТНОСНО ГРАФОРТ ОТ ДИСКЕТАТА.

#> TEXT EDIT

ПОД УПРАВЛЕНИЕ НА РЕДАКТОРА ИЗДАЙТЕ КОМАНДАТА POSITION:

Р

ДЪЛЖ. ПРОГРАМА *****
ПОЗИЦИЯ 34817
СВОБ. ПАМЕТ *****
ПРОМЯНА ПОЗИЦИЯ (Д/Н) : Д

ВЪВЕДИ НОВА ПОЗИЦИЯ : 3192

КРАИ

В ТЕКСТОВ РЕЖИМ, НО С ЕДИН БУФЕР УВЕЛИЧЕН С 16К, МОЖЕТЕ ДА РЕДАКТИРАТЕ И МНОГО ГОЛЕМИ ПРОГРАМНИ ФАЙЛОВЕ.

L. ВУЕ (В)

ЗА ИЗЛИЗАНЕ ОТ ТЕКСТОВИЯ РЕДАКТОР СЕ ИЗАДВА КОМАНДАТА В)УЕ.

II.10 КОМПИЛИРАНЕ НА ПРОГРАМИ

НОРМАЛНО ГРАФОРТ ПРИЕМА КОМАНДИ ПРЕЗ КЛАВИАТУРАТА, КОМПИЛИРА ГИ ВЕДНАГА И ГИ ИЗПЪЛНЯВА (АКО НЕ СА САМО ДЕФИНИЦИИ).

КОМПИЛАТОРЪТ МОЖЕ ДА ЧЕТЕ ЕДНА ПРОГРАМА КАКТО ОТ ТЕКСТОВИЯ БУФЕР НА РЕДАКТОРА, ТАКА И ДИРЕКТНО ОТ ФАЙЛ НА ДИСКЕТА. НОРМАЛНО АДРЕСА НА БУФЕРА НА РЕДАКТОРА Е 34817 (ПОЗИЦИЯ). ИМА ДВЕ КОМАНДИ КОИТО ПОЗВОЛЯВАТ КОМПИЛИРАНЕ НА ТЕКСТОВИ ФАЙЛОВЕ.

АКО ФАЙЛЪТ Е В ПАМЕТТА СЕ ИЗПОЛЗВА КОМАНДАТА MEMRD. ТЯ ИЗВЛИЧА ОТ СТЕКА ЧИСЛО, КОЕТО ИНТЕРПРТИРА КАТО АДРЕС В ПАМЕТТА НА ТЕКСТОВИЯ ФАЙЛ:

```
#> 34817 MEMRD
```

ВТОРИЯТ НАЧИН Е ЧРЕЗ КОМАНДАТА READ. ТЯ ЧЕТЕ И КОМПИЛИРА ФАЙЛ, КОИТО СЕ НАМИРА ВЪРХУ ДИСКЕТА. НЕКА ФАЙЛЪТ С ГРАФОРТ-ПРОГРАМАТА ВЪРХУ ДИСКЕТАТА СЕ НАРИЧА "ТЕСТ".

```
#> READ "ТЕСТ"
```

СЛЕД ИЗПЪЛНЕНИЕ НА КОМАНДАТА ФАЙЛЪТ "ТЕСТ" ЩЕ БЪДЕ ПРОЧЕТЕН ОТ ДИСКЕТАТА И ТРАНСЛИРАН НА МАШИЕН ЕЗИК. ОТ КОМПИЛАТОРА ГРАФОРТ.

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
READ          - - - - -
              ЧЕТЕ И КОМПИЛИРА ФАЙЛ
```

```
MEMRD        A - - - - -
              ЧЕТЕ И КОМПИЛИРА ОТ АДРЕС А
```

A. КОМЕНТАРИ

НАВСЯКЪДЕ В ЕДНА ПРОГРАМА ГРАФОРТ МОГАТ ДА БЪДАТ ВЛОЖЕНИ КОМЕНТАРИ. ДУМАТА "(" ДЕФИНИРА НАЧАЛОТО НА ЕДИН КОМЕНТАР, А ДУМАТА ")" - НЕГОВИЯ КРАИ.

```
60 ( ТОВА Е КОМЕНТАР )
```

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
- - - - -
НАЧАЛО НА КОМЕНТАР
```

11.11 ДУМИ ФОРМАТИРАЩИ ТЕКСТ

ДОСЕГА РАЗГЛЕДАХМЕ ТРИ ОТ ТЕЗИ ДУМИ:

PRINT - ИЗВЕЖДА ОГРАДЕНИЯ В КАВИЧКИ ТЕКСТ ОТ ТЕКУЩАТА ПОЗИЦИЯ НА КУРСОРА;

CR - ИЗВЕЖДА "RETURN", ПРИДВИЖВА КУРСОРА НА СЛЕДВАЩИЯ РЕД;

SPCE - ИЗВЕЖДА ЕДИНИЧЕН ИНТЕРВАЛ.

ЕТО И ОЩЕ НЯКОЛКО ДУМИ:

HTAB - ВЗИМА ЧИСЛО ОТ СТЕКА И ГО ПРИЕМА ЗА ХОРИЗОНТАЛНА ПОЗИЦИЯ НА КУРСОРА;

VTAB - ВЗИМА ЧИСЛО ОТ СТЕКА И ГО ПРИЕМА ЗА ВЕРТИКАЛНА ПОЗИЦИЯ НА КУРСОРА;

ВАЛИДНИТЕ СТОИНОСТИ ЗАВИСЯТ ОТ ГОЛЕМИНАТА НА СИМВОЛИТЕ (CHRSIZE) - СТАНДАРТНО ЗА HTAB ТЕ СА МЕЖДУ 0 И 39, А ЗА VTAB МЕЖДУ 0 И 23.

WINDOW - УСТАНОВЯВА ТЕКСТОВ ПРОЗОРЕЦ. ВЗИМА ЧЕТИРИ СТОИНОСТИ ОТ СТЕКА, КОИТО ИНТЕРПРЕТИРА ПО СЛЕДНИЯ НАЧИН:

ЛЯВА ПОЗИЦИЯ,
ШИРОЧИНА В СИМВОЛИ,
ГОРЕН РЕД,
ДОЛЕН РЕД.

ВСЯКО ИЗВЕЖДАНЕ И ПРИДВИЖВАНЕ НА ТЕКСТА СТАВА САМО В РАМКИТЕ НА ДЕФИНИРАНИЯ ПРОЗОРЕЦ, БЕЗ ДА ПРОМЕНЯ ОСТАНАЛИТЕ ЧАСТИ НА ЕКРАНА.

HOME - ИЗЧИСТВА ЕКРАНА В РАМКИТЕ НА ТЕКСТОВИЯ ПРОЗОРЕЦ.

CLEOP - ИЗЧИСТВА ЕКРАНА ОТ ТЕКУЩАТА ПОЗИЦИЯ ДО КРАЯ НА ТЕКСТОВИЯ ПРОЗОРЕЦ.

CLEOL - ИЗЧИСТВА ОТ ТЕКУЩАТА ПОЗИЦИЯ ДО КРАЯ НА РЕДА.

ERASE - ИЗЧИСТВА ЦЕЛИЯ ЕКРАН НЕЗАВИСИМО ОТ ТЕКСТОВИЯ ПРОЗОРЕЦ.

A. РЕЕК И РОКЕ

ЧЕТИРИ ДУМИ ПОЗВОЛЯВАТ ЗАПИСВАНЕ В ПАМЕТТА И ЧЕТЕНЕ ОТТАМ. ПОД "ДУМА В ПАМЕТТА" СЕ РАЗБИРАТ ДВА ПОРЕДНИ БАЙТА. ТАЗИ "ДУМА" НЕ ТРЯБВА ДА СЕ БЪРКА С ГРАФОРТ-ДУМИТЕ ОТ БИБЛИОТЕКАТА.

РОКЕW - ИЗВЛИЧА ДВЕ ЧИСЛА ОТ СТЕКА. ПЪРВОТО Е ЧИСЛО МЕЖДУ -32768 И 32767, А ВТОРОТО АДРЕС НА ДУМА В ПАМЕТТА, КЪДЕТО ТОВА ЧИСЛО ТРЯБВА ДА БЪДЕ ЗАПОМНЕНО.

РЕЕКW - ИЗВЛИЧА ЧИСЛО ОТ СТЕКА (АДРЕС НА ДУМА В ПАМЕТТА), А СЪДЪРЖАНИЕТО НА ДУМАТА ОТ ТОЗИ АДРЕС ВРЪЩА ОБРАТНО В СТЕКА.

РОКЕ - ИЗВЛИЧА ДВЕ ЧИСЛА ОТ СТЕКА - ПЪРВОТО СТОИНОСТ В ЕДИН БАЙТ (ОТ 0 ДО 255), А ВТОРОТО АДРЕС, НА КОИТО ДА БЪДЕ ЗАПОМНЕН ТОЗИ БАЙТ.

РЕЕК - ИЗВЛИЧА ЕДНО ЧИСЛО ОТ СТЕКА - АДРЕС НА БАЙТ В ПАМЕТТА. ВРЪЩА В СТЕКА СЪДЪРЖАНИЕТО НА ТОЗИ БАЙТ.

ДА ЗАПОМНИМ ЧИСЛОТО 23541 НА АДРЕС 2816:

#> 23541 2816 РОКЕW

#> 2816 РЕЕКW .

23541

ИЗВЛЕЧЕНОТО С РЕЕКW ЧИСЛО ОТ ДУМАТА НА АДРЕС 2816 Е ТОЧНО ЗАПАМЕТЕНОТО ТАМ - 23541.

#> 128 2820 РОКЕ

#> 2820 РЕЕК .

128

А ТОВА БЕШЕ АНАЛОГИЧЕН ПРИМЕР С БАЙТОВИТЕ КОМАНДИ РОКЕ И РЕЕК.

ОПЕРАТОР З Т А С К ОБЪСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

РОКЕW N A - - - РОКЕ WORD
 ЗАПОМНЯ ЧИСЛОТО НА АДРЕС A

РОКЕ N A - - - РОКЕ BYTE
 ЗАПОМНЯ БАЙТА НА АДРЕС A

РЕЕКW A - - N - - РЕЕК WORD
 ЧЕТЕ ЧИСЛО ОТ АДРЕС A

РЕЕК A - - N - - РЕЕК BYTE
 ЧЕТЕ БАЙТ ОТ АДРЕС A

В. ПРОМЕНЛИВИ

ГРАФОРТ-ДУМАТА VARIABLE ПОЗВОЛЯВА ДЕФИНИРАНЕ НА НОВИ ДУМИ И ДОБАВЯНЕТО ИМ В БИБЛИОТЕКАТА. ТЯ ИМА ДВА ФОРМАТА.

VARIABLE <ПРОМЕНЛИВА>

ДЕФИНИРА ПРОМЕНЛИВА С ИМЕ <ПРОМЕНЛИВА>.

#> LIST

CHK
ABS
CON

#> VARIABLE P1

#> LIST

P1
CHK
ABS
CON

НОВАТА ДУМА "P1" СЕ СЪСТОИ ОТ ДВЕ ЧАСТИ - ДВУБАЙТОВО МЯСТО ЗА ЗАПАЗВАНЕ НА ЧИСЛО И АДРЕС НА ВЪТРЕШНА ПОДПРОГРАМА, КОЯТО ИЛИ ВЗИМА ЧИСЛО ОТ СТЕКА И ГО СЛАГА В ПРОМЕНЛИВАТА ИЛИ ВЗИМА ЧИСЛО ОТ ПРОМЕНЛИВАТА И ГО СЛАГА В СТЕКА. ЗА ДА СЕ ЗАПИШЕ ЧИСЛО В ЕДНА ПРОМЕНЛИВА СЕ ИЗПОЛЗВА ДУМАТА "->". НА ПРАКТИКА ДУМАТА -> ПРОМЕНЯ САМО ЕДИН ВЪТРЕШЕН КЛЮЧ, КОИТО ПОКАЗВА ЧЕ ПРИ ПОСЛЕДВАЩО ПОЯВЯВАНЕ НА ИМЕ НА ПРОМЕНЛИВА, ВЪТРЕШНАТА ПОДПРОГРАМА ЗА КОЯТО ГОВОРИХМЕ ТРЯБВА ДА ВЗЕМЕ ЧИСЛО ОТ СТЕКА И ДА ГО ЗАПОМНИ В ПРОМЕНЛИВАТА. ПРИ НОРМАЛНО ПОЯВЯВАНЕ НА ИМЕ НА ПРОМЕНЛИВА СТОИНОСТТА НА ПРОМЕНЛИВАТА СЕ СЛАГА В СТЕКА.

#> 432 -> P1

#> P1 .
432

ВТОРИЯТ ФОРМАТ НА УПОТРЕБА НА ДУМАТА "VARIABLE" ПОЗВОЛЯВА ЗАДАВАНЕТО НА НАЧАЛНА СТОИНОСТ НА ПРОМЕНЛИВАТА:

<НАЧ.СТОИНОСТ> VARIABLE <ПРОМЕНЛИВА>

НАЧАЛНАТА СТОИНОСТ НЕ СЕ ВЗИМА ОТ СТЕКА, А ДИРЕКТНО ОТ ПРОГРАМНИЯ РЕД, ТАКА ЧЕ НЕ МОЖЕ ДА ИМА НА ТОВА МЯСТО АРИТМЕТИЧЕН ИЗРАЗ.

#> 17 VARIABLE P2

#> P2 .
17

СТОЙНОСТТА 17 БЕ ПРИСВОЕНА НА ПРОМЕНЛИВАТА П2 ПО ВРЕМЕ НА ДЕФИНИРАНЕТО. АКО НАЧАЛНА СТОЙНОСТ НЕ БЪДЕ ЗАДАДЕНА, ТО ПРОМЕНЛИВАТА СЕ ИНИЦИАЛИЗИРА С НУЛЕВА СТОЙНОСТ.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ .
=====ПРЕДИ=====СЛЕД=====

VARIABLE - - - - -
 ДЕКЛАРИРА ПРОМЕНЛИВА С ИМЕ
 ТОВА КОЕТО СЛЕДВА И ОПЦИОННО
 НАЧАЛНА СТОЙНОСТ ПРЕДИ НЕЯ

-> N - - X X X
 ЗАПАЗВА N В ПРОМЕНЛИВАТА
 СЛЕДВАЩА ->

II.13 НИЗОВЕ

A. ДЕФИНИРАНЕ НА НИЗОВЕ

ДУМАТА "STRING" СЕ ИЗПОЛЗВА ЗА ДЕФИНИРАНЕ НА НИЗОВИ ПРОМЕНЛИВИ
- ПРОМЕНЛИВИ, КОИТО СЪДЪРЖАТ СИМВОЛНА (ТЕКСТОВА) ИНФОРМАЦИЯ.

<РАЗМЕР> STRING <НИЗ>

ТЪЙ КАТО ГРАФОРТ ПРАВИ МНОГО МАЛКО ПРОВЕРКИ, ВНИМАТЕЛНО ТРЯБВА
ДА БЪДЕ ПРЕЦЕНЕН МАКСИМАЛНИЯТ РАЗМЕР (<РАЗМЕР>) НА НИЗА <НИЗ>.

#> 30 STRING НИЗ

#> LIST

НИЗ
CHS
ABS
SON
.
.

НИЗОВЕТЕ В ГРАФОРТ СА ИНДЕКСИРАНИ ОТ НУЛА ДО <РАЗМЕР>-1. КОГАТО
СЕ СРЕЩНЕ ДУМА ДЕФИНИРАНА ЧРЕЗ STRING (НИЗ) ТО СЕ ИЗВАНДА ЧИСЛО ОТ
СТЕКА, КОЕТО СЕ ПРИБАВЯ КЪМ НАЧАЛНИЯ АДРЕС НА НИЗА И ТАКА ПОЛУЧЕНИЯ
АДРЕС СЕ ВРЪЩА ОБРАТНО В СТЕКА.

#> 0 НИЗ .
-32240

#> 10 НИЗ .
-32230

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ====СЛЕД=====

STRING - - - - -
 СЛЕДВА СЕ ОТ ИМЕТО НА НИЗА И
 СЕ ПРЕДХОЖДА ОТ БРОЯ СИМВОЛИ

В. ИЗПОЛЗВАНЕ НА НИЗОВЕ

ЗА ЗАПОМНЯНЕ НА ТЕКСТ В ЕДИН НИЗ (ИЛИ НЯКЪДЕ В ПАМЕТТА) СЕ ИЗПОЛЗВА ДУМАТА "ASSIGN".

<АДРЕС> ASSIGN " <ТЕКСТ> "

ASSIGN ВЗИМА ЧИСЛО ОТ СТЕКА, ПРИЕМА ГО КАТО АДРЕС И ПОСТАВЯ ОТ ТОЗИ АДРЕС В ПАМЕТТА ТЕКСТА <ТЕКСТ>. ОБИКНОВЕНО АДРЕСА СЕ ЗАПИСВА В СТЕКА ЧРЕЗ ИЗПОЛЗВАНЕ НА ИМЕ НА НИЗ ПРЕДИ ASSIGN.

#> 0 НИЗ

#> ASSIGN " ДЕМО ТЕКСТ "

ЗА ДА СЕ ИЗВЕДЕ ЕДИН НИЗ ВЪРХУ ЕКРАНА СЕ ИЗПОЛЗВА ДУМАТА WRITELN.

<АДРЕС> WRITELN

WRITELN ВЗИМА АДРЕС ОТ СТЕКА И ИЗВЕЖДА НА ЕКРАНА ТЕКСТА ЗАПОЧВАЩ ОТ ТОЗИ АДРЕС.

#> 0 НИЗ WRITELN
ДЕМО ТЕКСТ

ТЕКСТОВА ИНФОРМАЦИЯ МОЖЕ ДА БЪДЕ ПРОЧЕТЕНА ОТ КЛАВИАТУРАТА И ДА БЪДЕ ЗАПОМНЕНА В НИЗ (ИЛИ КЪДЕ ДА Е В ПАМЕТТА). ЗА ЦЕЛТА СЕ ИЗПОЛЗВА ДУМАТА READLN.

<АДРЕС> READLN

READLN ВЗИМА АДРЕС ОТ СТЕКА, ЧЕТЕ ОТ КЛАВИАТУРАТА СИМВОЛИ ДО "RETURN" И ГИ ЗАПИСВА НА ТОЗИ АДРЕС В ПАМЕТТА (БЪКЛУЧИТЕЛНО И RETURN-СИМВОЛА ЗА КРАЙ НА НИЗА).

ЕТО И ПРИМЕР:

#> 0 НИЗ READLN
ДЕМО ТЕКСТ2
#>

"ДЕМО ТЕКСТ2" БЕШЕ ПРОЧЕТЕН В НИЗА "НИЗ".

#> 0 НИЗ WRITELN
ДЕМО ТЕКСТ2

```

#> 5 НИЗ READLN
ЧЕТЕНЕ
#> 0 НИЗ WRITELN
ДЕМО ЧЕТЕНЕ
#> 1 НИЗ WRITELN
ЕМО ЧЕТЕНЕ

```

```

ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

```

```

ASSIGN      A - - - - -
            СЛЕДВАЩИЯ ТЕКСТ НА АДРЕС А

READLN      A - - - - - READ LINE
            ЧЕТЕ ОТ КЛАВИАТ. НА АДРЕС А

WRITELN     A - - - - - WRITE LINE
            ПИШЕ ПО ЕКРАНА ОТ АДРЕС А

```

С. ПРЕОБРАЗУВАНЕ НА НИЗОВЕ

ЧЕСТО СЕ НАЛАГА ДА БЪДАТ ЧЕТЕНИ ЧИСЛА ОТ КЛАВИАТУРАТА. НО ОТ ТАМ МОГАТ ДА БЪДАТ ВЪВЕЖДАНИ САМО СИМВОЛИ, СЛЕДОВАТЕЛНО МОГАТ ДА СЕ ЧЕТАТ ТЕКСТОВЕ С READLN В НИЗОВЕ. "GETNUM" Е ДУМАТА, КОЯТО ПРЕОБРАЗУВА ЕДНА ПОРЕДИЦА ОТ ЦИФРОВИ СИМВОЛИ В ЧИСЛО. ТЯ ВЗИМА ОТ СТЕКА АДРЕСА НА НИЗА И СЕ ОПИТВА ДА ОТКРИЕ ПОРЕДИЦА ОТ ЦИФРИ ЗАПОЧВАЙКИ ОТ НАЧАЛОТО. АКО НЕ ОТКРИЕ ЗРЪЩА В СТЕКА НУЛА, В ПРОТИВЕН СЛУЧАЙ ВРЪЩА НАМЕРЕНОТО ЧИСЛО.

ТУК ИМА ОЩЕ ЕДИН ПРОБЛЕМ - ДАЛИ НУЛА В СТЕКА СЛЕД GETNUM Е НАМЕРЕНОТО ЧИСЛО ИЛИ ТОВА Е ИНДИКАЦИЯ ЗА ЛИПСА НА ЧИСЛО ? ЗА ТАЗИ ЦЕЛ СЕ ИЗПОЛЗВА ДУМАТА "VALID". VALID ВРЪЩА В СТЕКА НУЛА АКО НЕ Е НАМЕРЕНО ЧИСЛО ОТ GETNUM И РАЗЛИЧНО ОТ НУЛА АКО Е НАМЕРЕНО ВАЛИДНО ЧИСЛО.

```

#> 0 НИЗ READLN
234
#> 0 НИЗ GETNUM .
234
#> VALID .
142

```

```

#> 0 НИЗ READLN
ЧИСЛО
#> 0 НИЗ GETNUM .
0
#> VALID .
0

```


LEFT\$ - ВЗИМА ТРИ ЧИСЛА ОТ СТЕКА: АДРЕС НА ВХОДЕН НИЗ, АДРЕС НА ИЗХОДЕН НИЗ И БРОИ СИМВОЛИ. ВЗИМА ПОСОЧЕНИЯ БРОИ СИМВОЛИ ОТ ВХОДНИЯ НИЗ И ФОРМИРА ОТ ТЯХ ИЗХОДНИЯ НИЗ.

```
#> 0 НИЗ READLN
ЕЛИЗАБЕТ
#> 0 НИЗ PAD 5 LEFT$
```

```
#> PAD WRITELN
ЕЛИЗА
```

RIGHT\$ - КАТО LEFT\$, САМО ЧЕ ВЗИМА ПОСОЧЕНИЯ БРОИ СИМВОЛИ ОТДЯСНО.

```
#> 0 НИЗ PAD 3 RIGHT$
#> PAD WRITELN
БЕТ
```

MOVELN - ВЗИМА ДВА АДРЕСА ОТ СТЕКА И КОПИРА ВХОДНИЯ НИЗ В ИЗХОДНИЯ.

```
#> 0 НИЗ PAD MOVELN
#> PAD WRITELN
ЕЛИЗАБЕТ
```

CONCAT - ВЗИМА ОТНОВО ДВА АДРЕСА ОТ СТЕКА И ПРИСЪЕДИНЯВА ВТОРИЯ НИЗ КЪМ ПЪРВИЯ.

```
#> PAD READLN
ГРА
#> 0 НИЗ READLN
ФОРТ
#> PAD 0 НИЗ CONCAT
```

```
#> PAD WRITELN
ГРАФОРТ
```

COMPARE - ВЗИМА ДВА АДРЕСА НА НИЗОВЕ ОТ СТЕКА И СРАВНЯВА ДВАТА НИЗА. АКО ПЪРВИЯ Е ПО-ГОЛЯМ ВРЪЩА 1, АКО СА РАВНИ ВРЪЩА 0 И АКО ПЪРВИЯ Е ПО-МАЛЪК ВРЪЩА -1. ВНИМАВАЙТЕ ЗА БУКВИТЕ ОТ КИРИЛИЦАТА, ЗАЩОТО ТЕ НЕ СА В НАРАСТВАЩА ПОСЛЕДОВАТЕЛНОСТ ПО ВЪТРЕШЕН КОД.

```
#> PAD ASSIGN " CLASS "
#> 0 НИЗ ASSIGN " CLOSE "
#> PAD 0 НИЗ COMPARE .
```

-1

II.14 ДУМИ ЗА РАБОТА НАД ЕДИНИЧНИ СИМВОЛИ

GETC И PUTC СА ГРАФОРТ-ДУМИТЕ, КОИТО ЧЕТАТ ИЛИ ИЗВЕЖДАТ ЕДИНИЧЕН СИМВОЛ. GETC ЧЕТЕ СИМВОЛ ОТ КЛАВИАТУРАТА И ЗАПИСВА НЕГОВИЯ ASCII-КОД В СТЕКА. PUTC ВЗИМА ОТ СТЕКА ЕДИН ASCII-КОД И ИЗВЕЖДА СИМВОЛ ОТ ТЕКУЩАТА ЗА КУРСОРА ПОЗИЦИЯ.

```
#> 193 PUTC
A
#> GETC .
B
194
#> GETC DUP PUTC CR .
Г
231
```

GETC ИЗЧАКВА ДА БЪДЕ ВЪВЕДЕН СИМВОЛ ОТ КЛАВИАТУРАТА. АКО НЕ ИСКАТЕ ДА СПИРАТЕ ИЗПЪЛНЕНИЕТО НА ПРОГРАМАТА МОЖЕТЕ ДА ИЗПОЛЗВАТЕ GETKEY И CLRKEY.

