

PROGRAMY



INFORMATYKA
KOMPUTERY
SYSTEMY

Dodatek „Żołnierza Wolności” — zeszyty programów komputerowych Cena 80 zł

ORTOGRAFIA
JEZYK
KRZYŻÓWKA
M
OPTYKA
INŻYNIERIA
GRAFIKA
HANOI
MUS
ATRAKCJE
RULETKA
MUZYKA

W numerze:

AMSTRAD

- Prosta — str. 14
- Parabola — str. 15
- Elipsa — str. 16
- Hiperbola — str. 17
- Krzyżówka — str. 19
- Wieże Hanoi — str. 25

ATARI

- Mini-assembly — str. 27
- Test ortografii — str. 8

COMMODORE

- Muzyka 64 — str. 5

MERITUM

- Test z optyki — str. 22
- Ruch jednostajnie przyspieszony — str. 23
- Rzut poziomy — str. 23
- Rzut ukośny — str. 23
- Figury Lissajous — str. 24
- Test egzaminujący — str. 24

SHARP

- Komputerowa... ruletka — str. 3

SPECTRUM

- Program „Artysta” — str. 11

Okladka i rysunki: Michał Przybyłowski

Oddajemy do Waszych rąk pierwszy, specjalny numer „IKS-a”, poświęcony w całości programom komputerowym. Jak je wykorzystać?

Po pierwsze są one przeznaczone na konkretne typy mikrokomputerów. Wierzymy jednak, że wielu z naszych Czytelników potrafi je przetłumaczyć na potrzebny sobie dialekt Basica — a jeśli nie — pomoc zawsze znajdziecie w naszej redakcji.

Po drugie nasze programy są jedynie propozycjami rozwiązań. To tylko od użytkownika zależy jaką ostateczną postać otrzymają ich eksploatowane wersje — radzimy eksperymentować przede wszystkim z grafiką. Dotyczy to szczególnie programów edukacyjnych, które można rozbudować zgodnie z wymogami metodycznymi. Poza

tym będą one z pewnością częściej wykorzystywane przez uczniów na ich domowych komputerach niż w szkołach, w których jest zaledwie 3000 mikrokomputerów różnych typów.

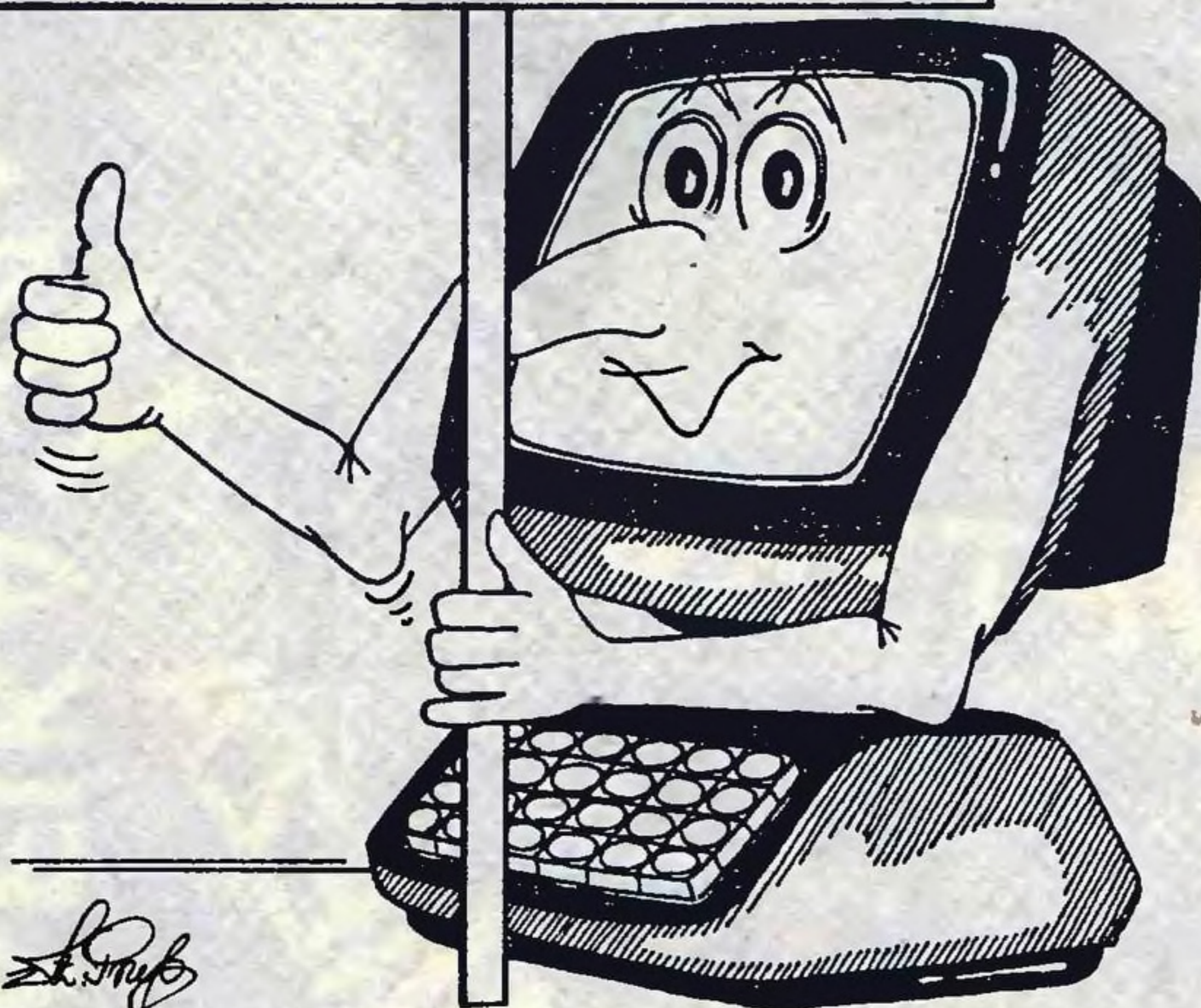
Po trzecie wprowadzanie do pamięci komputera długich programów z ich wydruków jest zajęciem żmudnym, łatwo popełnić błędy. Wiemy o tym. Ale UWAGA:

NIEKTÓRE Z PROGRAMÓW BĘDĄ EMITOWANE PRZEZ „RADIOKOMPUTER” — PROGRAM ROZGŁOSI HARCERSKIEJ POŚWIĘCONY MIKROINFORMATYCE. Które i kiedy — dowiedzie się w kolejnych audycjach: WTOREK, program IV, godz. 22.50.

A teraz życzymy już przyjemnej zabawy.

REDAKCJA

**DZIŚ WYŁĄCZNIE
PROGRAMY!**



Prezentujemy 22 uznane za najlepsze komputery w 1986 roku.

1. Bondwell BW 2

Mikroprocesor: Z80L, 4MHz

System operacyjny: CP/M2,2

Pamięć RAM: 64 KB

Pamięć masowa: RAM-Disc 256 lub
512 KB, Floppy-Disc
360 KB

Grafika: 640 X 200

Zasilanie: akumulator

Rozmiary: 28,4 X 7,8 X 31 cm

Waga: 3 kg

Cena: ok. 2400 marek



SHARP

PROGRAM

1

KOMPUTEROWA... RULETKA

Proponujemy Wam taką ruletkę, w której „bankierem” jest komputer, co nam daje *pewność*, że nie stracimy majątku.

W rzetelnej ruletce każdy numer jest jednakowo prawdopodobny. U nas zapewnia to *obiektywny* generator liczb losowych.

Program wywołuje etykieta „A” (RUN 40). Najpierw „kupujemy” taliony (tzw. „sztony”), potem zaś określamy wysokość stawki.

Typujemy:

- pojedyncze numery od 1 do 36,
- trójki numerów od 1 do 12,
- szóstki od 1 do 6 oraz
- tuziny od 1 do 3.

Oprócz tego możemy typować „połówki” typu:

- *parzyste* lub *nieparzyste*,
- *małe* (od 1 do 18) lub *duże* (od 19 do 36).

Ich rozmieszczenie podaje tablica: Gdy „coś” trafimy, to *wygrana* jest *krotnością* stawki:

- dla numerów mnożnik 36,
- dla trójki x 12,
- dla szóstki x 6,
- dla tuzina x 3.

Trafny wybór połówki daje wygraną w postaci podwojonej stawki. Jeśli przegramy, tracimy (niestety) stawkę.

Po zakończeniu typowania „rusza kuleczka”, co symbolizuje hasło RIEN NE VA PLUS (koniec typowania, zaś dosłownie „*nic więcej nie idzie*”).

Komputer wyświetla wygrany



numer (przypadkowy), a potem... oblicza. Rezultat jest podany pod postacią komunikatu: *wygrałeś... lub przegrałeś*. Jeśli zostanie wylosowane *zero*, to *wszystkie stawki zagarnia bankier*. Jest to jedyny legalny zysk kasyna.

Typować można *jedną, dwie lub wszystkie kombinacje jednocześnie*. Życzymy *rozbicia banku*.

```

40: "A": CLEAR :
    RANDOM : USING
    : WAIT 0: B=2E4
50: INPUT "KUP TAL
    ONY 100<T<10 0
    00 "; Z
60: IF Z<1000R Z>1
    E4 THEN BEEP 3,
    100, 500: GOTO 5
    0
70: INPUT "STAWKA
    OD 1 DO 500 ";
    Y
80: IF Y<10R Y>500
    GOTO "D"
90: INPUT "NUMER 0
    D 1 DO 36 "; N:
    IF N>36GOTO "D
    "
100: INPUT "TROJKA
    OD 1 DO 12 "; C
    : IF C>12GOTO "
    D"
110: INPUT "SZOSTKA
    OD 1 DO 6 "; S
    : IF S>6GOTO "D
    "
120: INPUT "TUZIN 0
    D 1 DO 3 "; D:
    IF D>3GOTO "D"
130: INPUT "PARZYST
    E ? T=1 N=0 ";
    P: IF P=1GOTO 1
    50
    
```

WYGRANE
(mnożniki
stawek)

Numer x 36

Trójka x 12

Szóstka x 6

Tuzin x 3

x 2 dla:

Parzyste

Nieparzyste

Poniżej 19

Powyżej 18

		0 BANKIERSKIE 0							
		Trójka 1			Trójka 2				
TUZIN 1	1	2	3	4	5	6	SZÓSTKA 1		1, 2, 18 19, 20, 36
		Trójka 3			Trójka 4			SZÓSTKA 2	
TUZIN 2	7	8	9	10	11	12	SZÓSTKA 3		
		Trójka 5			Trójka 6			SZÓSTKA 4	
TUZIN 3	13	14	15	16	17	18	SZÓSTKA 5		
		Trójka 7			Trójka 8			SZÓSTKA 6	
TUZIN 3	19	20	21	22	23	24	SZÓSTKA 5		
		Trójka 9			Trójka 10			SZÓSTKA 6	
TUZIN 3	25	26	27	28	29	30	SZÓSTKA 5		
		Trójka 11			Trójka 12			SZÓSTKA 6	
TUZIN 3	31	32	33	34	35	36	SZÓSTKA 6		

```

140: INPUT "NIEPARZ
    YSTE ? T=1 N=0
    "; R: IF (P+R)>
    1GOTO "D"
150: INPUT "1, 2... 1
    8 ? T=1 N=0 ";
    M: IF M=1GOTO 1
    70
160: INPUT "19, 20..
    .36 ? T=1 N=0
    "; O: IF (M+O)>1
    GOTO "D"
170: WAIT 0: PRINT "
    RIEN NE VA PLU
    S": BEEP 36, 60,
    10
    
```

```

180: X=(RND 37)-1:
    PRINT "N U M E
    R *"; X; " *"
190: BEEP 3, 200, 100
    0: IF X>0GOTO 2
    30
200: IF X=0 THEN
    PRINT "BANKIER
    SKIE ZERO":
    BEEP 3, 200, 100
    0
210: Q=SGN N+SGN C+
    SGN S+SGN D+
    SGN P
    
```

```

220: Q=Q+SGN R+SGN
      M+SGN O: B=B+Q*
      Y: W=-Q*Y: GOTO
      390
230: IF X<>NAND N>0
      GOSUB "C"
240: IF X=NLET F=36
      : GOSUB "B"
250: Q=INT ((X-1)/3
      )+1: IF Q<>CAND
      C>0GOSUB "C"
260: IF Q=CLET F=12
      : GOSUB "B"
270: Q=INT ((X-1)/6
      )+1: IF Q<>SAND
      S>0GOSUB "C"
280: IF Q=SLET F=6:
      GOSUB "B"
290: Q=INT ((X-1)/1
      2)+1: IF Q<>D
      AND D>0GOSUB "
      C"
300: IF Q=DLET F=3:
      GOSUB "B"

```

```

310: Q=X-INT (X/2)*
      2: IF (1+Q)*R=1
      GOSUB "C"
320: F=2: IF (1+Q)*P
      =1GOSUB "B"
330: IF Q*P=1GOSUB
      "C"
340: IF Q*R=1GOSUB
      "B"
350: IF M=1AND X<19
      GOSUB "B"
360: IF O=1AND X<19
      GOSUB "C"
370: IF M*X>18GOSUB
      "C"
380: IF O*X>18GOSUB
      "B"
390: W$="WYGRALES":
      IF W<0LET W$="
      PRZEGRALES"
400: WAIT :PRINT W$
      ;ABS W:Z=Z+W:
      IF Z<0GOTO 460
410: IF B<100THEN
      PRINT "BANK RO
      ZBITY":END

```

```

420: PRINT "MASZ":Z
      ;" TALONOW"
430: INPUT "GRASZ ?
      T=1 N=0 ";U:
      IF U=0THEN END
440: N=0: C=0: D=0: P=
      0: R=0: M=0: F=0
450: Q=0: Y=0: W=0: O=
      0: WAIT 0: GOTO
      70
460: PRINT "MASZ DL
      UG":ABS Z;:" T
      ALONOW":END
470: "B":W=W+F*Y: B=
      B-F*Y: RETURN
480: "C":W=W-Y: B=B+
      Y: RETURN
490: "D":PRINT "U W
      A Z A J !":
      BEEP 3, 100, 500
      :GOTO 70

```

JANUSZ MILLER



2. Bondwell BW 8

Mikroprocesor: 80C88, 4,77 MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 512 KB
Pamięć masowa: Floppy-Disc 720 lub 360 KB
Grafika: 640 X 200
Zasilanie: Sieć lub akumulator
Rozmiary: 28 X 7 X 30 cm
Waga: 4,9 kg
Cena: 3990 marek

3. Casio BP-1000

Mikroprocesor: Casio-eigen
System operacyjny: C61 BASIC
Pamięć RAM: 8—40 KB
Pamięć masowa: Floppy-Disc (3,5) 320 KB
Grafika: 192 X 32
Zasilanie: bateria
Rozmiary: 18,4 X 2,4 X 9,7 cm
Waga: 437 g
Cena: 1400 marek (bez dyskietek)



4. Datavue 25 II

Mikroprocesor: 80C88, 4,77MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 768 KB lub 1.2 MB
Pamięć masowa: 1 lub 2 Floppy-Disc po 360 KB
Grafika: 640x200, 320x200
Zasilanie: sieć lub akumulator
Waga: 5.5 kg
Cena: 8660 M

5. Epson HX-20

Mikroprocesor: 6301, 2.45MHz
System operacyjny: MBASIC
Pamięć RAM: 16 lub 32 KB
Pamięć masowa: mikrokaseta — 30 KB
Grafika: 120x32
Zasilanie: akumulator
Rozmiary: 29x4.4x21.5 cm
Waga: 1.6 kg
Cena: 1800 M



COMMODORE

PROGRAM 2

MUZYKA 64

Commodore-64 posiada duże możliwości muzyczne. Niestety oferowany Basic nie zawiera żadnych instrukcji (nie licząc POKE), które ułatwiałyby programowanie muzyki. Istniejące na rynku oprogramowanie muzyczne nie zmienia tej sytuacji, gdyż realizowane za ich pomocą kompozycje nie dają się wykorzystać w naszych własnych programach.

MUZYKA 64 może być traktowana jako autonomiczny program do tworzenia różnorodnych kompozycji lub też jako narzędzie programistyczne wspomagające realizację efektów dźwiękowych w naszych własnych programach. Osoby uczące się muzyki będą mogły wykorzystać go jako program edukacyjny.

Zasady postępowania się programem MUZYKA 64 zilustrowane są na przykładzie fragmentu utworu „NOBODY I KNOW” zespołu THE BEATLES.

MUZYKA 64 pozwala budować kompozycje na trzy głosy. W utworze muzycznym, który chcemy zaprogramować odszukujemy najkrócej trwający dźwięk (pauzę) — posłuży on jako jednostka czasu. Każdy dłużej trwający dźwięk wyrażany jest wielokrotnością podstawowej jednostki czasu. W przedstawionym fragmencie jest to ósemka. Programując dźwięki posługujemy się zamieszczoną tabelą kodową przyporządkowującą każdemu dźwiękowi dwie liczby (mniejszą i większą). Wybrane kody dźwięków wprowadzamy do programu w postaci danych typu DATE. W przedstawionym przykładzie zaprogramowanej muzyki (wiersze 349—670) każdy wiersz odpowiada jednej jednostce czasu i zawiera trzy dźwięki. Kompozycje kończymy wprowadzając wiersz zawierający trzy zera (wiersz 670).

W celu uzyskania zamierzonych efektów muzycznych należy korzystać z następujących reguł:

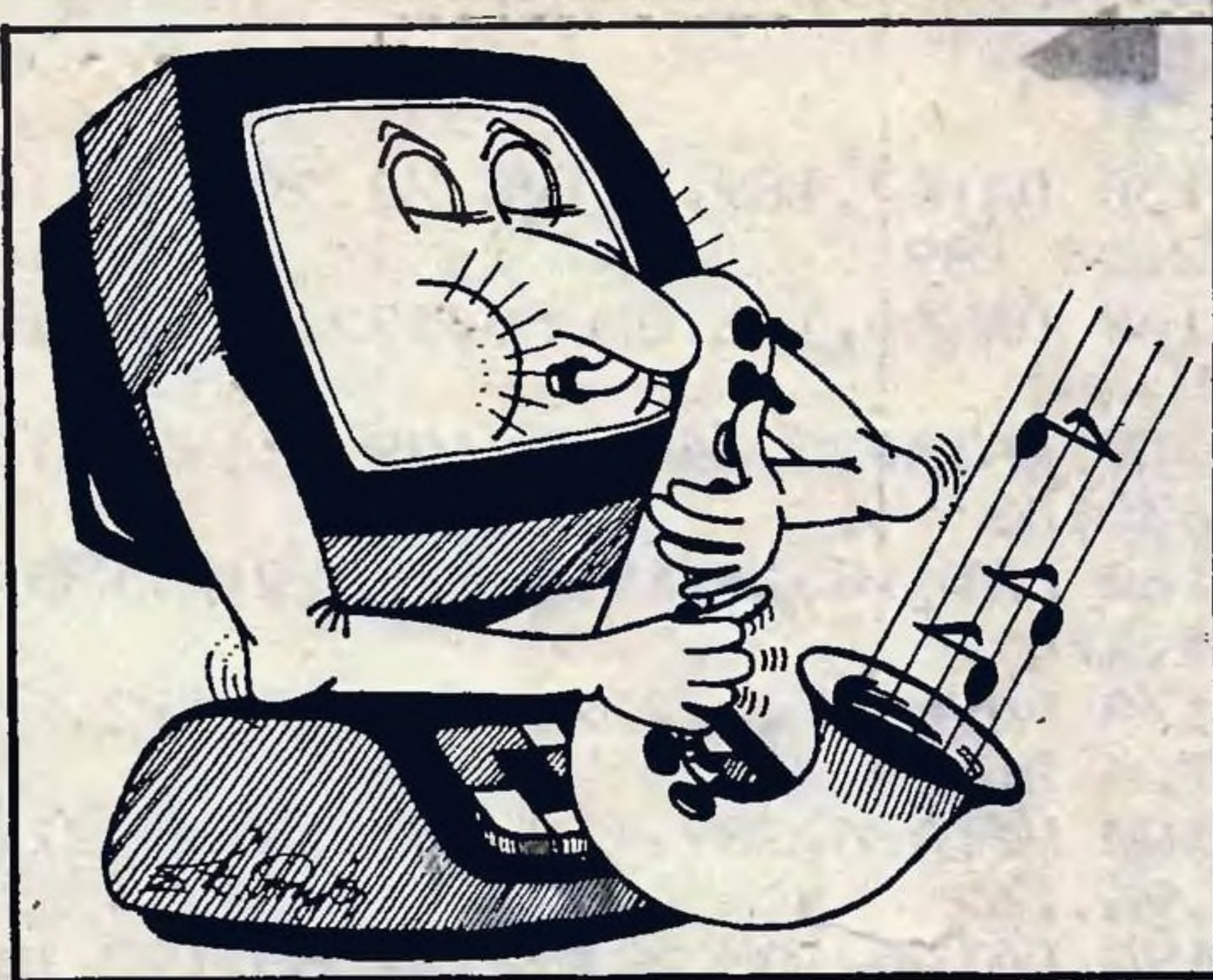
— kod o większej wartości używamy, gdy aktualnie programowany dźwięk ma być kontynuowany w następnej jednostce czasowej w tym samym głosie;

— kod o mniejszej wartości używamy, gdy aktualnie programowany dźwięk nie będzie kontynuowany w następnej jednostce czasowej w tym samym głosie;

— pauzę lub wyłączenie jakiegoś głosu w bieżącej jednostce czasowej otrzymujemy za pomocą kodu 192.

Program składa się z dwóch podprogramów oddzielnie wprowadzanych do pamięci mikrokomputera. Pierwszy z programów (program pomocniczy) inicjuje procedury zapisane w kodzie maszynowym, automatycznie ładuje z taśmy następny program i uruchamia go. Drugi program (program grający) zawiera sekwencje instrukcji realizujących odtworzenie dołączonej do niego kompozycji zapisanej w wierszach DATE. Jeśli chcemy wykorzystać właściwości programu MUZYKA 64 w naszym własnym programie musimy:

— zapisać na taśmie jako pierwszy program pomocniczy,



— wprowadzić do pamięci mikrokomputera program grający i stworzyć z niego procedurę wywoływaną w naszym własnym programie,

— dopisać własny program i zakodowaną kompozycję.

