

PROGRAMY 2

INFORMATYKA KOMPUTERY SYSTEMY

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



W numerze:

AMSTRAD

- Grafika „Wież Hanoi” — str. 12
- Technologia przenoszenia programów ze Spectrum na Amstrada — str. 21
- Rezystor — str. 24
- Hiperbole równoosiowe i sprzężone — str. 27
- Gra Hamiltona — str. 29

ATARI

- Automatyczny start programów napisanych w Basic-u na Atari 800 XL — str. 6
- Dysk — Menu — str. 7
- Rozszczepienie światła w pryzmacie — str. 8
- Regresja i korelacja — str. 18

COMMODORE

- „Życie” — str. 14

SHARP

- Dżungla — str. 16

SPECTRUM

- Kod maszynowy — str. 3
- Rubik — str. 10

Okladka i rysunki przy programach: Michał Przybyłowski

Oto drugi numer zeszytu programów komputerowych. Sądzymy, że każdy użytkownik komputera znajdzie w nim coś dla siebie. **Ale uwaga** — nasze programy proponujemy traktować jedynie jako propozycje. Każdy z nich można nie tylko poprawić zaopatrując na przykład w atrakcyjną grafikę, ale także przetłumaczyć na dialekt zrozumiały przez inny komputer. Zabieg ten choć wymaga nieco praktyki i wiedzy opłaci się sowicie.

I jeszcze przypominamy: niektóre z programów będą emitowane przez „Radiokomputer” — program rozgłośni harcerskiej. Słuchajcie go zawsze we wtorki o godzinie 22.50 w programie IV i w piątki o 16.00 również w programie V.

REDAKCJA

WARSZTATY KOMPUTEROWE'87

Pod patronatem Ogólnopolskiej Federacji Klubów Komputerowych Młodych Mistrzów Techniki ZSMP w dn. 19–21 marca br. w Pałacu Kultury i Nauki odbyły się warsztaty komputerowe INFOKLUB'87. Z ogólnej liczby 255 klubów, do Warszawy zaproszono dwadzieścia trzy. Stoisko Centralnej Składnicy Harcerskiej, punkt naprawy komputerów oraz wystawa firmy „LASER”, stanowiły dodatkowe atrakcje dla zwiedzających.

Jaki cel miała ta impreza?

Organizatorzy nie ukrywali, że tego typu spotkania mają służyć przede wszystkim edukacji informatycznej młodzieży oraz wspieraniu procesu komputeryzacji społeczeństwa i gospodarki narodowej. Wynikało to również z celów, jakie stawiają sobie same kluby.

Podczas trwania „warsztatów” mo-

żna było wymienić programy, skopiować dyskietki z ciekawymi grami, kupić specjalistyczną prasę, a nawet oddać do naprawy komputer i być świadkiem usunięcia usterki.

Przedstawiciele federacji wykazali dużo pomysłowości i inicjatywy w zorganizowaniu INFOKLUBU'87, ale jeszcze lepiej zaprezentowały się poszczególne kluby. Niektóre z nich przywiozły prawie wszystkie typy komputerów spotykanych w Polsce. Uzupełnieniem do tego był dodatkowy sprzęt z pełnym oprogramowaniem służącym zarówno do szkolenia, jak i zabawy.

Jakie mają osiągnięcia? Jakie plany? Na te pytania szukaliśmy odpowiedzi w bezpośrednich rozmowach z przedstawicielami klubów.

*Tekst i foto:
Rogoń Ryszard*

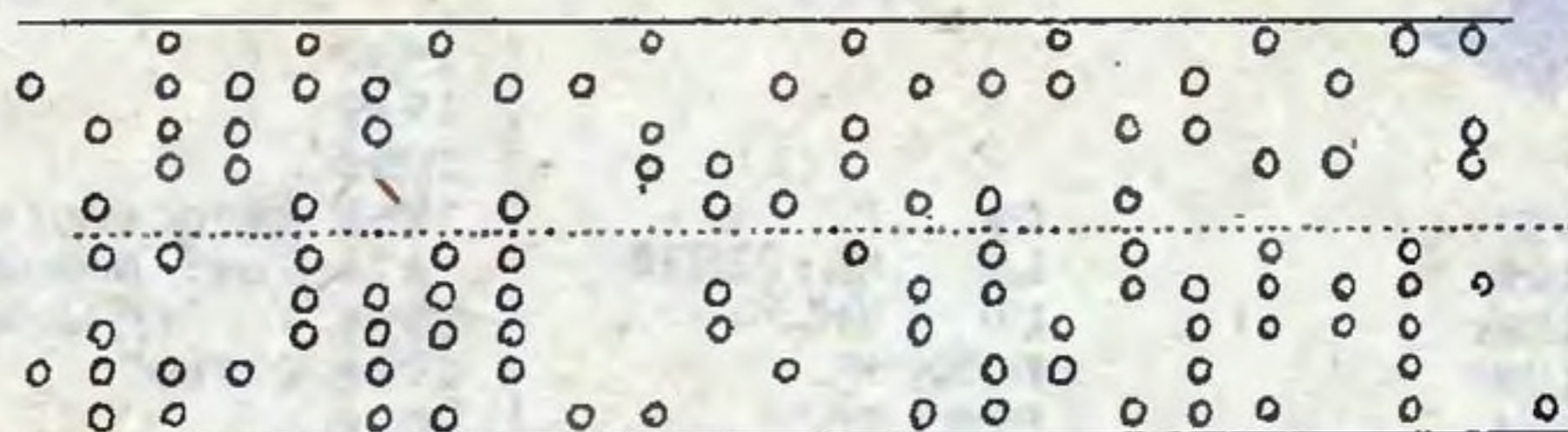


Ogólnopolska Federacja Klubów Komputerowych Młodych Mistrzów Techniki ZSMP skupia 255 klubów

PIERWSZY KLUB KOMPUTEROWY MMJ
POWSTAŁ W 1. ROKU W KATOWICACH
ZBIOROWYMI CZŁONKAMI FEDERACJI SA M. IN.
WARSZAWSKI KLUB UŻYTKOWNIKÓW
MIKROPROCESORÓW (OKOŁO 200 OSÓB
KORRESPONDENTÓW)
WIELEŃSKI KLUB UŻYTKOWNIKÓW
MIKROKOMPUTERÓW

SEKRETARIAT FEDERACJI
WARSZAWA, UL. ŚMOLNA 40
KOD 00-920





Asembler

Napisanie programu w języku wewnętrznym procesora jest zajęciem żmudnym i powinno się ograniczać do tych przypadków, gdzie nie można z różnych względów zastosować języka wysokiego poziomu. Niemniej zwolenników kodu maszynowego nie brakuje i stąd prezentowany program.

Pozwala on poznać strukturę obszaru Basic'a (opcja P), może służyć jako narzędzie do kontroli pracy w tym języku. Opcja Z wyświetla użyte w Basic'u zmienne, a po wyborze klawisza W otrzymamy podstawowe dane o programie. Inne używane klawisze: D — kasowanie ostatniego znaku podczas wpisywania numeru linii, BREAK — powrót do menu.