КОГАТО НАТИСНЕТЕ НЯКОИ КЛАВИШ, ASCII-КОДА НА СЪОТВЕТНИЯ СИМВОЛ СЕ ЗАПАМЕТАВА. GETKEY ЧЕТЕ ТОЗИ ASCII-КОД И ГО ПОСТАВЯ В СТЕКА. CLRKEY ИЗЧИСТВА ЗАПАМЕТАНИЯ ПРЕДИ СИМВОЛ. ВЪВЕДЕНИЯТ ПРЕЗ КЛАВИАТУРАТА КОД Е ВИНАГИ ПО-ГОЛЯМ ОТ 127. ТОВА МОЖЕ ДА БЪДЕ ИЗПОЛЗВАНО ЗА ДА СЕ ПРОБЕРИ ДАЛИ НЯКОИ КЛАВИШ Е БИЛ НАТИСНАТ.

```
#> : СИМВОЛ
#> CLRKEY
#> BEGIN
#> GETKEY
#> 128 <
#> WHILE
#> DROP
#> REPEAT
#> CLRKEY ;
```

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
----------	-----------	-----------

=====ПРЕДИ=====СЛЕД=====

GETC	- - - N - -	GET CHARACTER ЧЕТЕ СИМВОЛ ОТ КЛАВИАТУРАТА
------	-------------	--

GETKEY	- - - N - -	GET KEY ЧЕТЕ СИМВОЛ БЕЗ ЧАКАНЕ
--------	-------------	-----------------------------------

CLRKEY	- - - - -	CLEAR KEY ИЗЧИСТВА ПРЕДИ GETKEY
--------	-----------	------------------------------------

PUTC	N - - - - -	PUT CHARACTER ИЗВЕЖДА СИМВОЛ С КОД N
------	-------------	---

II.15 ЧИСЛА В ДРУГИ БРОЙНИ СИСТЕМИ

ЧЕТИРИ ГРАФОРТ-ДУМИ ПОЗВОЛЯВАТ СМЯНАТА НА ИЗПОЛЗВАНАТА БРОЙНА СИСТЕМА, КОЯТО ПРИ ЗАРЕЖДАНЕ НА ГРАФОРТ Е ДЕСЕТИЧНАТА.

HEX - УСТАНОВЯВА ШЕСТНАДЕСТИЧНА БРОЙНА СИСТЕМА;

BINARY - УСТАНОВЯВА ДВОИЧНА СИСТЕМА;

DECIMAL - ВРЪЩА ДЕСЕТИЧНАТА БРОЙНА СИСТЕМА.

МОГАТ ДА БЪДАТ ИЗПОЛЗВАНИ И КАКВИ ДА Е ДРУГИ БРОЙНИ СИСТЕМИ. ТОВА СТАВА ЧРЕЗ ЗАРЕЖДАНЕТО НА СИСТЕМНАТА ПРОМЕНЛИВА BASE.

КОГАТО СЕ ИЗПОЛЗВАТ БРОЙНИ СИСТЕМИ С ОСНОВА ПО-ГОЛЯМА ОТ 10, ЗА ЦИФРИТЕ ПО-ГОЛЕМИ ОТ 9 СЕ ИЗПОЛЗВАТ БУКВИ. ЗА ДА НЕ СЕ СБЪРКА ЕДНО ЧИСЛО ЗАПОЧВАЩО С БУКВА С ИМЕ, ТО ПРЕДИ НЕГО СЕ ПОСТАВЯ \$.

```
#> STACK 45
```

```
[45]
```

```
#> HEX
```

```
[2D]
```

```
#> BINARY
```

```
[1011011]
```

```
#> DECIMAL
```

```
[45]
```

```
#> 8 -> BASE
```

```
[55]
```

```
#> 12 -> BASE
```

```
[45]
```

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

BASE	ПРОМЕНЛИВА
	ЧИСЛОВА ОСНОВА ЗА СМЯТАНЕ

DECIMAL	10 -> BASE
---------	------------

BINARY	2 -> BASE
--------	-----------

HEX	16 -> BASE
-----	------------

II.16 ИЗПОЛЗВАНЕ НА ДОС ОТ ГРАФОРТ

ДОС-КОМАНДИ МОГАТ ДА СЕ ИЗДАВАТ КАКТО ДИРЕКТНО ОТ КЛАВИАТУРАТА, ТАКА И ОТ ДЕФИНИЦИИ НА ДУМИ. СТАНДАРТНО ДОС СЕ ИЗВИКВА ЧРЕЗ CR (ASCII-КОД 141) ПОСЛЕДВАНО ОТ [D] (ЕДНОВРЕМЕННО НАТИСНАТИ "МК" И "Д/Д") С ASCII-КОД 132.

```
CR 132 PUTC PRINT " <ДОС-КОМАНДА> " CR
```

ВЪВЕДЕТЕ:

```
#> CR 132 PUTC PRINT " CATALOG " CR
```

II.17 ДУМИ ЗА УПРАВЛЕНИЕ НА ПРОГРАМИ

RUN Е ДУМАТА, КОЯТО ПРЕДИЗВИКВА ИЗПЪЛНЕНИЕТО НА НАИ-ГОРНАТА ДУМА ОТ БИБЛИОТЕКАТА. СМИСЪЛЪТ НА ТАЗИ КОМАНДА Е ДА СЕ ИЗПЪЛНИ ТАЗИ ДУМА, КОГАТО НЕИНОТО ИМЕ Е ПРЕКАЛЕНО ДЪЛГО ИЛИ ПЪК ТО Е НЕПОЗНАТО.

AUTORUN Е ДУМА, КОЯТО ЧЕТЕ ОТ СТЕКА ЧИСЛО И АКО ТО Е РАЗЛИЧНО ОТ НУЛА ИЗПЪЛНЯВА ГОРНАТА ДУМА ОТ БИБЛИОТЕКАТА ВИНАГИ КОГАТО УПРАВЛЕНИЕТО ТРЯБВА ДА СЕ ВЪРНЕ НА ГРАФОРТ (ОЧАКВАТЕ ДА СЕ ПОЯВИ #>). АКО ПРОЧЕТЕНОТО ОТ СТЕКА ЧИСЛО Е НУЛА, ТОЗИ РЕЖИМ СЕ ПРЕКРАТЯВА.

ОПЕРАТОР	СТАСК	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

RUN - - - - -
 ИЗПЪЛНЯВА НАИ-ГОРНАТА ДУМА

AUTORUN N - - - - - N#0 ! N#0
 АКО N#0, АВТОМАТИЧЕН RUN

II.18 ЗАПАЗВАНЕ НА СИСТЕМАТА

КОМПИЛАТОРЪТ ГРАФОРТ СЕ НАМИРА ВЪРХУ ДИСКЕТАТА КАТО ИЗПЪЛНИМ ДВОИЧЕН ФАЙЛ ПОД ИМЕТО "ГРАФОРТ". ТОВА Е ФАЙЛЪТ КОИТО СЕ ИЗПЪЛНЯВА, КОГАТО ЗАРЕДИТЕ ДИСКЕТАТА.

ОПЕРАТОРЪТ SAVEPRG ЗАПАЗВА СИСТЕМАТА КАТО ДВОИЧЕН ФАЙЛ ПОДОБЕН НА "ГРАФОРТ". ПО ВСЯКО ВРЕМЕ ПО-КЪСНО ТОЗИ ФАЙЛ МОЖЕ ДА БЪДЕ ЗАРЕДЕН И ИЗПЪЛНЕН ЧРЕЗ DOS-КОМАНДАТА "BRUN".

СМИСЪЛЪТ НА ТАЗИ КОМАНДА Е В ЗАПАЗВАНЕТО НА ПОТРЕБИТЕЛСКИ ПРОГРАМИ (НАБОР ДУМИ В БИБЛИОТЕКАТА) НА ДИСКЕТА. СЛЕД ИЗДАВАНЕТО НА КОМАНДАТА SAVEPRG ЩЕ ВИ БЪДЕ ПОИСКАНО ИМЕТО НА ФАЙЛА КЪДЕТО ЩЕ БЪДЕ ЗАПАЗЕНА СИСТЕМАТА, А СЛЕД ТОВА И ЕДИН ДОПЪЛНИТЕЛЕН ВЪПРОС - ДАЛИ ИСКАТЕ АВТОМАТИЧНО (AUTORUN) ИЗПЪЛНЕНИЕ НА ГОРНАТА ДУМА ОТ БИБЛИОТЕКАТА СЛЕД ЗАРЕЖДАНЕ.

ГРАФОРТ ИЗПОЛЗВА МОДИФИЦИРАН DOS, ТАКА ЧЕ ВСЯКА СИСТЕМА ТРЯБВА ДА РАБОТИ ПОД НЕГОВО УПРАВЛЕНИЕ. ИЗПОЛЗВАЙТЕ КОПИЕ ОТ СИСТЕМНАТА ДИСКЕТА ЗА ДА ЗАПАЗВАТЕ ВАШИТЕ ПРОГРАМИ. АКО ИСКАТЕ СЪЗДАДЕНАТА ДИСКЕТА ДА ИЗПЪЛНЯВА АВТОМАТИЧНО САМО ВАШАТА ПРОГРАМА И ДА НЕ МОЖЕ ДА СЕ ИЗПОЛЗВА ЗА ДРУГО, КОМПИЛИРАЙТЕ ПРОГРАМАТА, ЗАПАЗЕТЕ КАТО "AUTORUN" СИСТЕМАТА ПОД ИМЕТО "ГРАФОРТ" И ИЗТРИЙТЕ ВСИЧКИ ОСТАНАЛИ ФАЙЛОВЕ ОТ ДИСКЕТАТА.

ПРИМЕР:

#> SAVEPRG

SAVE ФАЙЛ-ИМЕ :ГРАФОРТ

AUTORUN (Y/N) :Y

ОПЕРАТОР	СТАСК	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

SAVEPRG - - - - -
 ЗАПАЗВА СИСТЕМАТА НА ДИСК

II.19 ПРЕПОКРИВАЩИ СЕ СТРУКТУРИ (ОВЪРЛЕИ)

ГРАФОРТ-ПРОГРАМИТЕ МОГАТ ДА ЗАРЕЖДАТ И ИЗПЪЛНЯВАТ ДРУГИ ГРАФОРТ-ПРОГРАМИ ДОРИ КАТО ГИ ЗАРЕЖДАТ ВЪРХУ СЕБЕ СИ. ПРОГРАМНИ СЕГМЕНТИ, КОИТО ИЗПОЛЗВАТ ЕДНО И СЪЩО МЯСТО В ПАМЕТТА СЕ НАРИЧАТ ПРЕПОКРИВАЩИ СЕ (OVERLAY).

ЗА ДА ИЗПЪЛНИТЕ ЕДИН ТАКЪВ СЕГМЕНТ ВКЛЮЧЕТЕ ВЪВ ВИКАЩАТА ПРОГРАМА КОМАНДА READ, КОЯТО ДА ГО ПРОЧЕТЕ. САМИЯТ СЕГМЕНТ ТРЯБВА ДА ЗАПОЧВА С FORGET, ТАКА ЧЕ ДА ИЗТРИЕ НЕИЗПОЛЗВАНИТЕ ДУМИ ОТ БИБЛИОТЕКАТА (ДУМИТЕ, КОИТО ТРЯБВА ДА ПРЕПОКРИЕ). ТОИ ТРЯБВА ДА ЗАВЪРШВА С "CLOSE RUN" - CLOSE ЩЕ ЗАТВОРИ ДОС-ФАЙЛА, А RUN ЩЕ ИЗПЪЛНИ ПЪРВАТА ДУМА ОТ БИБЛИОТЕКАТА, КОЯТО НА ПРАКТИКА Е ГЛАВНАТА ДУМА НА ТОЗИ СЕГМЕНТ.

ДЕМОНСТРАЦИОННАТА ПРОГРАМА И ПОМОЩНОТО МЕНЮ ИЗПОЛЗВАТ ШИРОКО ТОЗИ МЕТОД НА РАБОТА - РАЗГЛЕДАЙТЕ ГИ ПОДРОБНО.

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
CLOSE      - - - - -
            ЗАТВАРЯ ВСИЧКИ ФАЙЛОВЕ
```

II.20 ПРЕМЕСТВАНЕ В ПАМЕТТА

ОПЕРАТОРЪТ MOVMEM ПРЕМЕСТВА БЛОКОВЕ ПАМЕТ ОТ ЕДНО МЯСТО НА ДРУГО.

```
<АДРЕС1> <АДРЕС2> <БРОИ> MOVMEM
```

ПРЕМЕСТВАТ СЕ <БРОИ> БАЙТОВЕ ОТ АДРЕС <АДРЕС1> НА АДРЕС <АДРЕС2>.

```
#> 2048 2816 200 MOVMEM
```

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
MOVMEM      A B N      - - -
            МЕСТИ N БАЙТА ОТ АДРЕС A НА B
```

II.21 ОТКРИВАНЕ АДРЕСИТЕ НА ДУМИ

ДУМАТА "'" (АПОСТРОФ) ЗАПАЗВА В СТЕКА АДРЕСА НА ИЗПЪЛНИМАТА ЧАСТ ОТ ДУМАТА, КОЯТО СЛЕДВА. СЛЕДВАЩАТА ДУМА НЕ СЕ ИЗПЪЛНЯВА. С #LIST СЕ ИЗВЕЖДАТ АДРЕСИТЕ НА НАЧАЛОТО НА ДЕФИНИЦИЯТА НА ВСЯКА ДУМА, А ТУК АДРЕСЪТ Е НА ИЗПЪЛНИМАТА ЧАСТ.

```
#> ' ERASE .
30822
```

ВИЖДАТЕ АДРЕСА НА ERASE, КАТО ЕКРАНА НЕ СЕ ЧИСТИ (ERASE НЕ СЕ ИЗПЪЛНЯВА).

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

- - - A - -
 A - АДРЕСА НА ДУМАТА СЛЕД '

II.22 ИЗПЪЛНЕНИЕ НА ПОДПРОГРАМИ НА МАШИЕН ЕЗИК

ЧРЕЗ ДОС-КОМАНДАТА "BLOAD" В ПАМЕТТА МОГАТ ДА СЕ ЗАРЕДЯТ ПОДПРОГРАМИ НА МАШИЕН ЕЗИК. ДУМАТА "CALL" ВЗИМА АДРЕС ОТ СТЕКА И ИЗПЪЛНЯВА ПОДПРОГРАМАТА ЗАПОЧВАЩА ОТ ТОЗИ АДРЕС. ПОДПРОГРАМАТА ТРЯБВА ДА ЗАВЪРШВА С RTS-ИНСТРУКЦИЯ (RETURN FROM SUBROUTINE).

РЕГИСТРИТЕ НА ПРОЦЕСОРА "A", "X", "Y" И "P" МОГАТ ПРЕДВАРИТЕЛНО ДА БЪДАТ ЗАРЕДЕНИ С ПОДХОДЯЩИ СТОЙНОСТИ. ТОВА СТАВА ЧРЕЗ ЗАРЕЖДАНЕ НА ТЕЗИ СТОЙНОСТИ В СИСТЕМНИТЕ ПРОМЕНЛИВИ НА ГРАФОРТ - AREG, XREG, YREG И PREG. ПРЕДИ ДА ПРЕДАДЕ УПРАВЛЕНИЕТО НА ПОДПРОГРАМАТА, ГРАФОРТ ЩЕ ЗАРЕДИ РЕГИСТРИТЕ, А СЛЕД ЗАВЪРШВАНЕТО И СТОЙНОСТИТЕ ИМ ЩЕ ЗАРЕДИ ОБРАТНО В СИСТЕМНИТЕ ПРОМЕНЛИВИ, ТАКА ЧЕ ТЕ ДА СА ДОСТЪПНИ ЗА АНАЛИЗ.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

CALL A - - - - -
 ЗАРЕЖДА РЕГИСТРИТЕ ОТ AREG,
 XREG, YREG, PREG; ИЗПЪЛНЯВА
 ПРОГРАМАТА ОТ АДРЕС A; ЗАРЕЖ-
 ДА ОТНОВО AREG XREG YREG PREG

PREG (ПРОМЕНЛИВА)

AREG (ПРОМЕНЛИВА)

XREG (ПРОМЕНЛИВА)

YREG (ПРОМЕНЛИВА)

II.23 КОМПИЛИРАНЕ НА ЧИСЛОВИ ТАБЛИЦИ

ДУМАТА ":", " SE ИЗПОЛЗВА ПРИ ДЕФИНИЦИИ НА ДУМИ И КОМПИЛИРА (ЗАПАЗВА В ЕДИН БАЙТ) ЧИСЛОТО, КОЕТО Я ПРЕДХОЖДА. ПЪРВОТО ЧИСЛО В ДЕФИНИЦИЯТА ТРЯБВА ДА Е РАЗЛИЧНО ОТ 10 (\$A) И ДА Е ПО-МАЛКО ОТ 128 (\$80).

:", " МОЖЕ ДА СЕ ИЗПОЛЗВА ЗА ГЕНЕРИРАНЕ НА ЧИСЛОВИ ТАБЛИЦИ ИЛИ ЗА ПИСАНЕ НА МАЛКИ ПРОГРАМКИ, НА МАШИЕН ЕЗИК.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
 =====ПРЕДИ====СЛЕД=====

- - - - - -
 КОМПИЛИРА ЕДИНИЧЕН БАЙТ

II.24 ПРИКЛЮЧВАНЕ НА РАБОТАТА С ГРАФОРТ

ДУМАТА "BYE" (ДОВИЖДАНЕ) СЕ ИЗПОЛЗВА ЗА ВЛИЗАНЕ В МОНИТОРА НА ПРАВЕЦ. ЗА ДА СЕ ВЪРНЕТЕ ОБРАТНО НАПИШЕТЕ 6000G.

#> BYE

*6000G

#>

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

BYE

ИЗХОД КЪМ МОНИТОРА НА ПРАВЕЦ

II.25 ГРАФИКИ С ГРАФОРТ

НЕЗАВИСИМО ОТ РЕЖИМА (ТЕКСТОВ ИЛИ ГРАФИЧЕН) ЕКРАНЪТ НА ПРАВЕЦ СЕ СЪСТОИ ОТ ЕДНИ И СЪЩИ ЕДИНИЦИ (ТОЧКИ). ЕКРАНЪТ Е РАЗДЕЛЕН НА 280 (ХОРИЗОНТАЛНО) ПО 192 (ВЕРТИКАЛНО) ТОЧКИ.

В ТЕКСТОВ РЕЖИМ ИМА 24 РЕДА ПО 8 ТОЧКИ ВСЕКИ И 40 КОЛОНИ ПО 7 ТОЧКИ ВСЯКА. ВСЕКИ СИМВОЛ Е ДЕФИНИРАН В ЕДНА МАТРИЦА ОТ 7 (ХОРИЗОНТАЛНО) ПО 8 (ВЕРТИКАЛНО) ТОЧКИ. В ДЕЙСТВИТЕЛНОСТ СИМВОЛИТЕ СА ДЕФИНИРАНИ В МАТРИЦА 5 X 7 ТОЧКИ, ТАКА ЧЕ ДА ОСТАНАТ ПО 2 ПРАЗНИ ТОЧКИ МЕЖДУ СИМВОЛИТЕ И ПО ЕДНА МЕЖДУ РЕДОВЕТЕ. НОРМАЛНО В РОМ ПАМЕТТА СЕ НАМИРАТ ДЕФИНИЦИИТЕ (ГРАФИЧНОТО ПОКРИТИЕ) ЗА ВСЕКИ ЕДИН СИМВОЛ И ЗА ВСЕКИ ПРОЧЕТЕН ОТ КЛАВИАТУРАТА ASCII-КОД СЕ НАМИРА СЪОТВЕТНАТА МАТРИЦА ОТ 7 X 5 ТОЧКИ, КОЯТО СЕ ИЗВЕЖДА ВЪРХУ ЕКРАНА.

В ГРАФИЧЕН РЕЖИМ СЕ РАБОТИ ИНДИВИДУАЛНО С ВСЯКА ЕДНА ОТ 53760-ТЕ ТОЧКИ ПО ЕКРАНА.

НОРМАЛНИЯТ ЗА ГРАФОРТ РЕЖИМ НА РАБОТА Е ГРАФИЧНИЯТ РЕЖИМ. ГЛЕДАЙКИ ДЕМОНСТРАЦИЯТА ВНЕ СТЕ ЗАБЕЛЯЗАЛИ НАЛИЧИЕТО НА ГОЛЕМИ И МАЛКИ БУКВИ, КАКТО И МНОГОТО РАЗЛИЧНИ ШРИТОВЕ. ТУК ТАБЛИЦАТА С ГРАФИЧНИТЕ СИМВОЛИ СЕ НАМИРА В РАМ-ПАМЕТТА (НА АДРЕС 2048) И ЛЕСНО МОЖЕ ДА БЪДЕ ЗАРЕДЕН ТАМ ДРУГ НАБОР СИМВОЛИ.

ГРАФИЧНИЯТ РЕЖИМ НА ГРАФОРТ СЕ РАЗЛИЧАВА МАЛКО ОТ НОРМАЛНИЯ ЗА ПРАВЕЦ. ТУК ХОРИЗОНТАЛНО ИМА НЕ 280, А 256 ТОЧКИ. ВСЯКА ТОЧКА ИМА КООРДИНАТИ (X,Y) ЗА X ОТ 0 ДО 255 И Y ОТ 0 ДО 191. ТОЧКАТА С КООРДИНАТИ (0,0) Е В ГОРНИЯ ЛЯВ ЪГЪЛ. ПРИ ГРАФОРТ И X-КООРДИНАТАТА МОЖЕ ДА БЪДЕ ЗАПАЗЕНА В ЕДИН БАЙТ, КОЕТО УСКОРЯВА ЗНАЧИТЕЛНО РАБОТАТА И ПРАВИ ДОСТЪПНА АНИМАЦИЯТА.