Przed uruchomieniem obydwu programów, składających się na program MUZYKA 64 należy stworzyć ich kopie.

```
5 REM ***** MUZYKA 64 *****
PROGRAM POMOOCNICZY *****
10 PRINT ".....PROSZE CZEKAC" :
SK=0
20 FOR I=1 TO 61 : READ W, N : POKE 49664 +
I, W : POKE 49920 + I, N : SK=SK+W+N : NEXT
30 DATA 4, 48, 4, 113, 4, 180, 4, 252, 5,
72, 5, 152, 5, 237, 6, 71, 6, 167, 7, 12
40 DATA 7, 119, 7, 233, 8, 97, 8, 225, 9,
104, 9, 247, 10, 143, 11, 48, 11, 218, 12
, 143
50 DATA 13, 78, 14, 24, 14, 239, 15, 210
, 16, 195, 17, 194, 18, 208, 19, 239, 21,
30
60 DATA 22, 96, 23, 180, 25, 29, 26, 156
, 28, 49, 29, 222, 31, 164, 33, 134, 35, 1
33
70 DATA 37, 161, 39, 222, 42, 61, 44, 19
2, 47, 105, 50, 59, 53, 55, 56, 97, 59, 18
8
80 DATA 63, 73, 67, 13, 71, 9, 75, 66, 79
, 188, 84, 122, 89, 127, 94, 21, 100, 117
90 DATA 106, 111, 112, 195, 119, 119, 1
26, 146, 134, 24
100 IFSK<>102191 THEN PRINT "BLAD W
DANYCH, LINIE: 30 - 90" : STOP
110 SK=0 : FOR I=49152 TO 49252 : READ W
: POKE I, W : SK=SK+W : NEXT
120 DATA 162, 0, 189, 0, 193, 32, 25, 19
2, 162
```



```

130 DATA7,189,0,193,32,25,192,16
2,14,189
140 DATA0,193,32,25,192,96,41,12
8,201
150 DATA128,240,25,189,0,193,41,
63,168
160 DATA185,0,195,157,0,212,185,
0,194
170 DATA157,1,212,189,113,192,15
7,4,212
180 DATA96,189,112,192,157,4,212
,96,162
190 DATA0,189,0,193,32,88,192,16
2,7
200 DATA189,0,193,32,88,192,162,
14,189
210 DATA0,193,32,88,192,96,41,64
,201,64
220 DATA240,6,189,114,192,157,4,
212,96
230 IFSK(>1160)THENPRINT"BLAD W
DANYCH, LINIE: 120 - 220":STOP
240 POKE631,76:POKE632,111:POKE6
33,13:POKE634,82:POKE635,117:POK
E636,13:POKE198,6:END

```

```

5 REM ***** MUZYKA 64 *****
PROGRAM GRAJACY *****
10 INPUT"TEMPO":T:S=54272
20 FORI=0TO24:POKES+I,0:NEXT:POK
ES+5,35:POKES+6,196:POKES+12,35
30 POKES+13,196:POKES+19,35:POKE
S+20,196,POKES+3,8:POKES+2,0:POK
ES+10,8
40 POKES+9,0:POKES+17,8:POKES+16
,0:POKES+24,15
50 FORI=0TO14STEP7:POKE49264+I,7
2:POKE49265+I,65:POKE49266+I,64:
NEXT
60 READG1,G2,G3:POKE49408,G1:POK
E49415,G2:POKE49422,G3:IFG1=0THEN
80
70 SYS49152:FORI=0TO50:NEXT:SYS49
215:FORI=0TO50:NEXT:GOTO60
80 POKES+4,64:POKES+11,64:POKES+
18,64:FORI=0TO50:NEXT:POKES+24,0
:END

```

DZWIEK	CZESTOTLIWOSC	KOD
C	65	1 65
C#	69	2 66
D	73	3 67
D#	78	4 68
E	82	5 69
F	87	6 70

F#	93	7	71
G	98	8	72
G#	104	9	73
A	110	10	74
A#	117	11	75
H	123	12	76
C	131	13	77
C#	139	14	78
D	147	15	79
D#	157	16	80
E	165	17	81
F	175	18	82
F#	185	19	83
G	196	20	84
G#	208	21	85
A	220	22	86
A#	233	23	87
H	246	24	88
* C	262	25	89
C#	277	26	90
D	294	27	91
D#	311	28	92
E	330	29	93
F	349	30	94
F#	370	31	95
G	392	32	96
G#	415	33	97
A	440	34	98
A#	466	35	99
H	494	36	100
C	523	37	101
C#	554	38	102
D	587	39	103
D#	622	40	104
E	659	41	105
F	698	42	106
F#	740	43	107
G	784	44	108
G#	831	45	109
A	880	46	110
A#	932	47	111
H	988	48	112
C	1047	49	113
C#	1109	50	114
D	1175	51	115
D#	1246	52	116
E	1319	53	117
F	1397	54	118
F#	1480	55	119
G	1568	56	120
G#	1661	57	121
A	1760	58	122
A#	1865	59	123
H	1976	60	124
C	2093	61	125

* SROTKOWE C

349 REM **NORODY I KNOW**

350 DATA 49, 77, 93
355 DATA 48, 77, 93
360 DATA 49, 192, 93
365 DATA 48, 192, 93
370 DATA 113, 77, 93
375 DATA 113, 77, 29
380 DATA 105, 192, 93
385 DATA 105, 192, 93
390 DATA 106, 79, 192
395 DATA 106, 79, 192
400 DATA 110, 192, 192
405 DATA 110, 192, 192
410 DATA 108, 84, 192
415 DATA 108, 84, 192
420 DATA 42, 192, 192
425 DATA 44, 192, 192
430 DATA 108, 77, 192
435 DATA 108, 77, 192
440 DATA 105, 192, 192
445 DATA 105, 192, 192
450 DATA 105, 84, 192
455 DATA 105, 84, 192
460 DATA 105, 192, 192
465 DATA 105, 192, 192

470 DATA 115, 79, 106
475 DATA 115, 79, 106
480 DATA 113, 79, 106
485 DATA 113, 79, 42
490 DATA 110, 84, 106
495 DATA 110, 84, 106
500 DATA 108, 84, 106
505 DATA 108, 84, 106
510 DATA 49, 77, 93
515 DATA 48, 77, 93
520 DATA 49, 192, 93
525 DATA 48, 192, 93
530 DATA 113, 77, 93
535 DATA 113, 77, 29
540 DATA 105, 192, 93
545 DATA 105, 192, 93
550 DATA 106, 87, 103
555 DATA 106, 87, 103
560 DATA 110, 192, 103
565 DATA 110, 192, 39
570 DATA 108, 84, 103
575 DATA 108, 84, 103

580 DATA 105, 192, 105
585 DATA 105, 192, 105
590 DATA 103, 79, 103
595 DATA 103, 15, 103
600 DATA 103, 79, 103
605 DATA 103, 15, 103
610 DATA 103, 79, 103
615 DATA 103, 15, 103
620 DATA 103, 79, 103
625 DATA 103, 15, 103
630 DATA 103, 84, 103
635 DATA 103, 20, 103
640 DATA 103, 84, 103
645 DATA 103, 20, 103
650 DATA 103, 84, 103
655 DATA 103, 20, 103
660 DATA 103, 84, 103
665 DATA 103, 20, 103
670 DATA 0, 0, 0

S. WASILEWSKI



Fragmenty utworu „Nobody I Know”

ATARI

PROGRAM 3

TEST ORTOGRAFII

Program „TEST ORTOGRAFII” jest kolejnym programem edukacyjnym publikowanym w „IKS-ie” na komputery ATARI.

Po uruchomieniu programu, na ekranie pojawi się MENU. Na początku użytkownika tego programu, należy najpierw wpisać do komputera wyrazy, które sprawiają nam trudności w pisaniu. (UWAGA — wyrazy te powinny być wpisane oczywiście poprawnie). Maksymalnie można ich wpisać 30 000. Program ten umożliwia używanie polskich liter: ą,ę,ć,ł,ó,ś,ż,ź. Uzyskuje się je przez naciśnięcie odpowiednio: CONTROL-A, CONTROL-E itd. Literę ż uzyskamy przez naciśnięcie CONTROL-”,”.

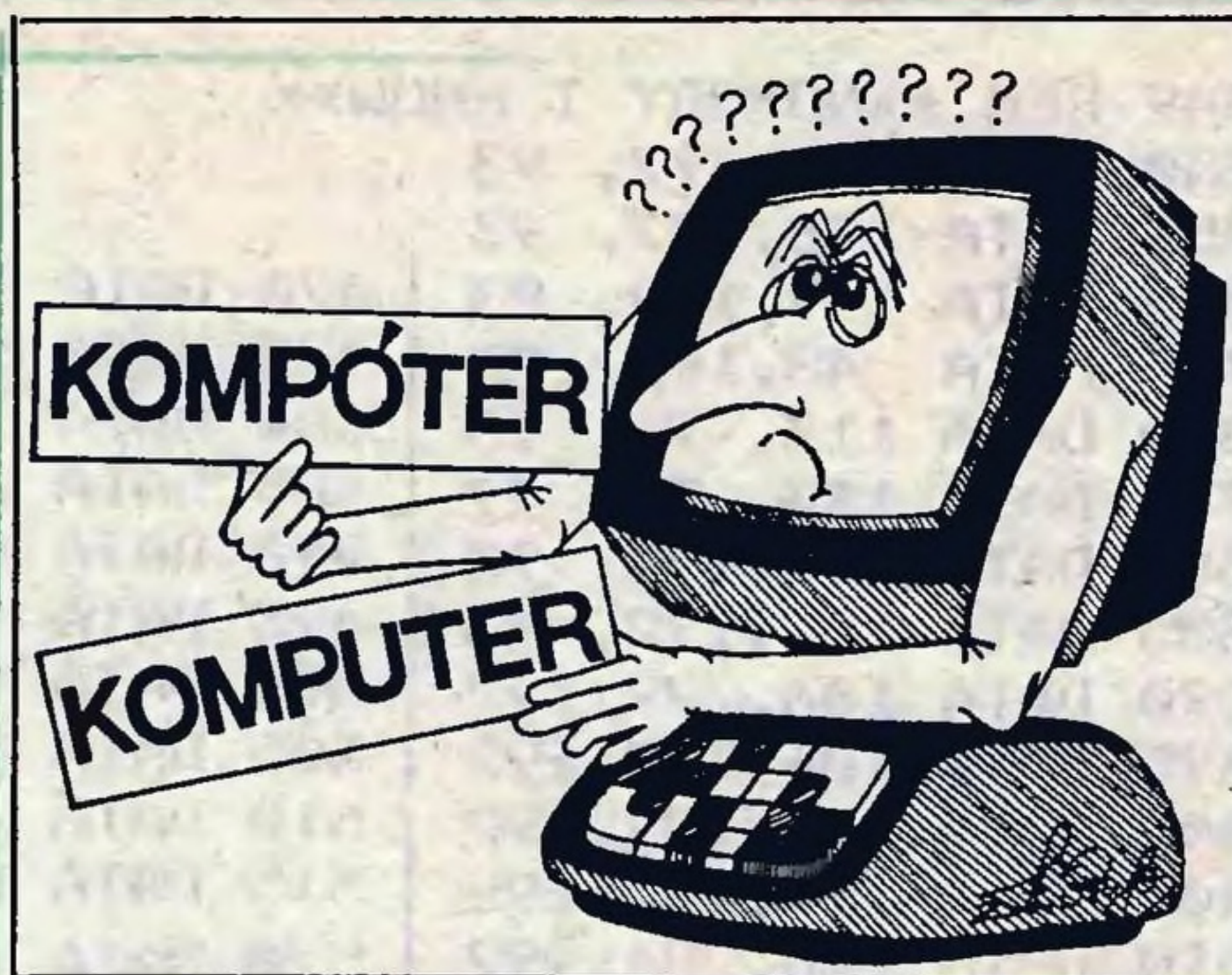
Po wpisaniu do komputera wyrazów, wybierając odpowiednie opcje o MENU, można przystąpić do testu lub nagrać wpisane do komputera wyrazy na taśmę magnetofonową. Zarejestrowane na taśmie wyrazy, przy późniejszym korzystaniu z programu można bezpośrednio wpisać do komputera.

Gdy przystąpimy do testu na ekranie będzie pokazywał się wyraz. Jeżeli unamy, że jest on napisany dobrze, to wciśniemy klawisz SPACE, jeżeli źle — dowolny klawisz (oprócz BREAK). Gdy nasza decyzja będzie poprawna, z głośnika wydobędzie się pojedynczy dźwięk, jeżeli zła — potrójny. Oprócz tego, na ekranie znajduje się licznik dobrych i złych odpowiedzi oraz zegar. Czas podawania odpowiedzi jest ograniczony. Jeżeli zegar osiągnie wartość 500, odpowiedź zostanie uznana za złą. Życzę owocnej nauki i miłej zabawy.

```

10 CLR
20 ? CHR$(125):? "Prosze czeka
c..."
30 OPEN #1,4,0,"K":GOSUB 1610
40 DIM A$(15),B$(18)
50 DL=PEEK(560)+PEEK(561)*256
60 POKE DL+6,7:POKE DL+7,6
70 ? CHR$(125):POKE 710,0:TRAP
100
80 SETCOLOR 1,10,10
90 RESTORE 1999:READ DAT
100 POSITION 8,3
110 ? "1 - TEST ORTOGRAFII"
120 POSITION 8,6:POKE 752,6
130 ? "2 - WPIS NOWYCH WYRAZOW
"
140 POSITION 8,9
150 ? "3 - WPIS WYRAZOW Z MAGN
"
160 POSITION 8,12

```



```

170 ? "4 - ZAPIS WYRAZOW NA MA
GN."
180 TRAP 180:GET #1,EE:E=EE-48
190 ON E GOTO 210,1040,1420,12
60
200 GOTO -9
210 LICZ=-1:MAX=500:DOB=0:ZL=0
220 ? CHR$(125):POKE 710,0:TRA
P 70
230 GOSUB 920
240 POSITION 1,5:?" ODPOWIED
ZI:"
250 POSITION 1,7:?" Dobre:
Ze:"
260 A=INT(RND(0)*DAT)+2000:M=0
270 POSITION 3,22
280 ? " SELECT - Powrt do ME
NU"
290 POSITION 4,3:?"CZAS :
"
300 LICZ=LICZ+1:POSITION 3,20
310 POSITION 23,3:?"Licznik :
";LICZ
320 RESTORE A:READ A$
330 FOR I=1 TO LEN(A$):TRAP 40
000
340 IF I>LEN(A$) THEN 500
350 IF A$(I,I)="" THEN 740
360 IF A$(I,I)="u" THEN 770
370 IF A$(I,I)="" THEN 800
380 IF A$(I,I)="h" THEN 860
390 IF LEN(A$)=I THEN 420
400 IF A$(I,I+1)="rz" THEN 830
410 IF A$(I,I+1)="ch" THEN 890
420 NEXT I
430 GOTO 500
440 LICZ=LICZ-1:GOTO 260
450 H=LEN(A$)-1:IF H=0 THEN RE
TURN
460 FOR HH=1 TO H
470 IF A$(HH,HH+1)="cc" THEN 2
60

```

```

480 NEXT HH
490 RETURN
500 GOSUB 450
510 POSITION 20,15: ? A$; "
"
520 POKE 764,255:LL=0
530 P=PEEK(764):LL=LL+1:V=V+1
540 SETCOLOR 0,5,V:IF V>100 TH
EN V=1
550 IF LL>MAX THEN 680
560 POSITION 10,3: ? LL
570 IF PEEK(53279)=5 THEN 70
580 IF P=255 THEN 530
590 IF P=33 AND M=0 THEN 640
600 IF P<>33 AND M=3 THEN 640
610 IF P=33 AND M=3 THEN 680
620 IF P<>33 AND M=0 THEN 680
630 STOP
640 FOR F=1 TO 50: SOUND 1,30,1
0,15
650 NEXT F: SOUND 1,0,0,0
660 DOB=DOB+1: POSITION 4,9: ? D
OB
670 GOTO 260
680 FOR FF=1 TO 3: FOR F=50 TO
100
690 SOUND 1,F,12,15: NEXT F
700 SOUND 1,0,0,0: FOR F=1 TO 3
0
710 NEXT F: NEXT FF
720 ZL=ZL+1: POSITION 12,9: ? ZL
730 GOTO 260
740 K=INT(RND(0)*5)
750 IF K=1 OR K=4 THEN A$(I,I)
="u": M=3
760 GOTO 420
770 K=INT(RND(0)*7)
780 IF K=3 OR K=6 THEN A$(I,I)
="": M=3
790 GOTO 420
800 K=INT(RND(0)*5)
810 IF K=1 OR K=3 THEN GOSUB 9
40
820 GOTO 420
830 K=INT(RND(0)*7)
840 IF K=2 OR K=5 THEN GOSUB 9
70
850 GOTO 420
860 K=INT(RND(0)*9)
870 IF K=5 OR K=8 THEN GOSUB 9
90
880 GOTO 420
890 K=INT(RND(0)*8)
900 IF K=2 OR K=5 THEN GOSUB 1
020
910 GOTO 420

```

```

920 POSITION 3,1: ? "TEST ORTOG
RAFII"
930 SETCOLOR 0,5,15: RETURN
940 B$=A$(I,LEN(A$))
950 A$(I+1,LEN(A$)+1)=B$
960 A$(I,I+1)="rz": M=3: RETURN
970 A$(I)=A$(I+1): M=3
980 A$(I,I)="": RETURN
990 B$=A$(I,LEN(A$))
1000 A$(I+1,LEN(A$)+1)=B$
1010 A$(I,I+1)="ch": M=3: RETURN
1020 A$(I)=A$(I+1): M=3
1030 A$(I,I)="h": RETURN
1040 ? CHR$(125): POKE 710,0: PO
KE 702,0
1050 POKE 752,0: POSITION 10,3
1060 ? " WPIS WYRAZOW ": GOSUB
920
1070 POSITION 4,20
1080 ? " RETURN --- MENU"
1090 POSITION 3,15: ? " UWAGA "
: ?
1100 ? "WYRAZ WPISUJ MALYMI LI
TERAMI"
1110 POSITION 5,8: ? "PODAJ WYR
AZ",
1120 INPUT A$: IF A$="" THEN 70
1130 ? CHR$(125): POSITION 2,4
1140 SETCOLOR 1,0,0: ? DAT+2000
;"D. "; A$
1150 ? : ? "G.1180"
1160 POSITION 0,2: POKE 842,1
1170 STOP
1180 POKE 842,12: DAT=DAT+1
1190 ? CHR$(125): POSITION 2,4
1200 ? 1999;"D. "; DAT
1210 ? : ? "G.1240"
1220 POSITION 0,2: POKE 842,1
1230 STOP
1240 POKE 842,12: SETCOLOR 1,10
,10
1250 GOTO 1040
1260 ? CHR$(125): POKE 710,0
1270 GOSUB 920: POSITION 7,5
1280 ? " ZAPIS WYRAZOW NA MAGN
ETOFON "
1290 POKE 702,64: POSITION 9,16
1300 ? "CZY JESTES PEWIEN ? [T
/N]"
1310 GET #1,EE
1320 IF EE=78 THEN 70
1330 IF EE=84 THEN 1350
1340 GOTO 1310
1350 RESTORE 1999: OPEN #2,9,0,
"C:"
1360 READ A$

```



```

1370 ? #2;A$
1380 IF A$="*+*" THEN CLOSE #2
:GOTO 1400
1390 GOTO 1360
1400 POSITION 12,10:? "O.K"
1410 FOR F=1 TO 300:NEXT F:GOT
0 70
1420 ? CHR$(125):POKE 710,0
1430 GOSUB 920:POSITION 7,4
1440 ? " WPIS WYRAZOW Z MAGNET
OFONU "
1450 POKE 702,64:POSITION 9,16
1460 ? "CZY JESTES PEWIEN ? [T
/N]"
1470 GET #1,EE
1480 IF EE=78 THEN 70
1490 IF EE=84 THEN 1510
1500 GOTO 1470
1510 SETCOLOR 1,0,0
1520 ? "}" :OPEN #2,5,0,"C:" :P=
1998
1530 INPUT #2;A$
1540 P=P+1:POSITION 2,4:? P;"D
." ;A$
1550 ? :? "G.1580":POSITION 0,
2
1560 POKE 842,1
1570 STOP
1580 POKE 842,12:? CHR$(125)
1590 IF A$="*+*" THEN CLOSE #2
:GOTO 70
1600 GOTO 1530
1610 REM GEN. POLSKICH LITER
1620 N=57344:M=32768
1630 FOR I=0 TO 1023:POKE M+I,

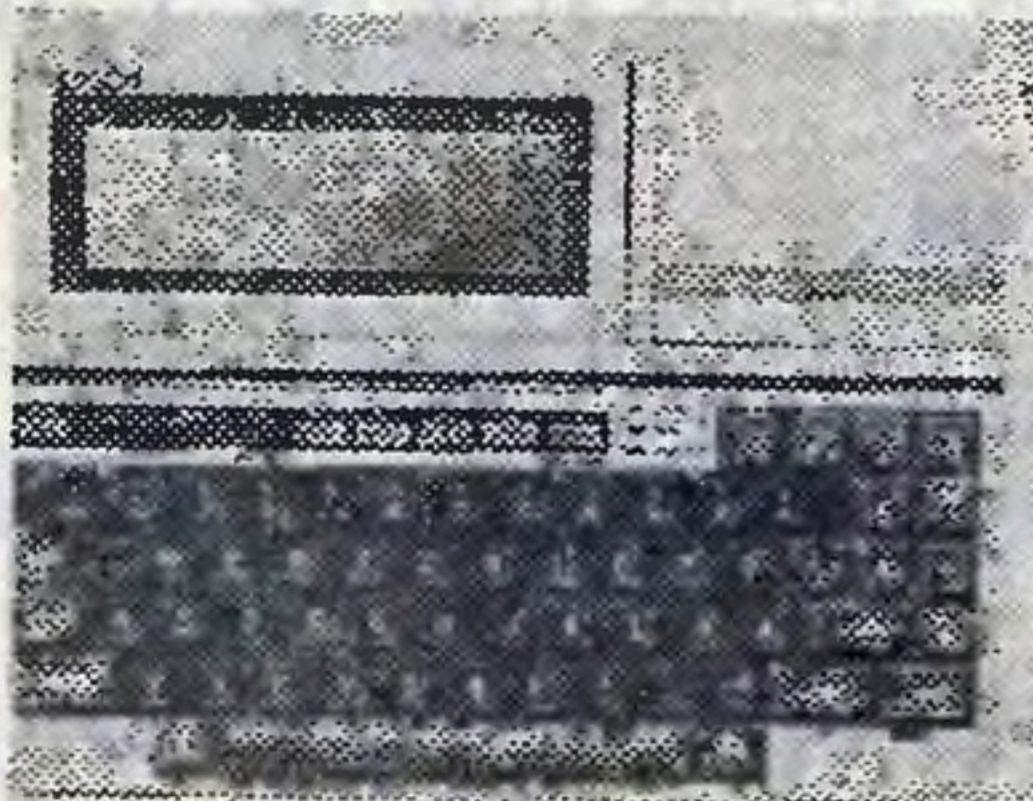
```

```

PEEK(N+I)
1640 NEXT I
1650 A=M+776:B=M+520
1660 FOR I=0 TO 207:POKE B+I,P
EEK(A+I)
1670 NEXT I
1680 A=M+976:B=M+512
1690 FOR I=0 TO 7:POKE B+I,PEE
K(A+I)
1700 NEXT I
1710 POKE M+513,12:POKE M+514,
24
1720 POKE M+515,124:POKE M+527
,3
1730 POKE M+537,12:POKE M+538,
24
1740 POKE M+539,60:POKE M+559,
6
1750 POKE M+611,28:POKE M+612,
56
1760 POKE M+625,12:POKE M+626,
24
1770 POKE M+627,124:POKE M+633
,12
1780 POKE M+634,24:POKE M+635,
60
1790 POKE M+664,6:POKE M+665,1
2
1800 POKE M+721,24:POKE M+722,
0
1810 POKE M+723,124:POKE 756,1
28
1820 RETURN
1999 DATA 0
31999 DATA *+*

```

OLGIERD NIEMYJSKI
LVI LO—Warszawa

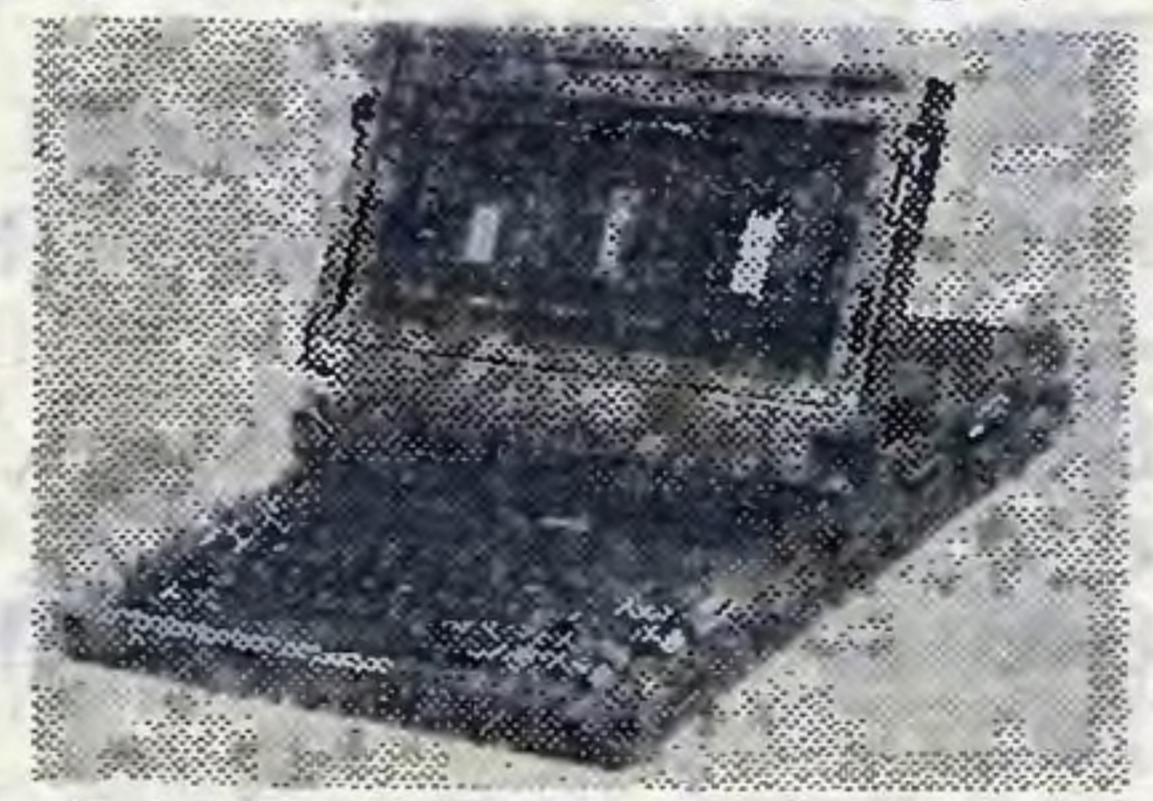


6. Epson PX-4

Mikroprocesor: Z80A, 3.68MHz
System operacyjny: CP/M 2.2
Pamięć RAM: 64 KB
Grafika: 240x64
Zasilanie: bateria lub akumulator
Rozmiary: 29.7x3.4x28.6 cm
Waga: 1.6 kg
Cena: 2500 M