Z poziomu interpretera Basic'a uruchamiamy program przez np. RANDOMIZE USR 64200, a opuszczamy go opcją Q. **Nie zapominajmy o obniżeniu RAMTOP komendą CLEAR 64199.**

Struktura programu w asemblerze powinna być szczególnie uporządkowana, gdyż tylko wtedy możliwy jest końcowy sukces. Spójrzmy na wydruk. Na początku wprowadzamy adres, od którego asembler będzie składował kod wynikowy (dyrektywa ORG). Rozkaz ENT nie jest konieczny, ale pozwala na uruchamianie programu z poziomu asemblera, co staje się wygodne przy testowaniu poszczególnych procedur lub fragmentów programu. Teraz wczytujemy skok do miejsca inicjalizacji kodu (START). W ten sposób tworzy się miejsce na „stałe fragmenty gry”. W tym przypadku są to definicje tekstów, etykiety, zmienne programu (numer linii i licznik przesunięć ekranu) oraz procedury. Cały ten blok nie tylko poprawia czytelność struktury kodu, lecz przede wszystkim znacznie ułatwia programowanie. Oczywiście blok operacyjny rozpoczyna etykieta START.

Użyte zmienne systemowe ROM są lub będą opisane w miesięczniku „IKS”, pozostaje więc krótko opisać procedury ROM, z których program korzysta. A zatem:

- 5633 — ustala kanał do transmisji danych (numer kanału w A),
- 3082 — wyprowadza teksty upo-

- rządowane jak na wydruku,
- 6683 — wyprowadza na używany kanał liczbę z zakresu 0...9999,
- 8202 — ustala pozycję dla wydruku tekstu (linia w C, kolumna w B),
- 8220 — ustala pozycję kolumny dla wydruku (w C),
- 8020 — sprawdza, czy przyciśnięto klawisz BREAK (lub CAPS + SPACE),
- 6584 — przeszukuje obszar programu lub zmiennych BASIC,
- 3582 — skroluje ekran o jedną linię w górę,
- 6654 — wydobywa liczbę umieszczoną cyfra po cyfrze w pamięci (do BC),
- 6510 — wyprowadza adres linii o podanym numerze w HL,
- 11547 — testuje, czy bajt w akumulatorze jest cyfrą,
- 11563 — umieszcza parę BC na stosie kalkulatora,
- 11747 — wyprowadza zawartość stosu kalkulatora na używany kanał.

Wszystkie adresy podane są w programie dziesiętnie, co może przeszkadzać „heksadecymalnym”. W takim wypadku trzeba po prostu podzielić adres przez 16 lub skorzystać z programu MONITOR.

Krzysztof MAMCARZ

```

10 ;** BASICDUMP '85
20 ;GENS3M2 assembler
30      ORG 64200
40      ENT 64200
50      JP  START
60 ;teksty
70 NAPISY      DEFB 128
80      DEFM "1985 BASI"
90      DEFM "C DUMP"
100     DEFB " "+128
110     DEFM "obszar prog"
120     DEFM "ramu pusty"
130     DEFB " "+128
140     DEFM "obszar zmie"
150     DEFM "nnych pusty"
160     DEFB " "+128
170     DEFM "koniec obsza"
180     DEFM "ru programu"
190     DEFB " "+128
200     DEFM "P budowa "
210     DEFM "programu"
220     DEFB " "+128
230     DEFM "Z lista "
240     DEFM "zmiennych"
250     DEFB " "+128
260     DEFM "W wskazniki"
270     DEFB " "+128
280     DEFM "Q basic"
290     DEFB " "+128
300     DEFM "od numeruJ "
    
```

```

310     DEFM " linii"
320     DEFB " "+128
330     DEFM "program"
340     DEFB " "+128
350     DEFM "zmienne"
360     DEFB " "+128
370     DEFM "wolne"
380     DEFB " "+128
390     DEFM "ramtop"
400     DEFB " "+128
410     DEFM "liczba"
420     DEFB " "+128
430     DEFM "(5 bajtow)"
440     DEFB " "+128
450     DEFM "linia nr."
460     DEFB " "+128
470     DEFM "dlug.linii"
480     DEFB " "+128
490     DEFM "koniec linii"
500     DEFB " "+128
510     DEFM "ktrl"
520     DEFB " "+128
    
```

```

530 ;etykiety
540 ;zmienna syst.
550 BORDCR      EQU 23624
560 KSTATE      EQU 23556
570 ATTR_T      EQU 23695
580 VARS         EQU 23627
590 PROG        EQU 23635
600 ELINE       EQU 23641
610 STKEND      EQU 23653
620 RAMTOP      EQU 23730
630 ; procedury ROM
640 OPENCH      EQU 5633
650 POMSG       EQU 3082
660 OUTNUM      EQU 6683
670 NUMBER      EQU 11547
680 AT          EQU 8202
690 TAB         EQU 8220
700 NEXT        EQU 6584
710 SCROLL      EQU 3582
720 STK_BC      EQU 11563
730 PR_FP       EQU 11747
740 BRTEST      EQU 8020
750 GETLN       EQU 6654
760 ADDRNLN     EQU 6510
770 ;zmienne programu
780 BUFOR       DEFS 5
790 LICZ        DEFB 10
800 ;procedura CLS
810 CLS         LD  A,2
820             CALL OPENCH
830             LD  HL,16384
840             LD  (HL),0
850             PUSH HL
860             POP  DE
870             INC  DE
880             LD  BC,6143
890             LDIR
900             RET
910 ;procedura PRTSTR(A)
920 ;we: A-nr. napisu
930 PRTSTR      LD  DE,NAPISY
940             JP  POMSG
950 ;procedura BORDER(A)
960 ;we: A-kolor ramki
970 BORDER      OUT (254),A
980             RLCA
990             RLCA
1000            RLCA
1010            LD  (BORDCR),A
1020            RET
1030 ;procedura KOLORY(A,C)
1040 ;we: A-paper
1050 ;      C-ink
1060 KOLORY     RLCA
1070            RLCA
1080            RLCA
    
```