II.26 ДВУМЕРНИ ГРАФИЧНИ ФУНКЦИИ

A. PLOT, LINE И FILL

PLOT ВЗИМА ДВЕ ЧИСЛА ОТ СТЕКА ИНТЕРПРЕТИРА ГИ КАТО X И Y КООРДИНАТИ И ИЗВЕЖДА НА ЕКРАНА ТОЧКА НА ТОВА МЯСТО.

LINE ВЗИМА СЪЩО X И Y КООРДИНАТИ ОТ СТЕКА И ЧЕРТАЕ ЛИНИЯ ОТ ПОСЛЕДНАТА ИЗВЕДЕНА ТОЧКА ДО ТОЧКАТА С ИЗВЛЕЧЕНИТЕ ОТ СТЕКА КООРДИНАТИ.

FILL ВЗИМА ОТ СТЕКА КООРДИНАТИТЕ НА ТОЧКА, КОЯТО РАЗГЛЕЖДА КАТО ЪГЪЛ НА ПРАВОЪГЪЛНА ОБЛАСТ ВЪРХУ ЕКРАНА. ЗА ПРОТИВОПОЛОЖЕН ЪГЪЛ СЕ СЧИТА ПОСЛЕДНАТА ИЗВЕДЕНА ТОЧКА.

POSN ОТНОВО ВЗИМА ОТ СТЕКА КООРДИНАТИ НА ТОЧКА, НО НЕ ИЗВЪРШВА НИКАКВО ВИДИМО ДЕЙСТВИЕ. ТАЗИ ТОЧКА СЕ МАРКИРА КАТО "ПОСЛЕДНА ИЗВЕДЕНА ТОЧКА" И МОЖЕ ДА БЪДЕ ИЗПОЛЗВАНА ОТ LINE И FILL.

- #> 0 40 18 24 WINDOW
- #> ERASE
- #> 0 0 PLOT
- #> 200 25 PLOT
- #> 50 100 LINE
- #> 100 10 PLOT 100 140 LINE
- #> 120 125 PLOT 200 75 FILL
- #> 225 50 POSN
- #> 250 125 LINE

С WINDOW ТЕКСТОВИЯ ПРОЗОРЕЦ СЕ ФИКСИРА МЕЖДУ 18-ТИ И 24-ТИ РЕД, ЗА ДА НЕ СЕ "ТЪРКАЛЯ" ТЕКСТА ПРЕЗ ГРАФИКИТЕ. С ERASE СЕ ИЗЧИСТВА ЕКРАНА.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

- PLOT X Y - - - -
 ИЗВЕЖДА ТОЧКАТА (X,Y)

- FILL X Y - - - -
 ЗАПЪЛВА ПРАВОЪГЪЛНА ОБЛАСТ
 ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА
 ПО ДИАГОНАЛА ДО (X,Y)

- LINE X Y - - - -
 ЧЕРТАЕ ЛИНИЯ ОТ ПОСЛЕДНАТА
 МАРКИРАНА ТОЧКА ДО (X,Y)

- POSN X Y - - - - POSITION
 ПОЗИЦИОНИРА ДАДЕНА ТОЧКА
 БЕЗ ДА Я ОТБЕЛЯЗВА

B. COLOR

ГРАФОРТ ПОЗВОЛЯВА ДА СЕ ЧЕРТАЕ И В ЦВЕТОВЕ. ЦВЕТЪТ СЕ ИЗБИРА ЧРЕЗ КОМАНДАТА COLOR. COLOR ВЗИМА ЧИСЛО ОТ СТЕКА И ГО ИЗПОЛЗВА ЗА ДА ИЗБЕРЕ ЦВЯТ. ОСЕМТЕ ЦВЯТА СЕ КОДИРАТ ЧРЕЗ СЛЕДНИТЕ ЦИФРИ:

0	---
1	ЗЕЛЕН
2	ВИОЛЕТОВ
3	БЯЛ
4	---
5	ОРАНЖЕВ
6	СИН
7	БЯЛ

ЦВЕТОВЕТЕ 1,2 И 3 ОТ ЕДНА СТРАНА И ЦВЕТОВЕТЕ 5,6 И 7 ОТ ДРУГА, НЕ Е ЗА ПРЕПОРЪЧВАНЕ ДА БЪДАТ СМЕСВАНИ ПОМЕЖДУ СИ.

#> : ТЕСТ.ЦВЕТОВЕ

#> 8 1 DO

#> I COLOR

#> I 20 * 0 + 0 PLOT

#> I 20 * 100 + 100 LINE

#> LOOP ;

С ДУМАТА "ТЕСТ.ЦВЕТОВЕ" ЩЕ БЪДАТ ИЗВЕДЕНИ 7 ЛИНИИ - ПЪРВИТЕ ТРИ В ЦВЕТОВЕТЕ 1,2 И 3, ЧЕТВЪРТАТА ЩЕ БЪДЕ ЧЕРНА (НИКАКЪВ ЦВЯТ) И ПОСЛЕДНИТЕ ТРИ В ЦВЕТОВЕ 5,6 И 7.

#> ERASE ТЕСТ.ЦВЕТОВЕ

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

COLOR	N - - - - -
	ЦВЯТ ЗА ЛИНИИ И СИМВОЛИ

C. UNPLOT, UNLINE И EMPTU

UNPLOT, UNLINE И EMPTU СА ПРОТИВОПОЛОЖНИТЕ ДУМИ НА PLOT, LINE И FILL.

UNPLOT ВЗИМА ОТ СТЕКА КООРДИНАТИТЕ НА ТОЧКАТА, КОЯТО ТРЯБВА ДА ИЗТРИЕ.

UNLINE ИЗТРИВА ЛИНИЯТА ОТ ПОСЛЕДНАТА ИЗВЕДЕНА (МАРКИРАНА) ТОЧКА ДО ТАЗИ С КООРДИНАТИ ЗАПИСАНИ В СТЕКА.

EMPTY ИЗТРИВА ПРАВОЪГЪЛНАТА ОБЛАСТ С ПРОТИВОПОЛОЖНИ ЪГЛИ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА И ТАЗИ, ЧИИТО КООРДИНАТИ СА ЗАПИСАНИ В СТЕКА.

АКО ЕДНА ТОЧКА, ЛИНИЯ ИЛИ ПРАВОЪГЪЛНА ОБЛАСТ Е ИЗВЕДЕНА В ДАДЕН ЦВЯТ, ТО ТЯ ТРЯБВА ДА БЪДЕ ИЗТРИТА ТОГАВА КОГАТО Е В СИЛА СЪЩИЯ ЦВЯТ. АКО ПО ВРЕМЕ НА ЧЕРТАНЕ НЕ СТЕ ЗАДАЛИ ЯВНО ЦВЯТ, ТО ЦВЕТА ПО ПРЕМЪЛЧАВАНЕ Е 3 (БЯЛ).

```
#> ERASE 3 COLOR
#> 50 50 PLOT
#> 100 50 PLOT
#> 50 50 UNPLOT
#> 0 0 PLOT 100 100 LINE
#> 50 0 PLOT 150 100 LINE
#> 0 0 UNPLOT 100 100 UNLINE
#> 25 75 PLOT 100 125 FILL
#> 175 25 PLOT 225 100 FILL
#> 25 75 UNPLOT 100 125 EMPTY
```

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
----------	-----------	-----------

=====ПРЕДИ====СЛЕД=====

UNPLOT	X Y - - - -	ИЗТРИВА ТОЧКАТА (X,Y)
UNLINE	X Y - - - -	ИЗТРИВА ЛИНИЯ ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА ДО (X,Y)
EMPTY	X Y - - - -	ИЗТРИВА ПРАВОЪГЪЛНА ОБЛАСТ ВН. FILL

D. INVERSE И NORMAL

ДУМИТЕ INVERSE И NORMAL НЕ РАБОТЯТ СЪС СТЕКА. INVERSE ПРЕДИЗВИКВА РАБОТА С ДОПЪЛНИТЕЛНИТЕ ЦВЕТОВЕ, А NORMAL ВРЪЩА СТАНДАРТНИЯ РЕЖИМ.

ЕТО И ЕДНА ТАБЛИЦА, КОАТО ДАВА СЪОТВЕТСТВИЕТО МЕЖДУ ОСНОВНИТЕ И ДОПЪЛНИТЕЛНИТЕ (ИЗПОЛЗВАНИТЕ ОТ INVERSE) ЦВЕТОВЕ. С "----" Е МАРКИРАНА ЛИПСАТА НА ЦВЯТ, КОЕТО МОЖЕ ДА БЪДЕ СЧИТАНО ЗА ЧЕРЕН ЦВЯТ.

	ОСНОВЕН ЦВЯТ		ДОПЪЛНИТЕЛЕН ЦВЯТ
0	---	3	БЯЛ
1	ЗЕЛЕН	2	ВИОЛЕТОВ
2	ВИОЛЕТОВ	1	ЗЕЛЕН
3	БЯЛ	0	---
4	---	7	БЯЛ
5	ОРАНЖЕВ	6	СИН
6	СИН	5	ОРАНЖЕВ
7	БЯЛ	4	---

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

INVERSE - - - - -
 ДОПЪЛНИТЕЛЕН ЦВЯТ ЗА ГРАФИКИ
 И ТЕКСТОВЕ (И БЯЛО И ЧЕРНО)

NORMAL - - - - -
 ВРЪЩА НОРМАЛНИ ЦВЕТОВЕ

Е. ORMODE И EXMODE

ГРАФОРТ ИМА ДВА РЕЖИМА НА ИЗВЕЖДАНЕ (ЗАПАЛВАНЕ) НА ТОЧКИ ПО ЕКРАНА - РЕЖИМ ORMODE И РЕЖИМ EXMODE.

В РЕЖИМ ORMODE PLOT, LINE И FILL ИЗВЕЖДАТ ВЪРХУ ЕКРАНА ВСИЧКИ ТОЧКИ В УКАЗАНИЯ ОТ COLOR ЦВЯТ И СЪОБРАЗЕНИ С NORMAL/INVERSE СЪСТОЯНИЕТО.

В РЕЖИМ EXMODE ПРИ ИЗВЕЖДАНЕ СЕ ВЗИМА ПРЕДВИД ПРЕДИШНОТО СЪСТОЯНИЕ НА ТОЧКИТЕ. АКО ЕДНА ТОЧКА Е БИЛА ЗАПАЛЕНА PLOT, LINE И FILL Я ИЗТРИВАТ И, ОБРАТНО, АКО НЕ Е БИЛА ЗАПАЛЕНА ТЕ Я ЗАПАЛВАТ. ОТНОВО ВСИЧКО Е СЪОБРАЗЕНО С COLOR И INVERSE/NORMAL.

АКО СМЕ В РЕЖИМ EXMODE ИЗТРИВАНЕТО НА ТОЧКА, ЛИНИЯ ИЛИ ОБЛАСТ СТАВА ПРОСТО ЧРЕЗ ПОВТОРНОТО ИМ ИЗЧЕРТАВАНЕ.

ЕТО И НЯКОЛКО ПРИМЕРА:

#> 100 50 POSN 150 100 FILL

#> 50 50 POSN 200 100 LINE

НЕКА СЕГА ИЗТРИЕМ ЛИНИЯТА:

#> 50 50 POSN 200 100 UNLINE

ЛИНИЯТА Е ИЗТРИТА, НО ПРАВОЪГЪЛНИКА НЕ Е ВЕЧЕ КАКТО ПРЕДИ.

#> ERASE EXMODE

#> 100 50 POSN 150 100 FILL

#> 50 50 POSN 200 100 LINE

СЕГА ЛИНИЯТА ПРЕМИНАВАЩА ПРЕЗ ПРАВОЪГЪЛНИКА СЕ ВИЖДА.

#> 50 50 PGN 200 100 LINE

НЕКА ДА ИЗТРИЕМ И ПРАВОЪГЪЛНИКА:

#> 100 50 PGN 150 100 FILL

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ====СЛЕД=====

ORMODE - - - - -
 . ПАЛИ ТОЧКИ НЕЗАВИСИМО ОТ
 ПРЕДИШНОТО ИМ СЪСТОЯНИЕ

EXMODE - - - - -
 ПРОМЕНЯ СЪСТОЯНИЕТО НА ТОЧКИТЕ

F. GREEK

ГРЕЕК Е ДУМА, КОЯТО ПОЗВОЛЯВА ДА СЕ ПРОВЕРИ ДАЛИ ЕДНА ТОЧКА Е ВЕЧЕ ЗАПАЛЕНА ИЛИ НЕ. ГРЕЕК ВЗИМА КООРДИНАТИТЕ НА ТОЧКАТА ЗА ТЕСТВАНЕ ОТ СТЕКА И ВРЪЩА ОБРАТНО РЕЗУЛТАТА ОТ ТЕСТА. НУЛА ОЗНАЧАВА, ЧЕ ТОЧКАТА НЕ Е БИЛА ЗАПАЛЕНА, А РАЗЛИЧНО ОТ НУЛА ЧИСЛО - ТОЧКАТА Е ЗАПАЛЕНА.

#> 3 COLOR 0 0 PLOT 100 100 LINE

#> 50 50 GREEK .
2

#> 30 80 GREEK .
0

ТОЧКАТА (50,50) ПРИНАДЛЕЖИ НА НАЧЕРТАНАТА ЛИНИЯ, А (30,80) Е ИЗВЪН НЕЯ.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ====СЛЕД=====

ГРЕЕК X Y - N - - N=0 ! N#0
 #0 - ЗАПАЛЕНА ТОЧКА

11.27 ГРАФИКИ ТИП "КОСТЕНУРКА"

ТОЗИ ТИП ГРАФИКА - ЧЕРТАЕНЕ НА ЛИНИИ ЧРЕЗ ОТНОСИТЕЛНИ ПОЛЯРНИ КООРДИНАТИ СЕ ПОДДЪРЖА СЪЩО ОТ ГРАФОРТ. ПРЕДСТАВЕТЕ СИ КОСТЕНУРКА В СРЕДАТА НА ЕКРАНА, КОЯТО ИМА НА ОПАШКАТА СИ МОЛИВ. КОСТЕНУРКАТА МОЖЕ ДА СЕ ПРИДВИЖВА С МИНИМАЛНА СЪПКА ЕДНА ТОЧКА. КОГАТО СЕ ДВИЖИ, ТЯ ОСТАВЯ СЛЕДА СЛЕД СЕБЕ СИ. МОЖЕТЕ ДА И КАЖЕТЕ ДА ЗАВИЕ НАЛЯВО ИЛИ НАДЯСНО, ДА СЕ ДВИЖИ КАТО ОСТАВЯ СЛЕДА ИЛИ БЕЗ ДА ОСТАВЯ СЛЕДА.

ДУМИТЕ ЗА ТОЗИ ТИП ГРАФИКА СЕ НАМИРАТ В ЕДИН ФАЙЛ НА ДИСКЕТАТА НАРЕЧЕН "КОСТЕНУРКА".

НЕКА КОМПИЛИРАМЕ ТЕЗИ ДУМИ:

```
#> READ "КОСТЕНУРКА"
```

СЕГА МОЖЕМ ДА ЗАПОЧНЕМ ДА РИСУВАМЕ:

```
#> RUN
```

ЕКРАНЪТ Е ИЗТРИТ, ОСТАВЕНИ СА САМО ЧЕТИРИ ЛИНИИ ЗА ТЕКСТА В ДОЛНИЯ МЪ КРАИ, ЗА ЦВЯТ Е ИЗБРАН 3 (БЯЛ), А КОСТЕНУРКАТА Е В СРЕДАТА НА ЕКРАНА СЪС СПУСНАТ МОЛИВ.

A. PENUP

ТОЗА Е ДУМАТА С КОЯТО СЕ ВДИГА МОЛИВА, ТАКА ЧЕ КОСТЕНУРКАТА ДА НЕ ОСТАВЯ СЛЕДА КОГАТО СЕ ДВИЖИ.

B. PENDOWN

ОБРАТНО НА PENUP, PENDOWN СПУСКА МОЛИВА.

C. MOVE

MOVE ДВИЖИ КОСТЕНУРКАТА В ИЗБРАНАТА ПОССКА.

<РАЗСТОЯНИЕ> MOVE

MOVE ВЗИМА ОТ СТЕКА РАЗСТОЯНИЕТО В ТОЧКИ И ПРЕМЕСТВА КОСТЕНУРКАТА. В ЗАВИСИМОСТ ОТ ПОЛОЖЕНИЕТО НА МОЛИВА ЩЕ БЪДЕ НАЧЕРТАНА ИЛИ НЕ ЛИНИЯ.

```
#> 50 MOVE
```

D. TURNTO

ЗА ДА ЗАВИЕ КОСТЕНУРКАТА НА НЯКАКЪВ ЪГЪЛ СЕ ИЗПОЛЗВА TURNTO.

<ЪГЪЛ> TURNTO

TURNTO ВЗИМА ОТ СТЕКА СТОИНОСТ ЗА ЪГЪЛА И УСТАНОВЯВА НОВАТА ПОССКА НА ДВИЖЕНИЕ ЗА КОСТЕНУРКАТА. ЗАДАВАНИЯТ ЪГЪЛ Е В ГРАДУСИ, КАТО НАРАСТВАНИЕТО Е ПО ЧАСОВНИКОВАТА СТРЕЛКА. 0 ГРАДУСА Е НАГОРЕ, 90 - НАДЯСНО, 180 - НАДОЛУ, 270 - НАЛЯВО.

```
#> 90 TURNTO
```

```
#> 75 MOVE
```

E. TURN

ТАЗИ ДУМА ЗАВЪРТА КОСТЕНУРКАТА НА ВЗЕТИЯ ОТ СТЕКА ЪГЪЛ, НО СПРЯМО ТЕКУЩАТА ПОССКА. ЗАВЪРТАНИЕТО С TURNTO Е АБСОЛЮТНО, А С TURN - ОТНОСИТЕЛНО. ПОССКАТА НА ЗАВЪРТАНЕ Е ОТНОВО ПО ЧАСОВНИКОВАТА СТРЕЛКА.

<ЪГЪЛ> TURN

#> 45 TURN

#> 50 MOVE

F. MOVETO

MOVETO ПРЕМЕСТВА КОСТЕНУРКАТА НА ТОЧКА С КООРДИНАТИ ПОСОЧЕНИТЕ В СТЕКА. АКО МОЛИВЪТ Е СПУСНАТ, ЩЕ БЪДЕ ИЗЧЕРТАНА ЛИНИЯ.

<X> <Y> MOVETO

#> PENUP

#> 0 0 MOVETO

#> 127 TURNTO

#> PENDOWN

#> 160 MOVE

ПРИМЕРИ.

#> TURTLE

#> 50 MOVE 90 TURN 50 MOVE 90 TURN

#> 50 MOVE 90 TURN 50 MOVE

МОЖЕ ДА СЕ НАПИШЕ И ТАКА:

#> 4 0 DO 50 MOVE 90 TURN LOOP

ИЛИ ДА СЕ ЗАПАЗИ КАТО ДУМА:

#> : КВАДРАТ

#> 4 0 DO

#> 50 MOVE

#> 90 TURN

#> LOOP ;

СЕГА КВАДРАТЪТ МОЖЕ ДА БЪДЕ НАЧЕРТАН НАВСЯКЪДЕ ПО ЕКРАНА.

#> TURTLE

#> PENUP 0 100 MOVETO PENDOWN КВАДРАТ

#> PENUP 55 100 MOVETO 30 TURNTO PENDOWN КВАДРАТ

#> PENUP 120 100 MOVETO 60 TURNTO PENDOWN КВАДРАТ

#> PENUP 190 100 MOVETO 90 TURNTO PENDOWN КВАДРАТ

ОПИТАЙТЕ СЛЕДНОТО:

#> TURTLE

#> 36 0 DO КВАДРАТ 10 TURN LOOP

А СЕГА ТАКА:

#> EXMODE TURTLE

#> 36 0 DO КВАДРАТ 10 TURN LOOP

#> ORMODE

А ЕТО И СЪДЪРЖАНИЕТО НА ФАЙЛА "КОСТЕНУРКА". ТОВА СА ДЕФИНИЦИИТЕ НА ТОКУ ДО РАЗГЛЕДАНИТЕ ДУМИ.

(КОСТЕНУРКА)

VARIABLE TURTLE.X
VARIABLE TURTLE.Y
VARIABLE TURTLE.ANG
VARIABLE TURTLE.PEN

: TURTLE.WALK
OVER OVER -> TURTLE.Y -> TURTLE.X
128 / SWAP 128 / SWAP
TURTLE.PEN IF LINE ELSE POSN THEN ;

: PENUP 0 -> TURTLE.PEN ;

: PENDOWN 1 -> TURTLE.PEN ;

: MOVETO
128 * 64 + SWAP
128 * 64 + SWAP
TURTLE.WALK ;

: MOVE
TURTLE.ANG 16 * 45 / PUSH
DUP I SIN * TURTLE.X + SWAP
I 32 + SIN * TURTLE.Y SWAP -
POP TURTLE.WALK ;

: TURNTD -> TURTLE.ANG ;

: TURN
TURTLE.ANG +
360 + 360 MOD TURNTD ;

: TURTLE
GR ERASE
0 40 20 24 WINDOW
3 COLOR PENUP
128 96 MOVETO
0 TURNTD
PENDOWN ;

II.28 СПЕЦИАЛНИ СИМВОЛИ НА ИЗХОД

ДВА УПРАВЛЯВАЩИ СИМВОЛА (НАТИСНАТИ ЗАЕДНО С "МК") СЕ УПОТРЕБЯВАТ ЗА ИЗВЕЖДАНЕ ОТ ПРОГРАМА.

[L] (ASCII-КОД 140) Е ЕКВИВАЛЕНТНО НА КОМАНДАТА HOME - ИЗТРИВА ВСИЧКО В ТЕКСТОВИЯ ПРОЗОРЕЦ.

[K] (ASCII-КОД 139) ПРЕДИЗВИКВА ПОЗИЦИОНИРАНЕ НЕ НА СЛЕДВАЩИЯ РЕД (КАКТО СЛЕД "RETURN" ИЛИ CR), А НА ПРЕДИШНИЯ. КОГАТО КУРСОРЪТ СТИГНЕ НАЙ-ГОРНИЯ РЕД ЩЕ СТАНЕ ОБРАТНО ПРИДВИЖВАНЕ НА ТЕКСТА ПО ЕКРАНА.

РАЗБИРА СЕ, МОЖЕТЕ ДА УПОТРЕБЯВАТЕ ТЕЗИ СИМВОЛИ И В ДИРЕКТЕН РЕЖИМ.

II.29 СМЯНА НА РАЗМЕРА И ЦВЕТА НА СИМВОЛИТЕ

ГРАФОРТ МОЖЕ ДА ИЗВЕЖДА СИМВОЛИТЕ В ОСЕМ РАЗЛИЧНИ РАЗМЕРА. ЗА ИЗБОР НА РАЗМЕР НА СИМВОЛИТЕ СЕ ИЗПОЗВА ДЪМАТА CHRSIZE. CHRSIZE ВЗИМА ЧИСЛО ОТ СТЕКА ЗА ДА ОПРЕДЕЛИ РАЗМЕРА. ВАЛИДНИТЕ ЧИСЛА СА МЕЖДУ 0 И 8. РАЗМЕРИТЕ 0 И 1 СА ЕДНАКВИ, НО ГРАФОРТ ИЗПОЛЗВА ВЪЗМОЖНОСТИТЕ СИ ЗА ЦВЕТНА ГРАФИКА ПРИ РАЗМЕРИ ОТ 1 ДО 8. СИМВОЛИТЕ С РАЗМЕРИ ОТ 2 ДО 8 СА ДВА ДО ОСЕМ ПЪТИ ПО-ГОЛЕМИ ОТ СИМВОЛИТЕ С РАЗМЕР 0 ИЛИ 1.