7. Epson PX-8

Mikroprocesor: Z80, 2.45MHz
System operacyjny: CP/M
Pamięć RAM: 64 KB lub 128 MB
Grafika: 480x64
Zasilanie: sieć lub akumulator
Rozmiary: 29.7x4.8x21.6 cm
Waga: 2.3 kg
Cena: 3000 M



8. Grid Case

Mikroprocesor: 80C86, 4.77MHz
System operacyjny: MS-DOS
2.11/3.2
Pamięć RAM: 128 KB lub 640 MB
Pamięć masowa: wewnętrzny Floppy-
-Disc 720 KB
Grafika: 640x200
Zasilanie: sieć lub akumulator
Rozmiary: 29x5.5x38 cm
Waga: 5.4 kg
Cena: 7000 M

SPECTRUM

PROGRAM 4

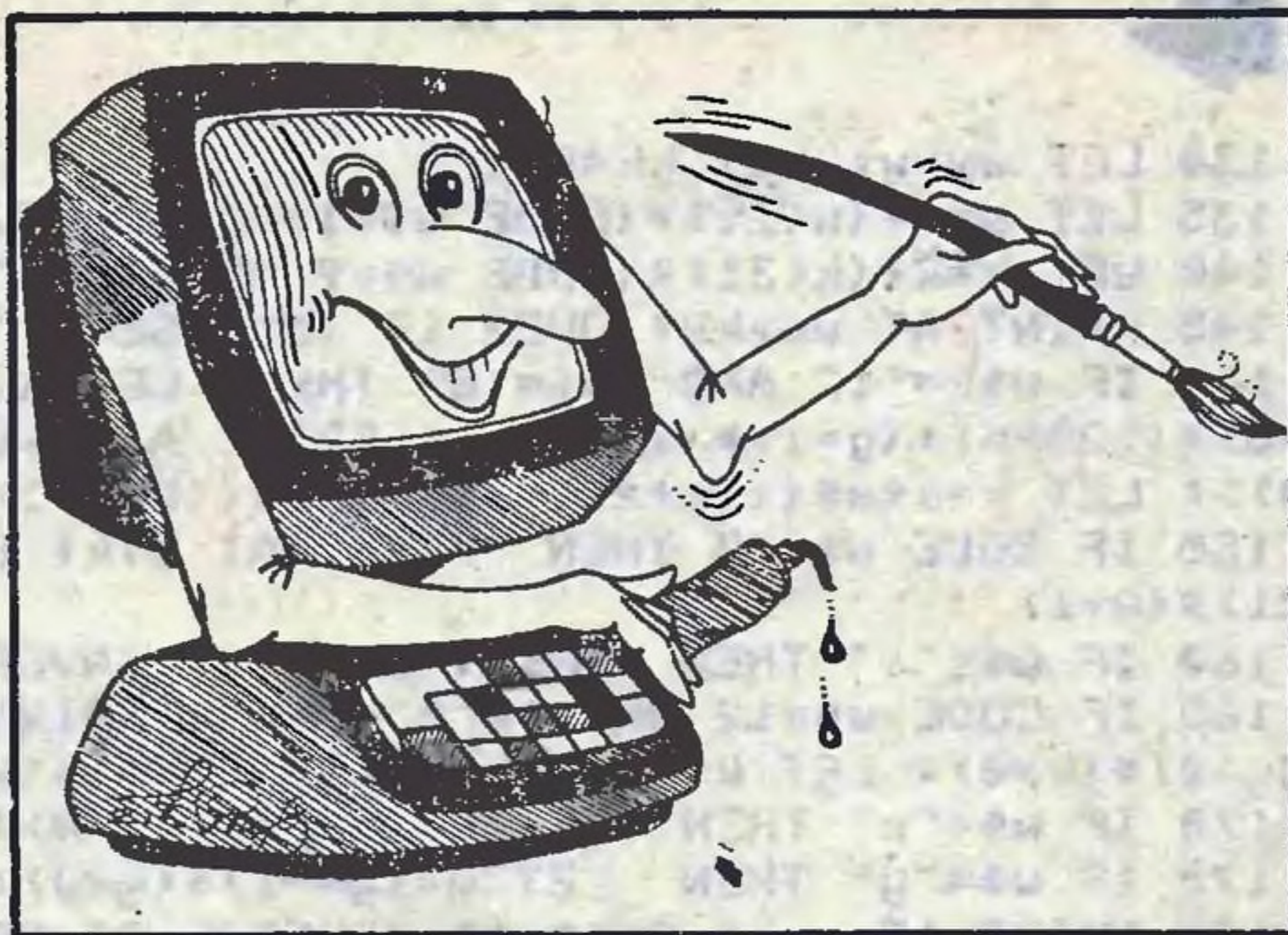
ARTYSTA

Program „Artysta” jest edytorem obrazu złożonego ze standardowych znaków graficznych. Program ten może być dobrym uzupełnieniem do programu „Kompens” publikowanego w numerze piątym „IKSa”. Pracuje w trzech trybach: graficznym, tekstowym i edycyjnym.

Tryb graficzny pozwala na budowanie obrazu ze znaków graficznych (klawisze 1,2...,8). Aktualną pozycję dostawionego znaku wskazuje kursor — kwadrat o podwyższonej jaskrawości. W tym trybie dostępne są następujące funkcje:

CAPS SHIFT (CS) i \leftarrow	— przesunięcie kursora we wskazanym kierunku (dodatkowo funkcję tę pełnią klawisze Q, W, E, R),
ENTER	— przejście na początek nowej linii,
SPACE	— usunięcie znaku na pozycji kursora,
DELETE	— przesunięcie kursora o jedno pole wstecz i usunięcie stojącego tam znaku,
CS i TRUE VIDEO	— kolor i tło znaku są zgodne z deklarowanymi,
CS i INVERS VIDEO	— kolor i tło znaku są dopełnieniem w stosunku do deklarowanych atrybutów,
GRAPHICS lub 9	— zmiana (negacja) zestawu znaków graficznych,
X, Y	— zmiana sposobu przesuwania się kursora po wpisaniu znaku, (x=1 w prawo, x=0 bez przesunięcia, X=1 w lewo Y=1 w górę, Y=0 bez przesunięcia, Y=1 w górę),
EDIT	— przejście do trybu edycyjnego.

O aktualnym rodzaju pracy informuje komunikat u dołu ekranu (napis: video, wartości X i Y oraz aktualny zestaw znaków).



W trybie edycyjnym dostępne są funkcje:

- J (LOAD) — wczytanie obrazu z taśmy,
- P (PRINT) — przejście do trybu tekstowego,
- S (SAVE) — zapisanie obrazu na taśmę,
- V (US) — skasowanie zawartości ekranu,
- Z (COPY) — wydruk zawartości ekranu,
- ENTER — powrót do trybu graficznego.

Tryb tekstowy pozwala na umieszczenie napisów (poziomych i pionowych) w tworzonym obrazie. Napisy mogą składać się ze znaków alfabetu ZX Spectrum i znaków UDG. Wykluczone są jedynie znaki dostępne w trybie „EXTEND MODE”. Komunikat informuje o rodzaju pracy podając tryb L lub C i sposób przemieszczania się kursora (kursor \emptyset — przesunięcie w prawo, kursor 1 przesunięcie w dół).

CAPS LOCK (CS i 2)	— zmiana trybu L na C i odwrotnie,
EXTEND MODE (CAPS SHIFT : SYMBOL SHIFT)	— zmiana sposobu przemieszczania się kursora (napis poziomy/napis pionowy),
\leftarrow	— przesunięcie kursora
DELETE	— skasowanie znaku na polu poprzedzającym kursor,
GRAPHICS	— przejście do wpisywania znaków UDG,
ENTER	— powrót do trybu edycji.

Struktura programu pozwala na łatwą rozbudowę (np. wersja kolorowa lub uzupełnienie instrukcjami DRAW, PLOT itd.). Dodatkowe bloki można zbudować oddzielnie i dołączyć w miarę potrzeby instrukcję merge „nazwa”.

```

5 REM PROGRAM "ARTYSTA"
10 BORDER 7: PAPER 7: INK 0: INVERSE 0: OVER 0: BRIGHT 0
CLS
20 LET p=0: LET w=0: LET k=0: LET g=0: LET x=1: LET y=0:
30 POKE 23658,0: GO SUB 1000
40 PRINT AT w;k OVER 1: INVERSE 0: BRIGHT 1: " "
100 LET w#=INKEY$: IF w#="" THEN GO TO 100
105 IF CODE w#=4 THEN LET i=0: GO SUB 1000
110 IF CODE w#=5 THEN LET i=1: GO SUB 1000
115 IF CODE w#=15 OR w#="9" THEN LET g=1-g: GO SUB 1000
120 IF CODE w#=7 THEN GO SUB 5000: GO TO 100
125 PRINT AT w;k OVER 1: INVERSE 0: BRIGHT 1: " "

```

```

130 LET ww=w: LET kk=k
135 LET w=w+(w<21)*(CODE w#=10 OR w#="w")-(w>0)*(CODE w#=11 OR w#="q")
140 LET k=k+(k<31)*(CODE w#=9 OR w#="r")-(k>0)*(CODE w#=8 OR w#="e")
145 PRINT AT ww,kk: OVER 1: INVERSE 0: " "
150 IF w#="1" AND w#<="8" THEN LET h=VAL w#-8*INT (VAL w#/8): LET w#=CHR$ ((g
=0)*(128+h)+(g=1)*(143-h)): PRINT AT w,k: w#: LET w=w-y*((w-y)<=21) AND ((w-y)>=
0): LET k=k+x*((k+x)<=31) AND ((k+x)>=0)
155 IF CODE w#=13 THEN PRINT AT w,k: OVER 1: INVERSE 0: " ": LET k=0: LET w=(w<
21)*(w+1)
160 IF w#=" " THEN PRINT AT w,k: INVERSE 0: BRIGHT 1: " "
165 IF CODE w#=12 OR w#="0" THEN PRINT AT w,k: INVERSE 0: OVER 1: " ": LET w=w-
(w>0)*(k=0): LET k=(k>0)*(k-1)+(w=0)*(k=0)*31: PRINT AT w,k: INVERSE 0: " "
170 IF w#="x" THEN LET x=(x=-1)-(x=0): GO SUB 1000
175 IF w#="y" THEN LET y=(y=-1)-(y=0): GO SUB 1000
180 PRINT AT w,k: OVER 1: INVERSE 0: BRIGHT 1: " "
185 GO TO 100
1000 REM KOMUNIKATY
1001 INVERSE i
1002 IF i=0 THEN PRINT #0: AT 0,0: "VIDEO "
1003 IF i=1 THEN PRINT #0: AT 0,0: "VIDEO "
1004 PRINT #0: AT 1,0: "X: "x:" " AT 1,5: "Y: "y:" "
1005 IF g=0 THEN PRINT #0: AT 0,8: "KLAWISZ: 1 2 3 4 5 6 7 8" AT 1,11: "ZNAK:
A"
1006 IF g=1 THEN PRINT #0: AT 0,8: "KLAWISZ: 1 2 3 4 5 6 7 8" AT 1,11: "ZNAK:
"
1007 RETURN
1008
5000 REM TRYB EDYCJI
5002 POKE 23658,0
5004 PRINT #0: AT 0,0: "J - LOAD S - SAVE P - PRINT Z - COPY V - CLS ENTER-
POWROT"
5006 LET w#=INKEY$: IF w#="" THEN GO TO 5006
5008 IF w#="s" THEN GO SUB 5200
5010 IF w#="j" THEN GO SUB 5100
5012 IF w#="z" THEN GO SUB 5300
5014 IF w#="v" THEN GO SUB 5400
5016 IF w#="p" THEN PAUSE 30: GO SUB 6000
5018 IF CODE w#=13 THEN GO SUB 1000: RETURN
5020 GO TO 5000
5100 REM LOAD
5102 PRINT #0: AT 0,0: "CZY LADOWAC OBRAZ Z TASY ? (T/N)
"
5104 IF INKEY$="n" THEN RETURN
5106 IF INKEY$<>"t" THEN GO TO 5104
5108 INPUT "PODAJ NAZWE ZBIORU "n$: IF n#="" OR LEN n#>10 THEN GO TO 5108
5110 LOAD n$ SCREEN$
5112 POKE 23658,0: RETURN
5114
5200 REM SAVE
5202 PRINT #0: AT 0,0: "CZY NAGRYWASZ OBRAZ ? (T/N)
"
5204 IF INKEY$="n" THEN RETURN
5206 IF INKEY$<>"t" THEN GO TO 5204
5208 INPUT "PODAJ NAZWE ZBIORU "n$: IF n#="" OR LEN n#>10 THEN GO TO 5208
5210 SAVE n$ SCREEN$
5212 PRINT #0: AT 0,0: "GOTOWE - WCISNIJ ENTER
"
5214 IF CODE INKEY$<>13 THEN GO TO 5214
5216 POKE 23658,0: RETURN
5218
5300 REM COPY
5302 PRINT #0: AT 0,0: "CZY DRUKOWAC OBRAZ ? (T/N)
"
5304 IF INKEY$="t" THEN COPY: RETURN
5306 IF INKEY$="n" THEN RETURN
5308 GO TO 5304

```

```

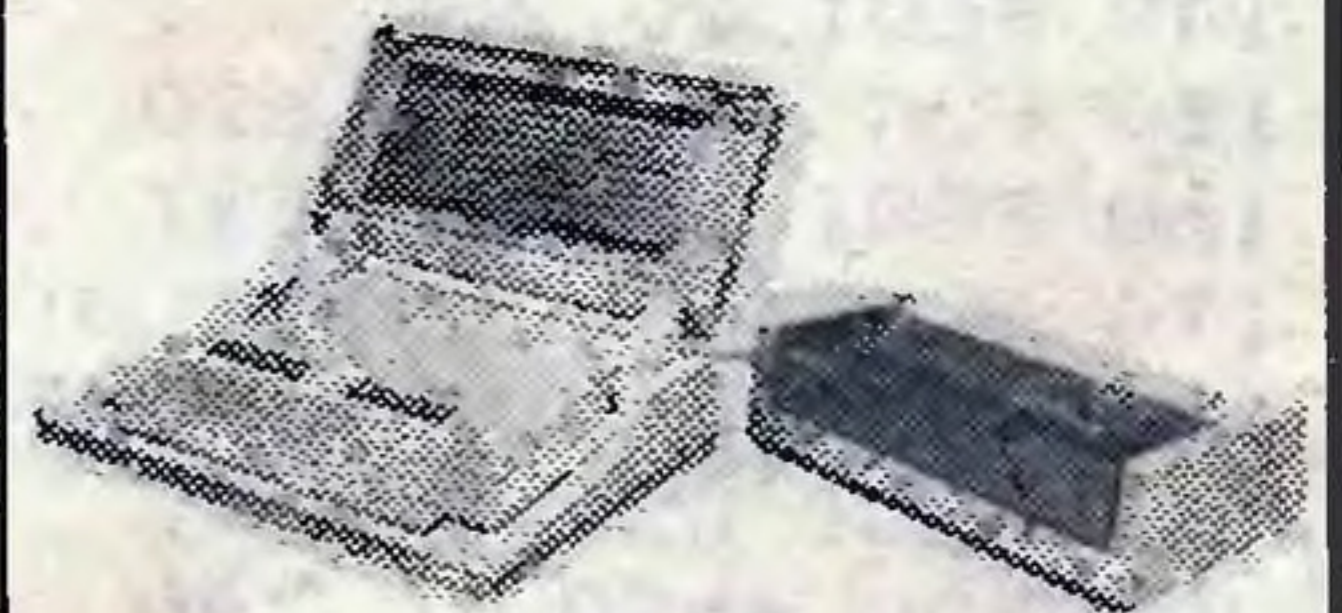
5310
5400 REM CLS
5402 PRINT #0;AT 0,0;"CZY SKASOWAC EKTAN ? (T/N)
"
5404 IF INKEY$="t" THEN CLS : RETURN
5406 IF INKEY$="n" THEN RETURN
5408 GO TO 5404
5410
6000 REM PRINT
6002 LET d=0
6004 IF p=0 THEN POKE 23658,0: PRINT #0;AT 0,0;"TRYB TEKSTOWY L  kursor:
ENTER - POWROT DO GRAFIKI "AT 0,24;d
6006 IF p=1 THEN POKE 23658,8: PRINT #0;AT 0,0;"TRYB TEKSTOWY C  kursor:
ENTER - POWROT DO GRAFIKI "AT 0,24;d
6008 LET w$=INKEY$: IF w$="" THEN GO TO 6008
6010 IF CODE w$=6 THEN LET p=1-p: PAUSE 50: GO TO 6004
6011 IF CODE w$=14 THEN LET d=1-d: PAUSE 30: GO TO 6004
6012 IF CODE w$=15 THEN GO SUB 6100: PRINT AT w,k; OVER 1; BRIGHT 1; INVERSE 0;
" ": GO TO 6004
6014 IF CODE w$=13 THEN POKE 23658,0: RETURN
6016 IF CODE w$=12 THEN PRINT AT w,k; OVER 1; INVERSE 0;" ": LET w=w-(d=0)*(w>0)
)*(k=0)-(d=1)*(w>0): LET k=(k>0)*(d=0)*(k-1)+(d=0)*(w=0)*(k=0)*31+(d=1)*k: PRIN
T AT w,k; INVERSE 0;" "
6018 IF w$)=" " AND w$<=" " THEN PRINT AT w,k;w$: LET w=w+d*((w+d)<=21): LET k=
k+(1-d)*((k+x)<=31)
6020 BEEP .05,12
6022 PRINT AT w,k; INVERSE 0; OVER 1;" "
6024 LET w=w+(w<21)*(CODE w$=10)-(w>0)*(CODE w$=11)
6026 LET k=k+(k<31)*(CODE w$=9)-(k>0)*(CODE w$=8)
6028 PRINT AT w,k; OVER 1; INVERSE 0; BRIGHT 1;" "
6030 GO TO 6008
6100 REM UDG
6102 PRINT #0;AT 0,0;"KLAWISZ:  ABCDEFGHIJKLMNOPQRSTUznak  G:  0 ZC
#EABCD"
6104 LET w$=INKEY$: IF w$="" THEN GO TO 6104
6106 IF CODE w$=13 THEN LET w$="": PAUSE 30: RETURN
618 LET w$=CHR$ (((w$)="a") AND (w$<="u"))*47+((w$)="A") AND (w$<="U"))*79+CODE
w$: IF w$)=" " AND w$<=" " THEN PRINT AT w,k;w$: LET w=w+d*(w<21): LET k=k+(1-
d)*(k<31): PAUSE 20: RETURN
6110 GO TO 6104

```

WIESŁAW RYCHLICKI



10. HP 110 (Hewlett Packard)
Mikroprocesor: 80C86, 5.33MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 96 KB lub 272 KB
Pamięć masowa: Floppy-Disc (3.5")
710 KB, dysk twardy
10 MB
Grafika: 480x128
Zasilanie: sieć lub akumulator
Rozmiary: 32x7x25 cm
Waga: 3.9 kg



11. HP Portable Plus (Hewlett Packard)
Mikroprocesor: 80C86, 5.33MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 256 KB lub 1.2 MB
Pamięć masowa: RAM-Disc 32 KB lub
1.1 MB, Floppy-
-Disc 710 KB, dysk
twardy 10 MB
Grafika: 480x200
Zasilanie: sieć lub akumulator
Rozmiary: 32x7.6x25 cm
Waga: 4.5 kg
Cena: 8500 M

9. Grid Compass
Mikroprocesor: 8086 i 8087
System operacyjny: MS-DOS, Grid-
-OS
Pamięć RAM: 512 KB
Pamięć masowa: Floppy-Disc 360 KB,
dysk twardy 10 MB
Grafika: 320x240, 512x256
Zasilanie: sieć lub akumulator
Rozmiary: 29x5x38 cm
Waga: 4.6 kg
Cena: 17 700 M



AMSTRAD

PROGRAMY 5-8

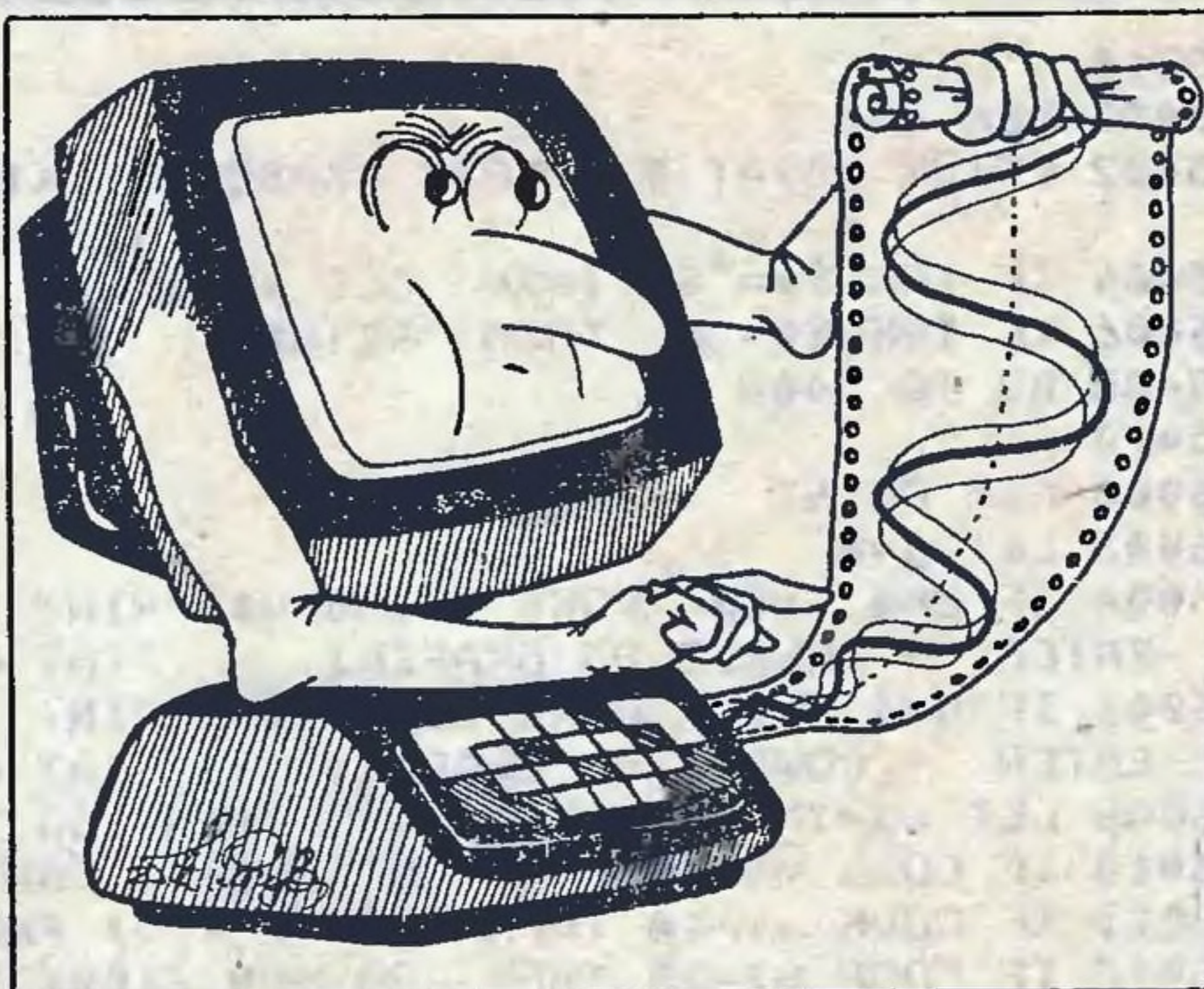
PROSTA

Program kreśli prostą o równaniu $y=ax+b$ w układzie współrzędnych OXY w zależności od podanych parametrów a i b; pozwala zaobserwować jak zmienia się rysowana prosta w zależności od zadanych parametrów. W przedstawionym układzie współrzędnych można narysować cztery proste.