3580 DUMP	CALL CLS	4010	JP START	4430	JR NZ,CTRL
3590	LD A,7	4020 CONT	CALL PR_ADR	4440	PUSH HL
3600	LD C,1	4030	PUSH HL	4450	LD A,13
3610	CALL KOLORY	4040	LD A,15	4460	CALL PRTSTR
3620	LD HL,(VARS)	4050	CALL PRTSTR	4470	POP HL
3630	LD DE,(PROG)	4060	POP HL	4480	CALL PRZES
3640	AND A	4070	LD B,(HL)	4490	INC HL
3650	SBC HL,DE	4080	INC HL	4500	CALL PR_ADR
3660	JR NZ,NRLIN	4090	LD C,(HL)	4510	PUSH HL
3670	LD A,1	4100	CALL OUTNUM	4520	LD A,14
3680	CALL PREDIT	4110	CALL PRZES	4530	CALL PRTSTR
3690	CALL BREAK	4120	CALL PR_ADR	4540	POP HL
3700	JP START	4130	CALL PRZES	4550	CALL PRZES
3710 NRLIN	LD BC,10	4140	INC HL	4560	INC HL
3720	CALL AT	4150	CALL PR_ADR	4570	LD B,4
3730	LD A,8	4160	PUSH HL	4580 FLOAT	PUSH BC
3740	CALL PRTSTR	4170	LD A,16	4590	CALL PR_ADR
3750	LD DE,BUFOR	4180	CALL PRTSTR	4600	CALL PRZES
3760 LINIA	CALL KURSOR	4190	POP HL	4610	INC HL
3770	CALL KLAU	4200	LD C,(HL)	4620	POP BC
3780	CP 13	4210	INC HL	4630	DJNZ FLOAT
3790	JR Z,KOMPL	4220	LD B,(HL)	4640	JR INLINE
3800	CALL CYFRA	4230	PUSH HL	4650 CTRL	CP 16
3810	CP "D"	4240	CALL STK_BC	4660	JR C,NORMAL
3820	CALL Z,DELETE	4250	CALL PR_FP	4670	CP 21
3830	JR LINIA	4260	POP HL	4680	JR NC,NORMAL
3840 KOMPL	LD A,255	4270	CALL PRZES	4690	PUSH HL
3850	LD (DE),A	4280	CALL PR_ADR	4700	LD A,18
3860	CALL CLS	4290	CALL PRZES	4710	CALL PRTSTR
3870	LD HL,BUFOR	4300	INC HL	4720	POP HL
3880	CALL GETLN	4310 INLINE	CALL PR_ADR	4730	LD A,(HL)
3890	LD H,B	4320	LD A,(HL)	4740	ADD A,201
3900	LD L,C	4330	CP 13	4750	RST 16
3910	CALL ADDRIN	4340	JR NZ,KOD14	4760	CALL PRZES
3920 LIN1	PUSH HL	4350	PUSH HL	4770	INC HL
3930	LD DE,(VARS)	4360	LD A,17	4780	CALL PR_ADR
3940	AND A	4370	CALL PRTSTR	4790 WYK	CALL PRZES
3950	SBC HL,DE	4380	POP HL	4800	INC HL
3960	POP HL	4390	CALL PRZES	4810	JR INLINE
3970	JR NZ,CONT	4400	INC HL	4820 NORMAL	RST 16
3980	LD A,3	4410	JR LIN1	4830	JR WYK
3990	CALL PREDIT	4420 KOD14	CP 14		
4000	CALL BREAK				



Wojewódzki Klub Komputerowy MMT „Sobótka” działa pod patronatem Uniwersytetu Mikołaja Kopernika w Toruniu. Siedzibą klubu jest Młodzieżowy Dom Kultury „Kameleon”. Może właśnie dlatego członkowie „Sobótki” mają tak zmienne szczęście. Dostając 31 grudnia 1986 roku 300 tys. złotych musieli wydać je właśnie w tym samym dniu. Jak kupić potrzebny sprzęt w tak krótkim czasie? Tej tajemnicy nikt nie chciał nam zdradzić.

Oprócz typowej działalności szkoleniowej dla młodzieży i dorosłych, klub prowadzi akcję „alfabetyzacji komputerowej” dla dzieci od lat 6 do 10.



ATARI

PROGRAM 2-3

AUTOMATYCZNY START PROGRAMÓW NAPISANYCH W BASICU NA ATARI 800XL

Programy firmowe dla mikrokomputerów ATARI 800XL napisane w kodzie maszynowym i dostępne na dyskietkach, np. gry, programy użytkowe, posiadają bardzo wygodny sposób uruchamiania. Po prostu: wkłada się dyskietkę do stacji dysków i włącza komputer. Natomiast uruchomienie programu napisanego w języku BASIC wymaga od użytkownika napisania komendy **LOAD** z odpowiednią nazwą zbioru i dopiero po wykonaniu jej przez mikrokomputer, komendą **RUN** można uruchomić załadowany program.

Przedstawiony poniżej program **AUTOSTART** umożliwia uruchamianie programów w **BASIC-u** w sposób tak wygodny jak uruchamianie programów firmowych. Istota jego działania polega na tym, że zapisuje on na dyskietce zbiór o nazwie **AUTORUN.SYS** zawierający program w kodzie maszynowym (linie 2000—2036). Dla dyskowego systemu operacyjnego (DOS) mikrokomputera ATARI zbiór o nazwie **AUTORUN.SYS** ma specjalne znaczenie. Zawarty w zbiorze program binarny (zapisany w kodzie maszynowym) jest ładowany i uruchamiany automatycznie po włączeniu mikrokomputera.

Po wprowadzeniu linii programu do pamięci należy go zapisać na dyskietkę, a następnie uruchomić komendą **RUN**. Program zatrzyma się i zażąda podania nazwy zbioru dyskowego zawierającego program w **BASIC-u**, który ma być automatycznie uruchamiany. Wówczas należy podać nazwę zbioru np.: **PROGRAM1.BAS** i wcisnąć klawisz **RETURN**. Następnie do stacji dysków należy włożyć dyskietkę zawierającą zbiór z podaną wcześniej nazwą i wcisnąć **START**. Program uzupełni kod maszynowy zawarty w liniach 2000—2036 o podaną nazwę zbioru i zapisze zaktualizowany kod do zbioru **AUTORUN.SYS**, po czym uruchomi się od początku. Analogiczne operacje program **AUTOSTART** będzie powtarzał z kolejnymi dyskietkami. Jeżeli chcemy przerwać działanie programu, należy w momencie, gdy żąda on podania nazwy zbioru napisać * (gwiazdkę) i wcisnąć **RETURN**. Gdy mamy już zapisany na dyskietce zbiór **AUTORUN.SYS** możemy automatycznie uruchomić program. W tym celu wyłączamy na kilka sekund mikrokomputer i po chwili włączamy go ponownie. Przy włączaniu komputera nie należy wciskać klawisza **OPTION**. Po kilku sekundach program „zamelduje się” na ekranie. W tym miejscu należy wymienić trzy wa-



runki, które muszą być spełnione, aby program **AUTOSTART** poprawnie działał:

1. W opisany wyżej sposób można uruchamiać programy w **BASIC-u** zapisane na dyskietkach komendą **SAVE** (zapisane komendą **LIST** generować będą przy próbie uruchomienia błąd nr 21).
2. Na dyskietce oprócz zbioru zawierającego uruchamiany program, zbioru **AUTORUN.SYS**, musi znajdować się **DOS** wraz z pakietem **DUP.SYS**.
3. Na dyskietce można zapisać TYLKO JEDEN zbiór o nazwie **AUTORUN.SYS**.