#> ABORT

#> HOME 2 CHRSIZE

#> HOME 1 COLOR

ЕТО И ЕДНА ТАБЛИЦА ДАВАЩА СЪОТВЕСТВИЕ МЕЖДУ ИЗБРАНИЯ РАЗМЕР И ВЪЗМОЖНОСТИТЕ ЗА ИЗВЕЖДАНЕ НА ТЕКСТ:

РАЗМЕР	КОЛОНИ	РЕДОВЕ	ЦВЯТ
0	40	24	НЕ
1	32	24	ДА
2	16	12	ДА
3	10	8	ДА
4	8	6	ДА
5	6	4	ДА
6	5	4	ДА
7	4	3	ДА
8	4	3	ДА

#> ABORT

#> HOME 8 CHRSIZE 2 COLOR

#> INVERSE

#> ABORT

ОПЕРАТОР S T A C K ОБЪЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

CHRSIZE N - - - - CHARACTER SIZE
ИЗБИРА РАЗМЕР НА СИМВОЛИТЕ ЗА
PRINT, WRITELN, PUTC, BLKSIZE

II.30 ИЗБОР НА ШРИФТ

СТАНДАРТНО НАБОРЪТ ГРАФИЧНИ СИМВОЛИ ЗА ПРАВЕЦ СЕ НАМИРА В ROM-ПАМЕТА. ГРАФОРТ ИЗПОЛЗВА ГРАФИЧНОТО ИЗВЕЖДАНЕ И НЕГОВИЯ СИМВОЛЕН НАБОР СЕ НАМИРА В RAM-ПАМЕТА НА АДРЕС 2048. СИМВОЛНИЯТ НАБОР ЗАЕМА 768 БАЙТА В ПАМЕТА ЗА 96 ГРАФИЧНИ СИМВОЛА ПО 8 БАЙТА ВСЕКИ. ВЪРХУ ДИСКЕТАТА ВСЕКИ ЕДИН ОТ НАБОРИТЕ (FONT) ЗАЕМА 5 СЕКТОРА. ВСИЧКИ ТЕ ЗАПОЧВАТ СЪС СТАНДАРТНИЯ ПРЕФИКС "CHR.":

ГОЛЕМИ И МАЛКИ ЛАТИНИЦА	ГОЛЕМИ И МАЛКИ КИРИЛИЦА	ГОЛЕМИ КИР + ЛАТ
CHR.SYS	CHR.CYR	CHR.CYS
CHR.STOP	CHR.CSTOP	CHR.SSTOP
CHR.SLANT	CHR.CSLANT	CHR.SSLANT
CHR.BYTE	CHR.CBYTE	CHR.SBYTE
CHR.GOTHIC	---	---
---	---	CHR.SPS

ТОВА СА СИМВОЛНИТЕ ШРИФТОВЕ, КОИТО СЕ НАМИРАТ И НА ДВЕТЕ ДИСКЕТИ. СТАНДАРТНИЯТ НАБОР (ТОЗИ, ЗАРЕЖДАН АВТОМАТИЧНО ОТ ГРАФОРТ) Е CHR.CYS. НА ВСЕКИ ЕДИН ОТ ГОРНИТЕ РЕДОВЕ СТОЯТ ЕДНОСТИЛНИ НАБОРИ СИМВОЛИ, НО В ПЪРВАТА КОЛОНА СА НАБОРИ С ГОЛЕМИ И МАЛКИ БУКВИ ЛАТИНИЦА, ВЪВ ВТОРАТА - НАБОРИ С ГОЛЕМИ И МАЛКИ БУКВИ КИРИЛИЦА, А В ТРЕТАТА - САМО ГОЛЕМИ БУКВИ И КИРИЛИЦА И ЛАТИНИЦА (КАКЪВТО Е СТАНДАРТНИЯТ НАБОР).

НА ДЕМОСТРАЦИОННАТА ДИСКЕТА ИМА ОЩЕ ДВА СПЕЦИАЛНИ СИМВОЛНИ НАБОРА:

CHR.MAXWELL
CHR.STUFF

ТОВА НЕ СА ТЕКСТОВИ СИМВОЛИ, А СИМВОЛИ ИЗПОЛЗВАНИ ЗА "СИМВОЛНА ГРАФИКА".

ГОРНИТЕ СИМВОЛНИ НАБОРИ МОГАТ ДА БЪДАТ ЗАРЕДЕНИ КЪДЕ ДА Е В ПАМЕТА ЧРЕЗ DOS-КОМАНДАТА BLOAD. НА АДРЕС 2816 ЗАПОЧВА ЕДНА СВОБОДНА ОБЛАСТ В ПАМЕТА - ДА ЗАРЕДИМ ТАМ ЕДИН ОТ CHR-ФАЙЛОВОТЕ:

#> CR 132 PUTC PRINT " BLOAD CHR.SSLANT,A2816 " CR

ЗА ДА СЕ ИЗПОЛЗВА ТОЗИ СИМВОЛЕН НАБОР, ГРАФОРТ ТРЯБВА ДА ЗНАЕ АДРЕСА МУ. ТОВА СТАВА ЧРЕЗ КОМАНДАТА CHRADR. CHRADR ВЗИМА АДРЕС ОТ СТЕКА И ПРЕДПОЛАГА ЧЕ ОТ ТОЗИ АДРЕС В ПАМЕТА СТОИ 768-БАЙТОВ СИМВОЛЕН НАБОР. ВСИЧКИ ИЗВЕЖДАНИ СИМВОЛИ ЩЕ БЪДАТ ВЗИМАНИ ОТТАМ.

#> 2816 CHRADR

ASCII-КОДОВЕТЕ НА ИЗВЕЖДАНИТЕ ВЪРХУ ЕКРАНА СИМВОЛИ СА ОТ 160 (A0) ДО 255 (FF). НЕКА ГИ РАЗПЕЧАТАМЕ:

```
#> 256, 160 DO I PUTC LOOP
```

ДРУГА ДУМА (CHRSET) ВРЪЩА В СТЕКА АДРЕСА НА СТАНДАРТНИЯ НАБОР - 2048. ДА СЕ ВЪРНЕМ ОБРАТНО КЪМ ТОЗИ НАБОР:

```
#> CHRSET CHRADR
```

МОЖЕТЕ ДА ИЗПОЛЗВАТЕ КАТО СТАНДАРТЕН НАБОР КОИ ДА Е ДРУГ ШРИФТ: ЗАРЕДЕТЕ СЪОТВЕТНИЯ ФАЙЛ НА АДРЕС 2048.

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====		

CHRADR	A - - - - -	CHAR ADDRESS ИЗБИРА СИМВОЛЕН НАБОР (АДРЕС А)
--------	-------------	---

CHRSET	ПРОМЕНЛИВА АДРЕС НА СТАНД.СИМВОЛЕН НАБОР
--------	---

II.31 СИМВОЛЕН РЕДАКТОР (С.РЕДАКТОР)

НА ДИСКЕТИТЕ СЕ НАМИРА ФАЙЛ С ИМЕ "С.РЕДАКТОР". ТАЗИ ПРОГРАМА ПОЗВОЛЯВА ЧЕТЕНЕ, МОДИФИЦИРАНЕ, СЪЗДАВАНЕ И ЗАПАЗВАНЕ НА ФИГУРИ И КАРТИНИ ДЕФИНИРАНИ КАТО БЛОКОВЕ ОТ СИМВОЛИ.

СИМВОЛНИЯТ РЕДАКТОР Е ЕДНА ОТ ГОЛЕМИТЕ ПРОГРАМИ, ТАКА ЧЕ ПРЕДИ ДА ГО КОМПИЛИРАТЕ ИЗЧИСТЕТЕ (С FORGET) НЕИЖНИТЕ ДУМИ ОТ БИБЛИОТЕКАТА.

```
#> READ " С.РЕДАКТОР "
```

```
#> HOME RUN
```

РЕДАКТОРЪТ НЕ ИЗЧИСТВА ЕКРАНА, ЗА ДА ПОЗВОЛИ КАРТИНИ СЪЗДАДЕНИ ЧРЕЗ ДРУГИТЕ СРЕДСТВА НА ЕЗИКА ДА МОГАТ ДА СЕ ОБРАБОТВАТ ОТ НЕГО.

И=ГОРЕ
И=ЛЯВО
К=ДЯСНО
М=ДОЛУ
Ф=ПЪЛНО
О=ПРАЗНО
Л=ЛИНИЯ
Ж=Х-ЛИНИЯ
Ц=ЦВЯТ
А=АДРЕС
Т=ТРАНСФЕР
Б=РАЗМ. БЛОК
Е=ИСТРИВАНЕ
Ч=ЧЕТЕНЕ
П=ПИСАНЕ
З=ЗАПАЗВАНЕ
В=ВЗЕМИ
С=СИМВОЛИ
Х=СТЪПКА
Я=ИЗХОД

ВЪВЕДИ КОМАНДА :

В ГОРНИЯ ЛЯВ ЪГЪЛ ЩЕ ВИДИТЕ МИГАЩА ТОЧКА - ТОВА Е КУРСОРЪТ ЗА ЧЕРТАЕНЕ.

А. ИЗБОР И ИЗВЕЖДАНЕ НА НАБОР СИМВОЛИ

НЕКА РАЗГЛЕДАМЕ СТАНДАРТНИЯ НАБОР СИМВОЛИ. НАТИСНЕТЕ "А" ЗА АДРЕС.

ВЪВЕДИ КОМАНДА : А

СИМВОЛЕН НАБОР
АДРЕС РАБОТНА ОБЛАСТ : 2816

ВИЖДАТЕ ИЗВЕДЕН АДРЕСА ПО ПРЕМЪЛЧАВАНЕ. НАПИШЕТЕ НОВИЯ АДРЕС - 2048. НАТИСНЕТЕ СЕГА "С" (ОТ СИМВОЛИ) ЗА ДА БИДИТЕ СИМВОЛНИЯ НАБОР.

ВЪВЕДИ КОМАНДА : С

ОБЕЗНАЧЕНИЕТО КЪМ КОНКРЕТЕН СИМВОЛ СТАВА ПО НЕГОВИЯ НОМЕР ОТ 0 ДО 95 (ВМЕСТО ДЕЙСТВИТЕЛНИТЕ ASCII-КОДОВЕ ОТ 160 ДО 255). НА ПЪРВИЯ РЕД СА СИМВОЛИ ОТ 0 ДО 31, НА ВТОРИЯ - ОТ 32 ДО 63 И НА ТРЕТИЯ - ОТ 64 ДО 95.

В. ИЗВЕЖДАНЕ НА СИМВОЛЕН БЛОК

МОЖЕМ ДА ИЗВЕЖДАМЕ В ПРАВОЪГЪЛЕН БЛОК ПОСЛЕДОВАТЕЛНОСТ ОТ СИМВОЛИ. НЕКА ПЪРВО ДЕФИНИРАМЕ БЛОКА:

ВЪВЕДИ КОМАНДА : Б

ХОРИЗОНТАЛЕН РАЗМЕР НА БЛОКА : 3
ВЕРТИКАЛЕН РАЗМЕР НА БЛОКА : 2

НЕКА ИЗВЕДЕМ ОТНОВО СИМВОЛНИЯ НАБОР НА ЕКРАНА:

ВЪВЕДИ КОМАНДА : С

А СЕГА ДА ЗАПЪЛНИМ БЛОКА С ПОСЛЕДОВАТЕЛНОСТ СИМВОЛИ ЗАПОЧВАЩИ ОТ "Б":

ВЪВЕДИ КОМАНДА : Ч

ВЪВЕДИ НОМЕР НА СИМВОЛА
ЗА ЧЕТЕНЕ : 66

БЛОКЪТ ЩЕ БЪДЕ ЗАПЪЛНЕН ТАКА:

ЕЦД
ЕФГ

ТОВА СА СИМВОЛИТЕ С НОМЕРА ОТ 66 ДО 71. БЯХА ВЗЕТИ ТОЛКОВА СИМВОЛА, КОЛКОТО СА НЕОБХОДИМИ ЗА ДА СЕ ЗАПЪЛНИ БЛОКА.

ЗА ДА БЪДЕ ИЗТРИТ БЛОКА СЕ ИЗПОЛЗВА КОМАНДАТА "Е":

ВЪВЕДИ КОМАНДА : Е

ИЗТРИВАНЕ (Д/Н) : Д

ДА ИЗБЕРЕМ ОТНОВО "РАБОТНИЯ" АДРЕС ЗА СИМВОЛНИЯ НАБОР:

ВЪВЕДИ КОМАНДА : А

СИМВОЛЕН НАБОР
АДРЕС РАБОТНА ОБЛАСТ : 2816

СЛОЖЕТЕ ДЕМОНСТРАЦИОННАТА ДИСКЕТА. НЕКА СЕГА ВЗЕМЕМ НОВ СИМВОЛЕН НАБОР:

ВЪВЕДИ КОМАНДА : В

ИМЕ НА ФАЙЛА ЗА ВЪВЕЖДАНЕ : CHR.STUFF

ТОЗИ НАБОР СИМВОЛИ ЩЕ БЪДЕ ЗАРЕДЕН НА АДРЕС 2816 В ПАМЕТТА. НЕКА ДА ВИДИМ СИМВОЛИТЕ:

ВЪВЕДИ КОМАНДА : С

ПОСТАВЕТЕ ОТНОВО (АКО СТЕ ГО ПРОМЕНИЛИ) РАЗМЕРА НА БЛОКА НА 3 X 2 И ПРОЧЕТЕТЕ (Ч) СИМВОЛ 78, А СЛЕД ТОВА 84 И 90.

ИЗЧИСТЕТЕ БЛОКА (Е) И ЗАДАЙТЕ НОВИ РАЗМЕРИ (Б) - 5 X 3. ПРОЧЕТЕТЕ (Ч) СИМВОЛИ 33, 48 И 63.

С. ДЕФИНИРАНЕ НА ФИГУРИ

МОЖЕТЕ ДА СЪДАВАТЕ ВАШИ СОБСТВЕНИ ФИГУРИ ИЛИ ДА ПРОМЕНИТЕ НЯКОИ ОТ ТЕЗИ КОИТО СА ВИ ПРЕДЛОЖЕНИ С ДЕМОНСТРАЦИОННАТА ДИСКЕТА. КЛАВИШИТЕ "И", "И", "К" И "М" КАКТО ОБИКНОВЕНО ПРИДВИЖВАТ КУРСОРА В ЧЕТИРИТЕ ПОСОКИ. С "♦" СЕ ИЗВЕЖДАТ ТОЧКИ (СЛОМНЕТЕ СИ PLOT), А С "О" СЕ ИЗТРИВАТ (UNPLOT). АНАЛОГИЧНО С "Λ" СЕ ЧЕРТАЕ ЛИНИЯ (LINE), А С "Ж" ТЯ СЕ ИЗТРИВА (UNLINE).

КОМАНДАТА "X" (СТЪПКА) СЕ ИЗПОЛЗВА ЗА ДЕФИНИРАНЕ НА НОВО ОТМЕСТВАНЕ В ЕДНА ОТ ЧЕТИРИТЕ ПОСОКИ ("И", "И", "К" И "М").

ПО ВРЕМЕ НА ЧЕРТАНЕТО МОЖЕТЕ ДА ПРОМЕНЯТЕ ЦВЕТА С КОИТО РАБОТИТЕ - ТОВА Е КОМАНДАТА "Ц".

ЗА ДА ПРЕНЕСЕТЕ КАРТИНАТА ОТ БЛОКА В БУФЕРА НА СИМВОЛНИЯ НАБОР (768-ТЕ БАЙТА ЗАПОЧВАЩИ ОБИКНОВЕНО ОТ АДРЕС 2816) ИЗПОЛЗВАЙТЕ "П".

КОМАНДАТА "Т" (ТРАНСФЕР) ВИ ПОМАГА ДА ПРЕМЕСТВАТЕ СИМВОЛИ ОТ ЕДИН НАБОР В ДРУГ.

ВЪВЕДИ КОМАНДА : Т

ВЪВЕДИ "ОТ" АДРЕС :

ВЪВЕДИ "НА" АДРЕС :

ВЪВЕДИ ДЪЛЖИНА (НОРМАЛНО 768) :

ТОЗА СА ТРИТЕ ИЗИСКВАНИ СТОИНОСТИ НА КОМАНДАТА "Т". С НЕИНА ПОМОЩ МОЖЕТЕ ДА ПРЕМЕСТВАТЕ И ИНДИВИДУАЛНИ СИМВОЛИ ИЛИ ГРУПИ СИМВОЛИ. ПОМНЕТЕ ЧЕ ВСЕКИ СИМВОЛ ЗАЕМА 3 БАЙТА, КОЕТО ЗНАЧИ ЧЕ ВТОРИЯ СИМВОЛ Е НА ОТМЕСТВАНЕ 3, ТРЕТИЯ НА ОТМЕСТВАНЕ 16 И Т.Н. ДЪЛЖИНАТА Е РАВНА НА БРОЯ НА СИМВОЛИТЕ УМНОЖЕН ПО 3.

Д. ЗАПАЗВАНЕ НА СИМВОЛЕН НАБОР

ЗА ДА ЗАПАЗИТЕ СИМВОЛЕН НАБОР ВЪРХУ ДИСКЕТАТА ИЗДАЙТЕ КОМАНДАТА "З".

ВЪВЕДИ КОМАНДА : З

ИМЕ НА ФАЙЛА ЗА ЗАПАЗВАНЕ : *****

ЗА ИЗЛИЗАНЕ ОТ СИМВОЛНИЯ РЕДАКТОР СЕ ИЗПОЛЗВА КОМАНДАТА "Я".

ПОМНЕТЕ ЧЕ СИМВОЛНИЯТ РЕДАКТОР Е ГОЛЯМА ПРОГРАМА И АКО НЕГОВИТЕ ДУМИ ОСТАНАТ В БИБЛИОТЕКАТА, ТО НЯМА ДА ИМА ДОСТАТЪЧНО МЯСТО ЗА ДРУГИ ПРОГРАМИ. ЗАТОВА СЛЕД ПРИКЛЮЧВАНЕ НА РАБОТА С РЕДАКТОРА ИЗДАВАЙТЕ ВИНАГИ КОМАНДАТА "FORGET":

#> FORGET X

II.32 БЛОКОВО ИЗВЕЖДАНЕ ОТ ГРАФОРТ

ИЗВЕЖДАНЕТО НА БЛОКОВЕ ОТ ПРОГРАМА НА ГРАФОРТ Е ПО СЪЩИЯ НАЧИН КАКТО И ПРИ СИМВОЛНИЯ РЕДАКТОР. ЗАРЕЖДА СЕ В ПАМЕТА СИМВОЛЕН НАБОР, ИЗБИРА СЕ РАЗМЕР НА БЛОК И ПОСЛЕДОВАТЕЛНОСТ ОТ СИМВОЛИ СЕ ИЗВЕЖДА НА ЕКРАНА ОТ ТЕКУЩАТА ПОЗИЦИЯ НА КУРСОРА.

НЕКА ДА ЗАРЕДИМ В ПАМЕТА CHR.STUFF:

```
#> CR 132 PUTC PRINT " BLOAD CHR.STUFF,A2816 " CR
```

МОЖЕТЕ ДА НАПИШЕТЕ "2816 CHRADR", НО НЯМА ДА ВИЖДАТЕ НА ЕКРАНА РАЗБИРАЕМИ СИМВОЛИ. ПРИ ИЗВЕЖДАНЕТО НА "СИМВОЛНА" ГРАФИКА СЕ ИЗБИРА СПЕЦИАЛИЗИРАНИЯ НАБОР, ИЗВЕЖДА СЕ БЛОКА И ОТНОВО СЕ ВЪЗСТАНОВЯВА НОРМАЛНИЯ НАБОР СИМВОЛИ.

А. РАЗМЕР НА БЛОК (BLKSIZE)

ДУМАТА BLKSIZE ВЗИМА ОТ СТЕКА ДВЕ ЧИСЛА: ХОРИЗОНТАЛЕН И ВЕРТИКАЛЕН РАЗМЕРИ НА БЛОКА.

```
#> 3 2 BLKSIZE
```

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

BLKSIZE	M N - - - -	BLOCK SIZE
		ИЗБИРА РАЗМЕР НА СИМВОЛНИЯ
		БЛОК: M-ДЪЛЖИНА, N-ВИСОЧИНА

В. ИЗВЕЖДАНЕ НА БЛОК (PUTBLK)

PUTBLK ИЗВЕЖДА БЛОК ВЪРХУ ЕКРАНА. ЗА ЦЕЛТА ТОИ ВЗИМА ОТ СТЕКА НОМЕРА НА НАЧАЛНИЯ СИМВОЛ ОТ ПОРЕДИЦАТА ЗА ИЗВЕЖДАНЕ. КАКТО И ПРИ СИМВОЛНИЯ РЕДАКТОР, НОМЕРАТА НА СИМВОЛИТЕ СА ОТ 0 ДО 95. АДРЕСЪТ НА СИМВОЛНИЯ НАБОР Е ЗАДАДЕН ЧРЕЗ CHRADR. ПОЗИЦИОНИРАНЕТО ПО ЕКРАНА СТАВА КАКТО И ПРИ ИЗВЕЖДАНЕТО НА ОБИКНОВЕН ТЕКСТ - ЧРЕЗ HTAB И VTAB. ЗА РАЗЛИКА ОТ PRINT, PUTBLK НЕ ПРОМЕНЯ ПОЗИЦИЯТА НА КУРСОРА СЛЕД ИЗВЕЖДАНЕ ПО ЕКРАНА.

```
#> HOME 2816 CHRADR 78 PUTBLK CHRSET CHRADR 12 VTAB
```

ДА ПОДГОТВИМ НЯКОЛКО ДУМИ, КОИТО ДА НИ СПЕСТЯТ МНОГОКРАТНО ПИСАНЕ:

```
#> : A 2816 CHRADR HOME ;
```

```
#> : B CHRSET CHRADR 12 VTAB ;
```

```
#> A 84 PUTBLK B
```

ДА НАПРАВИМ ОЩЕ ЕДИН ТЕСТ:

```
#> 5 3 BLKSIZE
```

```
#> A 100 0 DO 33 PUTBLK 48 PUTBLK 63 PUTBLK LOOP B
```

СТАВА ДОСТА БЪРЗО, НАЛИ ?

#> 10 3 BLKSIZE

#> A 5 VTAB 13 HTAB 2 PUTBLK Б

ЗА ДА СЕ ИЗТРИЕ ЕДИН БЛОК СЕ ИЗПОЛЗВА КОМАНДАТА UNBLK. ТЯ ИЗТРИВА БЛОК С РАЗМЕРИ ЗАДАДЕНИ В BLKSIZE ОТ ТЕКУЩАТА ПОЗИЦИЯ НА КУРСОРА:

#> 5 VTAB 13 HTAB UNBLK

РАЗБИРА СЕ, CHRSIZE МОЖЕ ДА БЪДЕ ИЗПОЛЗВАНА ЗА ДА СЕ УВЕЛИЧАТ РАЗМЕРИТЕ НА ФИГУРАТА:

#> A 3 CHRSIZE 3 COLOR 2 PUTBLK 0 CHRSIZE Б

ОПЕРАТОР	С Т А С К	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

PUTBLK	N - - - - -	PUT BLOCK ИЗВЕЖДА СИМВОЛЕН БЛОК
--------	-------------	------------------------------------

UNBLK	- - - - -	ИЗТРИВА СИМВОЛЕН БЛОК
-------	-----------	-----------------------

С. РЕЖИМ "ИЗКЛЮЧАЩО ИЛИ" (EXMODE)

РЕЖИМЪТ EXMODE МОЖЕ ДА БЪДЕ ИЗПОЛЗВАН ПРИ ИЗВЕЖДАНЕТО НА СИМВОЛНИ ГРАФИКИ. ПОВТОРНОТО ОТПЕЧАТВАНЕ НА ДАДЕН БЛОК ПРОСТО ЩЕ ИЗТРИЕ БЛОКА. И ВСЕ ПАК ИМА НЯКОЛКО СЪОБРАЖЕНИЯ, КОИТО ТРЯБВА ДА СЕ ИМАТ ПРЕДВИД.