Jeśli chcemy wykreślić linię (z założeń programowych $n=2$), to zmieniamy w programie wiersz 400 pisząc, zamiast $l=3$ $l=n-1$ i wiersz 410 pisząc zamiast $l=4$, $l=n$, gdzie za n wstawiamy liczbę naturalną, większą lub równą 2.

Aby rysowana prosta „zmieściła się” w oknie monitora, przyjęto ograniczenie na parametr b, a mianowicie $b \in \langle -18, 18 \rangle$

```
10 REM P R O S T A
20 CLS
30 GOSUB 540
40 CLS:BORDER 3
50 WINDOW #2,10,31,25,25
60 PAPER #2,13:PEN #2,0:CLS #2
70 WINDOW #3,35,40,1,1
80 PAPER #3,13:PEN #3,0:CLS #3
90 PRINT #3," 0"
100 WINDOW #1,1,8,1,3
110 BORDER 9
120 PAPER #1,13:PEN #1,0:CLS #1
130 PRINT #2,"      y= ax + b  "
140 PLOT 20,200
150 FOR s=20 TO 620
160 PLOT s,200:NEXT
170 PLOT 618,202:PLOT 618,199
180 PLOT 616,204:PLOT 616,197
190 PLOT 320,400
200 FOR s=20 TO 390
210 PLOT 320,s:NEXT
220 PLOT 318,388:PLOT 316,386
230 PLOT 322,388:PLOT 324,386
240 LOCATE 39,14:PRINT "x"
250 LOCATE 19,1:PRINT "y"
260 l=0
270 INPUT #1,"a=";a
280 GOTO 720
290 INPUT #1,"b=";b
300 IF b<-18 OR b>18 THEN 730
310 IF b>=0 THEN 430 ELSE 440
320 CLS #2
```



```
330 PRINT #2,"      y=";a;"x ";
      d#ABS(b)
340 FOR x=-320 TO 320
350 y=a*x*0.92+b*9.5
360 PLOT 320+x,200+y
370 NEXT
380 CLS #1:CLS #3:l=1+1
390 PRINT #3," ";l
400 IF l=3 THEN 450
410 IF l>=4 THEN 480
420 GOTO 270
430 d$="+":GOTO 320
440 d$="-":GOTO 320
450 ENV 3,=9,9000
460 SOUND 1,300,0,13,3
470 GOTO 270
480 LOCATE 9,24
490 PRINT " koniec -";
      " [DOWOLNY KLAWISZ]"
500 ENV 3,=9,9000
510 SOUND 1,100,0,13,3
520 IF INKEY$="" THEN 520
530 GOTO 40
540 t1$="P R O S T A "
550 T2$=" "+CHR$(164)+
      " "+" CJS "
560 MODE 0
570 ENV 1,=9,2000
580 ENT -1,6,3,1
590 ENV 2,127,0,0,127,0,
      0,127,0,0,127,0,0
600 ENV 3,=9,9000
610 FOR c=1 TO LEN(t1$)
620 LOCATE 2+c,10
630 FOR n2=LEN(t1$)
      TO c STEP -1
640 PRINT MID$(t1$,n2,1)
650 LOCATE 2+c,10
660 SOUND 135,20*n2,5,12,2,1
670 NEXT n2,c
```

```

680 SOUND 135,100,0,13,3,1,20
690 LOCATE 5,15:PRINT t2$
700 FOR c=1 TO 2000:NEXT
710 CLS:MODE 1:RETURN
720 LOCATE 10,23:GOTO 290
730 LOCATE 13,25
740 PRINT " PODAJ b";
    CHR$(180); "<-18,18>"
750 FOR w=1 TO 3500:NEXT
760 PRINT #2, "          y= ax + b  "
770 GOTO 290

```



PARABOLA, ELIPSA, HIPERBOLA

Program PARABOLA kreśli krzywą o równaniu $y=ax^2+bx+c$ w układzie współrzędnych OXY w zależności od podanych przez użytkownika parametrów a , b i c .

Program pozwala zaobserwować jak zmienia się rysowana parabola w zależności od zadanych parametrów a , b i c . W dowolnej części monitora w oknie przedstawione jest równanie $y=ax^2+bx+c$ aktualnie kreślonej paraboli.

W przedstawionym układzie współrzędnych można narysować $l=4$ parabole. Jeśli chcemy wykreślić $l=n \neq 4$ paraboli (ale z założeń programowych $l=n \geq 2$), to zmieniamy wiersz 460 pisząc zamiast $l=3$ $l=n-1$ i wiersz 470, pisząc zamiast $l=4$ $l=n$, gdzie za n wstawiamy konkretną liczbę naturalną większą lub równą 2. Aby rysowana parabola zmieściła się w oknie przyjęto ograniczenia na parametr b i c , a mianowicie $b \in \langle -9,9 \rangle$, $a \in \langle -9,9 \rangle$. Parametr a powinien być liczbą całkowitą z przedziału $\langle -9,9 \rangle$.

Konstrukcje pozostałych programów: „Elipsa” i „Hiperbola” oparte są dokładnie na tej samej zasadzie.

```

10 REM P A R A B O L A
20 CLS
30 GOSUB 640
40 CLS:BORDER 3
50 WINDOW #2,12,33,24,25
60 PAPER #2,13:PEN #2,0:CLS #2
70 WINDOW #3,35,40,1,1
80 PAPER #3,13:PEN #3,0:CLS #3
90 PRINT #3, " 0"
100 WINDOW #1,1,10,1,4
110 BORDER 9
120 PAPER #1,13:PEN #1,0:CLS #1
130 GOSUB 490
140 PLOT 100,220
150 FOR s=50 TO 620
160 PLOT s,220:NEXT
170 PLOT 618,222:PLOT 618,219
180 PLOT 616,224:PLOT 616,217
190 PLOT 320,400
200 FOR s=20 TO 390
210 PLOT 340,s:NEXT
220 PLOT 338,388:PLOT 336,386

```

```

230 PLOT 342,388:PLOT 344,386
240 LOCATE 39,13:PRINT "x"
250 LOCATE 20,1:PRINT "y"
260 l=0
270 INPUT #1,"a=";a
280 IF a=0 THEN 810
290 IF a>9 OR a<-9 THEN 850
300 INPUT #1,"b=";b
310 IF b>9 OR b<-9 THEN 890
320 INPUT #1,"c=";c
330 IF c>9 OR c<-9 THEN 930
340 IF b>=0 THEN 520 ELSE 540
350 IF c>=0 THEN 530 ELSE 550
360 CLS #2
370 PRINT #2, "          2  "
380 PRINT #2, " y=";a;
    "x ";d$b;"x ";e$c
390 FOR x=-300 TO 300
400 y=(a*(x)^2)/100+(b*x)/4+c*18
410 IF y>180 OR y<-180 THEN 430
420 PLOT 340+x,220+y
430 NEXT
440 CLS #1:CLS #3:l=1+1
450 PRINT #3, " ";l
460 IF l=3 THEN 560
470 IF l>=4 THEN 580
480 GOTO 270
490 PRINT #2, "          2  "
500 PRINT #2, " y= ax + bx + c  "
510 RETURN
520 d$="+":GOTO 350
530 e$="+":GOTO 360
540 d$=" ":GOTO 350
550 e$=" ":GOTO 360
560 ENV 3,=9,9000
570 SOUND 1,300,0,13,3:GOTO 270
580 LOCATE 9,23
590 PRINT " koniec -"
    "[DOWOLNY KLAWISZ]"
600 ENV 3,=9,9000
610 SOUND 1,100,0,13,3
620 IF INKEY$="" THEN 620
630 GOTO 40
640 t1$="P A R A B O L A"
650 T2$=CHR$(164)+" "+
    CHR$(32)+" CJS "
660 MODE 0
670 ENV 1,=9,2000:ENT -1,6,3,1
680 ENV 2,127,0,0,127,0,
    0,127,0,0,127,0,0
690 ENV 3,=9,9000
700 FOR c=1 TO LEN(t1$)
710 LOCATE 2+c,10
720 FOR n2=LEN(t1$) TO c STEP -1
730 PRINT MID$(t1$,n2,1)

```

```

740 LOCATE 2+c,10
750 SOUND 135,20*n2,5,12,2,1
760 NEXT n2,c
770 SOUND 135,100,0,13,3,1,20
780 LOCATE 5,15:PRINT t2$
790 FOR c=1 TO 2000:NEXT
800 CLS:MODE 1:RETURN
810 LOCATE 12,24
820 PRINT " WPISZ a"
      " ROZNE OD ZERA "
830 FOR w=1 TO 2000:NEXT
840 GOSUB 490:GOTO 270
850 LOCATE 12,24
860 PRINT "   PODAJ a";
      CHR$(180);"<-9,9> "
870 FOR w=1 TO 3500:NEXT
880 GOSUB 490:GOTO 270
890 LOCATE 12,24
900 PRINT "   PODAJ b";
      CHR$(180);"<-9,9> "
910 FOR w=1 TO 3500:NEXT
920 GOSUB 490:GOTO 300
930 LOCATE 12,24
940 PRINT "   PODAJ c";
      CHR$(180);"<-9,9> "
950 FOR w=1 TO 3500:NEXT
960 GOSUB 490:GOTO 320

```



```

10 REM E L I P S A
20 CLS
30 GOSUB 800
40 CLS:BORDER 3
50 WINDOW #2,11,31,21,25
60 PAPER #2,13:PEN #2,0:CLS #2
70 WINDOW #3,35,40,1,1
80 PAPER #3,13:PEN #3,0:CLS #3
90 PRINT #3," 0"
100 WINDOW #1,1,9,1,4
110 BORDER 9
120 PAPER #1,13:PEN #1,0:CLS #1
130 GOSUB 660
140 PLOT 100,240
150 FOR s=100 TO 600
160 PLOT s,240:NEXT
170 PLOT 598,242:PLOT 598,239
180 PLOT 596,244:PLOT 596,237
190 PLOT 340,400
200 FOR s=80 TO 390
210 PLOT 340,s:NEXT
220 PLOT 338,388:PLOT 336,386
230 PLOT 342,388:PLOT 344,386
240 LOCATE 38,12:PRINT "x"

```

```

250 LOCATE 20,1:PRINT "y"
260 l=0
270 INPUT #1,"a=";a
280 IF a=0 THEN 970
290 IF a<-9 OR a>9 THEN 1030
300 a=a*15
310 INPUT #1,"b=";b
320 IF b=0 THEN 1000
330 IF b<-9 OR b>9 THEN 1070
340 b=b*15:CLS #2
350 PRINT #2,"      2      2 "
360 PRINT #2,"      x      y "
370 PRINT #2,"      --- + --- = 1"
380 PRINT #2,"      2      2 "
390 PRINT #2,"      ";
      a/15;"      ";b/15;"
400 n=200
410 FOR x=0 TO -n STEP -1
420 k=(a-x)*(a+x)
430 IF k<0 THEN 500
440 y=(b/a)*SQR(k)
450 IF y>158 THEN 470
460 PLOT 340+x,240+y
470 y1=-(b/a)*SQR(k)
480 IF y1<-158 THEN 510
490 PLOT 340+x,240+y1
500 NEXT
510 FOR z=0 TO n
520 k1=(a-z)*(a+z)
530 IF k1<0 THEN 600
540 y2=(b/a)*SQR(k1)
550 IF y2>158 THEN 570
560 PLOT 340+z,240+y2
570 y3=-(b/a)*SQR(k1)
580 IF y3<-158 THEN 610
590 PLOT 340+z,240+y3
600 NEXT
610 CLS #1:CLS #3:l=1+1
620 PRINT #3," ";l
630 IF l=2 THEN 720
640 IF l>=3 THEN 740
650 GOTO 270
660 PRINT #2,"      2      2 "
670 PRINT #2,"      x      y "
680 PRINT #2,"      --- + --- = 1"
690 PRINT #2,"      2      2 "
700 PRINT #2,"      a      b "
710 RETURN
720 ENV 3,=9,9000
730 SOUND 1,300,0,13,3:GOTO 270
740 LOCATE 9,21
750 PRINT " koniec -"
      " [DOWOLNY KLAWISZ]"
760 ENV 3,=9,9000
770 SOUND 1,100,0,13,3
780 IF INKEY$="" THEN 780
790 ORIGIN 0,0:GOTO 40

```

```

800 t1$="E L I F S A"
810 T2$=" "+CHR$(164)+
      " "+" CJS "
820 MODE 0
830 ENV 1,=9,2000:ENT -1,6,3,1
840 ENV 2,127,0,0,127,
      0,0,127,0,0,127,0,0
850 ENV 3,=9,9000
860 FOR c=1 TO LEN(t1$)
870 LOCATE 2+c,10
880 FOR n2=LEN(t1$) TO c STEP -1
890 PRINT MID$(t1$,n2,1)
900 LOCATE 2+c,10
910 SOUND 135,20*n2,5,12,2,1
920 NEXT n2,c
930 SOUND 135,100,0,13,3,1,20
940 LOCATE 5,15:PRINT t2$
950 FOR c=1 TO 2000:NEXT
960 CLS:MODE 1:RETURN
970 LOCATE 11,21:PRINT
      "WPISZ a ROZNE OD ZERA"
980 FOR w=1 TO 2000:NEXT
990 GOSUB 660:GOTO 270
1000 LOCATE 11,21:PRINT
      "WPISZ b ROZNE OD ZERA"
1010 FOR w=1 TO 2000:NEXT
1020 GOSUB 660:GOTO 310
1030 LOCATE 11,21
1040 PRINT "   PODAJ a";
      CHR$(180);"<-9,9>"
1050 FOR w=1 TO 3500:NEXT
1060 GOSUB 660:GOTO 270
1070 LOCATE 11,21
1080 PRINT "   PODAJ b";
      CHR$(180);"<-9,9>"
1090 FOR w=1 TO 3500:NEXT
1100 GOSUB 660:GOTO 310

```

```

160 PLOT s,220:NEXT
170 PLOT 598,222:PLOT 598,219
180 PLOT 596,224:PLOT 596,217
190 PLOT 340,400
200 FOR s=50 TO 390
210 PLOT 340,s:NEXT
220 PLOT 338,388:PLOT 336,386
230 PLOT 342,388:PLOT 344,386
240 LOCATE 38,13:PRINT "x"
250 LOCATE 20,1:PRINT "y"
260 l=0
270 INPUT #1,"a=";a
280 IF a=0 THEN 820
290 IF a<-9 OR a>9 THEN 860
300 IF a=-1 OR a=1 THEN 860
310 a=a*10:CLS #2
320 PRINT #2,"      1  2"
      "      1  2"
330 PRINT #2,"   xy = -";a/10;
      "   xy = - -";a/10;
340 PRINT #2,"      2"
      "      2  "
350 n=220
360 FOR x=-n TO 0 STEP 1
370 y=(a*a)/(2*x)
380 IF y<-170 THEN 410
390 PLOT 340+x,220+y
400 NEXT
410 FOR x=n TO 0 STEP -1
420 y=(a*a)/(2*x)
430 IF y>170 THEN 460
440 PLOT 340+x,220+y
450 NEXT
460 CLS #1:CLS #3:l=1+l
470 PRINT #3," ";l
480 IF l=2 THEN 550
490 IF l>=3 THEN 580

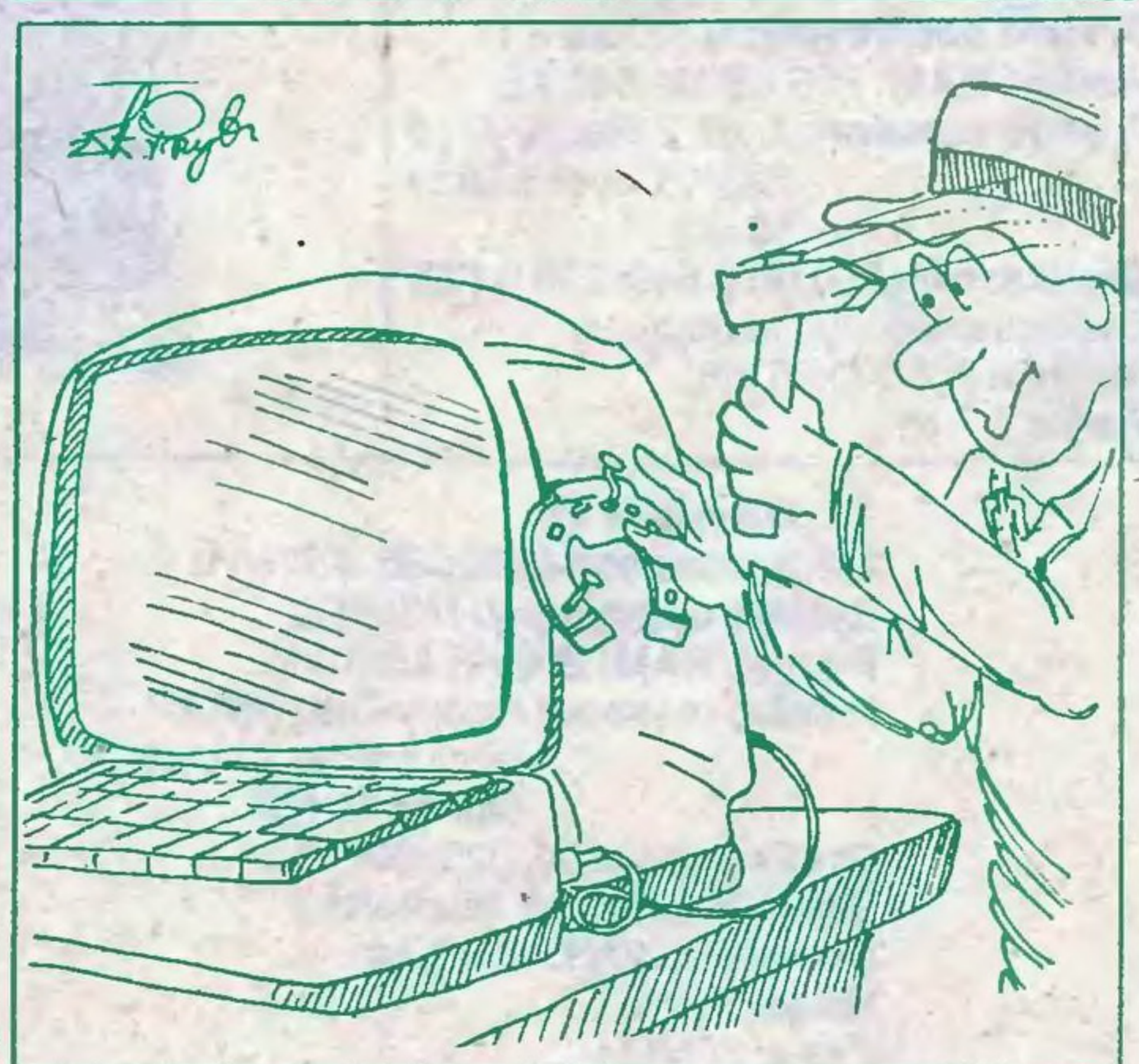
```



```

10 REM H I P E R B O L A
20 CLS
30 GOSUB 640
40 CLS:BORDER 3
50 WINDOW #2,9,33,23,25
60 PAPER #2,13:PEN #2,0:CLS #2
70 WINDOW #3,35,40,1,1
80 PAPER #3,13:PEN #3,0:CLS #3
90 PRINT #3,"  0"
100 WINDOW #1,1,9,1,4
110 BORDER 9
120 PAPER #1,13:PEN #1,0:CLS #1
130 GOSUB 510
140 PLOT 100,220
150 FOR s=100 TO 600

```



```

500 GOTO 270
510 PRINT #2, "      1  2";
      "      1  2 "
520 PRINT #2, " xy = - a";
      " xy = - - a "
530 PRINT #2, "      2";
      "      2 "
540 RETURN
550 ENV 3, =9, 9000
560 SOUND 1, 300, 0, 13, 3
570 GOTO 270
580 LOCATE 9, 22
590 PRINT "koniec -";
      " [DOWOLNY KLAWISZ]"
600 ENV 3, =9, 9000
610 SOUND 1, 100, 0, 13, 3
620 IF INKEY$="" THEN 620
630 ORIGIN 0, 0: GOTO 40
640 t1$="H I P E R B O L A"
650 t2$=" "+CHR$(164)+
      " "+" CJS "
660 MODE 0
670 ENV 1, =9, 2000
680 ENT -1, 6, 3, 1
690 ENV 2, 127, 0, 0, 127,
      0, 0, 127, 0, 0, 127, 0, 0

```

```

700 ENV 3, =9, 9000
710 FOR c=1 TO LEN(t1$)
720 LOCATE 2+c, 10
730 FOR n2=LEN(t1$)
      TO c STEP -1
740 PRINT MID$(t1$, n2, 1)
750 LOCATE 2+c, 10
760 SOUND 135, 20*n2, 5, 12, 2, 1
770 NEXT n2, c
780 SOUND 135, 100, 0, 13, 3, 1, 20
790 LOCATE 5, 15: PRINT t2$
800 FOR c=1 TO 2000: NEXT
810 CLS: MODE 1: RETURN
820 LOCATE 9, 23
830 PRINT " WPISZ a";
      " ROZNE OD ZERA "
840 FOR w=1 TO 2000: NEXT
850 GOSUB 510: GOTO 270
860 LOCATE 9, 23
870 PRINT "PODAJ a";
      CHR$(180); "<-9, -2> ";
      CHR$(118); " a";
      CHR$(180); "<2, 9>"
880 FOR w=1 TO 4000: NEXT
890 GOSUB 510: GOTO 270

```

**JACEK SZANIAWSKI
CEZARY SOBCZAK**



12. DG One, Modell 2 (Data General)
Mikroprocesor: 80C88, 4.77MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 256 KB lub 640 KB
Pamięć masowa: 1 lub 2 Floppy-Disc
 720 KB, dysk twardy
 10 MB
Grafika: 640x200 (EL), 640x256 (LCD)
Zasilanie: sieć lub akumulator
Rozmiary: 37x8x30 cm
Waga: 4.7 kg