Zwarunku trzeciego wynika, że program **AUTOSTART** jest pożyteczny, gdy na dysku mamy jeden program w **BASIC-u**. Co zrobić w sytuacji, gdy na dyskietce zapisanych jest np.: 20 lub więcej programów. Też można je wszystkie automatycznie uruchamiać. W tym celu wprowadzamy do pamięci mikrokomputera drugi program o nazwie **DYSK—MENU** (linie 1—1950). Wyświetla on wykaz zbiorów na dysku przyporządkowując jednocześnie każdemu zbiorowi pojedynczy znak — poczynając od znaku A np.:

```
PROGRAM1                PROGRAM2BAS
PROSTA BAS              PROGRAM4
```

itd. Chcąc uruchomić program zawarty w zbiorze **PROSTA.BAS** wystarczy tylko nacisnąć klawisz C. Wciśnięcie klawisza **RETURN** przekazuje sterowanie do translatora **BASIC-a**. Program **DYSK—MENU** musi być zapisany na dyskietkę zawierającą programy, które chcemy automatycznie uruchamiać komendą **SAVE**. Następnie programem **AUTOSTART** wpisujemy na tę samą dyskietkę zbiór **AUTORUN.SYS**. Gdy program **AUTOSTART** zażąda nazwy zbioru z programem do automatycznego startu należy podać nazwę zbioru, do którego wpisany został program **DYSK—MENU**. Od tej chwili możemy w komfortowy sposób uruchamiać dowolny z wielu programów zapisanych na dysku. Ma to jeszcze jedną zaletę. Nie musimy pamiętać, ani skrzętnie ewidencjonować zbiorów na dyskietce. Aby „zobaczyć” co na niej jest, wkładamy ją do stacji dysków, włączamy komputer i po chwili wszystko już wiemy.

```
1 REM *****
2 REM PROGRAM AUTOSTART
3 REM *****
100 DIM D(168),F$(16),L$(16)
110 ? CHR$(125):POKE 752,1
120 POSITION 0,0:? CHR$(17):FOR
LI=1 TO 38:? CHR$(18):NEXT LI
? CHR$(5);
122 POSITION 0,1:? "I PROGRAM Z
APISUJE ZBIOR AUTORUN.SYS I";
124 POSITION 0,2:? "I UMOZLIWIA
JACY AUTOMATYCZNY START I";
```

```
125 POSITION 0,3:? "I PROGRAMO
W NAPISANYCH W BASICU I";
126 POSITION 0,4:? CHR$(26):FOR
LI=1 TO 38:? CHR$(18):NEXT LI
? CHR$(3);
130 POSITION 3,6:? "Zbiór AUTORU
N.SYS zajmuje 2 sektory.";
150 POSITION 3,7:? "Podaj nazwe
zbioru zawierajacego"
151 POSITION 3,8:? "program, kto
ry ma miec automat.START"
152 INPUT F$
```

```
155 IF F$(1,1)="*" THEN GRAPHICS
0:END
160 L=LEN(F$)
162 FOR I=1 TO L:IF F$(I,1)=" "
THEN L=I-1:GOTO 170
165 NEXT I
170 FOR N=1 TO L
172 IF N=1 AND F$(N,N)<CHR$(65)
OR F$(N,N)>CHR$(90) THEN GOTO 11
00
174 IF N<>1 AND F$(N,N)<CHR$(48)
AND F$(N,N)>". " THEN GOTO 1100
```

```

175 IF N<>1 AND (F$(N,N)>CHR$(57
) AND F$(N,N)<CHR$(65) OR F$(N,N
)>CHR$(90)) THEN 1100
178 IF F$(N,N)="." AND (L>N+3 OR
P=1) THEN 1100
180 IF F$(N,N)="." THEN P=1
190 NEXT N
200 POSITION 3,10: ? "Wloz dyskiet
ke do stacji dyskow"
220 FOR I=1 TO 168: READ A:D(I)=A
: NEXT I
240 POSITION 3,12: ? "Nacisnij ST
ART "
242 IF PEEK(53279)=6 THEN GOTO 2
50
245 GOTO 242
250 FOR I=148 TO 137 STEP -1
260 J=J+1: IF J>L THEN D(I)=32: GO
TO 270
265 D(I)=ASC(F$(J,J))
270 NEXT I
280 GOSUB 1000

```

```

290 POSITION 3,14: ? "Prosze chwil
e poczekac..."
300 GOTO 1510
1000 TRAP 1500: CLOSE #1: OPEN #1,
8,0, "D:AUTORUN.SYS"
1010 FOR I=1 TO 168: PUT #1,D(I)
1020 NEXT I: CLOSE #1: RETURN
1100 ? CHR$(253); CHR$(254): RUN
1500 POSITION 3,20: ? "Wystapil b
lad numer "; PEEK(195)
1510 FOR T=1 TO 1100: NEXT T: ? CH
R$(253): RUN
2000 DATA 255,255,106,6,255,6,16
9,80
2002 DATA 141,0,3,169,1,141,1,3
2004 DATA 169,63,141,2,3,169,64,
141
2006 DATA 3,3,169,5,141,6,3,141
2007 DATA 5,3,169,0,141,4,3,141
2008 DATA 9,3,141,10,3,141,11,3
2010 DATA 169,12,141,8,3,32,89,2
28
2012 DATA 16,1,96,162,11,189,0,5

```

```

2014 DATA 157,0,3,202,16,247,32,
89
2016 DATA 228,48,6,32,6,5,108,12
2018 DATA 0,96,169,193,141,33,3,
169
2020 DATA 6,141,34,3,96,251,243,
51
2022 DATA 246,207,6,163,246,51,2
46,60
2024 DATA 246,76,228,243,172,255
,6,240
2026 DATA 9,185,234,6,206,255,6,
160
2028 DATA 1,96,140,33,3,169,228,
141
2030 DATA 34,3,169,155,160,1,96
2032 DATA 34,67,88,69,46,71,78,7
3,84,82,65,84,83,58,68,34,78,85,
82
2034 DATA 32,20,226,2,227,2
2036 DATA 106,6,224,2,225,2,180,
6

```



Program **DYSK—MENU** umożliwia wybór jednego z 26 programów, które mogą być automatycznie uruchamiane (litery A—Z). Jeżeli mamy na dysku więcej zbiorów (maksymalnie 64) należy w linii 1820 zmienić liczbę 91 na inną. Na przykład: liczba 101 zwiększa liczbę programów z 26 do 36. Należy jednak pamiętać, że

w tej sytuacji kolejnym programom będą przyporządkowane kolejne znaki (kody ATASCII), „większe” od liter alfabetu. Wybór programów do automatycznego startu będzie odbywał się poprzez „wybieranie” z klawiatury tych znaków, co nie zawsze jest wygodne gdyż np.: wybranie małej litery „a” wymaga użycia dwóch klawiszy (CAPS i A).

```

1 REM *****
2 REM PROGRAM DYSK-MENU
3 REM *****
1600 DIM A$(20), NZ$(12), TP$(768)
,P$(14)
1605 POKE 752,1
1610 CLOSE #1: OPEN #1,6,0, "D:*.*
"
1615 Y=65: J=1
1620 FOR X=1 TO 64: INPUT #1,A$
1630 IF A$(3,3)<"A" OR A$(3,3)>"
Z" THEN 1670
1640 TP$(J,J)=CHR$(Y): TP$(J+1,J+
11)=A$(3,13)
1650 J=J+12: Y=Y+1
1660 IF A$(1,1)=" " OR A$(1,1)="
*" THEN NEXT X
1670 Y=155: TP$(J,J)=CHR$(155)
1700 ? CHR$(125)
1710 POSITION 8,1: ? "WYKAZ ZBIOR
OW NA DYSKIETCE"