С ЦЕЛ ПЕСТЕНЕ НА ВРЕМЕ, ЕДИН СИМВОЛ НЕ СЕ ИЗВЕЖДА НА ЕКРАНА, АКО ТОИ Е ВЕЧЕ ТАМ. ЗА ДА СЕ ПРОВЕРИ ДАЛИ ТОИ Е ТАМ ИЛИ НЕ, СЕ ИЗПОЛЗВА ОБЛАСТТА НА ОБИКНОВЕНИЯ ТЕКСТОВ ЕКРАН, МАКАР ЧЕ НА ЕКРАНА СЕ ИЗВЕЖДА ОБЛАСТТА НА ГРАФИЧНИЯ. ЗАТОВА В РЕЖИМ EXMODE ТРЯБВА ДА СЕ ЧИСТИ ТЕКСТОВИЯ ЕКРАН. ТОВА СТАВА ЧРЕЗ ПОДПРОГРАМАТА НА МОНИТОРА НА АДРЕС -936 (" -936 CALL").

```

#> : DEMO.EXMODE

#> ERASE

#> 1 CHRSize

#> EXMODE

#> 0 0 PLOT 100 100 LINE

#> 4 0 DO

#>       3000 0 DO LOOP

#>       5 VTAB

#>       5 0 DO

#>           PRINT " ТОЗИ РЕД МОЖЕ ДА СЕ ИЗТРИЕ " CR

#>       LOOP

#>       -936 CALL

#> LOOP

#> 0 CHRSize ;

```

НЕКА ДА ИЗПЪЛНИМ ТАЗИ ПРОГРАМКА:

```
#> DEMO.EXMODE
```

II.33 ОБЕМНИ ГРАФИКИ

ЗА ДА МОЖЕ ГРАФОРТ ДА ИЗВЕДЕ НА ЕКРАНА ЕДИН ТРИМЕРЕН ОБЕКТ, ПЪРВО ТРЯБВА ДА СЕ ЗАРЕДИ В ПАМЕТТА ОПИСАНИЕТО МУ, А СЛЕД ТОВА ДА СЕ ВЪВЕДАТ КОМАНДИ КОИТО ДАВАТ ИНФОРМАЦИЯ КЪДЕ В ПАМЕТТА СЕ НАМИРА ТОВА ОПИСАНИЕ И КАК ТОЗИ ОБЕКТ ТРЯБВА ДА БЪДЕ ПОКАЗАН НА ЕКРАНА.

ЗА ВСИЧКИ ОБЕМНИ ОБЕКТИ, КОИТО ТЕКУЩО СА ДЕФИНИРАНИ ГРАФОРТ ИЗПОЛЗВА ЕДИН ВЪТРЕШЕН МАСИВ. ТАМ СА ЗАПОМНЕНИ АДРЕСИТЕ НА ОПИСАНИЯТА ЗА ВСЕКИ ЕДИН ОТ ОБЕКТИТЕ И ПАРАМЕТРИ ЗА ИЗВЕЖДАНЕ (ПОЗИЦИЯ, РОТАЦИЯ, СКАЛА И Т.Н.). МОГАТ ДА БЪДАТ УПРАВЛЯВАНИ ДО 16 ТАКИВА ОБЕКТА НАВЕДНАЖ. ТЕ СЕ НОМЕРИРАТ ОТ 0 ДО 15. ПО ТОЗИ НОМЕР ГРАФОРТ ЗНАЕ ЗА КОИ ОБЕКТ СЕ ОТНАСЯТ СЛЕДВАЩИТЕ КОМАНДИ.

ДА ИНИЦИАЛИЗИРАМЕ ПЪРВО СИСТЕМАТА:

```
#> ABORT
```

ДА ДЕФИНИРАМЕ ТЕКСТОВИЯ ПРОЗОРЕЦ В ДОЛНАТА ЧАСТ НА ЕКРАНА:

```
#> 0 40 20 24 WINDOW
```

ДА ЗАРЕДИМ СЕГА ОБРАЗА НА ЕДИН ТРИМЕРЕН ОБЕКТ - "XYZ" (ТРИТЕ ОСИ НА ТРИМЕРНАТА КООРДИНАТНА СИСТЕМА). ТОЗИ ФАЙЛ СЪДЕСТВУВА САМО НА ДЕМОНСТРАЦИОННАТА ДИСКЕТА.

```
#> CR 132 PUTC PRINT " BLOAD XYZ,A2816 " CR
```

НЕКА ДА ИНИЦИАЛИЗИРАМЕ И ВЪТРЕШНИЯ МАСИВ ЗА ГРАФИКИ С ТРИМЕРНИ ОБЕКТИ:

```
#> OBJERASE
```

СЕГА ТРЯБВА ДА ПРИСВОИМ НОМЕР (ОТ 0 ДО 15) НА ОБЕКТА, КОИТО ЩЕ ИЗВЕЖДАМЕ:

```
#> 0 OBJEST
```

ДУМАТА AUTODRAW ВЗИМА ЧИСЛО ОТ СТЕКА И АКО ТО Е 1, СЛЕД ВСЯКА ОБЕМНА ГРАФИЧНА КОМАНДА ОБЕКТА ЩЕ БЪДЕ ИЗВЕЖДАН НА ЕКРАНА. АКО ЧИСЛОТО Е НУЛА, РЕЖИМА AUTODRAW ЩЕ БЪДЕ ИЗКЛЮЧЕН.

```
#> 1 AUTODRAW
```

ОСТАВА ДА КАЖЕМ КЪДЕ В ПАМЕТТА СЕ НАМИРА ДЕФИНИЦИЯТА НА ОБЕКТА. ТОВА СТАВА ЧРЕЗ ДУМАТА OBJADR, КОЯТО ВЗИМА АДРЕСА НА ОБЕКТА ОТ СТЕКА:

```
#> 2816 OBJADR
```

ОБЕКТЪТ СЕ ПОЯВИ НА ЕКРАНА. ИМА 12 РАЗЛИЧНИ ДУМИ, КОИТО УПРАВЛЯВАТ ПОЗИЦИЯТА, РАЗМЕРА И ПОСОКАТА НА ТРИМЕРНИЯ ОБЕКТ.

НЕКА СЕГА ДА ЗАВЪРТИМ МАЛКО ОБЕКТА, ЗА ДА ГО ВИДИМ ПО-ДОБРЕ:

```
#> 14 YRST
```

```
#> 16 XROT
```

```
#> 6 SCALZ
```

```
ОПЕРАТОР      З Т А С К      ОБЯСНЕНИЯ  
=====ПРЕДИ=====СЛЕД=====
```

```
OBJERASE      - - -      - - -      В НАЧАЛОТО  
НА ТРИМЕРНИ ГРАФИЧНИ ПРОГРАМИ
```

```
OBJEST        N - -      - - -  
ИЗБИРА ТРИМЕРЕН ОБЕКТ
```

```
AUTODRAW      N - -      - - -      N#0: DRAW  
СЛЕД ВСЯКА ГРАФИЧНА КОМАНДА
```

```
OBJADR        A - -      - - -      УКАЗВА АДРЕС  
НА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ
```


А. ФОРМАТ НА ОБЕМНА ФИГУРА

ЗА ДА СЕ ДЕФИНИРА ЕДИН ТРИМЕРЕН ОБЕКТ СЕ ИЗПОЛЗВА ТРИМЕРНА КООРДИНАТНА СИСТЕМА С ОСИ ИМЕНОВАНИ "X", "Y" И "Z". СТРЕЛКИТЕ НА ТОКУ ДО ВИДЯНАТА ФИГУРА ПРЕДСТАВЯТ ПОСОКИТЕ НА ТРИТЕ ОСИ: X Е ОТЛЯВО НАДЯСНО; Y - ОТГОРЕ НАДОЛУ; Z - ОТЗАД НАПРЕД. ПРЕСЕЧНАТА ИМ ТОЧКА ИМА КООРДИНАТИ (0,0,0). В ТЕЗИ ПОСОКИ ПО ВСЯКА ЕДНА ОТ ОСИТЕ ИМА 256 ЕДИНИЦИ (ОТ -128 ДО 127).

ВСЕКИ ТРИМЕРЕН ОБЕКТ Е ДЕФИНИРАН И СЕ НАМИРА В СОБСТВЕНОТО СИ ТРИМЕРНО ПРОСТРАНСТВО. ПРИ ИЗВЕЖДАНЕ НА ПОВЕЧЕ ОБЕКТИ ПО ЕКРАНА, ТЕЗИ ПРОСТРАНСТВА МОГАТ ДА СЕ ПРЕПОКРИВАТ И ДОРИ ДА СЪВПАДАТ.

II.34 ПАРАМЕТРИ НА ФИГУРАТА

ЕДИН ОБЕКТ ВЕДНАЖ ЗАРЕДЕН В ПАМЕТТА И ИЗБРАН ЧРЕЗ ОБЈЕСТ И ОБЈАДР, МОЖЕ ДА БЪДЕ ВЪРТЯН, ПОЗИЦИОНИРАН, НАМАЛЯВАН ИЛИ УВЕЛИЧАВАН И ПРЕМЕСТВАН.

А. ВЪРТЕНЕ

ОБЕКТЪТ МОЖЕ ДА БЪДЕ ВЪРТЯН ОКОЛО КОЯ ДА Е ОТ ТРИТЕ ОСИ. ДУМИТЕ XROT, YROT И ZROT ИЗВЛИЧАТ ЧИСЛО ОТ СТЕКА И ВЪРТЯТ ОБЕКТА НА ИЗБРАНИЯ ЪГЪЛ. ЪГЛИТЕ СЕ ЗАДАВАТ В ЕДИНИЦИ МЕЖДУ 0 И 256, А НЕ МЕЖДУ 0 И 360. ТАКА 64 Е ЪГЪЛ ОТ 90 ГРАДУСА, 128 - 180, 192 - 270 И Т.Н. ДО 256 - 360 ГРАДУСА.

```
#> : Y.ВЪРТЕНЕ
```

```
#> 260 0 00
```

```
#> 1 YROT
```

```
#> ,4 +LOOP ;
```

ДА ИЗПЪЛНИМ ТАЗИ ДУМА:

```
#> Y.ВЪРТЕНЕ
```

ДА ВЪРНЕМ ПРЕДИШНИЯ ИЗГЛЕД:

```
#> 14 YROT
```

```
ОПЕРАТОР      S T A C K      ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====
```

```
XROT          N - - - - -      ИЗБИРА N
                ЗА ВЪРТЕНЕ ПО ОСТА X
```

```
YROT          N - - - - -      ИЗБИРА N
                ЗА ВЪРТЕНЕ ПО ОСТА Y
```

```
ZROT          N - - - - -      ИЗБИРА N
                ЗА ВЪРТЕНЕ ПО ОСТА Z
```

В. СКАЛА

РАЗМЕРЪТ НА ЕДИН ОБЕКТ МОЖЕ ДА БЪДЕ ПРОМЕНЯН В ШИРИНА ИЛИ ВИСОЧИНА ЧРЕЗ ДУМИТЕ SCALX И SCALY. И ДВЕТЕ ДУМИ ВЗИМАТ ЧИСЛО ОТ СТЕКА, ЗА ДА ОПРЕДЕЛЯТ СОТВЕТНАТА СКАЛА. ЧИСЛАТА ТРЯБВА ДА БЪДАТ МЕЖДУ -31 И +31. ВСЯКО ДРУГО ЗАДАДЕНО ЧИСЛО ЩЕ БЪДЕ "ВМЪКНАТО" В ТОЗИ ИНТЕРВАЛ. КОГАТО OBJERASE ИНИЦИАЛИЗИРА ВЪТРЕШНИЯ ТРИМЕРЕН МАСИВ, SCALX И SCALY СА И ДВЕТЕ ПО 16.

#> 25 SCALX

#> 3 SCALY

#> 4 SCALX

СКАЛА НУЛА ОЗНАЧАВА НУЛЕВ РАЗМЕР.

#> 0 SCALX

ОТРИЦАТЕЛНА СКАЛА ОЗНАЧАВА ОБРЪЩАНЕ НА ОБРАЗА:

#> -8 SCALX

ЕТО И ОЩЕ ЕДИН ПРИМЕР:

#> : ТЕСТ 12 -12 DO I SCALX LOOP ;

#> ТЕСТ

МНОГО ЧЕСТО СЕ НАЛАГА РАЗМЕРА НА ЦЕЛИЯ ОБЕКТ ДА БЪДЕ ПРОМЕНЯН. ДУМАТА SCALE ИЗВЛИЧА ЧИСЛО ОТ СТЕКА И ПРОМЕНЯ И "X"-СКАЛАТА И "Y"-СКАЛАТА.

#> 5 SCALE

#> 12 SCALE

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====		

SCALX	N - - - - -	X-СКАЛА (ШИРОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ
-------	-------------	---

SCALY	N - - - - -	Y-СКАЛА (ВИСОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ
-------	-------------	---

SCALE	N - - - - -	ИЗБИРА X И Y СКАЛИ ЗА ИЗБРАНИЯ ОБЕКТ
-------	-------------	---

С. ОБЕМНА ПЕРСПЕКТИВА

ЧЕТВЪРТАТА ДУМА SCALZ НЕ ПРОМЕНЯ РАЗМЕРА НА ОБЕКТИТЕ, А ВМЕСТО ТОВА ПРОМЕНЯ ПЕРСПЕКТИВАТА. ПО ПОДРАЗБИРАНЕ SCALZ Е НУЛА - НИКАКВА ПЕРСПЕКТИВА. SCALZ ВЗИМА ОТНОВО ОТ СТЕКА ЧИСЛО МЕЖДУ -31 И +31. ПРИ СТОИНОСТ РАЗЛИЧНА ОТ НУЛА СЕ ПОДДЪРЖА ПОГЛЕД В ПЕРСПЕКТИВА. АКО ЧИСЛОТО Е ПОЛОЖИТЕЛНО ПРЕДНАТА ЧАСТ НА ОБЕКТА ЩЕ БЪДЕ ПО-ГОЛЯМА.

#> 20 SCALZ Y.ВЪРТЕНЕ

#> -10 SCALZ Y.ВЪРТЕНЕ

#> 0 SCALZ Y.ВЪРТЕНЕ

ОПЕРАТОР	С Т А С К	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

SCALZ	N - - - - -	Z-СКАЛА
		(ПЕРСПЕКТИВА) ЗА ОБЕКТА

Д. ПОЗИЦИЯ

МЯСТОТО НА ОБЕМНАТА ФИГУРА ВЪРХУ ДВУМЕРНОТО ПРОСТРАНСТВО НА ЕКРАНА СЕ УКАЗВА ЧРЕЗ ДУМИТЕ XPOS И YPOS. ДВЕТЕ ДУМИ ВЗИМАТ ПО ЕДНО ЧИСЛО ОТ СТЕКА ЗА ДА ОПРЕДЕЛЯТ "X" И "Y" КООРДИНАТИТЕ ВЪРХУ ЕКРАНА. КАКТО СИ СПОМНЯТЕ X ТРЯБВА ДА Е МЕЖДУ 0 И 255, А Y - МЕЖДУ 0 И 191.

#> 5 SCALE

#> 50 XPOS

#> 40 YPOS

#> 200 XPOS

ДА СЕ ОПИТАМЕ ДА ИЗВЕДЕМ ОБЕКТА В САМИЯ КРАЙ НА ЕКРАНА:

#> 5 YPOS

ДА ГО ВЪРНЕМ ОТНОВО:

#> 96 YPOS

ОПЕРАТОР	С Т А С К	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

XPOS	N - - - - -	X-ПОЗИЦИЯ
		ВЪРХУ ЕКРАНА ЗА ОБЕКТА

YPOS	N - - - - -	Y-ПОЗИЦИЯ
		ВЪРХУ ЕКРАНА ЗА ОБЕКТА

Е. ТРАНСЛИРАНЕ

ТРАНСЛИРАНЕТО Е ПРИДВИЖВАНЕ НА ОБЕКТА НЕ ВЪРХУ ПЛОСКИЯ ЕКРАН, А В НЕГОВОТО СОБСТВЕНО ТРИМЕРНО ПРОСТРАНСТВО. ОБЕКТИТЕ МОГАТ ДА БЪДАТ ПРЕМЕСТВАНИ ПО ЕДНА ОТ ТРИТЕ ОСИ ЧРЕЗ ГРАФОРТ-ДУМИТЕ "XTRAN", "YTRAN" И "ZTRAN". ВАЖНО Е ПРИ ТРАНСЛИРАНЕТО ДА ЗАПАЗИТЕ ОБЕКТА В РАМКИТЕ НА ПРОСТРАНСТВОТО НА НЕГОВИЯ КУБ. В ПРОТИВЕН СЛУЧАЙ ЩЕ СЕ ПОЛУЧИ "ЗАЦИКЛЯНЕ" НА ФИГУРАТА.

ТЕКУЩИЯТ НИ ОБЕКТ (XYZ) ДОСТИГА ГРАНИЦИТЕ НА СВОЯ КУБ, ТАКА ЧЕ НЕ МОЖЕМ ДА ГО ТРАНСЛИРАМЕ БЕЗ ДА ГО "ЗАЦИКЛИМ":

#> 5 XTRAN

ДА ЗАРЕДИМ НОВ ОБЕКТ:

#> ERASE

#> CR 132 PUTS PRINT " BLOAD КЪЩА, АЗООО " CR

#> 1 ОБЈЕСТ 3000 ОБЈАДР

#> 20 XROT

#> 10 YROT

#> 8 SCALZ

#> 10 SCALE

СЕГА КЪЩАТА МОЖЕ ДА БЪДЕ ТРАНСЛИРАНА:

#> -50 ZTRAN

#> 50 ZTRAN

#> -25 XTRAN

СПИТАЙТЕ СЕГА С ПО-РАНО ДЕФИНИРАНАТА ДУМА Y.ВЪРТЕНЕ:

#> Y.ВЪРТЕНЕ

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

XTRAN N - - - - - ТРАНСЛИРА
 C N ОБЕКТА ПО X-ОСТА

YTRAN N - - - - - ТРАНСЛИРА
 C N ОБЕКТА ПО ОСТА Y

ZTRAN N - - - - - ТРАНСЛИРА
 C N ОБЕКТА ПО ОСТА Z

F. ЦВЯТ НА ОБЕКТА

АКО ИМАТЕ ЦВЕТЕН МОНИТОР, ВЕЧЕ СТЕ ЗАБЕЛЯЗАЛИ ЧЕ ВСЯКА ЕДНА ОТ ТРИТЕ ОСИ НА ОБЕКТА XYZ Е В РАЗЛИЧЕН ЦВЯТ. АКО ПО ВРЕМЕ НА ДЕФИНИРАНЕ НА ОБЕКТА НЕ СТЕ ЗАДАЛИ ЦВЯТ, ТО ТОИ МОЖЕ ДА БЪДЕ ЗАДАДЕН ПРЕДИ ИЗВЕЖДАНЕ ЧРЕЗ ДУМАТА OBJCOLOR. СТАНДАРТНИТЕ НОМЕРА НА ЦВЕТОВЕТЕ СЕ ИЗПОЛЗВАТ ЗА ИЗБОР:

#> 1 OBJCOLOR

#> 5 OBJCOLOR

И ТРИМЕРНИТЕ ГРАФИКИ МОГАТ ДА БЪДАТ ИЗВЕЖДАНИ В РАЗЛИЧНИ РЕЖИМИ ОПРЕДЕЛЯНИ ЧРЕЗ ДУМИТЕ INVERSE И NORMAL, ORMODE И EXMODE.

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====

OBJCOLOR	N - - - - -	ЦВЯТ ЗА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ
----------	-------------	------------------------------------

G. МАНИПУЛАТОР ЗА ФИГУРИ "ОГЛЕД".

НА ДИСКЕТАТА Е ЗАПИСАН ФАЙЛ С ИМЕ "ОГЛЕД". ТОИ СЪДЪРЖА ЕДНА ПРОГРАМА, КОЯТО ЩЕ ВИ ПОЗВОЛИ ДА РАЗБЕРЕТЕ ПО-ДОБРЕ ТОКУ ЩО РАЗГЛЕДАНИТЕ ГРАФОРТ-ДУМИ.

ДА КОМПИЛИРАМЕ ПРОГРАМАТА:

#> ABORT

#> READ " ОГЛЕД "

#> RUN

ОГЛЕД - МАНИПУЛАТОР ЗА ОБЕМНИ ФИГУРИ

РОТАЦИЯ			СКАЛА			ТРАНСЛАЦИЯ			ПОЗИЦИЯ	
1	2	3	4	5	6	7	8	9	:	-
X	Y	Z	X	Y	Z	X	Y	Z	X	Y

НАТИСНИ ГОРНИТЕ КЛАВИШИ ЗА ИЗБИРАНЕ,
А СЛЕД ТОВА НАТИСНИ

, . ЗА ЗАДВИЖВАНЕ
↓ ЗА ЗАМРАЗЯВАНЕ
↓ ЗА ВЪЗСТАНОВЯВАНЕ

ИЛИ НАТИСНИ

ESC ЗА ВЪЗСТАНОВЯВАНЕ НА ВСИЧКО
CTRL S ЗА ВРЕМЕННО СПИРАНЕ
/ ЗА ТЕЗИ ИНСТРУКЦИИ
Я ЗА КРАИ.

ФИГУРА В ПАМЕТТА ИЛИ НА ДИСК?

ВЪВЕДИ АДРЕС НА ОБРАЗА : 2816
ИМЕ НА ФАЙЛА С ОБРАЗА :

НАТИСНИ RETURN ЗА НАЧАЛО...

ТОВА Е ПЪРВИЯТ ЕКРАН. ПЪРВО ТРЯБВА ДА ОТГОВОРИТЕ ДАЛИ
ДЕФИНИЦИЯТА СЕ НАМИРА В ПАМЕТТА (П) ИЛИ Е ЗАПИСАНА НА ДИСКЕТАТА (Д).
ВЪВЕДЕТЕ "Д". СЛЕД ТОВА НАТИСНЕТЕ RETURN, ЗА ДА ПРИЕТЕ ПРЕДЛОЖЕНИЯ
ВИ АДРЕС (2816). ВЪВЕДЕТЕ И ИМЕТО НА ФАЙЛА - "КЪДА". НАТИСНЕТЕ
RETURN.

ROT
0
0
0

СКЛ
16
16
0

TRH
0
0
0

POS
128
96

?