13. Kaypro 2000

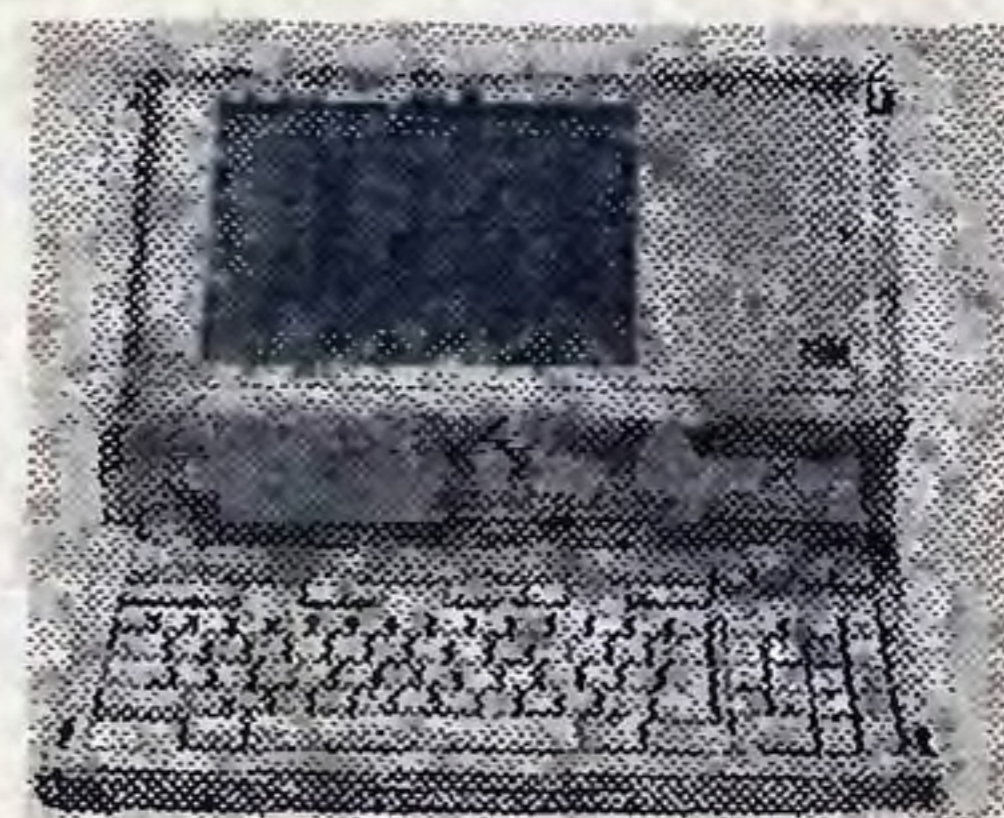
Mikroprocesor: 8088, 4.77MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 768 KB
Pamięć masowa: Floppy-Disc 720 KB
Grafika: 640x200
Zasilanie: sieć lub akumulator
Rozmiary: 33x7x28.7 cm
Waga: 5 kg
Cena: 4350 M



14. NEC 8401 BD
Mikroprocesor: Z80A, 3.99MHz
System operacyjny: CP/M 2.2
Pamięć RAM: 64 KB lub 196 KB
Pamięć masowa: Floppy-Disc 320 KB
Grafika: 480x128
Zasilanie: bateria
Rozmiary: 30x7x21 cm
Waga: 2.7 kg
Cena: 1990 M

15. Olivetti M 22

Mikroprocesor: 2x80C88, 4.77MHz
System operacyjny: MS-DOS
Pamięć RAM: 256 KB lub 1 MB
Pamięć masowa: Floppy-Disc 360 KB,
 opcjonalnie dysk
 twardy 10 MB
Grafika: 640x200, 320x200
Zasilanie: sieć lub akumulator
Rozmiary: 40x9.1x34.4 cm
Waga: 7.5 kg
Cena: 7050 M



AMSTRAD

PROGRAM 9

KRZYŻÓWKA

Program KRZYŻÓWKA jest narzędziem do tworzenia najpopularniejszej formy rozrywek umysłowych. Głównym efektem tej zabawy są wydruki ułożonego diagramu, kontrolnego układu haseł oraz haseł wraz ze współrzędnymi do późniejszego ich zdefiniowania. Rozmiary krzyżówki są opcjonalne, lecz ograniczone do prostokąta 25x22 krutek, co w zasadzie powinno wystarczyć. W pola diagramu można wpisywać pełny alfabet polski. Litery kłopotliwe tzn. Ć, Ł, Ź... itp. wczytuje się przez CONTROL C, L, Z... itd. Przy wpisywaniu liter należy posługiwać się kursorem, który jest nieodłącznym atrybutem diagramu i spełnia rolę wskaźnika. Cursor porusza się w czterech kierunkach po naciśnięciu klawisza ze strzałką. Dodatkowe funkcje pozwalają na umieszczenie w diagramie wyrazów lub ich fragmentów w poziomie lub w pionie. Funkcje te umożliwią wpis jedynie wtedy, gdy wyraz będzie pasował do aktualnej konfiguracji liter. Kropka oznacza, że pole jest puste, zaś gwiazdka symuluje pole niewykorzystane przez autora (tzw. czarne).

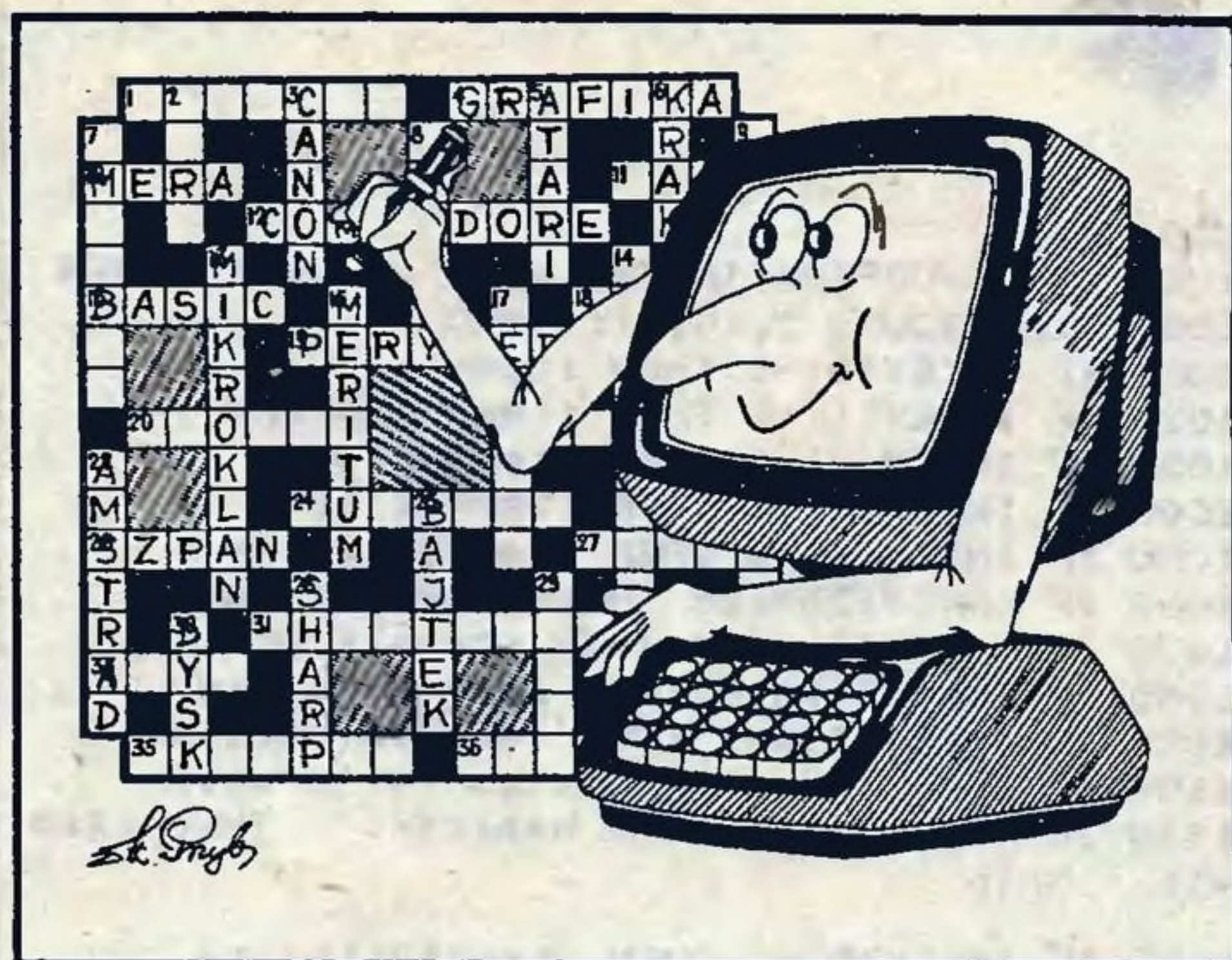
W przypadku przerwania pracy koncepcyjnej program pozwala na zapis dotychczasowego stanu na dyskietce, a potem na ewentualne powtórne wczytanie krzyżówki i kontynuowanie pracy. Należy podawać maksymalnie 8-literowy identyfikator krzyżówki bez uwzględniania typu zbioru, gdyż program automatycznie dodaje rozszerzenie KRZ. Jeśli krzyżówka posiada hasło będzie ono również zapamiętane na dyskietce.

Hasło wczytuje się na początku programu (RETURN ignoruje), lecz można je potem modyfikować w dowolnym momencie. Co kilka sekund hasło jest weryfikowane a na ekranie podawane są brakujące w diagramie litery. Jeśli w danym momencie wszystkie litery hasła występują już w diagramie, to ukazuje się komunikat „hasło kompletne”.

Pełny zestaw możliwości programu wraz z klawiszami wywołującymi poszczególne moduły uzyskuje się po naciśnięciu klawisza TAB.

Trudno przedstawić wszystkie rozwiązania ze względu na brak miejsca, lecz krótko chciałbym wskazać na potężną pracę systemu przerwań, który regularnie weryfikuje hasło. Przerwania dostępne w Basic'u są bardzo wygodnym środkiem. Ich poznanie nie nastęrcza większych trudności, a pozwala na niezwykle śmiało posunięcia.

```
10 ' ** krzyzowka '86
20 DEFINT a-z: DIM krz$(25,22): ZONE 12
30 ON BREAK GOSUB 1520
40 diak$="6962583646346071"
50 nor$="ACELNOSZ": char=140
60 '
70 MODE 1: BORDER 0: PAPER 0: PEN 3
80 INK 0,0: INK 2,20: INK 1,21: INK 3,6
```



```
90 WINDOW #1,1,40,25,25
100 PAPER #1,2: PEN #1,0
110 WINDOW #3,29,40,2,23
120 PAPER #3,1: PEN #3,0: CLS #3
130 EVERY 50,3 GOSUB 8090
140 'definicje polskich liter
150 SYMBOL AFTER 140
160 SYMBOL 140,24,60,98,98,126,98,103,1
170 SYMBOL 141,62,104,200,192,192,102,60,0
180 SYMBOL 142,254,98,104,120,106,99,254,6
190 SYMBOL 143,240,100,104,112,98,102,254,0
200 SYMBOL 144,202,234,242,218,206,198,194,0
210 SYMBOL 145,56,84,214,198,198,68,56,0
220 SYMBOL 146,62,100,96,60,6,102,60,0
230 SYMBOL 147,254,38,76,24,50,102,254,0
240 'definicje klawiszy
250 FOR a=0 TO 7
260 KEY char+1,CHR$(char)
270 b=VAL(MID$(diak$,2*a+1,2))
280 n=ASC(MID$(nor$,a+1,1))
290 KEY DEF b,1,n,0,char+1
300 char=char+1:NEXT
310 'haslo i wymiary diagramu
320 GOSUB 10050: flag=0
330 WHILE flag>=0: CLS #1
340 IF flag=0 THEN INPUT #1,"szer.krzyzowki
(2..25) ",sz ELSE INPUT #1,"wys.krzyzowki (
2..22) ",wys
350 IF sz<2 OR sz>26 THEN ELSE flag=1: PEN #
1,3
360 IF wys<2 OR wys>22 THEN ELSE flag=-1
370 WEND
380 FOR a=1 TO wys: FOR b=1 TO sz: krz$(b,a)=
".":NEXT b,a
390 krz$(0,1)=STR$(wys)
400 krz$(0,0)=STR$(sz)
410 WINDOW #2,3,sz+2,2,wys+1
420 PAPER #2,2: PEN #2,0: CLS#2: CLS #3
430 LOCATE 3,1: flag=1
440 WHILE flag<=sz
450 PRINT CHR$(flag+64);: flag=flag+1
460 WEND
470 LOCATE 1,2: flag=1
480 WHILE flag<=wys
490 IF flag<10 THEN PRINT " ";
500 PRINT MID$(STR$(flag),2)
510 flag=flag+1
520 WEND
530 GOSUB 1500: WINDOW SWAP 0,2
540 ON BREAK GOSUB 1510
550 FOR a=1 TO wys
560 FOR b=1 TO sz: PRINT krz$(b,a);
570 NEXT b,a
580 IF haslo$="" THEN GOSUB 9000 ELSE EVERY
500,2 GOSUB 8000
590 x=1:y=1: xp=1: yp=1: GOSUB 6000
```

```

600 'menu
1000 klaw$=UPPER$(INKEY$): IF klaw$="" THEN
1000 ELSE SOUND 1,10,4:item=0
1010 IF INKEY(0)=0 THEN item=1:'^
1020 IF INKEY(1)=0 THEN item=2:'prawo
1030 IF INKEY(2)=0 THEN item=3:'dol
1040 IF INKEY(8)=0 THEN item=4:'lewo
1050 IF INKEY(6)=0 THEN item=5:'enter
1060 IF INKEY(2)=128 THEN item=6:'pion
1070 IF INKEY(1)=128 THEN item=7:'poz
1080 IF INKEY(10)=0 THEN item=8:'zapis
1090 IF INKEY(11)=0 THEN item=9:'odczyt
1100 IF INKEY(9)=0 THEN item=10:'druk.
1110 IF INKEY(68)=0 AND haslo$<>"" THEN item
=11: 'help

1120 IF INKEY(5)=0 THEN item=12:'haslo
1130 IF item>0 THEN ON item GOSUB 2000,2030,
2050,2070,2090,3150,3000,5000,4000,7000,9040
,10000 : IF item=9 THEN 410 ELSE 1000
1140 IF klaw$>CHR$(139) AND klaw$<CHR$(148)
THEN item=32
1150 IF klaw$="." OR (klaw$)="A" AND klaw$<=
"Z") OR item=32 THEN krz$(yp, xp)=klaw$:GOSUB
2030
1160 GOTO 1000

1500 PAPER #1,0:CLS#1:PAPER #1,2:RETURN
1510 WINDOW SWAP 0,2
1520 CALL &BBO0:END
2000 'kierunki
2010 x=x-1:IF x=0 THEN x=wys
2020 GOSUB 6000:RETURN
2030 y=y+1:IF y=sz+1 THEN y=1
2040 GOSUB 6000:RETURN
2050 x=x+1:IF x=wys+1 THEN x=1
2060 GOSUB 6000:RETURN
2070 y=y-1:IF y=0 THEN y=sz
2080 GOSUB 6000:RETURN
2090 'enter
2100 krz$(yp, xp)="*":GOSUB 2030:RETURN
2110 ENT 1,50,10,30,20,5,40
2120 SOUND 1,30,50,15,,1:RETURN
3000 'poziom
3010 flag=0:dlug=sz:wsp=y:off=1
3020 PRINT #1,"wyraz ";:IF flag=0 THEN PRINT
#1,"poziomo..."; ELSE PRINT #1,"pionowo..."
;
3030 INPUT #1,wyraz$:wyraz$=UPPER$(wyraz$)
3040 FOR a=1 TO LEN(wyraz$):s$=MID$(wyraz$,a
,1): IF s$<"A" OR (s$>"Z" AND s$<CHR$(140))
OR s$>CHR$(147) THEN PRINT#1,"wyraz zawiera
zabroniony znak...":AFTER 200 GOSUB 1500:RE
TURN
3050 NEXT
3060 IF wsp+LEN(wyraz$)>dlug+1 THEN PRINT#1,
"za dlugi wyraz...":AFTER 200 GOSUB 1500:RE
TURN
3070 b=0:c=0:FOR a=1 TO LEN(wyraz$)
3080 IF krz$(y+b,x+c)="." OR krz$(y+b,x+c)=M
ID$(wyraz$,a,1) THEN c=c+flag:b=b+off:NEXT E
LSE IF krz$(y+b,x+c)="*" THEN PRINT#1,"czarn
e pole na drodze...":AFTER 200 GOSUB 1500:RE
TURN ELSE PRINT#1,"konfuzja liter...":AFTER 20
0 GOSUB 1500:RETURN
3090 FOR a=1 TO LEN(wyraz$)
3100 krz$(yp, xp)=MID$(wyraz$,a,1)
3110 IF flag=0 THEN GOSUB 2030 ELSE GOSUB 20
50
3120 NEXT
3130 IF (yp>1 AND flag=0) OR (xp>1 AND flag=
1) THEN krz$(yp, xp)="*":IF flag=0 THEN GOSUB
2030 ELSE GOSUB 2050
3140 GOSUB 1500:RETURN
3150 ' pion
3160 flag=1: dlug=wys: wsp=x :off=0
3170 GOTO 3020

```

```

4000 'odczyt tablicy
4010 a=REMAIN(2):b=REMAIN(3)
4020 INPUT #1,"ODCZYT- nazwa pliku...";plik$

4030 IF plik$="" THEN GOSUB 1500:RETURN
4040 plik$=plik$+".krz"
4050 OPENIN plik$
4060 FOR a=0 TO 21
4070 FOR b=0 TO 25
4080 INPUT #9,krz$(b,a)
4090 NEXT b,a
4100 CLOSEIN
4110 wys=VAL(krz$(0,1)):sz=VAL(krz$(0,0)):ha
slo$=krz$(0,2)
4120 IF haslo$<>"" THEN EVERY 500,2 GOSUB 80
00
4130 EVERY 50,3 GOSUB 8090
4140 WINDOW SWAP 0,2:CLS:CLS#1
4150 GOSUB 2110:RETURN
5000 'zapis tablicy
5010 a=0:WHILE a=0:INPUT #1,"ZAPIS- nazwa pl
iku...";plik$
5020 IF LEN(plik$)<=8 THEN a=1
5030 WEND
5040 IF plik$="" THEN GOSUB 1500:RETURN
5050 plik$=plik$+".krz":krz$(0,2)=haslo$
5060 a=REMAIN(2):b=REMAIN(3)
5070 OPENDOUT plik$
5080 FOR a=0 TO 21
5090 FOR b=0 TO 25
5100 WRITE #9,krz$(b,a)
5110 NEXT b,a
5120 CLOSEOUT:GOSUB 1500
5130 IF haslo$<>"" THEN EVERY 500,2 GOSUB 80
00
5140 EVERY 50,3 GOSUB 8090
5150 GOSUB 2110:RETURN
6000 'kursor
6010 LOCATE yp,xp:PRINT krz$(yp,xp);
6020 LOCATE y,x
6030 PRINT CHR$(22);CHR$(1);
6040 PEN 3:PRINT "_";
6050 PRINT CHR$(22);CHR$(0);
6060 PEN 0:yp=y:xp=x:RETURN
7000 'drukarka
7010 opc$="DKH"
7020 a=REMAIN(2):b=REMAIN(3)
7030 INPUT #1,"Diagram / Krzyzowka / Hasla";
a$:a$=UPPER$(a$)
7040 a$=UPPER$(a$)
7050 PRINT#8,CHR$(27);CHR$(64):'reset
7060 PRINT#8,CHR$(27);CHR$(80):'pica
7070 PRINT#8,CHR$(27);CHR$(71):'enfaza
7080 ON INSTR(opc$,a$) GOSUB 7120,7380,7490
7090 IF haslo$<>"" THEN EVERY 500,2 GOSUB 80
00
7100 EVERY 50,3 GOSUB 8090
7110 GOSUB 1500:GOSUB 2110:RETURN
7120 'Gemini Btar diagram
7130 PRINT#8,SPC(4);
7140 FOR a=1 TO sz
7150 PRINT #8," ";UPPER$(CHR$(64+a));:NEXT
7160 PRINT#8,CHR$(27);CHR$(55);
7170 PRINT#8,CHR$(27);CHR$(62).
7180 FOR a=1 TO wys
7190 PRINT#8,STRING$(4,CHR$(255));
7200 FOR b=1 TO sz
7210 IF a=1 AND b=1 THEN PRINT#8,CHR$(218);C
HR$(196);
7220 IF a=1 AND b<>1 THEN PRINT#8,CHR$(203);
CHR$(196);
7230 IF a<>1 AND b<>1 THEN PRINT#8,CHR$(215)
;CHR$(196);
7240 IF a<>1 AND b=1 THEN PRINT#8,CHR$(198);
CHR$(196);
7250 NEXT b
7260 IF a=1 THEN PRINT#8,CHR$(183) ELSE PRIN
T#8,CHR$(181)

```

```

7270 PRINT#8, CHR$(27);CHR$(61);a; IF a<10 T
HEN PRINT#8, " ";
7280 PRINT#8, CHR$(27);CHR$(62);
7290 FOR b=1 TO sz
7300 PRINT#8, CHR$(179);
7310 IF krz$(b,a)="*" THEN PRINT#8,CHR$(219)
; ELSE PRINT#8,CHR$(255);
7320 NEXT b;PRINT#8,CHR$(179);NEXT a
7330 PRINT#8,STRING$(4,CHR$(255));
7340 FOR a=1 TO sz
7350 IF a=1 THEN PRINT#8,CHR$(211);CHR$(205)
; ELSE PRINT#8,CHR$(207);CHR$(205);
7360 NEXT;PRINT#8,CHR$(188)
7370 PRINT#8,CHR$(27);CHR$(61);RETURN

7380 FOR a=1 TO wys
7390 FOR b=1 TO sz
7400 IF krz$(b,a)<="Z" THEN PRINT#8,krz$(b,a)
);GOTO 7420
7410 c=ASC(krz$(b,a));GOSUB 7440
7420 PRINT#8, " ";NEXT b;PRINT#8;NEXT a
7430 RETURN
7440 PRINT#8,MID$(nor$,c-139,1);
7450 IF c=140 OR c=142 THEN PRINT#8,CHR$(8);
";";
7460 IF c=141 OR c>143 THEN PRINT#8,CHR$(8);
"";
7470 IF c=143 THEN PRINT#8,CHR$(8);"/";
7480 RETURN
7490 'hasla
7500 PRINT#8,"POZIOMO:"
7510 FOR a=1 TO wys;FOR b=1 TO sz
7520 IF krz$(b,a)<>"." AND krz$(b,a)<>*" TH
EN b$=krz$(b,a);c$=CHR$(64+b);dlug=sz;off=1;
GOSUB 7590
7530 NEXT b,a
7540 PRINT#8,"PIONOWO:"
7550 FOR b=1 TO sz;FOR a=1 TO wys
7570 IF krz$(b,a)<>"." AND krz$(b,a)<>*" TH
EN d=a;b$=krz$(b,a);dlug=wys;off=0;GOSUB 759
0
7580 NEXT a,b;RETURN
7590 IF off=1 THEN b=b+1; IF b>dlug THEN RET
URN
7600 IF off=0 THEN a=a+1; IF a>dlug THEN RET
URN
7610 WHILE krz$(b,a)<>"." AND krz$(b,a)<>*"
AND ((b<=dlug AND off=1) OR (a<=dlug AND of
f=0))
7620 b$=b$+krz$(b,a)
7630 IF off=1 THEN b=b+1 ELSE a=a+1
7640 WEND
7650 IF LEN(b$)<2 THEN RETURN