```

```

720 J=1: Y=3
730 IF TP$(J,J)=CHR$(155) THEN
GOTO 1800
1740 POSITION 4,Y: ? TP$(J,J); " "
: TP$(J+1,J+11): J=J+12
1750 IF TP$(J,J)=CHR$(155) THEN
GOTO 1800
1760 POSITION 22,Y: ? TP$(J,J); "
": TP$(J+1,J+11): J=J+12: Y=Y+1: GOT
O 1730
1800 CLOSE #1: OPEN #1,4,0, "K:"
1810 POSITION 15,22: ? "WYBIERASZ
?"
1820 GET #1,B: IF B=155 THEN END
: IF B<=64 OR B>=91 THEN GOTO 182
0
1830 J=1
1840 IF TP$(J,J)=CHR$(B) THEN GO
TO 1900
1850 IF TP$(J,J)=CHR$(155) THEN
GOTO 1950

```

```

1860 J=J+12: GOTO 1840
1900 P$(1,2)="D:": Y=2
1901 FOR X=0 TO 7
1902 IF TP$(J+1+X,J+1+X)=" " THE
N GOTO 1904
1903 Y=Y+1: P$(Y,Y)=TP$(J+1+X,J+1
+X): NEXT X
1904 Y=Y+1: P$(Y,Y)="."
1905 IF TP$(J+9,J+11)=" " THEN
P$(Y,Y)=" ": GOTO 1910
1906 P$(Y+1,Y+3)=TP$(J+9,J+11)
1910 POSITION 6,20: ? "URUCHAMIAM
PROGRAM "; P$: RUN P$
1950 POSITION 0,0: ? "zle wybrale
s program !!! ": GOTO 1800

```

Opracował H. KRASUSKI

Stoisko Mikołaja Przychody było nie mniej oblegane od innych. „Złote rączki” mistrza potrafiły w ciągu kilku minut usunąć różnorodne usterki z komputera. Za dodatkową zapłatą można było również wymienić rodzime układy w Meritum na podobne z inicjałami renomowanych firm elektronicznych. W odczuciu większości uczestników utworzenie punktu naprawy komputerów było znakomitym pomysłem organizatorów.



ATARI

PROGRAM

4

Rozszczepienie światła w pryzmacie

Program ten, umożliwia zaobserwowanie zjawiska rozszczepienia światła w pryzmacie. Jako dane, należy podać rodzaj materiału, z którego wykonany jest pryzmat, kąt łamiący pryzmatu oraz kąt padania wiązki światła białego na pryzmat. Zakończenie prezentacji rozszczepienia światła jest sygnalizowane krótkim sygnałem dźwiękowym.

W tym momencie, aby przejść do następnej części programu, należy wcisnąć klawisz SELECT. W tej części

```
0 CLR DEG
1 MX=0: DIM V$(40), PR(5,3), LL(5)
5 GRAPHICS 0: POKE 710, 12: POKE 709, 0
6 POL=0
7 FOR I=1 TO 5: FOR II=1 TO 3
8 PR(I, II)=0: NEXT II: NEXT I
10 H=170: Z=16: Q=0.25: N1=1: N2=1.5
15 POSITION 5,3: POKE 752,6
16 ? "
20 POSITION 5,4
21 ? " ROZSCZEPNIENIE SWIATLA
23 POSITION 5,5
24 ? " W PRYZMACIE
25 POSITION 5,6
26 ? "
30 POSITION 10,10: ? "1- Crown lekki"
32 POSITION 10,12: ? "2- Flint lekki"
34 POSITION 10,14: ? "3- Flint ciezki"
36 POSITION 8,19
38 ? "PODAJ RODZAJ PRYZMATU"
39 GOSUB 9000
40 OPEN #1,4,0, "K": GET #1,ZZ
41 CLOSE #1: SK=ZZ-49
42 IF SK=0 THEN 50
43 IF SK=1 THEN 50
44 IF SK=2 THEN 50
45 GOTO 40
50 RESTORE 5000+100*SK
60 GRAPHICS 0: POKE 710, 12: POKE 709, 0
65 POSITION 6,3: TRAP 65
66 ? "Podaj kat lamiacz pryzmatu"
68 POSITION 18,5: INPUT GAMA
69 IF GAMA>110 OR GAMA<1 THEN 65
70 POSITION 10,9: TRAP 70
71 ? "Podaj kat padania"
72 POSITION 18,11: INPUT AL
73 IF AL>90 OR AL<0 THEN 70
74 IF MX=1 THEN 110
75 POSITION 12,20: POKE 752,6
76 ? " DANE PRYZMATU": TRAP 40000
85 FOR M=1 TO 6: POKE 755,2
87 POSITION 15,18: ? " SELECT "
89 SOUND 1,50,10,15
90 FOR M1=1 TO 130: NEXT M1
95 SOUND 1,0,0,0: POKE 755,0
100 FOR M1=1 TO 70: NEXT M1
105 NEXT M
110 GA=GAMA/2
200 GRAPHICS 11: COLOR 15
500 REM RYSOWANIE PRYZMATU
510 APO=H*SIN(GA)/COS(GA)
515 IF APO>160 THEN H=H*0.8: GOTO 510
520 APX=(160-AP0)*Q: APY=191-Z
530 BPX=APX+APO*Q: BPY=191-Z-H
540 CPX=BPX+APO*Q: CPY=APY
550 PLOT APX, APY: DRAWTO BPX, BPY:
DRAWTO CPX, CPY
560 FOR R=Z TO 1 STEP -1: K=191-R
```

programu, można uzyskać dane dotyczące rozszczepienia i załamania poszczególnych promieni w pryzmacie. Dalsze informacje dotyczące obsługi programu będą ukazywane podczas wykonywania programu. Program ten, działa także w TURBO BASIC'u XL, co powoduje znaczne przyspieszenie wykonywanych obliczeń.

Posiadaczom odbiorników czarno-białych poleciłbym dokonanie w linii 200 zmiany trybu graficznego z 11 na 9. Można dokonać też

```
565 COLOR R: PLOT 0, K: DRAWTO 78, K
570 NEXT R
590 COLOR 15
600 REM ***** ***** ***** *****
*
610 A=SIN(AL-GA)/COS(AL-GA)
620 BX1=APX/Q+APO/2: BY1=H/2+Z
630 B=BY1-A*BX1
640 FOR F=0 TO BX1
650 Y=A*F+B
660 TRAP 680
665 IF Y<=16 THEN 680
670 PLOT F*Q, -Y+191
680 NEXT F
690 TRAP 40000
700 FOR PROMIEN=1 TO 5
790 READ COL, N2
792 LL(PROMIEN)=N2
795 COLOR COL
1000 REM == PROMIEN W PRYZMACIE
==
1010 O1=N1*SIN(AL)/N2
1020 BETA=ATN(O1/SQR(1-O1^2))
1025 PR(PROMIEN, 1)=BETA
1030 A=SIN(BETA-GA)/COS(BETA-GA)
1040 B=BY1-A*BX1
1050 A1=SIN(90+GA)/COS(90+GA)
1060 B1=Z-(A1*(CPX/Q))
1070 CY1=(A1*B-B1*A)/(A1-A)
1080 CX1=(CY1-B1)/A1
1110 A11=(CY1-BY1)/(CX1-BX1)
1120 B11=BY1-A11*BX1
1130 FOR F=BX1 TO CX1
1145 Y=A*F+B
1147 IF Y<=16 THEN POL=10: GOTO 3000
1150 TRAP 1170
1160 PLOT F*Q, -Y+191
1170 NEXT F
1180 TRAP 40000
1300 REM == PROMIEN ZA PRYZMATE
M ==
1310 BETAB=ABS(BETA-GAMA)
1315 PR(PROMIEN, 2)=BETAB
1320 O3=N1/N2
1330 GR=ATN(O3/SQR(1-O3^2))
1340 IF BETAB>GR THEN 1500
1350 O2=N2*SIN(BETAB)/N1
1360 DEL=ATN(O2/SQR(1-O2^2))
1365 PR(PROMIEN, 3)=DEL
1370 IF BETA-GAMA>=0 THEN GOSUB 2240
1375 IF BETA-GAMA<0 THEN GOSUB 2230
1380 B=CY1-A*CX1
```