[[[XROT СТОИН.: 0 СЪПКА: 0

ТОВА Е СЛЕДВАЩИЯ ЕКРАН, КАТО СЕ ПРЕДПОЛАГА ЧЕ СТЕ НАТИСНАЛИ "1". НА ДОЛНИЯ (ИНФОРМАЦИОНЕН) РЕД ЩЕ БЪДЕ ИЗВЕДЕН ПЪРВО НАТИСНАТИЯТ КЛАВИШ ЗАЕДНО С ГРАФОРТ ДУМАТА, КОЯТО Е СВЪРЗАНА С НЕГО:

- [1] XROT
- [2] YROT
- [3] ZROT
- [4] SCALX
- [5] SCALY
- [6] SCALZ
- [7] XTRAN
- [8] YTRAN
- [9] ZTRAN
- [:] XPOS
- [-] YPOS

ВЕДНАГА СЛЕД ТОВА СЕ ИЗВЕЖДА ТЕКУЩАТА СТОИНОСТ НА ДУМАТА И СТЬПКАТА С КОЯТО ТЯ СЕ ПРОМЕНЯ. ЗА ДА ПРОМЕНИТЕ СТЬПКАТА ИЗПОЛЗВАЙТЕ КЛАВИШИТЕ "<" И ">" В РЕЖИМ КИРИЛИЦА (ОТБЕЛЯЗАНИ СА НА ПОМОЩНОТО МЕНЮ КАТО ", " И ". "). СЛЕД КАТО УСТАНОВИТЕ ЖЕЛАНАТА СТЬПКА, МОЖЕТЕ ДА ПРЕМИНЕТЕ КЪМ УСТАНОВЯВАНЕ НА СТЬПКА ЗА НЯКОИ ОТ ДРУГИТЕ ПАРАМЕТРИ ЧРЕЗ НАТИСКАНЕ НА НЯКОИ ОТ КЛАВИШИТЕ НА ГОРНИЯ РЕД (1, 2, 3 ... :, -).

В ДЯСНО НА ЕКРАНА ЩЕ ВИЖДАТЕ ТЕКУЩИТЕ СТОИНОСТИ ЗА ВСЕКИ ЕДИН ОТ ПАРАМЕТРИТЕ. НА ФИГУРАТА ТУК СА ПОКАЗАНИ НАЧАЛНИТЕ СТОИНОСТИ. КОГАТО НАТИСНЕТЕ "ESC" (ESC), ТЕЗИ СТОИНОСТИ ЩЕ БЪДАТ ВЪЗСТАНОВЕНИ. АКО НАТИСНЕТЕ "D", ЩЕ БЪДЕ ВЪЗСТАНОВЕНА СТОИНОСТТА ПО ПРЕМЪЛЧАВАНЕ САМО ЗА ТЕКУЩО ИЗБИРАНИЯ ПАРАМЕТЪР. ПРИ НАТИСКАНЕ НА "Q", СТОИНОСТТА НА СТЬПКАТА ЩЕ БЪДЕ НУЛИРАНА И ИЗБИРАНИЯ ПАРАМЕТЪР ЩЕ ПРЕСТАНЕ ДА СЕ ПРОМЕНЯ.

ПО ВСЯКО ВРЕМЕ ЧРЕЗ НАТИСКАНЕ НА "?" В РЕЖИМ КИРИЛИЦА (ОТБЕЛЯЗАН ЧРЕЗ "/") ЩЕ МОЖЕТЕ ДА ВИДИТЕ СЛЕДНИЯ ЕКРАН:

РОТАЦИЯ	СКАЛА	ТРАНСЛАЦИЯ	ПОЗИЦИЯ
1 2 3	4 5 6	7 8 9	: -
X Y Z	X Y Z	X Y Z	X Y

НАТИСНИ ГОРНИТЕ КЛАВИШИ ЗА ИЗБИРАНЕ,

А СЛЕД ТОВА НАТИСНИ

- , . ЗА ЗАДВИЖВАНЕ
- ♦ ЗА ЗАМРАЗЯВАНЕ
- D ЗА ВЪЗСТАНОВЯВАНЕ

ИЛИ НАТИСНИ

- ESC ЗА ВЪЗСТАНОВЯВАНЕ НА ВСИЧКО
- CTRL S ЗА ВРЕМЕННО СПИРАНЕ
- / ЗА ТЕЗИ ИНСТРУКЦИИ
- Я ЗА КРАИ.

"RETURN" ЗА ПРОДЪЛЖЕНИЕ...

ЗА ДА СПРЕТЕ ВРЕМЕННО НЕПРЕКЪСНАТОТО ДВИЖЕНИЕ НА ОБРАЗА ПО ЕКРАНА НАТИСНЕТЕ ЕДНОВРЕМЕННО "МК" И "C/S" (CTRL S). КОИ ДА Е НАТИСНАТ КЛАВИШ ЩЕ ВЪЗОБНОВИ ДВИЖЕНИЕТО.

ЗА ИЗХОД ОТ МАНИПУЛАТОРА ВЪВЕДЕТЕ "Я/Q".

НЕ ЗАБРАВЯЙТЕ ДА ИЗТРИЕТЕ ДУМИТЕ НА "ОГЛЕД" ОТ БИБЛИОТЕКАТА - ТОВА Е ЕДНА МНОГО ГОЛЯМА ПРОГРАМА:

#> FORGET DATA

II.35 РЕДАКТОР ЗА ФИГУРИ (Ф.РЕДАКТОР)

"Ф.РЕДАКТОР" Е ФАЙЛ, КОИТО СЪДЪРЖА ПРОГРАМА ЗА ДЕФИНИРАНЕ НА ТРИМЕРНИ ОБЕКТИ. ПРЕДИ ДА Я КОМПИЛИРАТЕ, ИЗТРИЙТЕ ОТ БИБЛИОТЕКАТА ВСИЧКИ НОВОВЪВЕДЕНИ ДУМИ - ПРОГРАМАТА Е ГОЛЯМА.

ДА КОМПИЛИРАМЕ ПРОГРАМАТА:

#> ABORT

#> READ " Ф.РЕДАКТОР "

#> RUN

СЛЕДВА НАЧАЛНИЯТ ЕКРАН:

A=АДРЕС
Ч=ЧЕТЕНЕ
З=ЗАПИС
Р=РСТИРАНЕ
П=ПОЗИЦИЯ
С=СКАЛА
Ж=ПРОЗОРЕЦ
Л=ЛИСТ
Ц=ЦВЯТ
М=РЕЖИМ
Б=ВЪВЕЖДАНЕ
Н=НУЛИРАНЕ
Я=КРАЙ

ВЪВЕДИ КОМАНДА :

A. ИЗБОР НА АДРЕС И ФИГУРА

ПЪРВО ТРЯБВА ДА ИЗБЕРЕМ РАБОТНА ОБЛАСТ В ПАМЕТТА - НАТИСНЕТЕ "А" (АДРЕС).

ВЪВЕДИ АДРЕС НА ФАЙЛА :2816

ВЪВЕДЕТЕ НОВ АДРЕС ИЛИ ПРИЕТЕ (КАКТО В СЛУЧАЯ) ПРЕДЛОЖЕНИЯ.
НЕКА СЕГА ПРОЧЕТЕМ (Ч) ОТ ДИСКЕТАТА ФАЙЛА "XYZ":

ИМЕ НА ФАЙЛА ЗА ЧЕТЕНЕ :XYZ

В. ИЗБОР НА ГЛЕНА ТОЧКА

ЗА ДА ЗАВЪРТИТЕ ОБЕКТА НАТИСНЕТЕ "P":

РОТИРАНЕ [X (...) ДО Z (...)] :

ТУК СЕ ВЪВЕЖДАТ ИМЕ НА ОСТА, ПОСЛЕДВАНО ОТ ЪГЪЛА НА ВЪРТЕНЕ. В СЛУЧАЯ НЕКА ВЪВЕДЕМ "Y16".

РОТИРАНЕ [X (...) ДО Z (...)] :Y16

ДА НАТИСНЕМ ОЩЕ ВЕДНАЖ "P" И ДА ОТГОВОРИМ С "X16".

РОТИРАНЕ [X (...) ДО Z (...)] :X16

ЗА ДА ПРОМЕНИМ РАЗМЕРИТЕ СЕ ИЗПОЛЗВА "C" (СКАЛА).

СКАЛА [(...) ИЛИ X,Y,Z (...)] :

ЗА ДА СЕ ПРОМЕНЯТ X И Y ЕДНОВРЕМЕННО (SCALE) ПРОСТО ВЪВЕДЕТЕ СТОИНОСТТА. В ПРОТИВЕН СЛУЧАИ, ВЪВЕДЕТЕ ИМЕТО НА ОСТА ПОСЛЕДВАНО ОТ ЖЕЛАНАТА СКАЛА. ПЪРВОНАЧАЛНО СКАЛАТА Е 8.

ЗА ДА СМЕНИТЕ ПОЗИЦИЯТА НА ОБЕКТА ВЪРХУ ЕКРАНА, НАТИСНЕТЕ "P" (ПОЗИЦИЯ).

ПОЗИЦИЯ [X (...) ИЛИ Y (...)] :

ПЪРВОНАЧАЛНО ПОЗИЦИЯТА Е (64,48).

АКО ЦВЯТ НЕ Е ПОССОЧЕН ВЪВ ФАИЛА С ФИГУРАТА, МОЖЕТЕ ДА ИЗБЕРЕТЕ И ЦВЯТ ЧРЕЗ "C".

НАЧАЛЕН ЦВЯТ (0-7) :

ЧРЕЗ НАТИСКАНЕ НА "M" (РЕЖИМ) МОЖЕТЕ ДА ИЗБЕРЕТЕ РЕЖИМ НА ИЗВЕЖДАНЕ "X" (EXMODE) ИЛИ "I" (ORMODE):

РЕЖИМ (X)=ИЗКЛЮЧВАЩ ИЛИ (I)=ИЛИ :

С. ДАННИ ЗА ФАИЛА С ФИГУРАТА

ВЪВЕДЕТЕ "A" (ЛИСТ) ЗА ДА ВИДИТЕ ЧИСЛАТА, КОИТО ДЕФИНИРАТ ФИГУРАТА. МОЖЕТЕ ДА НАТИСКАТЕ RETURN, ЗА ДА ВИДИТЕ ВСИЧКИ ДАННИ ИЛИ ДА НАТИСНЕТЕ ЕДНОВРЕМЕННО "MK" И "C/C" (CTRL-C ИЛИ [C]) ЗА ДА СПРЕТЕ:

(RETURN=ОЩЕ, CTRL-C=ИЗХОД) :

ЕДИН ЕЛЕМЕНТ В ТОЗИ РЕДАКТОР ИМА СЛЕДНИЯ ФОРМАТ:

```
ИЦ X      Y      Z      АДР
!! !      !      !      !
!! !      !      !      +-----> АДРЕС НА ЕЛЕМЕНТА
!! !      !      !
!! !      !      +-----> Z-КООРДИНАТА НА ТОЧКАТА
!! !      !
!! !      +-----> Y-КООРДИНАТА НА ТОЧКАТА
!! !
!! +-----> X-КООРДИНАТА НА ТОЧКАТА
!!
!+-----> ЦВЯТ ЗА ТОЧКАТА
!
+-----> Д (ДВИЖЕНИЕ MOVE); Ч (ЧЕРТАНЕ DRAW)
```

ВСЕКИ ЕЛЕМЕНТ ЗАЕМА 4 БАЙТА В ПАМЕТТА. ЕДНА ДЕФИНИЦИЯ ТРЯБВА ДА ЗАПОЧВА С "Д" - ДВИЖЕНИЕ ДО ПЪРВАТА ТОЧКА. "Ч" ЧЕРТАЕ ЛИНИЯ ОТ ПРЕДИШНАТА ТОЧКА.

КОМАНДАТА "Ж" (ПРОЗОРЕЦ) ЩЕ ДЕФИНИРА ТЕКСТОВИЯ ПРОЗОРЕЦ В ДОЛНИЯ КРАИ НА ЕКРАНА, КАТО ЗА ГОРЕН КРАИ ЩЕ ПРИЕМЕ ВЪВЕДЕНИЯ РЕД:

ГОРЕН КРАИ НА ПРОЗОРЕЦА (0-23) :

КОМАНДАТА "Ж" ИЗЧИСТВА ЕКРАНА, КОГАТО ПОСТАВЯ НОВ ТЕКСТОВ ПРОЗОРЕЦ.

Д. СЪЗДАВАНЕ НА НОВИ ФИГУРИ

ДА СЪЗДАДЕМ СОБСТВЕНА ФИГУРА - КУБ. ПЪРВО ТРЯБВА ДА ИЗТРИЕМ СТАРАТА ДЕФИНИЦИЯ - НАТИСНЕТЕ "Н" (НУЛИРАНЕ):

ИЗТРИВАНЕ НА ФАЙЛА (Д/Н) :

ОТГОВОРЕТЕ С "Д" ЗА ДА ИЗТРИЕТЕ ОТ ПАМЕТТА СТАРАТА ДЕФИНИЦИЯ. АКО СТАРАТА ФИГУРА ВИ ДРАЗНИ, ВЪВЕДЕТЕ "Ж":

ГОРЕН КРАИ НА ПРОЗОРЕЦА (0-23) :14

ДА ВЪВЕДЕМ И ПЕРСПЕКТИВА ЗА ОБЕКТА, ТАКА ЧЕ ДА МОЖЕМ ДА ВИЖДАМЕ ВСИЧКИТЕ МУ СТРАНИ:

СКАЛА [(...) ИЛИ X,Y,Z (...)] :28

ВЪВЕДЕТЕ "В" (ВЪВЕЖДАНЕ).

(Д)ВИЖ, (Ч)ЕРТ, (-)АНУЛ, (CR)КРАИ :

ПО НАТАТЪК ЗА ЦВЯТ ЩЕ ВЪВЕЖДАМЕ ПРОСТО RETURN, КОЕТО ЩЕ НИ ПОЗВОЛИ ПО-КЪСНО ДА МОЖЕМ ДА ИЗБИРАМЕ ЦВЯТ С OBJCOLOR.

1.	(D)	X = -127	Y = 127	Z = 127
2.	(4)	X = 127	Y = ---	Z = ---
3.	(4)	X = ---	Y = ---	Z = -127
4.	(4)	X = -127	Y = ---	Z = ---
5.	(4)	X = ---	Y = ---	Z = 127
6.	(4)	X = ---	Y = -127	Z = ---
7.	(4)	X = 127	Y = ---	Z = ---
8.	(4)	X = ---	Y = ---	Z = -127
9.	(4)	X = -127	Y = ---	Z = ---
10.	(4)	X = ---	Y = ---	Z = 127
11.	(D)	X = 127	Y = 127	Z = ---
12.	(4)	X = ---	Y = -127	Z = ---
13.	(D)	X = ---	Y = 127	Z = -127
14.	(4)	X = ---	Y = -127	Z = ---
15.	(D)	X = -127	Y = 127	Z = ---
16.	(4)	X = ---	Y = -127	Z = ---

С "----" Е ОЗНАЧЕНО НАТИСКАНЕТО САМО НА RETURN, ЗАЩОТО НЕОБХОДИМАТА НИ СТОИНОСТ Е ПРЕДЛОЖЕНА ОТ РЕДАКТОРА (ТОВА Е ПРЕДИШНАТА СТОИНОСТ ЗА СЪОТВЕТНАТА КООРДИНАТА).

Е. ЗАПАЗВАНЕ НА ФАЙЛА С ФИГУРАТА

НАТИСНЕТЕ "З" (ЗАПИС), ЗА ДА ЗАПАЗИТЕ ДЕФИНИРАНАТА ФИГУРА НА ДИСКЕТА. В ОТГОВОР НА СЪОБЩЕНИЕТО:

ИМЕ НА ФАЙЛА ЗА ЗАПИС :

МОЖЕТЕ ДА ДАДЕТЕ КАКВО ДА Е ИМЕ. ДЕМОНСТРАЦИОННАТА ДИСКЕТА СЪДЪРЖА ВЕЧЕ ТАЗИ ФИГУРА ПОД ИМЕТО "КУБ".

II.36 НАЧИНИ ЗА ОБЕМНО ИЗВЕЖДАНЕ

КОГАТО СЕ ДВИЖАТ ТРИМЕРНИ ОБЕКТИ ОТ ГРАФОРТ-ПРОГРАМА, ВМЕСТО AUTODRAW, СЕ ИЗПОЛЗВА DRAW. DRAW ИЗВЕЖДА ТЕКУЩИЯ ОБЕКТ ДЕФИНИРАН ЧРЕЗ OBJADR И OBJEST. ИЗВЕЖДАНЕТО ЧРЕЗ DRAW ИМА СВОИТЕ ПРЕДИМСТВА - МОГАТ ДА БЪДАТ ИЗДАДЕНИ МНОГО ГРАФИЧНИ КОМАНДИ ПРЕДИ ДА БЪДЕ ИЗЧЕРТАН ОБЕКТА НА ЕКРАНА (ПРИ AUTODRAW ВСЯКА ГРАФИЧНА КОМАНДА ПРЕДИЗВИКВА НЕЗАБАВНОТО ИЗВЕЖДАНЕ НА ОБЕКТА).

ОПЕРАТОР	СТАК	ОБЯСНЕНИЯ
=====	ПРЕДИ=====	СЛЕД=====
DRAW	- - - - -	ЧЕРТАЕ ИЗБРАНИТЕ ОБЕКТИ

НОРМАЛНО DRAW ИЗВЕЖДА ВСИЧКИ ОБЯВЕНИ ОБЕКТИ, КАТО ПРЕДВАРИТЕЛНО ИЗТРИВА ПРЕДИШНИТЕ КОПИЯ НА ОБЕКТИТЕ ВЪРХУ ЕКРАНА. КОМАНДАта UNDRAW ИЗТРИВА БЛОК (ВЖ. BLKSIZE) И ПРЕДОТВРАТЯВА АВТОМАТИЧНОТО ИЗТРИВАНЕ ОТ СЛЕДВАЩАТА DRAW-КОМАНДА.

КОМАНДАта OFF ПРЕДИЗВИКВА ИЗТРИВАНЕ НА ОБЕКТА ОТ СЛЕДВАЩАТА DRAW-КОМАНДА.

ОПЕРАТОР S T A C K ОБЪСНЕНИЯ
 =====ПРЕДИ=====СЛЕД=====

UNDRAW - - - - -
 ТРИЕ БЛОК И СПИРА АВТОМАТИЧНО
 ТРИЕНЕ ОТ СЛЕДВАЩИЯ DRAW

OFF - - - - -
 СЛЕДВАЩАТА DRAW-КОМАНДА
 ЩЕ ИЗТРИЕ ИЗБРАНИЯ ОБЕКТ

НОРМАЛНО ГРАФОРТ ИЗПОЛЗВА ДВЕТЕ ГРАФИЧНИ СТРАНИЦИ ПРИ ИЗВЕЖДАНЕТО НА ОБЕКТТЕ. SEQUENCE Е КОМАНДАТА КОЯТО ОПРЕДЕЛЯ ДАЛИ АВТОМАТИЧНО ЩЕ БЪДАТ ИЗПОЛЗВАНИ И ДВАТА ГРАФИЧНИ ЕКРАНА ИЛИ НЕ. SEQUENCE ВЗИМА ЧИСЛО ОТ СТЕКА И АКО ТО Е НУЛА СПИРА АВТОМАТИЧНАТА СМЯНА НА ЕКРАНИТЕ; АКО ЧИСЛОТО Е РАЗЛИЧНО ОТ НУЛА - ВЪЗСТАНОВЯВА АВТОМАТИЧНИЯ РЕЖИМ.

КОГАТО АВТОМАТИЧНАТА СМЯНА Е СПРЯНА ЧРЕЗ "0 SEQUENCE", ИЗБОРА НА ГРАФИЧНА СТРАНИЦА ЗА ИЗВЕЖДАНЕ ВЪРХУ ЕКРАНА СТАВА ЧРЕЗ ДУМАТА SCREEN. ДВЕТЕ СТРАНИЦИ СА НОМЕРИРАНИ 0 И 1. SCREEN ВЗИМА ОТ СТЕКА НОМЕРА НА ГРАФИЧНАТА СТРАНИЦА, КОЯТО ТРЯБВА ДА БЪДЕ ИЗВЕДЕНА.

ОПЕРАТОР S T A C K ОБЪСНЕНИЯ
 =====ПРЕДИ=====СЛЕД=====

SCREEN N - - - - - N = [0!1]
 ИЗБОР НА ГРАФИЧЕН ЕКРАН (0,1)

SEQUENCE N - - - - - N = [0!1]
 N=0: СПИРА СМЯНАТА НА ГРАФ. ЕКРАНИ

II.37 ПРОФИЛ

ЕТО ОЩЕ ЕДНА ПРОГРАМА НАМИРАЩА СЕ НА ДИСКЕТАТА, КОЯТО ВИ ПОЗВОЛЯВА ДА СЪЗДАВАТЕ ТРИМЕРНИ ОБЕКТИ. ТАЗИ ПРОГРАМА Е ЗАПИСАНА КАТО ФАЙЛ С ИМЕ "ПРОФИЛ". ИЗЧИСТЕТЕ БИБЛИОТЕКАТА ОТ ИЗЛИШНИ ДУМИ. ДА КОМПИЛИРАМЕ ПРОГРАМАТА:

#> READ " ПРОФИЛ "

#> RUN

A. ПАРАМЕТРИ

ЩЕ ДЕФИНИРАМЕ ЕДИН КОНУС. ЕТО И ПЪРВИЯТ ЕКРАН:

[ПРОФИЛ ГЕНЕРАТОР]

ТАЗИ ПРОГРАМА ГЕНЕРИРА ОБЕКТИ С МНОГОЪГЪЛНО СЕЧЕНИЕ И ЗАДАДЕН ПРОФИЛ ЗА ОБЕМНИ ГРАФОРТ ПРИЛОЖЕНИЯ.

ПРОФИЛ-ДАНКТЕ СА X, Y ДВОЙКИ В ИНТЕРВАЛА ОТ -127 ДО 127.

БРОИ НА СТРАНИТЕ НА МНОГОЪГЪЛНИКА : 20

ТОЗИ БРОЙ ЩЕ ДЕФИНИРА СЕЧЕНИЕТО НА КОНУСА. ЗА ДА БЪДЕ ТО ИДЕАЛНА ОКРЪЖНОСТ, БРОЯ НА СТРАНИТЕ ТРЯБВА ДА Е БЕЗКРАЕН. НО В КОМПЮТЪРА НЯМА ТОЛКОВА ПАМЕТ.

АДРЕС НА ОБЕКТОВИЯ ФАЙЛ : 2016

А ТУК ПРОСТО НАТИСНЕТЕ RETURN.

В. ВЪВЕЖДАНЕ НА ДАННИ ОТ КЛАВИАТУРАТА

ДАННИ ОТ [К]ЛАВИАТУРАТА ИЛИ ОТ [Д]ИСК ?
НАПИШЕТЕ "К" ЗА ВХОД ОТ КЛАВИАТУРА.

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") :

Y-КООРДИНАТАТА Е ВЕРТИКАЛНАТА ПОЗИЦИЯ СЪС СТОИНОСТИ ОТ -128 ДО 127. X-КООРДИНАТАТА МОЖЕ ДА БЪДЕ СЧИТАНА ЗА РАДИУС. И НЕИНИТЕ СТОИНОСТИ СА МЕЖДУ -128 И 127, НО СЕ ИЗПОЛЗВАТ САМО КАТО ПОЛОЖИТЕЛНИ.

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : 0,-128

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : 32,-64

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : 64,0

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : 96,64

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : 127,127

ВЪВЕДИ ЧИФТ X,Y (КРАИ="К") : К

ГЕНЕРИРАНЕ НА ФАЙЛА С ФИГУРАТА (824 БАЙТА) ...

ЗАБЕЛЕЖЕТЕ, ЧЕ КОНУСЪТ ИМА ДВАДЕСЕТ ВЕРТИКАЛНИ ЛИНИИ, КОИТО СЕ СПУСКАТ ОТ ВЪРХА ПО ПОВЪРХНОСТТА МУ (ИЗБРАХМЕ 20 СТРАНИ НА МНОГОЪГЪЛНИКА) И ИМА 4 ОКРЪЖНОСТИ И ТОЧКА НА ВЪРХА (ВЪВЕДОХМЕ ПЕТ ЕЛЕМЕНТА).