```

```

7660 IF off=1 THEN PRINT#8,a;"-";c$;" "; ELS
E PRINT#8,CHR$(64+b);"-";d$;" ";
7670 FOR e=1 TO LEN(b$)
7680 c=ASC(MID$(b$,e,1))
7690 IF c<140 THEN PRINT#8,MID$(b$,e,1); ELS
E GOSUB 7440
7700 NEXT e
7710 PRINT#8;RETURN
8000 'przerwania
8010 CLS#3;PEN #3,3;PRINT#3,"HASLO:";PEN #3,
0;PRINT #3,haslo$
8020 tekst$=haslo$
8030 por$=SPACE$(LEN(tekst$))
8040 FOR a=1 TO wys;FOR b=1 TO sz
8050 IF krz$(b,a)="." OR krz$(b,a)="*" THEN
ELSE c=INSTR(tekst$,krz$(b,a));IF c>0 THEN M
ID$(tekst$,c,1)=" "
8060 NEXT b,a
8070 IF tekst$<>por$ THEN PRINT#3;PEN #3,3;P
RINT#3,"BRAKI:";PEN #3,0;PRINT#3,tekst$ ELSE
PRINT#3,,"haslo","kompletne"
8080 RETURN
8090 POKE &B632,255;RETURN
9000 'help
9010 CLS#3;PRINT#3," HELP",,;FOR a=6 TO 11
;PRINT#3,CHR$(1);CHR$(a);NEXT;PRINT#3," ku
rsor",,"ENTER *";PRINT#3
9020 PRINT#3,"wyrz:","poziomo CTL";CHR$(1);
CHR$(9);"pionowo CTL";CHR$(1);CHR$(10)
9030 PRINT#3,"F7 zapis","FB odczyt","
F3 zm.hasla","COPY druk.,"TAB help";R
ETURN
9040 'help
9050 a=REMAIN(2);GOSUB 9000
9060 WHILE INKEY$="" :WEND:CLS#3
9070 EVERY 500,2 GOSUB 8000;RETURN
10000 'zmiana hasla
10010 CLS#3;IF haslo$<>"" THEN a=REMAIN(2)
10020 PRINT#3;PRINT#3,haslo$;GOSUB 10050
10030 CLS#3; IF haslo$<>"" THEN EVERY 500,2
GOSUB 8000
10040 GOSUB 2110;RETURN
10050 'haslo
10060 flag=1
10070 WHILE flag=1: PRINT #3,"haslo...";PRIN
T;INPUT #3,haslo$
10080 a=1;flag=0;WHILE a<=LEN(haslo$)
10090 a$=MID$(haslo$,a,1)
10100 IF a$=" " OR (a$>="A" AND a$<="Z") OR
(a$>=CHR$(140) AND a$<=CHR$(147)) THEN a=a+1
ELSE a=LEN(haslo$)+1;flag=1
10110 WEND
10130 WEND;haslo$=UPPER$(haslo$);RETURN

```

POZIOMO:

- 1 -A KONSERWACJA
- 3 -A NUBIA
- 3 -G ELAND
- 4 -E NIZ
- 5 -A EUBEA
- 5 -G ARTEL
- 7 -A TRIAS
- 7 -G MARSZ
- 8 -E YMA
- 9 -A TOPAZ
- 9 -G KOLEJ
- 11 -A RATYFIKACJA

PIONOWO:

- A- 1 KONTESTATOR
- C- 1 NABAB
- C- 7 IMPET
- D- 5 ERA
- E- 1 ELANA

```

E- 7 SYZYF
G- 1 WIEZA
G- 7 MAKAK
H- 5 REA
I- 1 CHART
I- 7 RYLEC
K- 1 ANDALUZYJKA

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

```

	A	B	C	D	E	F	G	H	I	J	K
1											
2	■		■		■		■		■		
3					■						
4	■		■				■		■		
5					■						
6	■	■		■	■		■		■	■	
7					■						
8	■		■				■		■		
9					■						
10	■		■		■		■		■		
11											

Krzysztof MAMCARZ

MERITUM

PROGRAMY 10-15

PROGRAMY EDUKACYJNE Z FIZYKI

```
10 REM TEST Z OPTYKI, AUTOR JAN GOLLA
20 CLEAR 500:CLS

30 PRINT@468,CHR$(23)"TEST Z OPTYKI"
40 PRINT@910,"W CELU KONTYNUACJI"
50 PRINT@974,"NACISNIJ KLAWISZ SPACJA"
60 IF INKEY$(">") " GOTO 60
70 FOR L=1 TO 5:'ILOSC PYTAN
80 READ P$
90 FOR K=1 TO 4:'ILOSC MOZLIWYCH ODPO.
100 READ A$(K)
110 NEXT P=4:'NR. POPRAWNEJ ODPOWIEDZI
120 REM LOSOWA ZMIANA KOLEJNOSC ODPO.
130 FOR K=1 TO 5
140 RANDOM:W1=RND(4):W2=RND(4)
150 IF W2=P THEN P=W1:GOTO 170
160 IF W1=P THEN P=W2
170 B$=A$(W1):A$(W1)=A$(W2):A$(W2)=B$
180 NEXT K
190 CLS
200 PRINT@18,"PYTANIE NR"IL
210 PRINT:PRINT P$:PRINT
220 FOR K=1 TO 4
230 PRINT K:A$(K)
240 NEXT K:PRINT
250 INPUT"PODAJ NUMER ODPOWIEDZI"IX
260 IF X(">")P GOTO 300
270 PRINT"ODPOWIEDZIALES DOBRZE"
280 T=T+1:GOTO 320
300 PRINT"ZLE!!!!"
310 PRINT"ODPOWIEDZ DOBRA MA NUMER"IP
320 IF INKEY$(">") " GOTO 320
330 NEXT L
340 REM OCENA WYNIKOW TESTU
350 IF T=5 THEN O$="BARDZO DOBRA"
360 IF T=4 THEN O$="DOBRA"
370 IF T=3 THEN O$="DOSTATECZNA"
380 IF T<3 THEN O$="NIEDOSTATECZNA"
390 IF T=0 OR T>4 THEN I$="PYTAN"
400 IF T=1 THEN I$="PYTANIE"
410 IF T=2 OR T=3 THEN I$="PYTANIA"
420 CLS
430 PRINT@262,CHR$(23)I
440 PRINT"ODPOWIEDZIALES DOBRZE NA"IT
450 PRINT@340,I$
```



```
460 PRINT
470 PRINT"OTRZYMUJESZ OCENE "ID$
480 IF INKEY$(">") " GOTO 480 ELSE RUN
490 REM PYTANIE NR 1
500 DATA "ZJAWISKO PODWOJNEGO ZALAMANIA
SWIATLA ZACHODZI:"
510 REM ODPOWIEDZI
520 DATA WE WSZYSTKICH CIALACH PRZE
ZROCZYSTYCH
530 DATA WE WSZYSTKICH PRZEZROCZYSTYCH
CIALACH STALYCH
540 DATA WE WSZYSTKICH KRYSZTALACH
550 DATA W CIALACH ANIZOTROPOWYCH
560 REM PYTANIE NR 2
570 DATA "O WSPOLCZ. ZALAMANIA SWIATLA
DECYDUJE:"
580 DATA GESTOSC OBU OSRODKOW
590 DATA PREDKOSC ROZCHODZENIA SIE
SWIATLA W OSRODKACH
600 DATA SPREZYSTOSC OBU OSRODKOW
610 DATA STOSUNEK SIN. KATOW PADANIA
I ZALAMANIA
620 REM PYTANIE NR 3

630 DATA "FOTOMETR TO PRZYRZAD
SLUZACY DO:"
640 DATA USTALANIA CZASU NASWIETLANIA
BLONY FOTOGRAFICZNEJ
650 DATA POMIARU PREDKOSCI SWIATLA
660 DATA POMIARU STRUMIENIA SWIETLNEGO
670 DATA POROWNYWANIA NATEZEN DWOCH
ZRODEL SWIATLA
680 REM PYTANIE NR 4
690 DATA "W SZKLE PREDKOSC SWIATLA
JEST:"
710 DATA DLA WSZYSTKICH BARW
JEDNAKOWA
720 DATA NAJWIEKSZA DLA SWIATLA
FIOLETOWEGO
740 DATA NAJWIEKSZA DLA ROZNYCH BARW
ZALEZNIE OD GATUNKU SZKLA
```

```

750 DATA NAJWIEKSZA DLA SWIATLA
      CZERWONEGO
760 REM PYTANIE NR 5
770 DATA "OBRAZ OTRZYMANY W SOCZEWCE
      ROZPRASZAJACEJ JEST:"
780 DATA ZAWSZE POWIEKSZONY I
      RZECZYWISTY
790 DATA POWIEKSZONY LUB POMNIEJSZONY
      ZALEZNIIE OD OGNISKOWEJ
800 DATA ZAWSZE POMNIEJSZONY I
      RZECZYWISTY
810 DATA ZAWSZE POMNIEJSZONY I
      POZORNY

```



```

10 REM RUCH JEDNOSTAJNIE PRZYSPIESZONY
20 REM AUTOR JAN GOLLA
30 CLS
40 DIM V(10):DIM S(10)
50 FOR X=1 TO 7:PRINT:NEXT
60 PRINT CHR$(23);"RUCH JEDNOSTAJNIE PRZYSPIESZONY"
70 PRINT@B46;"NACISNIJ KLAWISZ SPACJA"
80 IF INKEY$("<")" " GOTO 80
90 CLS
100 INPUT"PODAJ PRZYSPIESZENIE 'A'"#A
110 CLS
120 PRINT TAB(12)"PRZYSPIESZENIE 'A' WYNDOSI "#A
130 PRINT" T="," V="," S="," DS="
140 PRINT
150 FOR T=1 TO 10
160 V(T)=A*T:S(T)=(A*T*T)/2:DS=S(T)-(A*(T-1)*(T-1))/2
170 PRINT T,V(T),S(T),DS
180 NEXT T
190 PRINT@902;"W CELU KONTYNUACJI NACISNIJ KLAWISZ SPACJA"
200 IF INKEY$("<")" " GOTO 200
210 CLS:PRINT@0;"V":GOSUB 400
220 FOR T=1 TO 10
230 ON ERROR GOTO 500
240 SET(T*9,44-V(T)/10)
250 NEXT T
260 IF INKEY$("<")" " GOTO 260
270 CLS:PRINT@0;"S":GOSUB 400
280 FOR T=1 TO 10
290 ON ERROR GOTO 520
300 SET(T*9,44-S(T)/10)
310 NEXT T
320 IF INKEY$("<")" " GOTO 320
330 RUN
400 FOR X=0 TO 41
410 SET(X,44):SET(0,44-X)
420 NEXT X
430 FOR X=42 TO 100
440 SET(X,44)
450 NEXT X
460 PRINT@945;"T"

```

```

470 RETURN
500 IF INKEY$("<")" " GOTO 500
510 RESUME 270
520 IF INKEY$("<")" " GOTO 520
530 RUN

```



```

10 REM RZUT POZIOHY
20 REM AUTOR JAN GOLLA
30 CLS
40 FOR X=1 TO 7:PRINT:NEXT
50 PRINT CHR$(23);TAB(9)"RZUT POZIOHY"
60 PRINT@896;TAB(7)"NACISNIJ KLAWISZ SPACJA"
70 IF INKEY$("<")" " GOTO 70
80 CLS
90 INPUT"PODAJ WYSOKOSC Z KTOREJ RZUCAMY CIALO(H (48)"#H
100 IF H<=0 OR H)=48 GOTO 90
110 INPUT"PODAJ PREDKOSC POCZATKOWA V0(OKOLO 100)"#V0
120 CLS:PRINT TAB(49)"H=";INT(H*10)/10
130 PRINT TAB(48)"V0=";INT(V0*10)/10
140 T=SQR(2*H/9.81):D=V0*T:VK=SQR(V0*V0+2*9.81*H)
150 PRINT:PRINT:PRINT TAB(45)"T=SQR(2*H/9.81)",
160 PRINT TAB(49)"T=";INT(T*10)/10
170 PRINT:PRINT TAB(49)"D=V0*T"
180 PRINT TAB(49)"D=";INT(D*10)/10
190 PRINT:PRINT TAB(42)"VK=SQR(V0*V0+2*9.81*H)",
200 PRINT TAB(48)"VK=";INT(VK*10)/10
210 PRINT:PRINT TAB(48)"NACISNIJ"
220 PRINT TAB(48)"SPACJE"
230 FOR X=0 TO 47:SET(X,47):SET(0,47-X):NEXT
240 FOR X=48 TO 84:SET(X,47):NEXT
250 ON ERROR GOTO 370
260 XS=0:YS=0:FOR T=0 TO 84 STEP .5
270 X=V0/10*T
280 Y=9.81*T*T/20
290 IF X>84 OR Y>H GOTO 310
300 RESET(XS,47-H+YS):SET(X,47-H+Y):XS=X:YS=Y:NEXT
310 FOR T=0 TO 84 STEP 0.5
320 X=V0/10*T:Y=9.81*T*T/20
330 IF X>84 OR Y>H GOTO 350
340 SET(X,47-H+Y):NEXT
350 IF INKEY$("<")" " GOTO 350
360 GOTO 80
370 IF INKEY$("<")" " GOTO 370
380 RESUME 60

```



```

10 REM RZUT UKOSNY
20 REM AUTOR JAN GOLLA
25 CLS
30 FOR X=1 TO 7:PRINT:NEXT
40 PRINT CHR$(23);TAB(10)"RZUT UKOSNY"
50 PRINT@910;"NACISNIJ KLAWISZ SPACJA"
60 IF INKEY$("<")" " GOTO 60

```

70 CLS
 80 INPUT "PODAJ PRĘDKOŚĆ POZATKOWĄ V0 (OKOŁO 100) " ; V0
 90 PRINT "PODAJ KĄT (A) NACHYLENIA WEKTORA PRĘDKOŚCI POZ. "
 100 INPUT "DO POZIOMU W STOPNIACH (0 < A < 90) " ; A
 110 CLS
 120 PRINT:PRINT TAB(5) "V0=" ; V0:PRINT TAB(5) "A=" ; A
 130 A=A*3.14/180
 140 T=2*V0*SIN(A)/9.81
 150 D=V0*V0*SIN(2*A)/9.81
 160 H=V0*V0*SIN(A)*SIN(A)/(2*9.81)
 170 PRINT:PRINT
 180 PRINT "T=2*V0*SIN(A)/9.81",TAB(40) "T=" ; T
 190 PRINT
 200 PRINT "D=V0*V0*SIN(2*A)/9.81",TAB(40) "D=" ; D
 210 PRINT
 220 PRINT "H=V0*V0*SIN(A)*SIN(A)/(2*9.81)",TAB(40) "H=" ; H
 230 PRINT@B40,"W CELU KONTYNUACJI NACISNIJ KLAWISZ SPACJA"
 240 IF INKEY\$(">") " GOTO 240
 250 CLS:FOR X=0 TO 47:SET(X,47):SET(0,47-X):NEXT
 260 FOR X=48 TO 127:SET(X,47):NEXT
 270 ON ERROR GOTO 420
 280 XS=0:YS=0
 290 FOR T=0 TO 40 STEP .5
 300 X=V0*COS(A)*T/4
 310 Y=(V0*SIN(A)*T-9.81*T*T)/8
 320 RESET(XS,47-YS):SET(X,47-Y):XS=X:YS=Y:NEXT
 330 ON ERROR GOTO 400
 340 FOR T=0 TO 127 STEP .5
 350 X=V0*COS(A)*T/4
 360 Y=(V0*SIN(A)*T-9.81*T*T)/8
 370 SET(X,47-Y):NEXT
 380 IF INKEY\$(">") " GOTO 380
 390 GOTO 70
 400 IF INKEY\$(">") " GOTO 400
 410 RESUME 70
 420 RESUME 330



10 REM FIGURY LISSAJOUS
 20 REM AUTOR JAN GOLLA
 25 CLS
 30 FOR X=1 TO 7:PRINT:NEXT
 40 PRINT CHR\$(23)
 50 PRINT TAB(8) "FIGURY LISSAJOUS"
 60 PRINT@B96,TAB(8) "NACISNIJ SPACJE"
 70 IF INKEY\$(">") " GOTO 70
 80 CLS
 90 INPUT "PODAJ STOSUNEK CZĘSTOTLIWOŚCI
 DRGAŃ K" ; K
 100 INPUT "PODAJ PRZESUNIĘCIE FAZOWE
 DRGAŃ F (W RADIANACH) " ; F
 110 PRINT@774,"JESLI CHCESZ ROZPOCZĄC
 RYSOWANIE TO:"
 120 PRINT TAB(20) "NACISNIJ SPACJE"
 130 IF INKEY\$(">") " GOTO 130
 140 CLS
 150 PRINT "K=" ; K:PRINT "F=" ; F
 160 FOR T=1 TO 600

170 X=63+38*SIN(.05*T)
 180 Y=23+21*SIN(K*.05*T+F)
 190 RESET(XS,47):RESET(18,YS)
 200 SET(18,Y):SET(X,47):SET(X,Y)
 210 XS=Y:YS=Y
 220 IF INKEY\$=" " GOTO 80
 230 NEXT
 240 IF INKEY\$(">") " GOTO 240
 250 GOTO 80



10 REM PROGRAM W TRYBIE KONWERSACYJNYM DO TWORZENIA
 20 REM TESTU EGZAMINACYJNEGO
 30 REM AUTOR JAN GOLLA
 35 REM PRZYGOTOWANIE PYTAŃ EGZAMINACYJNYCH
 40 CLEAR 10000
 50 CLS
 60 INPUT "Z JAKIEGO MATERIAŁU JEST TEN TEST" ; Z\$
 70 INPUT "PODAJ ILOŚĆ PYTAŃ" ; N
 80 INPUT "PODAJ ILOŚĆ ODPOWIEDZI" ; M
 85 IF M<2 GOTO 80
 90 INPUT "ZA ILE DOBRYCH ODPOWIEDZI BĘDZIE OCENA DOSTATECZNA" ; A
 91 INPUT "ZA ILE DOBRYCH ODPOWIEDZI BĘDZIE OCENA DOBRA" ; B
 92 INPUT "ZA ILE DOBRYCH ODPOWIEDZI BĘDZIE OCENA BARDZO DOBRA" ; C
 120 CLS:DIM P\$(N),D\$(N,M),P(N)
 130 FOR X=1 TO N
 140 PRINT@66,"PODAJ" ; X;"PYTANIE":INPUT P\$(X):P(X)=M
 150 FOR Y=1 TO M-1
 160 PRINT@66,"PODAJ ZŁĄ ODPOWIEDZ":INPUT D\$(X,Y)
 170 CLS:NEXT
 180 PRINT@66,"PODAJ DOBRĄ ODPOWIEDZ":INPUT D\$(X,Y)
 190 CLS:NEXT
 200 REM URUCHOMIENIE PRZYGOTOWANEGO PROGRAMU
 210 CLS:T=0
 220 PRINT@460,CHR\$(23) "TEST Z " ; Z\$
 230 PRINT@910,"W CELU KONTYNUACJI"
 240 PRINT@976,"NACISNIJ KLAWISZ SPACJA"
 250 IF INKEY\$(">") " GOTO 250
 260 CLS:FOR L=1 TO N
 270 FOR K=1 TO 5
 280 RANDOM:W1=RND(M):W2=RND(M)
 290 IF W1=P(L) THEN P(L)=W2:GOTO 310
 300 IF W2=P(L) THEN P(L)=W1
 310 B\$=D\$(L,W1):D\$(L,W1)=D\$(L,W2):D\$(L,W2)=B\$
 320 NEXT:CLS
 330 PRINT@18,CHR\$(23) "PYTANIE NR" ; L
 340 PRINT:PRINT P\$(L):PRINT
 350 FOR K=1 TO M
 360 PRINT K ; D\$(L,K):NEXT:PRINT
 370 INPUT "PODAJ NUMER POPRAWNEJ ODPOWIEDZI" ; X
 380 IF X(>)P(L) GOTO 400
 390 PRINT "ODPOWIEDZIAŁES DOBRZE":T=T+1:GOTO 410
 400 PRINT "ZŁE!! ODPOWIEDZ DOBRA MA NR" ; P(L)
 410 IF INKEY\$(">") " GOTO 410 ELSE NEXT
 420 IF T<A THEN O\$="NIEDOSTATECZNA"
 430 IF T=A AND T<B THEN O\$="DOSTATECZNA"
 440 IF T=B AND T<C THEN O\$="DOBRA"
 450 IF T=C THEN O\$="BARDZO DOBRA"
 460 CLS:PRINT CHR\$(23)
 470 PRINT@264,"ODPOWIEDZIAŁES DOBRZE NA"
 480 PRINT@330,T;"SPOŚRÓD" ; N;"PYTAŃ"
 490 PRINT:PRINT "OTRZYMUJESZ OCENĘ " ; O\$
 500 IF INKEY\$(">") " GOTO 500 ELSE 200

JAN GOLLA

AMSTRAD

PROGRAM 16

WIEŻA HANOI

Mamy trzy pręty i n krążków o różnych średnicach. Krążki mogą być nakładane na pręty: tworząc w ten sposób wieże. Założmy, że n krążków zostało umieszczonych początkowo na pręcie A w kolejności malejących średnic (na rysunku zilustrowano tę sytuację dla $n = 3$).

Naszym zadaniem jest przesunięcie n krążków z pręta A na pręt C z zachowaniem pierwotnego porządku. Zasady przesuwania są następujące:

- (1) w każdym kroku przesuwa się dokładnie jeden krążek z jednego pręta na inny,
- (2) krążek nie może być nigdy nałożony na krążek o mniejszej średnicy,
- (3) pręt B może być użyty jako „magazyn pomocniczy”.

Dane do programu:

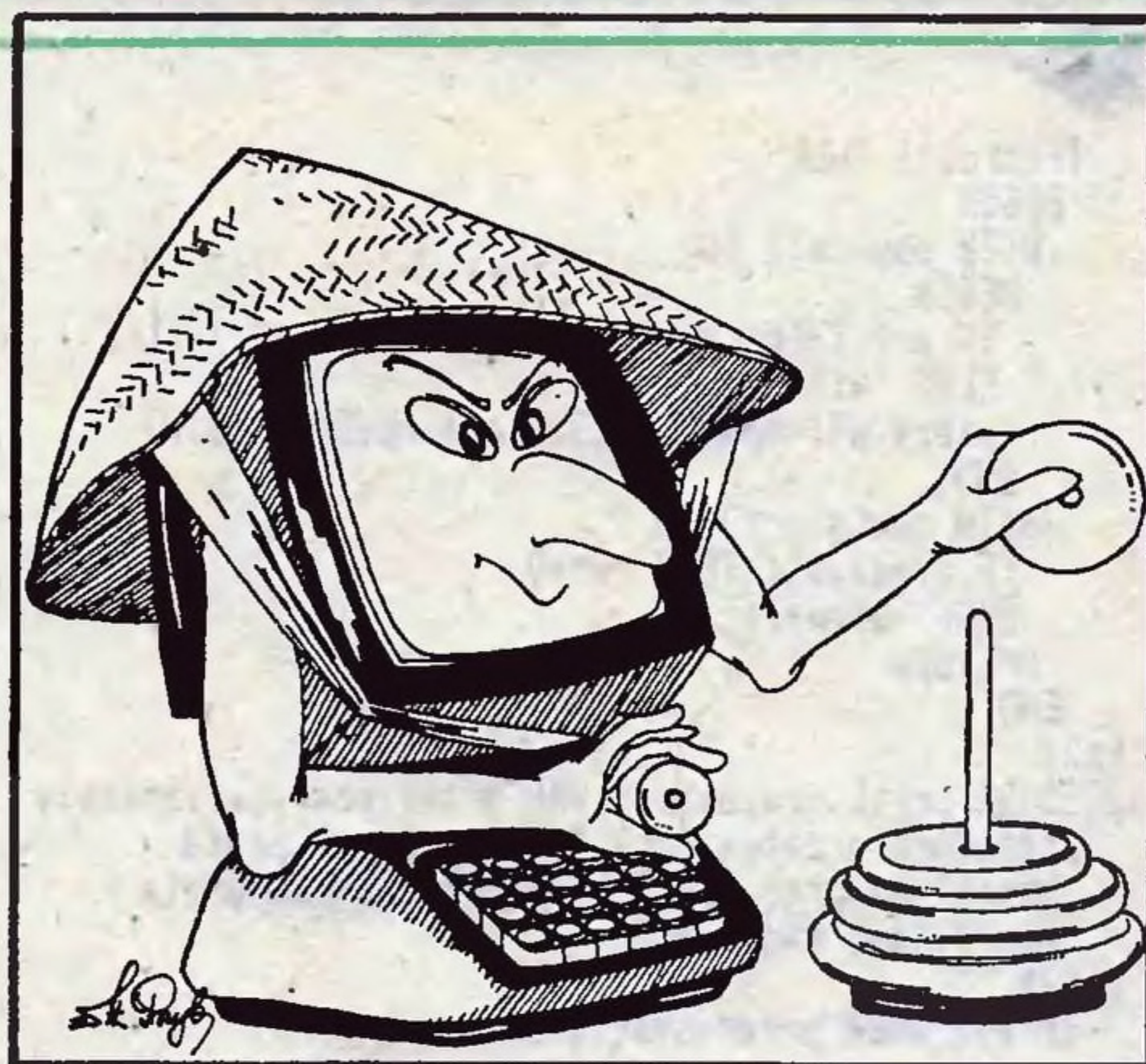
- ilość krążków (1—5) maksymalną ilość krążków można zwiększyć zmieniając w programie wartość stałej MAXILKR,
- żądanie wyświetlenia stanu wież po przełożeniu krążka (1 — tak, 0 — nie).