wielu zmian dotyczących ilości rozpatrywanych kolorów, lecz pozostawiam to jedynie dociekliwym użytkownikom komputerów ATARI.

Olgierd NIEMYJSKI

```
1390 FOR F=CX1 TO 79/Q
1400 Y=A*F+B
1405 IF Y<=Z THEN 1430
1410 TRAP 1430
1420 PLOT F*Q, -Y+191
1430 NEXT F
1440 TRAP 40000
1499 GOTO 2000
1500 REM ** ODBICIE W PRYZMACIE
**
1510 A=SIN(GA+BETAB)/COS(GA+BETA
B)
1520 B=CY1-A*CX1
1521 PR(PROMIEN, 3)=-10
1522 IF BETAB<90-GA THEN GOSUB 2100
1524 IF BETAB>90-GA THEN GOSUB 2200
1526 IF BETAB=90-GA THEN 2300
1530 FOR F=CX1 TO T STEP TT
1540 Y=A*F+B
1545 IF Y<=16 THEN POP: GOTO 2400
1550 PLOT F*Q, -Y+191
1560 NEXT F
2000 GOTO 3000
2100 T=APX/Q: TT=-0.5: RETURN
2200 T=CPX/Q: TT=0.2: RETURN
2230 A=SIN(GA-DEL)/COS(GA-DEL)
2235 RETURN
2240 A=SIN(GA+DEL)/COS(GA+DEL)
2245 RETURN
2300 DRAWTO CX1*Q, 191-Z: GOTO 3000
2400 PLOT CX1*Q, -CY1+191
2410 DRAWTO F*Q, 191-Z
3000 NEXT PROMIEN
3010 SOUND 1,80,10,15
3020 FOR M=1 TO 50: NEXT M
3030 SOUND 1,0,0,0
3040 IF PEEK(53279)=5 THEN MX=1
GOTO 6000
3050 GOTO 3040
4990 REM ** CROWN LEKKI **
5000 DATA 4,1,5.118
5010 DATA 15,1,5.153
5020 DATA 12,1,5.186
5030 DATA 8,1,5.267
5040 DATA 10,1,5.312
5090 REM ** FLINT LEKKI **
5100 DATA 4,1,6.020
5110 DATA 15,1,6.085
5120 DATA 12,1,6.145
5130 DATA 8,1,6.307
5140 DATA 10,1,6.404
```

```

5190 REM ** FLINT CIEZKI **
5200 DATA 4,1.7406
5210 DATA 15,1.7515
5220 DATA 12,1.7685
5230 DATA 8,1.7922
5240 DATA 10,1.8045
6000 REM ** ** ** **
6010 GRAPHICS 8:COLOR 1:POKE 710,12
6015 POKE 709,0:POKE 752,6
6020 XL=58:HL=100:ZL=20
6030 PLOT ZL,ZL+HL:DRAWTO ZL+XL,ZL
6040 DRAWTO ZL+2*XL,ZL+HL
6050 DRAWTO ZL,ZL+HL
6060 PLOT 0,85:DRAWTO ZL+XL/2,ZL+HL/2
6070 DRAWTO 49+55,70-4
6080 DRAWTO 49+55+50+ZL,70-4+14
6090 COL=-1
6100 AT=-0.577:QT=130:WT=1:ET=78
6110 GOSUB 8000
6120 AT=0.5:QT=150:WT=25:ET=2
6130 GOSUB 8000
6140 TX=ZL+XL:TY=ZL:I1=332:I2=390
6150 GOSUB 8500
6155 PX=8:PY=10:V$="GAMMA":GOSUB 8700
6160 TX=ZL+XL/2:TY=ZL+HL/2
6170 I1=245:I2=292:GOSUB 8500
6180 PX=0:PY=66:V$="ALFA":GOSUB 8700
6190 I1=57:I2=95:GOSUB 8500
6200 PX=5:PY=80:V$="BETA":GOSUB 8700
6210 TX=49+55:TY=70-4
6220 I1=272:I2=300:GOSUB 8500
6230 PX=10:PY=80:V$="BETAB":GOSUB 8700
6240 I1=82:I2=117:GOSUB 8500
6250 PX=16:PY=59:V$="DELTA":GOSUB 8700
6255 PX=9:PY=100:V$="N2":GOSUB 8700
6260 PX=3:PY=30:V$="N1=1":GOSUB 8700
6270 PX=25:PY=5
6272 IF SK=0 THEN V$="CROW LEKK I"
6274 IF SK=1 THEN V$="FLINT LEKK I"
6276 IF SK=2 THEN V$="FLINT CIEZKI"
6278 FOR I=3 TO 14
6280 PLOT 197,I:DRAWTO 202+LEN(V$)*8,I
6282 NEXT I
6284 GOSUB 8700
6295 REM +*=-+*=-+*=-+*=-+*=-+*=-+*=-