ИМЕ НА ОБЕКТОВИЯ ФАЙЛ :

АКО ИСКАТЕ ДА ЗАПАЗИТЕ ФИГУРАТА, МОЖЕТЕ ДА ЗАДАДЕТЕ ИМЕТО НА ФАЙЛА. САМО RETURN В ОТГОВОР НА ТОВА СЪОБЩЕНИЕ ЩЕ ЗАВЪРШИ ПРОГРАМАТА БЕЗ ЗАПАЗВАНЕ НА ФАЙЛА. ВЪВЕДЕТЕ НОМЕ ИЛИ ERASE.

С. ВЪВЕЖДАНЕ НА ДАННИ ОТ ДИСКЕТАТА

"ПРОФИЛ" МОЖЕ ДА ЧЕТЕ СПИСЪК КООРДИНАТИ И ОТ ФАЙЛ НА ДИСКЕТАТА. ТЕКСТОВИЯ ФАЙЛ "БОКАЛ2" СЪДЪРЖА КООРДИНАТИТЕ НА ГОЛЯМАТА "ЧАША" ОТ ДЕМОНСТРАЦИЯТА. КОГАТО РАБОТИТЕ С ТЕКСТОВИЯ РЕДАКТОР, МОЖЕТЕ ДА ПОГЛЕДНЕТЕ КАКВО СЪДЪРЖА ФАЙЛ "БОКАЛ2". ЗАБЕЛЕЖЕТЕ, ЧЕ ПОСЛЕДНИЯТ ЕЛЕМЕНТ В НЕГО Е "К".

НАПИШЕТЕ ОТНОВО:

#> RUN

ПРИ БРОИ НА СТРАНИТЕ 3, ОТГОВОРЕТЕ С "Д" НА ВЪПРОСА:

ДАНИИ ОТ [К]ЛАВИАТУРАТА ИЛИ ОТ [Д]ИСК ?Д
ТЕКСТОВИЯТ ФАЙЛ ТРЯБВА ДА ИМА КАТО ПОСЛЕДЕН СВОИ ЕЛЕМЕНТ "К"
ПОСЛЕДВАН ОТ RETURN. ЗА ТОВА ВИ ПОДСЕЩА СЪОБЩЕНИЕТО:

ПОСЛЕДНИЯТ ЕЛЕМЕНТ ТРЯБВА ДА Е "К (CR)"

ИМЕ НА ФАЙЛА С ДАНИИ : БОКАЛ2

СКОРО ЩЕ СЕ ПОЯВИ СЪОБЩЕНИЕТО:

ГЕНЕРИРАНЕ НА ФАЙЛА С ФИГУРАТА (2724 БАЙТА) ...

КАКТО И ПРЕДИ МОЖЕТЕ ДА ЗАПАЗИТЕ НА ДИСКЕТАТА ГЕНЕРИРАНАТА ОТ
"ПРОФИЛ" ФИГУРА.

D. СЪОБРАЖЕНИЯ ЗА ПАМЕТТА

"ПРОФИЛ" ГЕНЕРИРА МНОГО БЪРЗО ГОЛЕМИ И ОБЕМИСТИ ТРИМЕРНИ
ОБЕКТИ. НАЛАГА СЕ ДА СЕ ПРАВИ ПРОВЕРКА ДАЛИ МЯСТОТО В ПАМЕТТА Е
ДОСТАТЪЧНО. АКО ТОВА МЯСТО НЕ ДОСТИГА ЩЕ ПОЛУЧИТЕ СЪОБЩЕНИЕТО:

НЯМА ДОСТАТЪЧНО МЯСТО.
(ТРЯБВАТ NNNNN БАЙТА.)

NNNNN Е ТОЧНИЯТ РАЗМЕР В БАЙТОВЕ, КОИТО СА НЕОБХОДИМИ ЗА
ГЕНЕРИРАНЕ НА ФИГУРАТА.

II.38 МУЗИКА

ДВЕТЕ ДУМИ, С КОИТО СЕ УПРАВЛЯВА МУЗИКАЛНИЯ СИНТЕЗАТОР НА
ГРАФОРТ СА VOICE И NOTE.

A. VOICE

VOICE ИЗБИРА ТЕМБЪРА, С КОИТО ДА БЪДАТ ИЗСВИРЕНИ НОТИТЕ. ДУМАТА
ВЗИМА ЧИСЛО ОТ СТЕКА И ГО ИЗПОЛЗВА ЗА ИЗБИРАНЕ НА ТЕМБЪР. ВАЛИДНИТЕ
СТОИНОСТИ СА МЕЖДУ -6 И 2 - ОБЩО 9 РАЗЛИЧНИ ТЕМБЪРА.

СПЕРАТОР	S	T	A	S	K	ОБЯСНЕНИЯ
=====	ПРЕДИ	=====	СЛЕД	=====		

VOICE	N	-	-	-	-	
	ИЗБИРА ТЕМБЪР (ОТ -6 ДО 2)					

B. NOTE

ДУМАТА NOTE ПРЕДИЗВИКВА ИЗСВИРВАНЕТО НА ОПРЕДЕЛЕНА НОТА С
ОПРЕДЕЛЕНА ПРОДЪЛЖИТЕЛНОСТ. NOTE ВЗИМА ОТ СТЕКА ВИСОЧИНАТА НА НОТАТА
И НЕИНАТА ПРОДЪЛЖИТЕЛНОСТ. И ДВЕТЕ СТОИНОСТИ СА МЕЖДУ 2 И 255.

ПО-ГОЛЕМИЯТ НОМЕР ЗА ВИСОЧИНА НА НОТАТА ИЗБИРА ПО-НИСКА НОТА.
ПО-ГОЛЕМИЯТ НОМЕР ЗА ПРОДЪЛЖИТЕЛНОСТ Е ПО-ДЪЛГА НОТА.

ОПЕРАТОР S T A C K ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====

NOTE M N - - - -
 M - НОТА; N - ПРОДЪЛЖИТЕЛНОСТ

С. ОПРЕДЕЛЯНЕ ПРОДЪЛЖИТЕЛНОСТТА И ВИСОЧИНАТА

255 МОЖЕ ДА СЕ СЧИТА ЗА ЦЯЛА НОТА, 128 ЗА ПОЛОВИНКА, 64 - ЗА ЧЕТВЪРТИНКА И Т.Н.

НОТА	ОКТАВА1	ОКТАВА2	ОКТАВА3	ОКТАВА4
ЛА (A)	243	124	62	31
ЛА# (A#)	234	117	58	29
СИ (B)	221	110	55	27
ДО (C)	209	104	52	26
ДО# (C#)	197	98	49	24
РЕ (D)	186	93	46	23
РЕ# (D#)	175	87	43	21
МИ (E)	166	83	41	20
ФА (F)	156	78	39	19
ФА# (F#)	147	73	36	18
СОЛ (G)	139	69	34	17
СОЛ# (G#)	131	65	32	16

Д. ПОЛЕЗНИ МУЗИКАЛНИ ДУМИ

ЕТО И ЕДНА ДУМА, КОЯТО ЩЕ ГЕНЕРИРА ВСИЧКИ НОТИ И ЩЕ ГИ ЗАПОМНИ В НИЗА "НОТИ".

#> 50 STRING НОТИ

#> : ИЗЧИСЛЯВАНЕ.НОТИ

#> 24870

#> 48 0 DO

#> DUP 100 / I НОТИ РОКЕ

#> DUP 18 / -

#> DUP 1655 / -

#> LOOP DROP ;

НОТИТЕ ЗАПИСАНИ В НИЗА "НОТИ" СА НОМЕРИРАНИ ОТ 0 ДО 47. ДА ВЗЕМЕМ ДО (С) ОТ ПЪРВАТА ОКТАВА - ТОВА Е 3-ТАТА ПОРЕД НОТА (ЗАПОЧВАМЕ ОТ НУЛА).

#> 3 НОТИ РЕЕК

209

ДА ИЗСВИРИМ НОТАТА С ПРОДЪЛЖИТЕЛНОСТ ПОЛОВИН НОТА.

#> 3 НОТИ РЕЕК 128 NOTE

ДА ДЕФИНИРАМЕ ОЩЕ ЕДНА ДУМА:

#> : НОТА НОТИ РЕЕК ;

#> 3 НОТА 128 NOTE

А ЕТО И ВСИЧКИ НОТИ ОТ ЧЕТИРИТЕ ОКТАВИ:

#> 48 0 DO 1 НОТА 32 NOTE LOOP

ТОЗА Е НАЧАЛОТО НА ПОПУЛЯРНА МЕЛОДИЯ:

#> : МЕЛОДИЯ

#> 12 НОТА 50 NOTE

#> 12 НОТА 50 NOTE

#> 14 НОТА 100 NOTE

#> 12 НОТА 100 NOTE

#> 17 НОТА 100 NOTE

#> 16 НОТА 200 NOTE ;

#> МЕЛОДИЯ

ПРИЛОЖЕНИЕ А
СПИСЪК НА ДУМИТЕ ОТ БИБЛИОТЕКАТА ПО РЕДА НА ПОЯВЯВАНЕТО ИМ В
РЪКОВОДСТВОТО.

ОПЕРАТОР	S T A C K	ОБЯСНЕНИЯ
=====ПРЕДИ=====СЛЕД=====		
STACK	- - - - -	ДИСПЛЕИ ИЛИ НЕ НА СТЕКА
	N - - - - -	ИЗВЕЖДА ЧИСЛОТО N
LIST	- - - - -	ИЗВЕЖДА ДУМИТЕ ОТ БИБЛИОТЕКАТА
SWAP	M N - N M -	РАЗМЕСТВА ГОРНИТЕ 2 ЧИСЛА
DROP	N - - - - -	МАХА ЧИСЛО ОТ СТЕКА
DUP	N - - N N -	ДУБЛИРА СТОИНОСТТА
OVER	M N - M N M	КОПИРА 2-РАТА СТОИНОСТ
PICK	..M N - ..M N -	КОПИРА N-ТАТА СТОИНОСТ
CHS	M - - N - -	CHANGE SIGN ПРОМЯНА НА ЗНАКА N = -M
ABS	N - - M - -	ABSOLUTE АБСОЛЮТНА СТОИНОСТ M=!N!
SGN	M - - N - -	SIGN ЗНАК НА ЧИСЛОТО N=-1,0,1
RND	- - - N - -	RANDOM СЛУЧАЙНО ЧИСЛО -32K ДО 32K
RNDS	- - - N - -	RANDOM BYTE СЛУЧАЙНО ЧИСЛО 0 ДО 255
MIN	M N - N - -	N=MIN(M,N) МИНИМУМ НА ДВЕ ЧИСЛА
MAX	M N - N - -	N=MAX(M,N) МАКСИМУМ НА ДВЕ ЧИСЛА
MOD	M N - N - -	ОСТАТЪК СЛЕД M/N
	M N - N - -	N=M/N

*	M N - N - -	N=M*N
+	M N - N - -	N=M+N
-	M N - N - -	N=M-N
SIN	M - - N - -	N=SIN(M) СКАЛАРЕН СИНУС (-128,127)
ABORT	- - - - -	РЕСТАРТИРА ГРАФОРТ
BELL	- - - - -	ИЗДАВА МЕЛОДИЧЕН ЗВУК
"	- - - - -	ОБГРАНДА ТЕКСТ В ПРОГРАМАТА
SPACE	- - - - -	SPACE ИЗВЕЖДА ПРАЗНО (КОД 160)
CR	- - - - -	CARRIAGE RET. ВРЪЩАНЕ НА КУРСОРА (КОД 141)
;	- - - - -	КРАИ НА ДЕФИНИЦИЯ НА ДУМА
:	- - - - -	НАЧАЛО НА ДЕФИНИЦИЯ НА ДУМА
FORGET	- - - - -	ИЗТРИВА БИБЛИОТЕКАТА ДО ДУМАТА СЛЕД FORGET
LOOP	- - - - -	DO...LOOP АКО Е < ОТ КРАИ -> ОБРАТНО
+LOOP	N - - - - -	DO...+LOOP ИЗПОЛЗВА N КАТО НАРАСТВАНЕ
DO	M N - - - - -	DO... LOOP N - НАЧАЛО, M - КРАИ
I	- - - N - -	ВРЪЩА СТОИНОСТ НА ВЪТР.ЦИКЪЛ
J	- - - N - -	СТОИНОСТ НА ВЪНШЕН ЦИКЪЛ
K	- - - N - -	ЗА ЭТОРИ ВЪНШЕН ЦИКЪЛ
POP	- - - - -	МАХА ОТ ВЪЗВРАТНИЯ СТЕК

PULL - - - N - -
 ВЪЗВРАТЕН СТЕК -> СТЕК

PUSH N - - - - -
 СТЕК -> ВЪЗВРАТЕН СТЕК

<> N M - N - - N=0,1
 1, АКО ЧИСЛАТА НЕ СА РАВНИ

= N M - N - - N=0,1
 1, АКО СА РАВНИ

> N M - N - - N=0,1
 1, АКО N > M

< N M - N - - N=0,1
 1, АКО N < M

>= N M - N - - N=0,1
 1, АКО N >= M

<= N M - N - -
 1, АКО N <= M

OR M N - N - - N = M OR N
 ПОБИТОВО ИЛИ

AND M N - N - - N = M AND N
 ПОБИТОВО И

IF N - - - - - IF...THEN
 IF...ELSE...THEN
 N#0 СЛЕД IF, N=0-THEN (ELSE)

THEN - - - - - - IF...THEN
 КРАИ НА IF...THEN ФАЗА

ELSE - - - - - - IF..ELSE..THEN
 РАЗДЕЛЯ IF...THEN ФАЗАТА

BEGIN - - - - - -
 ВЪЗВРАТ ОТ REPEAT, UNTIL

UNTIL N - - - - - BEGIN...UNTIL
 ВРЪЩА КОГАТО N = 0

WHILE N - - - - -
 N=0 СЛЕД REPEAT, N#0 - WHILE

REPEAT - - - - - -
 BEGIN ... WHILE ... REPEAT

CASE: N - - - - - CASE: THEN
 ИЗПЪЛНЯВА N-ТАТА ДЪНА (ОТ 0)

GR - - - - - - GRAPHICS
 ГРАФИЧЕН РЕЖИМ

TEXT - - - - -
 ТЕКСТОВ РЕЖИМ

PRGTOP - - - A - - PROGRAM TOP
 АДРЕС НА ВЪРХА НА БИБЛИОТ.

\$LIST - - - - -
 ИЗВЕЖДА ДУМИТЕ С АДРЕСИТЕ ИМ

EDIT - - - - -
 ВХОД В ТЕКСТОВИЯ РЕДАКТОР

READ - - - - -
 ЧЕТЕ И КОМПИЛИРА ФАЙЛ

MEMRD A - - - - -
 ЧЕТЕ И КОМПИЛИРА ОТ АДРЕС А

(- - - - -
 НАЧАЛО НА КОМЕНТАР

VTAB N - - - - - VERT. TAB
 ПОЗИЦИОНИРА N-ТИ РЕД

HTAB M - - - - - HORIZ. TAB
 ПОЗИЦИОНИРА M-ТА КОЛОНА

WINDOW L W T B - - - LEFT WIDTH
 TOP BOTTOM
 ТЕКСТОВ ПРОЗОРЕЦ

PRINT - - - - -
 ИЗВЕЖДА СЛЕДВАЩИЯ ТЕКСТ

HOME - - - - -
 ИЗТРИВА ТЕКСТА В ПРОЗОРЕЦА

CLEOF - - - - -
 CLEAR TO THE END OF PAGE

CLEOL - - - - -
 CLEAR TO THE END OF LINE

ERASE - - - - -
 ИЗТРИВА ГРАФИЧНИТЕ ЕКРАНИ

POKEW N A - - - - POKE WORD
 ЗАПОМНЯ ЧИСЛОТО НА АДРЕС А

POKE N A - - - - POKE BYTE
 ЗАПОМНЯ БАЙТА НА АДРЕС А

PEEKW A - - N - - PEEK WORD
 ЧЕТЕ ЧИСЛО ОТ АДРЕС А

PEEK A - - N - - PEEK BYTE
 ЧЕТЕ БАЙТ ОТ АДРЕС A

VARIABLE - - - - - -
 ДЕКЛАРИРА ПРОМЕНЛИВА С ИМЕ
 ТОВА КОЕТО СЛЕДВА И ОПЦИОННО
 НАЧАЛНА СТОЙНОСТ ПРЕДИ НЕЯ

-> N - - X X X
 ЗАПАЗВА N В ПРОМЕНЛИВАТА
 СЛЕДВАЩА ->

STRING - - - - - -
 СЛЕДВА СЕ ОТ ИМЕТО НА НИЗА И
 СЕ ПРЕДХОЖДА ОТ БРОЯ СИМВОЛИ

ASSIGN A - - - - -
 СЛЕДВАЩИЯ ТЕКСТ НА АДРЕС A

READLN A - - - - - READ LINE
 ЧЕТЕ ОТ КЛАВИАТ. НА АДРЕС A

WRITELN A - - - - - WRITE LINE
 ПИШЕ ПО ЕКРАНА ОТ АДРЕС A

GETNUM A - - N - - N=0, #0
 СТРИНГА ОТ АДРЕС A В ЧИСЛО

VALID - - - N - - N=0, #0
 0, АКО НЕ Е ЧИСЛО СЛЕД GETNUM

PAD - - - A - -
 A -АДРЕСА НА 120 БАЙТОВ БУФЕР

GETC - - - N - - GET CHARACTER
 ЧЕТЕ СИМВОЛ ОТ КЛАВИАТУРАТА

GETKEY - - - N - - GET KEY
 ЧЕТЕ СИМВОЛ БЕЗ ЧАКАНЕ

CLRKEY - - - - - - CLEAR KEY
 ИЗЧИСТВА ПРЕДИ GETKEY

PUTC N - - - - - PUT CHARACTER
 ИЗВЕЖДА СИМВОЛ С КОД N

BASE ПРОМЕНЛИВА
 ЧИСЛОВА ОСНОВА ЗА СМЯТАНЕ

DECIMAL 10 -> BASE

BINARY 2 -> BASE

HEX 16 -> BASE

RUN - - - - - -
 ИЗПЪЛНЯВА НАЙ-ГОРНАТА ДЪМА

AUTORUN N - - - - - N=0 ! N#0
АКО N#0, АВТОМАТИЧЕН RUN .

SAVEPRG - - - - -
ЗАПАЗВА СИСТЕМАТА НА ДИСК

CLOSE - - - - -
ЗАТВАРЯ ВСИЧКИ ФАЙЛОВЕ

MOVMEM A B N - - -
МЕСТИ N БАЙТА ОТ АДРЕС А НА В

- - - A - - -
А - АДРЕСА НА ДУМАТА СЛЕД '

CALL A - - - - -
ЗАРЕЖДА РЕГИСТРИТЕ ОТ AREG,
XREG, YREG, PREG; ИЗПЪЛНЯВА
ПРОГРАМАТА ОТ АДРЕС А; ЗАРЕЖ-
ДА ОТНОВО AREG XREG YREG PREG

PREG (ПРОМЕНЛИВА)

AREG (ПРОМЕНЛИВА)

XREG (ПРОМЕНЛИВА)

YREG (ПРОМЕНЛИВА)

' - - - - -
КОМПИЛИРА ЕДИНИЧЕН БАЙТ

BYE - - - - -
ИЗХОД КЪМ МОНИТОРА НА ПРАВЕЦ

PLOT X Y - - - - -
ИЗВЕЖДА ТОЧКАТА (X,Y)

FILL X Y - - - - -
ЗАПЪЛВА ПРАВОЪГЪЛНА ОБЛАСТ
ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА
ПО ДИАГОНАЛА ДО (X,Y)

LINE X Y - - - - -
ЧЕРТАЕ ЛИНИЯ ОТ ПОСЛЕДНАТА
МАРКИРАНА ТОЧКА ДО (X,Y)

POSN X Y - - - - - POSITION
ПОЗИЦИОНИРА ДАДЕНА ТОЧКА
БЕЗ ДА Я ОТБЕЛЯЗВА

COLOR N - - - - -
ЦВЯТ ЗА ЛИНИИ И СИМВОЛИ

UNPLOT X Y - - - - -
ИЗТРИВА ТОЧКАТА (X,Y)

UNLINE X Y - - - -
 ИЗТРИВА ЛИНИЯ ОТ ПОСЛЕДНАТА
 МАРКИРАНА ТОЧКА ДО (X,Y)

EMPTY X Y - - - -
 ИЗТРИВА ПРАВОЪГЪЛНА ОБЛАСТ
 ВЖ. FILL

INVERSE - - - - -
 ДОПЪЛНИТЕЛЕН ЦВЯТ ЗА ГРАФИКИ
 И ТЕКСТОВЕ (И БЯЛО И ЧЕРНО)

NORMAL - - - - -
 ВРЪЩА НОРМАЛНИ ЦВЕТОВЕ

ORMODE - - - - -
 ПАЛИ ТОЧКИ НЕЗАВИСИМО ОТ
 ПРЕДИШНОТО ИМ СЪСТОЯНИЕ

EXMODE - - - - -
 ПРОМЕНЯ СЪСТОЯНИЕТО НА ТОЧКИТЕ

GREEK X Y - N - - N=0 ! N#0
 #0 - ЗАПАЛЕНА ТОЧКА

CHRSIZE N - - - - - CHARACTER SIZE
 ИЗБИРА РАЗМЕР НА СИМВОЛИТЕ ЗА
 PRINT, WRITELN, PUTC, BLKSIZE

CHRADR A - - - - - CHAR ADDRESS
 ИЗБИРА СИМВОЛЕН НАБОР (АДРЕС A)

CHRSET ПРОМЕНЛИВА
 АДРЕС НА СТАНД.СИМВОЛЕН НАБОР

BLKSIZE M N - - - - BLOCK SIZE
 ИЗБИРА РАЗМЕР НА СИМВОЛНИЯ
 БЛОК: M-ДЪЛЖИНА, N-ВИСОЧИНА

PUTBLK N - - - - - PUT BLOCK
 ИЗВЕЖДА СИМВОЛЕН БЛОК

UNBLK - - - - -
 ИЗТРИВА СИМВОЛЕН БЛОК

OBJERASE - - - - - В НАЧАЛОТО
 НА ТРИМЕРНИ ГРАФИЧНИ ПРОГРАМИ

OBJECT N - - - - -
 ИЗБИРА ТРИМЕРЕН ОБЕКТ

AUTODRAW N - - - - - N#0: DRAW
 СЛЕД ВСЯКА ГРАФИЧНА КОМАНДА

OBJADR A - - - - - УКАЗВА АДРЕС
 НА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ

XROT	N - - - - -	ИЗБИРА N ЗА ВЪРТЕНЕ ПО ОСТА X
YROT	N - - - - -	ИЗБИРА N ЗА ВЪРТЕНЕ ПО ОСТА Y
ZROT	N - - - - -	ИЗБИРА N ЗА ВЪРТЕНЕ ПО ОСТА Z
SCALX	N - - - - -	X-СКАЛА (ШИРОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ
SCALY	N - - - - -	Y-СКАЛА (ВИСОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ
SCALE	N - - - - -	ИЗБИРА X И Y СКАЛИ ЗА ИЗБРАНИЯ ОБЕКТ
SCALZ	N - - - - -	Z-СКАЛА (ПЕРСПЕКТИВА) ЗА ОБЕКТА
XPOS	N - - - - -	X-ПОЗИЦИЯ ВЪРХУ ЕКРАНА ЗА ОБЕКТА
YPOS	N - - - - -	Y-ПОЗИЦИЯ ВЪРХУ ЕКРАНА ЗА ОБЕКТА
XTRAN	N - - - - -	ТРАНСЛИРА С N ОБЕКТА ПО X-ОСТА
YTRAN	N - - - - -	ТРАНСЛИРА С N ОБЕКТА ПО ОСТА Y
ZTRAN	N - - - - -	ТРАНСЛИРА С N ОБЕКТА ПО ОСТА Z
OBJCOLOR	N - - - - -	ЦВЯТ ЗА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ
DRAW	- - - - -	ЧЕРТАЕ ИЗБРАНИТЕ ОБЕКТИ
UNDRAW	- - - - -	ТРИЕ БЛОК И СПИРА АВТОМАТИЧНО ТРИЕНЕ ОТ СЛЕДВАЩИЯ DRAW
OFF	- - - - -	СЛЕДВАЩАТА DRAW-КОМАНДА НЕ ИЗТРИЕ ИЗБРАНИЯ ОБЕКТ
SCREEN	N - - - - -	N = [0!1] ИЗБОР НА ГРАФИЧЕН ЕКРАН (0,1)
SEQUENCE	N - - - - -	N = [0!1] N=0: СПИРА СМЯНАТА НА ГРАФ. ЕКРАНИ

VOICE N - - - -
 ИЗБИРА ТЕМБЪР (ОТ -6 ДО 2)

NOTE M N - - - -
 M - НОТА; N - ПРОДЪЛЖИТЕЛНОСТ

ПРИЛОЖЕНИЕ В
СПИСЪК НА ДУМИТЕ ОТ БИБЛИОТЕКАТА ПО АЗБУЧЕН РЕД.