Kontynuacja wyświetlania wyników (po wyświetleniu 16 wierszy), wciśnięcie dowolnego klawisza. Program napisany jest w Pascalu.

S. R.

N. Wirth: „Algorytmy + struktury danych = programy”, WNT, Warszawa 1980, s. 172, ćwiczenie 3.1.

```
PROGRAM wieza_hanoi;
{$a-}
CONST
  maxilkr=5; {maksymalna ilosc krazkow}
  maxilw=16; {maksymalna ilosc wyswietlanych
             wierszy}
TYPE
  znakp='a'..'c'; {nazwa pretu}
  wyswiez=1..maxilkr; {maksymalna wysokosc wiezy
                     mierzona iloscia krazkow}
  ilw=1..maxilw; {ilosc wyswietlonych wierszy}
  opiswiaz=ARRAY [znakp] OF {opis wiezy}
  RECORD
    w:wyswiez; {wskaznik wierzchołka wiezy}
    wieza:ARRAY [wyswiez] OF wyswiez
  END;
VAR
  ilkr:integer; {zadana ilosc krazkow}
  start, {pret, z ktorego przesuujemy krazki}
  cel,   {pret, na ktory przesuujemy krazki}
  posr:znakp; {pret - magazyn pomocniczy}
  opw:opiswiaz;
  druk:integer;
  ilwierszy:ilw;
PROCEDURE ustawstocz;
  {procedura ustawiajaca poczatkowy stan wiezy}
  VAR
    s:znakp; {zmienne robocze}
    i:wyswiez;
  BEGIN
    WITH opw['a'] DO
      BEGIN
```



```

    w:=maxilkr-ilkr+1;
    FOR i:=w TO maxilkr DO
      wieza[i]:=i-w+1
    END;
    opw['b'].w:=0;
    opw['c'].w:=0
  END;
PROCEDURE czyscekran;
  {procedura kasujaca na zadanie (wciśnięcie
  dowolnego klawisza) poprzedni obraz po
  wyswietleniu maksymalnej ilosci wierszy}
  VAR
    x:char; {zmienne robocze}
  BEGIN
    IF ilwierszy>=maxilw THEN
      BEGIN
        REPEAT
          UNTIL keypressed;
          READ(kbd,x);
          clrscr;
          ilwierszy:=0
        END
      END;
PROCEDURE drukopw;
  {procedura drukujaca numery krazkow
  znajdujacych sie na poszczegolnych pretach}
  VAR
    s:znakp; {zmienne robocze}
    i:wyswiez;
  BEGIN
    FOR s:='a' TO 'c' DO
      WITH opw[s] DO
        BEGIN
          WRITE(' ':10,'wieza ',s,' ');
          IF w>0 THEN
            FOR i:=w TO maxilkr DO
              WRITE(' ':2,wieza[i]:2);
            WRITELN
          END;
          ilwierszy:=ilwierszy+3
        END;
PROCEDURE przen(start,cel:znakp);
  {procedura przenoszaca pojedynczy krazek
  z pretu "start" na pret "cel"}
  BEGIN
    czyscekran;
    WRITELN('przenies krazek z pretu ',
            start,' na pret ',cel);
    ilwierszy:=ilwierszy+1;
```

```

IF druk=1 THEN
BEGIN
WITH opw[cel] DO
BEGIN
IF w=0 THEN w:=maxilkr
ELSE w:=w-1;
wieza[w]:=opw[start].wieza[opw[start].w]
END;
WITH opw[start] DO
IF w=maxilkr THEN w:=0
ELSE w:=w+1;
drukopw
END
END;
PROCEDURE przel(n:wyswierz; VAR start,posr,cel:znakp);
{procedura przesuwajaca "n" krazkow z preta
"start" na pret "cel" z wykorzystaniem preta
"posr" jako magazynu}
BEGIN
IF n=1 THEN przen(start,cel)

ELSE
BEGIN
przel(n-1,start,cel,posr);
przen(start,cel);
przel(n-1,posr,start,cel)
END
END;

BEGIN
REPEAT

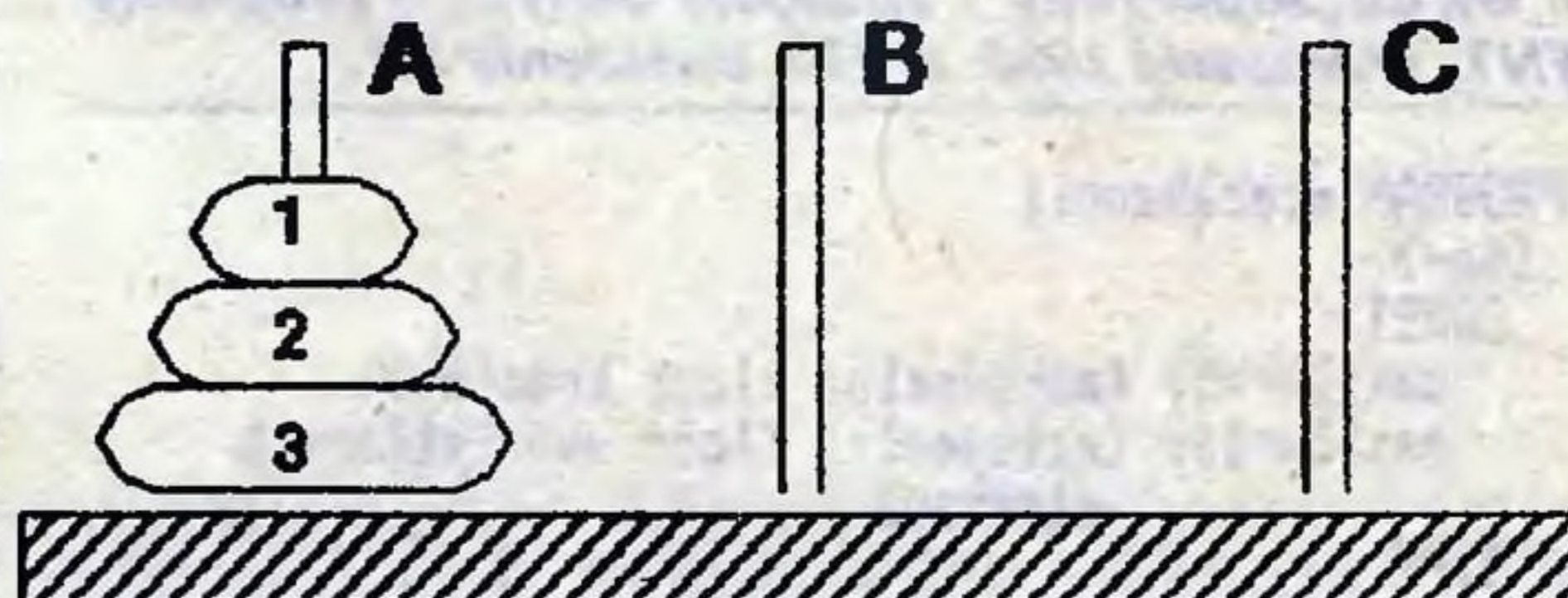
WRITELN('podaj ilosc krazkow (<= ',maxilkr);
READLN(ilkr);
WRITELN('czy chcesz wyswietlac stanywiez po');
WRITELN('przełożeniu kazdego krazka? 1-tak,0-nie');
READLN(druk)
UNTIL (ilkr>0) AND (ilkr<=maxilkr);
clrscr;
ilwierszy:=0;
IF druk=1 THEN
BEGIN
ustawstpocz;
WRITELN('stan początkowy :');
ilwierszy:=ilwierszy+1;
drukopw
END;
start:='a';
cel:='c';
posr:='b';
przel(ilkr,start,posr,cel)
END.

```

```

stan początkowy :
wieza a : 1 2 3
wieza b :
wieza c :
przenies krazek z preta a na pret c
wieza a : 2 3
wieza b :
wieza c : 1
przenies krazek z preta a na pret b
wieza a : 3
wieza b : 2
wieza c : 1
przenies krazek z preta c na pret b
wieza a : 3
wieza b : 1 2
wieza c :
przenies krazek z preta a na pret c
wieza a :
wieza b : 1 2
wieza c : 3
przenies krazek z preta b na pret a
wieza a : 1
wieza b : 2
wieza c : 3
przenies krazek z preta b na pret c
wieza a : 1
wieza b :
wieza c : 2 3
przenies krazek z preta a na pret c
wieza a :
wieza b :
wieza c : 1 2 3

```



STEFAN ROZMUS



16. Pro Lite (Texas Instruments)
Mikroprocesor: 80C88, 4.77MHz
System operacyjny: MS-DOS 2.12,
CP/M 86
Pamięć RAM: 256 KB lub 768 KB
Pamięć masowa: Floppy-Disc 720 KB
Grafika: 640x200
Zasilanie: akumulator
Rozmiary: 29x7.5x33 cm
Waga: 4.8 kg
Cena: 11 300 M



ATARI

PROGRAM 17

MINIASSEMBLER

Prezentowany program nazwany MINIASSEMBLER realizuje dwie zasadnicze funkcje: asemblacji i disassemblacji. Proces asemblacji polega na tłumaczeniu programu napisanego za pomocą nazw mnemonicznych rozkazów mikroprocesora 6502 na kody cyfrowe tych rozkazów, które mogą być w takiej postaci wykonywane przez mikroprocesor. Natomiast disassemblacja jest procesem odwrotnym. Należy zaznaczyć, że program MINIASSEMBLER nie może i nie powinien zastępować prawdziwego (firmowego) Asemblera i Disassemblera oraz nie jest narzędziem przeznaczonym do tworzenia dużych programów w języku maszynowym. Uwzględniając jednak fakt, że przeciętny użytkownik mikrokomputera ma duże trudności ze „zdobyciem” firmowego asemblera, program ten jest szczególnie przydatny dla początkujących programistów w języku maszynowym. Program posiada jeszcze inne funkcje, które zostały dalej dokładnie opisane. Możliwe obszary zastosowań tego programu rozciągają się od pisania i testowania małych programów w języku maszynowym, poprzez „sklejanie” programów w BASIC-u z procedurami w języku maszynowym aż do modyfikacji posiadanego oprogramowania (np. bardzo łatwo za pomocą tego programu „spolszczyć” firmowe programy, zmieniać zakodowane w nich stałe itp.

Po wprowadzeniu programu do pamięci należy go uruchomić instrukcją RUN. Program wyświetli MENU, które informuje o następujących opcjach programu:

M — drukuj pamięć (heksadecymalnie).

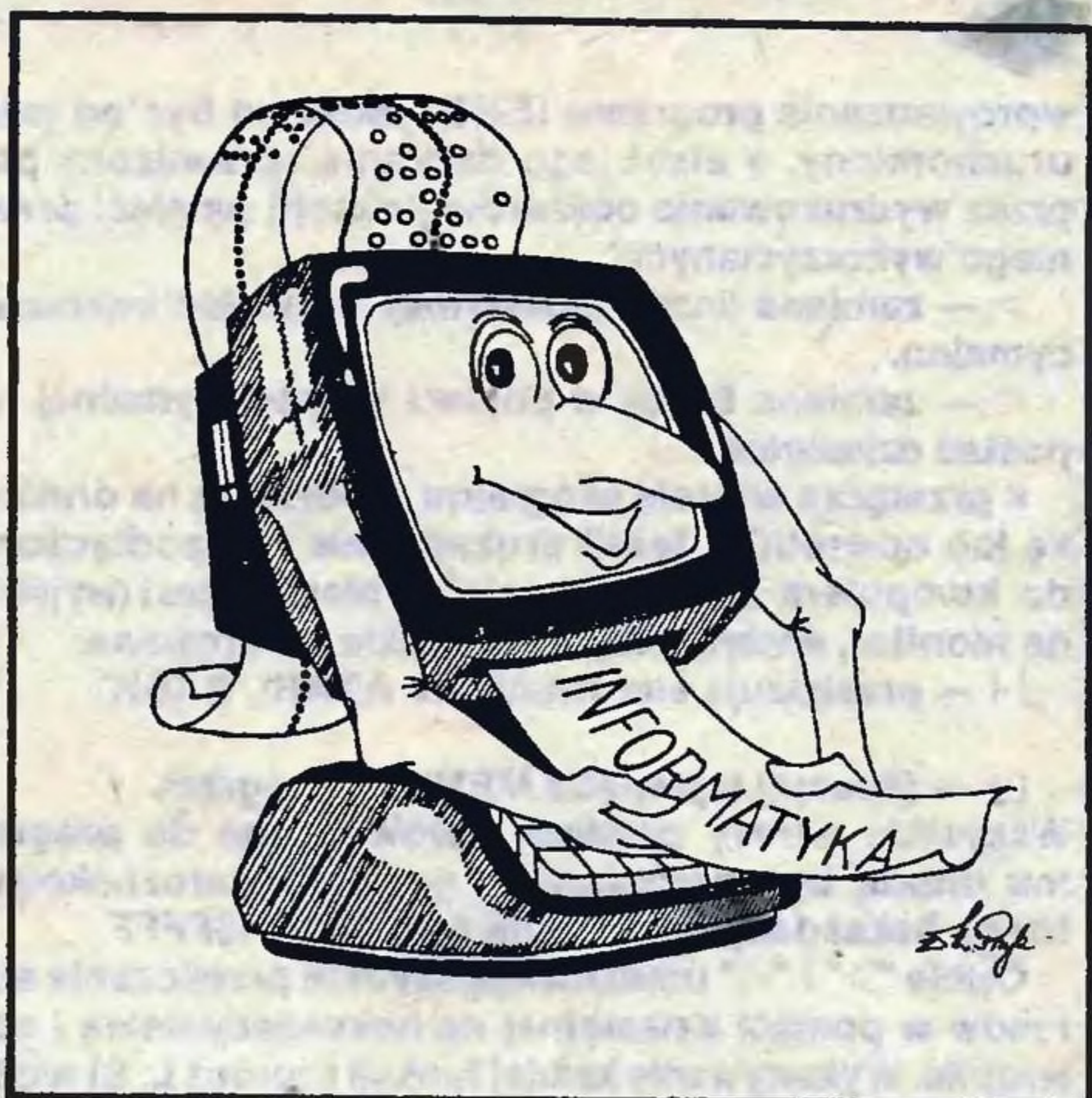
Począwszy od określonego przez użytkownika adresu z zakresu \$0000—\$FFFF wyświetlana jest wierszami zawartość pamięci. W jednym wierszu wyprowadzana jest zawartość 8 bajtów. Każdy wiersz zawiera: adres (heksadecymalnie) pierwszego z 8 bajtów, 8 liczb heksadecymalnych będących zawartością każdego z 8 bajtów i 8 odpowiadających tym zawartościom znaków ATASCII ujętych w (")

B — drukuj pamięć (binarnie)

każdy bajt pamięci jest wyprowadzany w postaci liczby heksadecymalnej i odpowiadającej jej liczby binarnej.

D — wykonaj disassemblację określonego obszaru pamięci.

Wykonywana jest disassemblacja obszaru RAM, którego adres początkowy określa użytkownik. Dla każdego rozkazu podawany jest jego adres (heksadecymalnie), nazwa mnemoniczna rozkazu mikroprocesora, zawartość operandu oraz używane rejestry. Dla każdego rozkazu będącego skokiem względnym podawany jest adres bezwzględny pamięci, do którego przekazywane jest sterowanie programu przy wykonaniu tego skoku. Dla bajtów, których zawartość nie może być zamieniona na nazwę mnemoniczną rozkazu podawany jest znak ATASCII ujęty w apostrofy, natomiast w miejsce nazwy mnemonicznej wyprowadzane są znaki: ???.



P — wpisz do pamięci zawartość heksadecymalną; wyświetlony jest adres bajtu pamięci i jego zawartości w postaci dwuznakowej liczby heksadecymalnej. Następnie użytkownik ma możliwość zmiany tej zawartości poprzez podanie dwuznakowej wartości heksadecymalnej, która zostanie wpisana w miejsce wyświetlonej. Starą wartość można pozostawić wciskając tylko klawisz RETURN. Wówczas program „pokaże” zawartość następnego bajtu pamięci.

T — wprowadź tekst do pamięci.

Pod wskazany adres pamięci wprowadzony jest tekst, który użytkownik wpisuje z klawiatury. Koniec tekstu sygnalizuje wcisnięcie klawisza RETURN.

L — Ładuj do pamięci (Load).

Program podaje wielkość wolnego obszaru RAM w postaci dwóch adresów — dolnego i górnego. Następnie użytkownik ma możliwość wprowadzenia w wolny obszar RAM informacji z dowolnego urządzenia podając specyfikację zbioru np.: D: nazwa zbioru. ext, C lub E. Za pomocą tej opcji można wprowadzić do pamięci komputera zbiory danych, programy w BASIC-u, jak i w kodzie maszynowym. Po zakończeniu ładowania program wyprowadza (w postaci heksadecymalnej) liczbę załadowanych bajtów.

S — wyprowadź z pamięci (SAVE). Użytkownik wprowadza do komputera adres pocz. obszaru RAM, którego zawartość będzie wyprowadzona na określone urządzenie oraz liczbę bajtów do wyprowadzenia (heksadecymalnie). Możliwości jak w opcji L.

A — uruchamia miniassembler.

Użytkownik określa adres RAM, do którego mają być wprowadzone rozkazy w kodzie maszynowym, powstałe w wyniku asemblacji. Asemblacja dokonywana jest wierszami (liniami) tzn. użytkownik wprowadza instrukcję programu w języku asemblera, która jest natychmiast tłumaczona na kod maszynowy, który następnie wraz z operandem jest wyświetlony na monitorze. Składnia tworzonych instrukcji jest analogiczna do obowiązującej w opracowanych dla ATARI asemblerach jak ASEMBLER EDITOR lub MAC65. Ponieważ asemblacja dokonywana jest na pojedynczych wierszach programu. Używanie adresów symbolicznych (etykiet) jest niestety niemożliwe. Po zakończeniu

wprowadzania programu (END) może on być od razu uruchomiony, a efekt jego działania sprawdzony poprzez wydrukowanie obszarów (lokacji) pamięci przez niego wykorzystanych.

> — zamiana liczby dziesiętnej na postać heksadecymalną.

< — zamiana liczby w postaci heksadecymalnej na postać dziesiętną.

przełącza wyjście programu z monitora na drukarkę lub odwrotnie. Jeżeli drukarka nie jest podłączona do komputera automatycznie wybierane jest wyjście na monitor, a użycie tej opcji będzie ignorowane.

! — przekazuje sterowanie do ATARI, BASIC

□ — (spacja) wyświetla MENU — program. Wszystkie adresy pamięci wprowadzone do programu muszą być podawane w postaci czteroznakowej liczby heksadecymalnej z zakresu \$0000-\$FFFF.

Opcje ">" i "<" umożliwiają szybko przeliczanie adresów w postaci dziesiętnej na heksadecymalną i odwrotnie. Wykonywanie każdej funkcji (oprócz L, S) może być przerwane po wciśnięciu klawisza ESCAPE (w trakcie jej realizacji) lub RETURN przed rozpoczęciem realizacji.

Konstrukcja programu umożliwiła łatwe rozbudowanie go o dowolne funkcje potrzebne użytkownikowi. W tym celu w linie 830—990 należy wpisać instrukcję wywołującą podprogram opracowany przez użytkownika, a w linie 11000—29000 tenże podprogram. Przy konstruowaniu własnych podprogramów przydatne mogą być następujące podprogramy z prezentowanego programu. Oto one:

linia 100 — podprogram ADRES
— pobiera z klawiatury podany przez użytkownika czteroznakowy adres obszaru pamięci (w post. heksadecymalnej), dokonuje jego zamiany na liczbę dziesiętną, którą umieszcza w zmiennej AD,

linia 200 — podprogram HEXDEC
— zamienia czteroznakową liczbę heksadecymalną zawartą w zmiennej HEX na liczbę dziesiętną, którą umieszcza w zmiennej N,

linia 300 — podprogram DECHEX
— zamienia liczbę dziesiętną zawartą w zmiennej DEC na liczbę heksadecymalną, którą umieszcza w zmiennej HEX\$. Długość łańcucha HEX\$ (wartość 2 lub 4) znajduje się po wykonaniu podprogramu w zmiennej L,

linia 400 — podprogram POBZ
— pobiera z klawiatury znak, którego kod umieszcza w zmiennej A,

linia 500 — podprogram DECBIN
— zamienia liczbę dziesiętną zawartą w zmiennej DEC na ośmioznakową liczbę binarną, którą umieszcza w BINS,

linia 600 — podprogram BINDEC
— zamienia liczbę binarną zawartą w BIN\$ na liczbę dziesiętną, którą umieszcza w DEC,

linia 700 — podprogram BREAK
— bada czy został wciśnięty klawisz ESCAPE i jeżeli tak, to przerywa realizowaną w chwili naćnięcia funkcję.

Przy wprowadzaniu programu do pamięci komputera proponuję zwrócić szczególną uwagę na cztery problemy:

1. Niektóre linie programu są bardzo długie (taki sposób zapisywania zmniejsza pamięć zajmowaną przez program), aby mogły być przyjęte przez kompu-

ter należy przed rozpoczęciem wprowadzenia programu wykonać instrukcję POKE 82,0

2. Znak „strzałka\ w Inwersie wideo” linia 840, 850... itp. uzyskuje się po wciśnięciu klawiszy ESC/CTRL-2.

3. Wartość zmiennej tekstowej ML\$ bardzo istotna dla poprawnego działania opcji L i S w linii 30030 składa się z następujących sześciu znaków: {h, klawisz ATARI-SHIFT-2, CTRL-P, L, V, klawisz ATARI — małe d} stała ML\$ zawiera trzyzobkowy podprogram w kodzie maszynowym, który przekazuje sterowanie do procedury systemu operacyjnego obsługującej urządzenia zewnętrzne.

4. Problem komentarzy — proponuje nie wprowadzać linii komentarzy ponieważ stanowią one jedynie informacje dla czytelnika, a im większa ich ilość w programie, tym większa pracochłonność związana z wprowadzeniem programu do pamięci komputera i wolniejsze działania programu.