```

```

6300 PX=24:PY=20
6305 V$="WYBIERZ KOLOR":GOSUB 8700
6310 PX=25:PY=30
6315 V$="DLA KTOREGO":GOSUB 8700
6320 PX=24:PY=40
6325 V$="CHCESZ UZYSKAC":GOSUB 8700
6330 PX=28:PY=50
6335 V$="DANE":GOSUB 8700
6340 PX=25:PY=70
6345 V$="1-CZERWONY":GOSUB 8700
6350 PX=25:PY=80
6355 V$="2-ZOLTY":GOSUB 8700
6360 PX=25:PY=90
6365 V$="3-ZIELONY":GOSUB 8700
6370 PX=25:PY=100
6375 V$="4-NIEBIESKI":GOSUB 8700
6380 PX=25:PY=110
6385 V$="5-FIOLETOWY":GOSUB 8700
6389 POKE 764,255:GOSUB 9000
6390 OPEN #1,4,0,"K":GET #1,I
6395 CLOSE #1:II=I-48:COLOR 0
6400 FOR I=20 TO 118
6405 PLOT 190,I:DRAWTO 305,I
6410 NEXT I
6415 PX=23:PY=30
6420 V$="GAMMA":GOSUB 8700
6425 PX=29:PY=30:V$=STR$(GAMA)
6430 GOSUB 8700
6435 PX=23:PY=40:V$="ALFA ="
6440 GOSUB 8700
6445 PX=29:PY=40:V$=STR$(AL)
6450 GOSUB 8700
6455 PX=23:PY=50:V$="BETA ="
6460 GOSUB 8700
6465 PX=29:V$=STR$(PR(II,1))
6470 GOSUB 8700:IF POL=10 THEN 6520
6475 PX=23:PY=60:V$="BETAB="
6480 GOSUB 8700
6485 PX=29:V$=STR$(PR(II,2))
6490 GOSUB 8700
6491 IF PR(II,3)=-10 THEN 6520
6495 PX=23:PY=70:V$="DELTA="
6500 GOSUB 8700
6505 PX=29:V$=STR$(PR(II,3))
6510 GOSUB 8700
6520 PX=23:PY=85:V$="WSP. ZALAMA NIA"
6525 GOSUB 8700
6530 PX=25:PY=95:V$="PRYZMATU"
6535 GOSUB 8700
6540 PX=25:PY=105:V$="N2="
6545 GOSUB 8700
6550 PX=28:V$=STR$(LL(II))
6555 GOSUB 8700:GOSUB 9000
6560 IF PR(II,3)=-10 THEN GOSUB 8900
6615 ? " SELECT - Powrot do MEN U" : ?

```

```

6620 ? " START - Powrot do PRO GRAMU"
6625 SA=PEEK(53279)
6630 IF SA=6 THEN 5
6635 IF SA=5 THEN 8800
6640 GOTO 6625
7000 END
8000 REM =====
8010 FOR F=WT TO QT STEP 10
8013 COLOR 1
8020 FOR F1=1 TO 5
8030 YT=AT*(F+F1)+ET
8040 PLOT F+F1,ZL+HL-YT
8050 NEXT F1
8060 NEXT F
8070 RETURN
8500 REM ++ ++ ++ ++ ++ ++ ++ ++
8505 DEG
8510 FOR I=11 TO 12 STEP 5
8520 PLOT TX+15*SIN(I),TY+15*COS(I)
8530 NEXT I
8540 RETURN
8700 REM == == == == == == == ==
8710 DL=PEEK(560)+PEEK(561)*256
8720 D1=PEEK(DL+4)+PEEK(DL+5)*256
8730 FOR UU=1 TO LEN(V$)
8740 D2=57344+((ASC(V$(UU,UU))-32)*8)
8750 D3=D1+PY*40+PX+UU-1
8760 FOR UUU=0 TO 7
8770 POKE D3+UUU*40,PEEK(D2+UUU)
8780 NEXT UUU:NEXT UU
8790 RETURN
8800 REM *****
8810 COLOR 0:?:?:?:?
8815 FOR I=30 TO 115
8820 PLOT 185,I:DRAWTO 319,I
8825 NEXT I
8830 IF PR(II,3)=-10 THEN GOSUB 8950
8840 GOTO 6300
8899 REM =====
8900 PX=4:PY=135:COLOR 1:GOSUB 8950
8905 V$="CALKOWITE ODBICIE W PRYZMACIE !!"
8910 GOSUB 8700
8930 RETURN
8950 FOR I=130 TO 147
8960 PLOT 26,I:DRAWTO 293,I
8970 NEXT I
8980 RETURN
9000 FOR I=1 TO 7
9005 SOUND 0,20,10,15:NEXT I
9010 SOUND 0,0,0,0:RETURN

```

Siedzibą klubu „MERI-ZAP” jest pracownia techniki cyfrowej w Zespole Szkół Technicznych w Ostrowie Wielkopolskim. W rozmowie z Mariuszem Rudowiczem dowiedzieliśmy się, że klub posiada bogatą bibliotekę programów edukacyjnych, narzędziowych i gier.

Członkowie sami opracowują programy dydaktyczne z fizyki i elektroniki. W najbliższej przyszłości za pomocą „Meritum” mają zamiar skomputeryzować również szkolną bibliotekę.