ОПЕРАТОР	STACK	ОБЯСНЕНИЯ	СТР.
=====ПРЕДИ=====СЛЕД=====			
"	- - - - -	ОБГРАЖДА ТЕКСТ В ПРОГРАМАТА	13
\$LIST	- - - - -	ИЗВЕЖДА ДУМИТЕ С АДРЕСИТЕ ИМ	31
'	- - - A - -	A - АДРЕСА НА ДУМАТА СЛЕД '	51
(- - - - -	НАЧАЛО НА КОМЕНТАР	38
*	M N - N - -	$N=M*N$	11
+	M N - N - -	$N=M+N$	11
+LOOP	N - - - - -	DO...+LOOP ИЗПОЛЗВА N КАТО НАРАСТВАНЕ	16
,	- - - - -	КОМПИЛИРА ЕДИНИЧЕН БАЙТ	52
-	M N - N - -	$N=M-N$	11
->	N - - X X X	ЗАПАЗВА N В ПРОМЕНЛИВАТА СЛЕДВАЩА ->	42
.	N - - - - -	ИЗВЕЖДА ЧИСЛОТО N	8
/	M N - N - -	$N=M/N$	11
:	- - - - -	НАЧАЛО НА ДЕФИНИЦИЯ НА ДУМА	14
;	- - - - -	КРАЙ НА ДЕФИНИЦИЯ НА ДУМА	14
<	N M - N - -	$N=0,1$ 1, АКО $N < M$	19
<=	N M - N - -	$N=0,1$ 1, АКО $N <= M$	19
<>	N M - N - -	$N=0,1$ 1, АКО ЧИСЛАТА НЕ СА РАВНИ	19
=	N M - N - -	$N=0,1$ 1, АКО СА РАВНИ	19

>	N M - N - - N=0,1 1, АКО N > M	19
>=	N M - N - - N=0,1 1, АКО N >= M	19
ABORT	- - - - - РЕСТАРТИРА ГРАФОРТ	12
ABS	N - - M - - ABSOLUTE АБСОЛЮТНА СТОИНОСТ M=!N!	11
AND	M N - N - - N = M AND N ПОБИТОВО И	20
AREG	(ПРОМЕНЛИВА)	52
ASSIGN	A - - - - - СЛЕДВАЩИЯ ТЕКСТ НА АДРЕС A	44
AUTODRAW	N - - - - - N#0: DRAW СЛЕД ВСЯКА ГРАФИЧНА КОМАНДА	71
AUTORUN	N - - - - - N#0 ! N#0 АКО N#0, АВТОМАТИЧЕН RUN	50
BASE	ПРОМЕНЛИВА ЧИСЛОВА ОСНОВА ЗА СМЯТАНЕ	49
BEGIN	- - - - - ВЪЗВРАТ ОТ REPEAT, UNTIL	25,26
BELL	- - - - - ИЗДАВА МЕЛОДИЧЕН ЗВУК	29
BINARY	2 -> BASE	49
BLKSIZE	M N - - - - - BLOCK SIZE ИЗБИРА РАЗМЕР НА СИМВОЛНИЯ БЛОК: M-ДЪЛЖИНА, N-ВИСОЧИНА	68
BYE	- - - - - ИЗХОД КЪМ МОНИТОРА НА ПРАВЕЦ	53
CALL	A - - - - - ЗАРЕЖДА РЕГИСТРИТЕ ОТ AREG, XREG, YREG, PREG; ИЗПЪЛНЯВА ПРОГРАМАТА ОТ АДРЕС A; ЗАРЕЖ- ДА ОТНОВО AREG XREG YREG PREG	52
CASE:	N - - - - - CASE: THEN ИЗПЪЛНЯВА N-ТАТА ДУМА (ОТ 0)	27
CHRADR	A - - - - - CHAR ADDRESS ИЗБИРА СИМВОЛЕН НАБОР (АДРЕС A)	63

CHRSET	ПРОМЕНЛИВА АДРЕС НА СТАНД.СИМБОЛЕН НАБОР	63
CHRSIZE	N - - - - CHARACTER SIZE ИЗБИРА РАЗМЕР НА СИМБОЛИТЕ ЗА PRINT, WRITELN, PUTC, BLKSIZE	62
CHS	M - - N - - CHANGE SIGN ПРОМЯНА НА ЗНАКА N = -M	11
CLEOL	- - - - - CLEAR TO THE END OF LINE	39
CLEOP	- - - - - CLEAR TO THE END OF PAGE	39
CLOSE	- - - - - ЗАТВАРЯ ВСИЧКИ ФАЙЛОВЕ	51
CLRKEY	- - - - - CLEAR KEY ИЗЧИСТВА ПРЕДИ GETKEY	48
COLOR	N - - - - ЦВЯТ ЗА ЛИНИИ И СИМБОЛИ	54
CR	- - - - - CARRIAGE RET. ВРЪЩАНЕ НА КУРСОРА (КОД 141)	13,39
DECIMAL	10 -> BASE	49
DO	M N - - - - DO... LOOP N - НАЧАЛО, M - КРАЙ	16
DRAW	- - - - - ЧЕРТАЕ ИЗБРАНТЕ ОБЕКТИ	82
DROP	N - - - - МАХА ЧИСЛО ОТ СТЕКА	9
DUP	N - - N N - ДУБЛИРА СТОИНОСТТА	9
EDIT	- - - - - ВХОД В ТЕКСТОВИЯ РЕДАКТОР	32
ELSE	- - - - - IF..ELSE..THEN РАЗДЕЛЯ IF...THEN ФРАЗАТА	22
EMPTY	X Y - - - - ИЗТРИВА ПРАВОЪГЪЛНА ОБЛАСТ ВЖ. FILL	55
ERASE	- - - - - ИЗТРИВА ГРАФИЧНИТЕ ЕКРАНИ	39
EXMODE	- - - - - ПРОМЕНЯ СЪСТОЯНИЕТО НА ТОЧКИТЕ	57,69

FILL	X Y - - - - ЗАПЪЛВА ПРАВОЪГЪЛНА ОБЛАСТ ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА ПО ДИАГОНАЛА ДО (X,Y)	53
FORGET	- - - - - ИЗТРИВА БИБЛИОТЕКАТА ДО ДУМАТА СЛЕД FORGET.	15
GETC	- - - N - - GET CHARACTER ЧЕТЕ СИМВОЛ ОТ КЛАВИАТУРАТА	48
GETKEY	- - - N - - GET KEY ЧЕТЕ СИМВОЛ БЕЗ ЧАКАНЕ	48
GETNUM	A - - N - - N=0, #0 СТРИНГА ОТ АДРЕС A В ЧИСЛО	45
GREEK	X Y - N - - N=0 ! N#0 #0 - ЗАПАЛЕНА ТОЧКА	57
GR	- - - - - GRAPHICS ГРАФИЧЕН РЕЖИМ	30
HEX	16 -> BASE	49
HOME	- - - - - ИЗТРИВА ТЕКСТА В ПРОЗОРЕЦА	39
HTAB	M - - - - - HORIZ.TAB ПОЗИЦИОНИРА M-ТА КОЛОНА	39
I	- - - N - - ВРЪЩА СТОИНОСТ НА ВЪТР.ЦИКЪЛ	16
IF	N - - - - - IF...THEN IF...ELSE...THEN N#0 СЛЕД IF, N=0-THEN (ELSE)	21
INVERSE	- - - - - ДОПЪЛНИТЕЛЕН ЦВЯТ ЗА ГРАФИКИ И ТЕКСТОВЕ (И БЯЛО И ЧЕРНО)	56
J	- - - N - - СТОИНОСТ НА ВЪНШЕН ЦИКЪЛ	16
K	- - - N - - ЗА ВТОРИ ВЪНШЕН ЦИКЪЛ	16
LINE	X Y - - - - ЧЕРТАЕ ЛИНИЯ ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА ДО (X,Y)	53
LIST	- - - - - ИЗВЕЖДА ДУМИТЕ ОТ БИБЛИОТЕКАТА	7

LOOP	- - - - - DO...LOOP АКО Е < ОТ КРАИ -> ОБРАТНО	16
MAX	M N - N - - N=MAX(M,N) МАКСИМУМ НА ДВЕ ЧИСЛА	11
MEMRD	A - - - - - ЧЕТЕ .И КОМПИЛИРА ОТ АДРЕС А	38
MIN	M N - N - - N=MIN(M,N) МИНИМУМ НА ДВЕ ЧИСЛА	11
MOD	M N - .N - - ОСТАТЪК СЛЕД M/N	11
MOVMEM	A B N - - - МЕСТИ N БАЙТА ОТ АДРЕС А НА В	51
NORMAL	- - - - - ВРЪЩА НОРМАЛНИ ЦВЕТОВЕ	56
NOTE	M N - - - - M - НОТА; N - ПРОДЪЛЖИТЕЛНОСТ	85
OBJADR	A - - - - - УКАЗВА АДРЕС. НА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ	71
OBJCOLOR	N - - - - - ЦВЯТ ЗА ИЗБРАНИЯ ТРИМЕРЕН ОБЕКТ	76
OBJECT	N - - - - - ИЗБИРА ТРИМЕРЕН ОБЕКТ	71
OBJERASE	- - - - - В НАЧАЛОТО НА ТРИМЕРНИ ГРАФИЧНИ ПРОГРАМИ	71
OFF	- - - - - СЛЕДВАЩАТА DRAW-КОМАНДА НЕ ИЗТРИЕ ИЗБРАНИЯ ОБЕКТ	82
OR	M N - N - - N = M OR N ПОБИТОВО ИЛИ	20
ORMODE	- - - - - ПАЛИ ТОЧКИ НЕЗАВИСИМО ОТ ПРЕДИШНОТО ИМ СЪСТОЯНИЕ	57
OVER	M N - M N M КОПИРА 2-ПАТА СТОИНОСТ	9
PAD	- - - A - - А -АДРЕСА НА 120 БАЙТОВ БУФЕР	46
PEEK	A - - N - - PEEK BYTE ЧЕТЕ БАЙТ ОТ АДРЕС А	41

PEEKW	A - - N - - PEEK WORD ЧЕТЕ ЧИСЛО ОТ АДРЕС A	41
PICK	..M N - ..M N - КОПИРА N-ТАТА СТОИНОСТ	9
PLOT	X Y - - - - ИЗВЕЖДА ТОЧКАТА (X,Y)	53
POKE	N A - - - - POKE BYTE ЗАПОМНЯ БАЙТА НА АДРЕС A	41
POKEW	N A - - - - POKE WORD ЗАПОМНЯ ЧИСЛОТО НА АДРЕС A	41
POP	- - - - - МАХА ОТ ВЪЗВРАТНИЯ СТЕК	18
POSN	X Y - - - - POSITION ПОЗИЦИОНИРА ДАДЕНА ТОЧКА БЕЗ ДА Я ОТБЕЛЯЗВА	53
PREG	(ПРОМЕНЛИВА)	52
PRGTOP	- - - A - - PROGRAM TOP АДРЕС НА ВЪРХА НА БИБЛИОТ.	31
PRINT	- - - - - ИЗВЕЖДА СЛЕДВАЩИЯ ТЕКСТ	13,39
PULL	- - - N - - ВЪЗВРАТЕН СТЕК -> СТЕК	18
PUSH	N - - - - - СТЕК -> ВЪЗВРАТЕН СТЕК	18
PUTBLK	N - - - - - PUT BLOCK ИЗВЕЖДА СИМВОЛЕН БЛОК	68
PUTC	N - - - - - PUT CHARACTER ИЗВЕЖДА СИМВОЛ С КОД N	48
READ	- - - - - ЧЕТЕ И КОМПИЛИРА ФАЙЛ	38
READLN	A - - - - - READ LINE ЧЕТЕ ОТ КЛАВИАТ. НА АДРЕС A	44
REPEAT	- - - - - BEGIN ... WHILE ... REPEAT	26
RND	- - - N - - RANDOM СЛУЧАЙНО ЧИСЛО -32K ДО 32K	11
RNDVB	- - - N - - RANDOM BYTE СЛУЧАЙНО ЧИСЛО 0 ДО 255	11

RUN	- - - - -	50
	ИЗПЪЛНЯВА НАИ-ГОРНАТА ДЪМА	
SAVEPRG	- - - - -	50
	ЗАПАЗВА СИСТЕМАТА НА ДИСК	
SCALE	N - - - - - ИЗБИРА X И Y СКАЛИ ЗА ИЗБРАНИЯ ОБЕКТ	73
SCALX	N - - - - - X-СКАЛА (ШИРОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ	73
SCALY	N - - - - - Y-СКАЛА (ВИСОЧИНА) ЗА ИЗБРАНИЯ ОБЕКТ	73
SCALZ	N - - - - - Z-СКАЛА (ПЕРСПЕКТИВА) ЗА ОБЕКТА	74
SCREEN	N - - - - - N = [0!1] ИЗБОР НА ГРАФИЧЕН ЕКРАН (0,1)	82
SEQUENCE	N - - - - - N = [0!1] N=0: СПИРА СМЯНАТА НА ГРАФ. ЕКРАНИ	82
SGN	M - - - N - - - SIGN ЗНАК НА ЧИСЛОТО N=-1,0,1	11
SIN	M - - - N - - - N=SIN(M) СКАЛАРЕН СИНУС (-128,127)	11
SPACE	- - - - - SPACE ИЗВЕЖДА ПРАЗНО (КОД 160)	13,39
STACK	- - - - - ДИСПЛЕИ ИЛИ НЕ НА СТЕКА	7
STRING	- - - - - СЛЕДВА СЕ ОТ ИМЕТО НА НИЗА И СЕ ПРЕДХОЖДА ОТ БРОЯ СИМВОЛИ	43
SWAP	M N - - - N M - РАЗМЕСТВА ГОРНИТЕ 2 ЧИСЛА	9
TEXT	- - - - - ТЕКСТОВ РЕЖИМ	30
THEN	- - - - - IF...THEN КРАИ НА IF...THEN ФАЗА	21,22,27
UNBLK	- - - - - ИЗТРИВА СИМВОЛЕН БЛОК	68
UNDRAW	- - - - - ТРИЕ БЛОК И СПИРА АВТОМАТИЧНО ТРИЕНЕ ОТ СЛЕДВАЩИЯ DRAW	82

UNLINE	X Y - - - -	35
	ИЗТРИВА ЛИНИЯ ОТ ПОСЛЕДНАТА МАРКИРАНА ТОЧКА ДО (X,Y)	
UNPLOT	X Y - - - -	35
	ИЗТРИВА ТОЧКАТА (X,Y)	
UNTIL	N - - - - BEGIN...UNTIL	25
	ВРЪЩА КОГАТО N = 0	
VALID	- - - N - - N=0, #0	45
	0, АКО НЕ Е ЧИСЛО СЛЕД GETNUM	
VARIABLE	- - - - -	42
	ДЕКЛАРИРА ПРОМЕНЛИВА С ИМЕ ТОВА КОЕТО СЛЕДВА И ОПЦИОННО НАЧАЛНА СТОЙНОСТ ПРЕДИ НЕЯ	
VOICE	N - - - - -	85
	ИЗБИРА ТЕМБЪР (ОТ -6 ДО 2)	
VTAB	N - - - - - VERT. TAB	39
	ПОЗИЦИОНИРА N-ТИ РЕД	
WHILE	N - - - - -	26
	N=0 СЛЕД REPEAT, N#0 - WHILE	
WINDOW	L W T B - - - - LEFT WIDTH	39
	TOP BOTTOM	
	ТЕКСТОВ ПРОЗОРЕЦ	
WRITELN	A - - - - - WRITE LINE	44
	ПИШЕ ПО ЕКРАНА ОТ АДРЕС A	
XPOS	N - - - - - X-ПОЗИЦИЯ	74
	ВЪРХУ ЕКРАНА ЗА ОБЕКТА	
XREG	(ПРОМЕНЛИВА)	52
XROT	N - - - - - ИЗБИРА N	72
	ЗА ВЪРТЕНЕ ПО ОСТА X	
XTRAN	N - - - - - ТРАНСЛИРА	75
	С N ОБЕКТА ПО X-ОСТА	
YPOS	N - - - - - Y-ПОЗИЦИЯ	74
	ВЪРХУ ЕКРАНА ЗА ОБЕКТА	
YREG	(ПРОМЕНЛИВА)	52
YROT	N - - - - - ИЗБИРА N	
	ЗА ВЪРТЕНЕ ПО ОСТА Y	
YTRAN	N - - - - - ТРАНСЛИРА	75
	С N ОБЕКТА ПО ОСТА Y	
ZROT	N - - - - - ИЗБИРА N	72
	ЗА ВЪРТЕНЕ ПО ОСТА Z	

ZTRAN

N - - - - - ТРАНСПИРА
С N ОБЕКТА ПО ОСТА Z

75

ПРИЛОЖЕНИЕ С
СПРАВОЧНИК НА ДИСКЕТИТЕ.

Т ДЪЛ.	ИМЕ	ИЗТРИВАНЕ
=====		
B 006	ГРАФОРТ	НЕ
B 011	ГРЕДАКТОР1	ДА (64K)
B 011	ГРЕДАКТОР2	ДА (48K)
T 021	С.РЕДАКТОР	НЕ
T 024	Ф.РЕДАКТОР	НЕ
T 023	ОГЛЕД	НЕ
T 015	ПРОФИЛ	НЕ
T 004	КОСТЕНУРКА	НЕ
T 013	КОМП.МУЗ	НЕ
T 003	ИЗП.МУЗ	НЕ
T 008	ЗВУЦИ	ДА
T 004	СТРИНГ.ДУМИ	НЕ
A 002	A ***** СИМВОЛНИ НАБОРИ **	
B 005	CHR.SYS	ДА
B 005	CHR.CYR	ДА
B 005	CHR.CYS	НЕ
B 005	CHR.SLANT	ДА
B 005	CHR.CSLANT	ДА
B 005	CHR.SSLANT	ДА
B 005	CHR.STOP	ДА
B 005	CHR.CSTOP	ДА
B 005	CHR.SSTOP	ДА
B 005	CHR.BYTE	ДА
B 005	CHR.CBYTE	ДА
B 005	CHR.SBYTE	ДА
B 005	CHR.GOTHIC	ДА
B 005	CHR.EPS	ДА
A 002	A *** ИНФОРМАЦИОННИ ФАЙЛОВЕ **	
T 012	ПОМОЩ	НЕ
T 020	ЧИСЛА	НЕ
T 017	ПРЕХОДИ	НЕ
T 008	ВХОД/ИЗХОД	НЕ
T 011	ТЕКСТ	НЕ
T 007	ГРАФИКИ	НЕ
T 008	ГРАФИКИ.2	НЕ
T 006	ГРАФИКИ.С	НЕ
T 015	ГРАФИКИ.3	НЕ
T 008	ОБЩИ	НЕ
A 002	A ***** ДРУГИ ФАЙЛОВЕ *****	

Т ДЪЛ.	ИМЕ	ИЗТРИВАНЕ
Е 036	ГРАФОРТ	НЕ
В 011	ГФРЕДАКТОР1	ДА (64К)
В 011	ГФРЕДАКТОР2	ДА (48К)
Т 021	С.РЕДАКТОР	НЕ
Т 024	Ф.РЕДАКТОР	НЕ
Т 023	СГЛЕД	НЕ
Т 015	ПРОФИЛ	НЕ
Т 004	КОСТЕНЪРКА	НЕ
Т 013	КОМП.МУЗ	НЕ
Т 003	ИЗП.МУЗ	НЕ
Т 008	ЗВУЦИ	ДА
Т 004	СТРИНГ.ДУМИ	НЕ
А 002	А ***** СИМВОЛНИ НАБОРИ **	
В 005	CHR.SYS	ДА
В 005	CHR.CYR	ДА
В 005	CHR.CYS	НЕ
В 005	CHR.SLANT	ДА
В 005	CHR.CSLANT	ДА
В 005	CHR.SSLANT	ДА
В 005	CHR.STOP	ДА
В 005	CHR.CSTOP	ДА
В 005	CHR.SSTOP	ДА
В 005	CHR.BYTE	ДА
В 005	CHR.CBYTE	ДА
В 005	CHR.SBYTE	ДА
В 005	CHR.GOTHIC	ДА
В 005	CHR.SPS	ДА
В 005	CHR.MAXWELL	ДА
В 005	CHR.STUFF	ДА
А 002	А **** ГРАФОРТ ДЕМОНСТРАЦИИ **	
Т 002	ВЪПРОС	ДА
Т 018	НАЧАЛО	ДА
Т 008	МЕНЮ	ДА
Т 008	ГРАФИКА1	ДА
Т 010	ГРАФИКА2	ДА
Т 017	ГРАФИКА3	ДА
Т 012	ТЕКСТ.ДЕМО	ДА
Т 018	ФОРТ.ОПИСАНИЕ	ДА
Т 011	ПРИЛЕП	ДА
Т 011	ПИАНО	ДА
Т 003	ПЕСЕН1.ДЕМО	ДА
Т 009	ПОЛЕТ.ДЕМО	ДА
Т 004	ЧАСОВНИК	ДА
Т 003	БЪЛГАРИЯ.ДЕМО	ДА
Т 010	ЗВУЦИ.ДЕМО	ДА
А 002	А ***** ОБЕЛНИ ФИГУРИ ****	
В 002	ТЕТРАЕДЪР	ДА
В 002	ХУЗ	ДА
В 002	БАТ	ДА
В 002	КУБ	ДА
В 002	КЪЩА	ДА
В 010	ЕСКАЛ	ДА
В 012	ЕСКАЛ1	ДА

T 003	БОКАЛ2	ДА
B 002	СПЪТНИК	ДА
B 003	СОВАЛКА	ДА
B 002	К	ДА
A 002	A ***** ДРУГИ ФАЙЛОВЕ *****	
T 006	ПОЛЕТ	ДА
T 004	ПЕСЕН1	ДА
B 002	ПЕСЕН1.И	ДА
T 003	БЪЛГАРИЯ	ДА
B 002	БЪЛГАРИЯ.И	ДА