Program zajmuje ok. 10 KB pamięci RAM.

```
1 REM *****
*****
2 REM PROGRAM ASSEMBLER
3 REM *****
*****
10 GOSUB 30000
20 GOTO 30150
95 REM *****
*****
96 REM PODPROGRAM "POBIERZ ADRES
"
97 REM *****
*****
100 POKE KEY,STR-N1:?"Adres poc
z. obszaru":INPUT HEX$:IF LEN(H
EX$)<N4 THEN ? :?"Bład w danyc
h wej.":POP :GOTO
110 GOSUB HEXDEC:IF EF=N1 THEN G
OTO ADRES
120 AD=N:RETURN
195 REM *****
*****
196 REM PODPROGRAM "ZAMIANA HEX
-> DEC"
197 REM *****
*****
200 N=N0:EF=N0:FOR I=N1 TO LEN(H
EX$):NIB$=HEX$(I,I):IF NIB$<"0"
THEN EF=N1
210 IF NIB$<="9" THEN N=N*16+VAL
(NIB$):GOTO 240
220 IF NIB$<"A" OR NIB$>"F" THEN
EF=N1
230 N=N*16+ASC(NIB$)-55
240 NEXT I:RETURN
295 REM *****
*****
296 REM PODPROGRAM "ZAMIANA DEC
-> HEX"
297 REM *****
*****
300 HEX$=""
```

```

310 TEMP=DEC-DEC=INT(DEC/16)*TEM
P=TEMP-DEC*16:IF TEMP<N10 THEN H
EX$(L,L)=STR$(TEMP):GOTO 330
320 HEX$(L,L)=CHR$(TEMP+55)
330 IF L<N1 THEN L=L-N1:GOTO 31
0
340 RETURN
395 REM *****
*****
396 REM PODPROGRAM "POBIERZ ZNAK
"
397 REM *****
*****
400 OPEN #N5,N4,N0,"K:":GET #N5.
A:CLOSE #N5:RETURN
495 REM *****
*****
496 REM PODPROGRAM "ZAMIANA DEC
-> BIN"
497 REM *****
*****
500 BIN$="00000000":NIB$="1":N=S
TR/N2:FOR I=N1 TO N8:IF DEC>N-1
THEN BIN$(I,I)=NIB$:DEC=DEC-N
510 N=N/N2:NEXT I:RETURN
595 REM *****
*****
596 REM PODPROGRAM "ZAMIANA BIN
-> DEC"
597 REM *****
*****
600 DEC=N0:L=STR/N2:FOR J=N1 TO
N8:DEC=DEC+L*VAL(BIN$(J,J)):L=L/
N2:NEXT J:RETURN
695 REM *****
*****
696 REM PODPROGRAM "BADANIE KL.
ESCAPE"
697 REM *****
*****
700 IF PEEK(KEY)=28 THEN ? :? :?
BREAK$:POP:GOTO START
710 RETURN
795 REM *****
*****
796 REM BADANIE WYBRANEJ OPCJI
797 REM *****
*****
800 ? :? ">":
820 GOSUB POBZ
821 IF CHR$(A)=" " THEN GOTO 301
80
830 IF CHR$(A)="M" THEN ? "Druko
wanie pamieci":GOTO 1000
840 IF CHR$(A)="B" THEN ? CHR$(2
53);"Drukowanie pamieci/binarnie
/":GOTO 2000
850 IF CHR$(A)="!" THEN ? "!";CH
R$(253):END
860 IF CHR$(A)="D" THEN ? CHR$(2
53);"Disasemblacja":GOTO 3000
870 IF CHR$(A)="P" THEN ? CHR$(2
53);"Pisz do pamieci":GOTO 4000
880 IF CHR$(A)="T" THEN ? CHR$(2
53);"Pisz tekst do pamieci":GOTO
5000
890 IF CHR$(A)="A" THEN ? CHR$(2
53);"Asemblacja":GOTO 6000
900 IF CHR$(A)="L" THEN ? CHR$(2
53);"Laduj do pamieci":GOTO 7000
910 IF CHR$(A)="S" THEN ? CHR$(2
53);"Wyprowadz z pamieci":GOTO 8
000
920 IF CHR$(A)="#" THEN ? CHR$(2
53);"#":GOTO 9000
930 IF CHR$(A)=">" THEN ? CHR$(2
53);"Zamiana dec -> hex":GOTO 10
000
940 IF CHR$(A)="<" THEN ? CHR$(2
53);"Zamiana hex -> dec":GOTO 11
000
950 IF CHR$(A)=CHR$(125) THEN 30
160
990 GOTO 820
995 REM *****
*****
996 REM PODPROGRAM
997 REM DRUKUJE ZAWARTOSC PAMIEC
I
998 REM *****
*****
1000 GOSUB ADRES
1010 ? #V: ? #V;HEX$;" ";:FOR I=A
D TO AD+N7:DEC=PEEK(I):L=N2:GOSU
B DECHEX: ? #V;HEX$;" ";:NEXT I: ?
#V;" ";
1020 FOR I=AD TO AD+N7:DEC=PEEK(
I):IF V=N4 THEN 1040
1030 ? #V;CHR$(27);:IF DEC=EOL T
HEN DEC=46
1035 GOTO 1050
1040 DEC=DEC-(DEC>127)*128:IF DE
C<32 THEN DEC=46
1050 ? #V;CHR$(DEC);
1060 NEXT I: ? #V;" ";:AD=AD+N8:D
EC=AD:L=N4:GOSUB DECHEX:GOSUB BR
EAK
1070 GOTO 1010
1995 REM *****
*****
1996 REM PODPROGRAM
1997 REM DRUKUJE ZAWARTOSC PAMIE
CI
1998 REM W POSTACI BINARNEJ
1999 REM *****
*****
2000 GOSUB ADRES
2010 ? #V: ? #V;HEX$;" ";:TRAP S
TART:DEC=PEEK(AD):L=N2:GOSUB DEC
HEX:DEC=PEEK(AD):GOSUB DEC BIN
2020 ? #V;HEX$;" ";:BIN$;:AD=AD+N
1:DEC=AD:L=N4:GOSUB DECHEX:GOSUB
BREAK
2030 GOTO 2010
2995 REM *****
*****
2996 REM PODPROGRAM DISASEMBLACJ
I
2997 REM *****
*****
3000 GOSUB ADRES:TRAP N0:D1=-STR
3010 DEC=AD:L=N4:GOSUB DECHEX: ?
#V;HEX$;" ";:DEC=PEEK(AD):D0=DE
C:L=N2:GOSUB DECHEX
3011 ? #V;HEX$;" ";:C=A(D0):M=IN
T(C/1000)
3070 IF M>N0 THEN 3090
3080 ? #V;" ";:IF V=4 THEN 3084
3082 ? #V;CHR$(27);:IF D0=EOL TH
EN D0=46
3083 GOTO 3085
3084 D0=D0-(D0>127)*128:IF D0<32
THEN D0=46
3085 ? #V;CHR$(D0);" " ???":
AD=AD+N1:GOTO 3230
3090 B=INT(C/100)-M*N10:P=INT(C/
N10)-M*100-B*N10:Q=C-M*1000-B*10
0-P*N10:IF B=N1 THEN ? #V;"
";:GOTO 3150
3110 D1=PEEK(AD+N1):DEC=D1:L=N2:
GOSUB DECHEX: ? #V;HEX$;" ";:IF B
=N2 THEN ? #V;" ";:GOTO 3150
3130 D2=PEEK(AD+N2):DEC=N2:L=N2:
GOSUB DECHEX: ? #V;HEX$;" ";:D1=D
1+D2*STR
3150 IF M<N4 OR M>13 OR M=11 OR
M=N7 THEN 3170
3160 IF D1>127 THEN D1=-STR-D1)
3165 D1=AD+D1+N2
3170 ? #V;" ";:N$(M*N3-N2,M*N3)
:" ";:L$(P*N3+N1,P*N3+N3);
3180 DEC=D1:L=N2:IF DEC>STR-N1 T
HEN L=N4
3190 GOSUB DECHEX:IF D1=-STR THE
N ? #V:GOTO 3205
3200 ? #V;" ";HEX$;T$(Q*N3+N1,Q*
N3+N3)

```

Podstawowe wiadomości o programowaniu w języku maszynowym można znaleźć w książce pod red. W. M. Miguta „ATARI BASIC KAW”, Kraków 1985 oraz w numerach 2, 3, 5 miesięcznika „KOMPUTER” z ubiegłego roku.



ELWAO 800 Junior

```

3205 D1=-STR:AD=AD+B
3230 GOSUB BREAK:GOTO 3010
3995 REM *****
*****
3996 REM PODPROGRAM WPISUJE DO P
AMIEMI
3997 REM LICZBY HEKSADECYMALNE
3998 REM *****
*****
4000 GOSUB ADRES
4010 TRAP N0:DEC=AD:L=N4:GOSUB D
ECHEX:? HEX$;" ";DEC=PEEK(AD):
L=N2:GOSUB DECHEX:? HEX$;" ";K=
N0
4030 GOSUB POBZ
4047 IF A=ESC THEN ? :? :? BREAK
$:GOTO START
4050 IF A=EOL THEN GOSUB HEXDEC:
POKE AD,N:AD=AD+N1:? :GOTO 4010
4060 IF A>126 THEN IF K>0 THEN ?
CHR$(A);:K=K-1
4070 IF (A>47 AND A<58) OR (A>64
AND A<71) THEN IF K<N2 THEN ? C
HR$(A);:K=K+N1:HEX$(K,K)=CHR$(A)
4080 GOTO 4030
4995 REM *****
*****
4996 REM PODPROGRAM WPISUJE DO P
AMIEMI
4997 REM TEXT
4998 REM *****
*****
5000 GOSUB ADRES
5010 TRAP N0:DEC=AD:L=N4:GOSUB D
ECHEX:? :? HEX$;" ";:K=N0
5030 GOSUB POBZ
5040 IF A=EOL THEN POKE AD,EOL:?
:GOTO START
5050 IF A=ESC THEN ? :? :? BREAK
$:GOTO START
5060 IF A=126 THEN AD=AD-N1:GOTO
5010
5070 ? CHR$(27);CHR$(A);:POKE AD
,A:AD=AD+1:K=K+1:IF K>15 THEN 50
10
5080 GOTO 5030
5995 REM *****
*****
5996 REM PODPROGRAM ASEMBLACJI
5997 REM *****
*****
6000 GOSUB ADRES:TRAP N0:?
6005 I=N1:DEC=N0:HEX$="":EDIT$=H
EX$:OP$=HEX$:L=N1
6010 GET #N2,A:EDIT$(I,I)=CHR$(A
):IF A>EOL THEN I=I+N1:GOTO 601
0
6020 B=LEN(EDIT$):A=N0:IF B<N3 T
HEN 6230
6030 FOR I=N1 TO B:IF EDIT$(I,I)
<>" " THEN A=I:I=B
6040 NEXT I:EDIT$=EDIT$(A):OP$(N
1,N3)=EDIT$(A,A+N2):C=N0:IF OP$=
"END" THEN GOTO START
6050 FOR I=N1 TO 168 STEP N3:IF
N$(I,I+N2)=OP$ THEN C=(I+N2)/N3:
I=STR
6060 NEXT I:IF C=N0 THEN 6230
6070 TRAP 6210:X=N0:Y=N0:D=N0:H=
N0:Q=N0:FOR I=A+N3 TO B:IF EDIT$
(I,I)<>" " THEN D=I:I=B
6080 NEXT I:OP$(N1,B-D)=EDIT$(D,
B):D=N1
6090 IF OP$(D,D)="A" THEN X=N3:D
=D+N1:L=N1:GOTO 6210
6100 IF OP$(D,D)="#" THEN X=N1:D
=D+N1
6105 IF OP$(D,D)="(" THEN X=N2:D
=D+N1
6110 IF OP$(D,D)="$" THEN H=N1:D
=D+N1
6120 IF OP$(D,D)="$" THEN Q=N1:D
=D+N1

```

```

6130 OP$=OP$(D);D=LEN(OP$):IF D<
N3 THEN 6160
6140 IF OP$(D-N2,D)=","X" THEN Y
=N1:D=D-N3:GOTO 6160
6145 IF OP$(D-N2,D)="),Y" THEN Y
=N2:D=D-N3:GOTO 6160
6150 IF OP$(D-N1,D)=","X" THEN Y=
N3:D=D-N2:GOTO 6160
6155 IF OP$(D-N1,D)="),Y" THEN Y=
N4:D=D-N2
6160 IF OP$(D,D)=")" THEN Y=N5:D
=D-N1
6170 OP$=OP$(N1,D)
6180 IF H THEN HEX$=OP$:GOSUB HE
XDEC:DEC=N:GOTO 6200
6190 IF Q THEN BIN$=OP$:GOSUB BI
NDEC:GOTO 6200
6195 DEC=VAL(OP$)
6200 L=N2+((DEC>=STR) OR (LEN(HE
X$)=N4))
6210 K=C*1000+L*100+X*N10+Y:FOR
I=N0 TO STR-N1:IF A(I)=K THEN B=
I:I=STR:NEXT I:GOTO 6300
6220 NEXT I
6230 ? CHR$(253);CHR$(28);"Synta
x Error!";:FOR I=N0 TO EOL:NEXT
I:? CHR$(156);:GOTO 6005
6300 POKE AD,B:IF L=N2 THEN POKE
AD+N1,DEC
6310 IF L=3 THEN POKE AD+N1,DEC-
INT(DEC/STR)*STR:POKE AD+N2,INT(
DEC/STR)
6315 ? CHR$(28);CHR$(156);:J=L:D
EC=AD:L=N4:GOSUB DECHEX:? HEX$;"
";
6320 FOR I=N0 TO J-N1:DEC=PEEK(A
D+I):L=N2:GOSUB DECHEX:? HEX$;"
";:NEXT I
6330 AD=AD+J:POKE 85,18:? :EDIT$;
:GOTO 6005
6995 REM *****
*****
6996 REM PODPROGRAM WPROWADZA DO
PAMIEMI
6997 REM DANE Z URZADZENIA ZEW.
(LOAD)
6998 REM *****
*****
7000 ? :? "Wolna pamiec od adr:
$";:DEC=PEEK(144)+STR*PEEK(145):
L=N4:GOSUB DECHEX:? HEX$;" - $";
7010 DEC=PEEK(741)+STR*PEEK(742):
L=N4:GOSUB DECHEX:? HEX$
7020 ? "Adres obszaru do ladowan
ia ? $";:INPUT HEX$:IF LEN(HEX$)
<>4 THEN ? :? "Blad w danych wej
.":GOTO START
7030 GOSUB HEXDEC:BH=INT(N/STR):
BL=N-BH*STR
7040 ? "Podaj specyfikacje zbior
u ";:INPUT SPEC$:TRAP START:GS=N
0:IF SPEC$(1,1)="C" THEN GS=STR/
N2
7050 OPEN #N1,N4,GS,SPEC$:POKE I
OBASE+N2,N7:POKE IOBASE+N8,N0:PO
KE IOBASE+N9,PEEK(742)-PEEK(145)
7060 POKE IOBASE+N4,BL:POKE IOBA
SE+N5,BH:I=USR(ADR(ML$)):CLOSE #
N1
7070 SL=PEEK(IOBASE+N8):SH=PEEK(
IOBASE+N9):DEC=SL+STR*SH:L=4:GOS
UB DECHEX:? "Zaladowano bajtow:
$";HEX$:GOTO START
7995 REM *****
*****
7996 REM PODPROGRAM WYPROWADZA Z
PAMIEMI
7997 REM DANE NA URZADZENIE ZEW.
(SAVE)
7998 REM *****
*****
8000 ? "Adres - skad wyprowadzac
? ";:INPUT HEX$:IF LEN(HEX$)<>4
THEN ? :? "Blad w danych wej. "
:GOTO START

```

```

8010 GOSUB HEXDEC:BH=INT(N/STR):
BL=N-BH*STR
8020 ? "Liczba bajtow do wyprowa
dzenia ? ";:INPUT HEX$:IF LEN(HE
X$)<>4 THEN ? :? "Blad w danych
wej. ":GOTO START
8030 GOSUB HEXDEC:SH=INT(N/STR):
SL=N-SH*STR
8040 ? "Podaj specyfikacje zbior
u ";:INPUT SPEC$:TRAP START:GS=N
0:IF SPEC$(1,1)="C" THEN GS=STR/
N2
8050 OPEN #N1,N8,GS,SPEC$:POKE I
OBASE+N2,N10+N1:POKE IOBASE+N8,S
L:POKE IOBASE+N9,SH
8060 POKE IOBASE+N4,BL:POKE IOBA
SE+N5,BH:I=USR(ADR(ML$)):CLOSE #
N1:GOTO START
8995 REM *****
*****
8996 REM PODPROGRAM "WYJSCIE NA
DRUKARKE
8997 REM *****
*****
9000 IF V=N3 THEN V=N4:? "Wydruk
na drukarke":GOTO START
9010 IF V=N4 THEN V=N3:? "Wyjsci
e na monitor":GOTO START
9995 REM Zamiana dec -> hex
10000 TRAP START:? "Liczba dzies
ietna ";:INPUT DEC:L=N4:GOSUB DE
CHEX:? "Heksadecymalna ";HEX$:G
OTO START
10995 REM Zamiana hex -> dec
11000 ? "Liczba heksadecymalna "
;:INPUT HEX$:GOSUB HEXDEC:IF EF=
N1 THEN GOTO START
11010 ? "Dziesietna ";N:GOTO ST
ART
29995 REM *****
*****
29996 REM PODPROGRAM INICJUJACY
29997 REM DZIALANIE PROGRAMU
29998 REM *****
*****
30000 RESTORE 30000:POKE 559,0
30010 READ N0,N1,N2,N3,N4,N5,N6,
N7,N8,N9,N10,N12,ESC,ADRES,EOL,H
EXDEC,STR,DECHEX,POBZ,DECBIN,BIN
DEC,BREAK,KEY,STAR
30020 DATA 0,1,2,3,4,5,6,7,8,9,1
0,12,27,100,155,200,256,300,400,
500,600,700,764,800
30030 DIM ML$(N6),SPEC$(N10+N5):
ML$="h LV":ML$(2,2)=CHR$(162):M
L$(3,3)=CHR$(16):ML$(6,6)=CHR$(2
28)
30031 IOBASE=848:V=N3
30100 DIM N$(171),A(STR-N1),L$(N
12),T$(N10+N8),HEX$(4),NIB$(N1),
BIN$(N8),BREAK$(N10+N1),EDIT$(N1
0*N4),OP$(N12)
30110 N$(N1,99)="ADCANDASLBCCBCS
BEQBIBMBNEBPLBRKBVCBVSCLCCLDCL
ICLVCMPCPCPYDECDEXDEYEDORINCINXI
NYJMPJSRLDLDXLDYL
30120 N$(100,171)="NOPORAPHAPHP
LAPLPROLRTIRTSSBCSECESEIESTASTX
STYXTAYTSXTXATXSTYAROR???"
30140 L$=" # ( A ":T$="
X)),Y,X ,Y ) ":L$(2,2)=CHR$(31)
:L$(3,3)=CHR$(30):L$(5,5)=CHR$(3
0)
30141 L$(6,6)=CHR$(31):L$(8,8)=C
HR$(31):L$(9,9)=CHR$(30):T$(2,2)
=CHR$(31):T$(3,3)=CHR$(30)
30142 BREAK$="** BREAK **":FOR I
=N0 TO STR-N1:READ A:A(I)=A:NEXT
I:RETURN
30150 OPEN #N2,N12,N0,"E":OPEN
#N3,N8,N0,"E":TRAP 30160:OPEN #
N4,N8,N0,"P":GOTO 30170
30160 CLOSE #N4:OPEN #N4,N8,N0,"
E:"

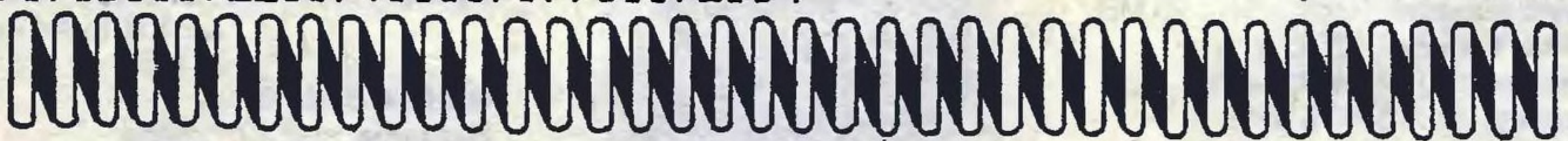
```

```

30170 GRAPHICS N0:SETCOLOR N1,N0
,N10:SETCOLOR N2,N12+N2,N2:SETCO
LOR N4,N12+N2,N2
30180 PRINT CHR$(125):PRINT :PRI
NT "
      ASEMBLER WIERSZO
WY      ":PRINT :PRINT "
      M E N U":P
30182 ? "M - Drukuj zawartosc pa
mieci"
30183 ? "B - Drukuj zawartosc pa
mieci/binarnie/"
30184 ? "D - Disasemblacja"
30185 ? "P - Pisz do pamieci (he
ksadecymalnie)"
30186 ? "T - Pisz tekst do pamie
ci"
30187 ? "A - Asemblacja"
30188 ? "L - Laduj do pamieci -
LOAD"
30189 ? "S - Wyprowadz z pamieci
- SAVE"
30190 ? "> - Zamiana dec -> hex"
30191 ? "< - Zamiana hex -> dec"
30192 ? "# - Wyjscie na drukarke
"
30193 ? "! - Sterowanie do ATARI
BASIC"
30194 ? " - Powrot do M E N U"
30195 ? :? :? "
W
YBIERASZ ? ":POKE 82,N0:GOTO ST
ART
30200 DATA 11100,35221,0,0,0,352
00,3200,0,37100,35210,3130,0,0,3
5300,3300,0,10200,35222,0,0,0,35
203,3203,0,14100
30210 DATA 35304,0,0,0,35303,330
3,0,29300,2221,0,0,7200,2200,402
00,0,39100,2210,40130,0,7300,230
0,40300,0,8200
30220 DATA 2222,0,0,0,2203,40203
,0,44100,2304,0,0,0,2303,40303,0
,41100,24221,0,0,0,24200,33200,0
,36100,24210
30230 DATA 33130,0,28300,24300,3
3300,0,12200,24222,0,0,0,24203,3
3203,0,16100,24304,0,0,0,24303,3
3303,0,42100,1221
30240 DATA 0,0,0,1200,56200,0,38
100,1210,56130,0,28325,1300,5630
0,0,13200,1222,0,0,0,1203,56203,
0,46100,1304
30250 DATA 0,0,0,1303,56303,0,0,
47221,0,0,49200,47200,48200,0,23
100,0,53100,0,49300,47300,48300,
0,4200,47222,0,0
30260 DATA 49203,47203,48204,0,5
5100,47304,54100,0,0,47303,0,0,3
2210,30221,31210,0,32200,30200,3
1200,0,51100,30210
30270 DATA 50100,0,32300,30300,3
1300,0,5200,30222,0,0,32203,3020
3,31204,0,17100,30304,52100,0,32
303,30303,31304
30280 DATA 0,20210,18221,0,0,202
00,18200,21200,0,27100,18210,221
00,0,20300,18300,21300,0,9200,18
222,0,0,0,18203
30290 DATA 21203,0,15100,18304,0
,0,0,18303,21303,0,19210,43221,0
,0,19200,43200,25200,0,26100
30300 DATA 43210,34100,0,19300,4
3300,25300,0,6200,43222,0,0,0,43
203,25203,0,45100,43304,0,0,0,43
303,25303,0

```

Oprac. H. KRASUSKI



17. Sharp PC 5000

Mikroprocesor: 8088, 4.77MHz
System operacyjny: MS-DOS 2.0
Pamięć RAM: 128 KB lub 256 KB
Pamięć masowa: Floppy-Disc 360 KB
Grafika: 640x80
Zasilanie: sieć lub akumulator
Rozmiary: 32.6x8.7x30.5 cm
Waga: 4.4 kg
Cena: 5000 M

18. Tandy 102

Mikroprocesor: 80C85, 2.4MHz
System operacyjny: ROM-based
Pamięć RAM: 24 KB lub 32 KB
Pamięć masowa: Mikrofloppy 100 KB
Grafika: 240x64
Zasilanie: sieć lub bateria
Rozmiary: 30x4.2x21.5 cm
Waga: 1.6 kg
Cena: 1400 M



19. Tandy 200

Mikroprocesor: 80C85, 2.4MHz
System operacyjny: ROM-based
Pamięć RAM: 24 KB lub 72 KB
Pamięć masowa: Mikrofloppy 100 KB
Grafika: 240x128
Zasilanie: sieć, bateria lub akumulator
Rozmiary: 29.9x5.6x21.6 cm
Waga: 2.1 kg
Cena: 2500 M



20. Toshiba T 1100

Mikroprocesor: 80C88, 4.77MHz
System operacyjny: MS-DOS 2.11
Pamięć RAM: 256 KB lub 512 KB
Pamięć masowa: Floppy-Disc 20 MB
Grafika: 640x200, 320x200
Zasilanie: sieć lub akumulator
Rozmiary: 31x6.6x30.5 cm
Waga: 4.2 kg
Cena: 4700 M



22. Zenith Z 181 Portable PC

Mikroprocesor: 80C88, 4.77MHz
System operacyjny: MS-DOS 3.1
Pamięć RAM: 640 KB lub 1.2 MB
Pamięć masowa: 2 Floppy-Disc po 720 KB, dysk twardy 20 MB

Grafika: 640x200
Zasilanie: sieć lub bateria
Rozmiary: 29x8x34 cm
Waga: 6 kg



21. Zenith Z 171

Mikroprocesor: 80C88, 4.77MHz
System operacyjny: MS-DOS 3.1
Pamięć RAM: 256 KB lub 640 KB
Grafika: 640x200
Zasilanie: sieć lub akumulator
Rozmiary: 33x24.1x16.8 cm
Waga: 6 kg
Cena: 6780 M



„IKS” — dodatek „Żołnierza Wojskość”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogołek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Mliier, Michał Przybyłowski, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 813664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 8570. Nr. ind. 361682. K-81