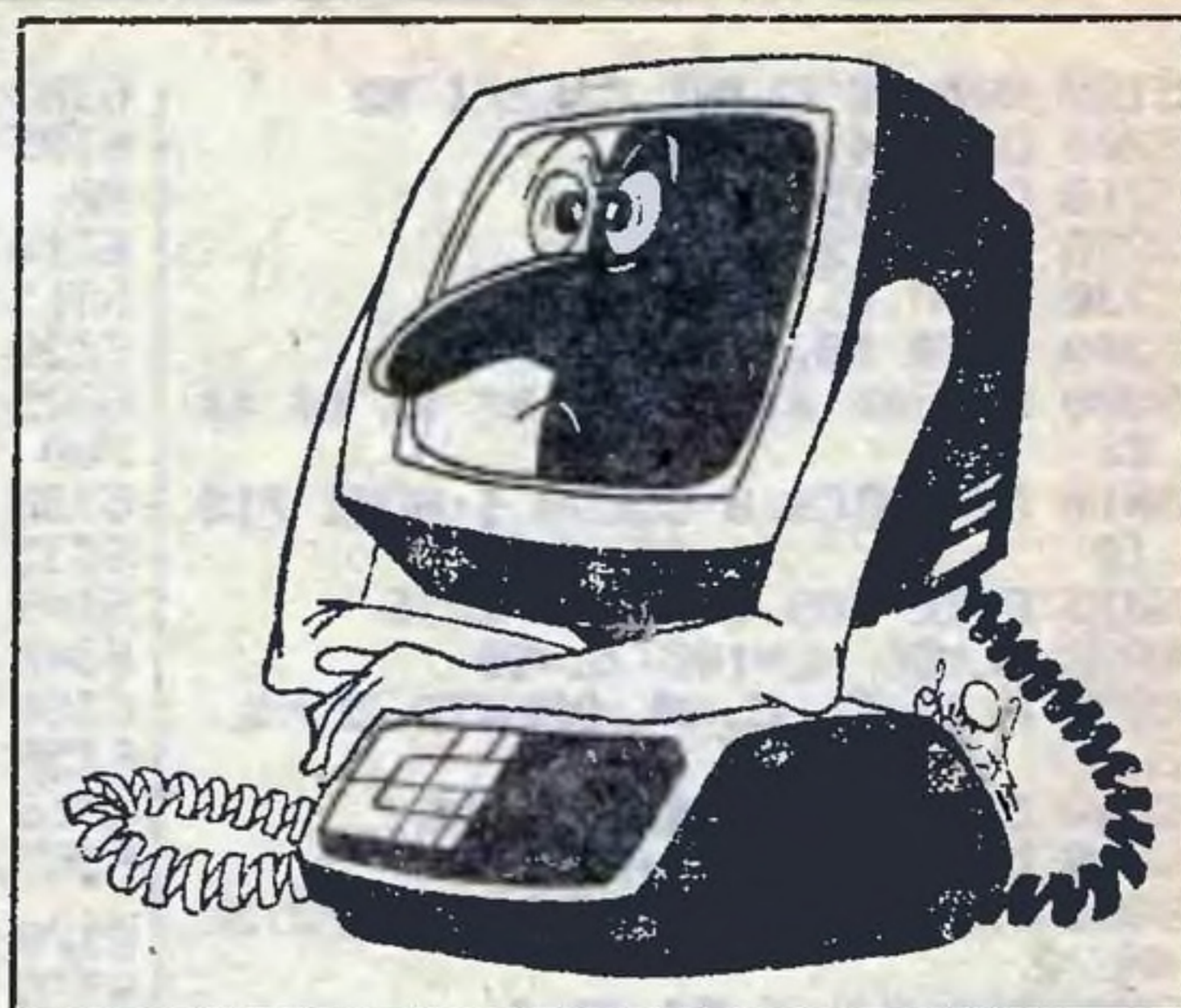
15



SPECTRUM

PROGRAM 5

RUBIK



Prezentowana gra polega na zamianie koloru przedstawionej na rys. 1 planszy na kolor przeciwny (negatyw). Po wczytaniu i uruchomieniu gry, na ekranie ukazuje się plansza (rys. 1). Wciśnięcie dowolnego klawisza, odpowiadającego jednej z widocznych na planszy liter, powoduje zaczernienie pola z wybraną literą i pól sąsiednich „na krzyż”. Np. wciśnięcie klawisza „G” i „Z” sprawia, że plansza przybie-

ra wygląd jak na rys. 2. Jeżeli teraz zostanie naciśnięty klawisz „H”, a potem „O”, to plansza ma wygląd jak na rys. 3. Widać stąd, że odpowiednie pola dotychczas białe, zostaną zaczernione, a pola czarne staną się białe.

Celem gry jest zamiana koloru całej planszy — jak na rys. 4. Oczywiście można dążyć do innych efektów, np. do otrzymania szachownicy, jak na rys. 5.

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

RYS. 1

WCISNIĘCIE
O POWODUJE
ROZPOCZĘCIE
GRY OD PO-
CZĄTKU

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

RYS. 2

WCISNIĘCIE
O POWODUJE
ROZPOCZĘCIE
GRY OD PO-
CZĄTKU

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

RYS. 3

WCISNIĘCIE
O POWODUJE
ROZPOCZĘCIE
GRY OD PO-
CZĄTKU

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

RYS. 4

WCISNIĘCIE
O POWODUJE
ROZPOCZĘCIE
GRY OD PO-
CZĄTKU

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

RYS. 5

WCISNIĘCIE
O POWODUJE
ROZPOCZĘCIE
GRY OD PO-
CZĄTKU

```

1 REM Gra "SPECTRUM-RUBIK"
10 BORDER 0: PAPER 0: INK 0: CLS
100 GO SUB 1000
200 IF INKEY$="a" OR INKEY$="A" THEN GO SUB 3000: GO SUB 3010: GO SUB 3050: GO
TO 100
210 IF INKEY$="b" OR INKEY$="B" THEN GO SUB 3000: GO SUB 3010: GO SUB 3020: GO
SUB 3060: GO TO 100
220 IF INKEY$="c" OR INKEY$="C" THEN GO SUB 3010: GO SUB 3020: GO SUB 3030: GO
SUB 3070: GO TO 100
230 IF INKEY$="d" OR INKEY$="D" THEN GO SUB 3020: GO SUB 3030: GO SUB 3040: GO
SUB 3080: GO TO 100
240 IF INKEY$="e" OR INKEY$="E" THEN GO SUB 3030: GO SUB 3040: GO SUB 3090: GO
TO 100
250 IF INKEY$="f" OR INKEY$="F" THEN GO SUB 3000: GO SUB 3050: GO SUB 3060: GO
SUB 3100: GO TO 100
260 IF INKEY$="g" OR INKEY$="G" THEN GO SUB 3010: GO SUB 3050: GO SUB 3060: GO
SUB 3070: GO SUB 3110: GO TO 100
270 IF INKEY$="h" OR INKEY$="H" THEN GO SUB 3020: GO SUB 3060: GO SUB 3070: GO
SUB 3080: GO SUB 3120: GO TO 100
280 IF INKEY$="i" OR INKEY$="I" THEN GO SUB 3030: GO SUB 3070: GO SUB 3080: GO
SUB 3090: GO SUB 3130: GO TO 100
290 IF INKEY$="j" OR INKEY$="J" THEN GO SUB 3040: GO SUB 3080: GO SUB 3090: GO
SUB 3140: GO TO 100
300 IF INKEY$="k" OR INKEY$="K" THEN GO SUB 3050: GO SUB 3100: GO SUB 3110: GO

```

PIOTR ININSKI 1986.11.14

```

SUB 3150: GO TO 100
310 IF INKEY$="l" OR INKEY$="L" THEN GO SUB 3060: GO SUB 3100: GO SUB 3110: GO
SUB 3120: GO SUB 3160: GO TO 100
320 IF INKEY$="m" OR INKEY$="M" THEN GO SUB 3070: GO SUB 3110: GO SUB 3120: GO
SUB 3130: GO SUB 3170: GO TO 100
330 IF INKEY$="n" OR INKEY$="N" THEN GO SUB 3080: GO SUB 3120: GO SUB 3130: GO
SUB 3140: GO SUB 3180: GO TO 100
340 IF INKEY$="o" OR INKEY$="O" THEN GO SUB 3090: GO SUB 3130: GO SUB 3140: GO
SUB 3190: GO TO 100
350 IF INKEY$="p" OR INKEY$="P" THEN GO SUB 3100: GO SUB 3150: GO SUB 3160: GO
SUB 3200: GO TO 100
360 IF INKEY$="q" OR INKEY$="Q" THEN GO SUB 3110: GO SUB 3150: GO SUB 3160: GO
SUB 3170: GO SUB 3210: GO TO 100
370 IF INKEY$="r" OR INKEY$="R" THEN GO SUB 3120: GO SUB 3160: GO SUB 3170: GO
SUB 3180: GO SUB 3220: GO TO 100
380 IF INKEY$="s" OR INKEY$="S" THEN GO SUB 3130: GO SUB 3170: GO SUB 3180: GO
SUB 3190: GO SUB 3230: GO TO 100
390 IF INKEY$="t" OR INKEY$="T" THEN GO SUB 3140: GO SUB 3180: GO SUB 3190: GO
SUB 3240: GO TO 100
400 IF INKEY$="u" OR INKEY$="U" THEN GO SUB 3150: GO SUB 3200: GO SUB 3210: GO
TO 100
410 IF INKEY$="w" OR INKEY$="W" THEN GO SUB 3160: GO SUB 3200: GO SUB 3210: GO
SUB 3220: GO TO 100
420 IF INKEY$="x" OR INKEY$="X" THEN GO SUB 3170: GO SUB 3210: GO SUB 3220: GO
SUB 3230: GO TO 100
430 IF INKEY$="y" OR INKEY$="Y" THEN GO SUB 3180: GO SUB 3220: GO SUB 3230: GO
SUB 3240: GO TO 100
440 IF INKEY$="z" OR INKEY$="Z" THEN GO SUB 3190: GO SUB 3230: GO SUB 3240: GO
TO 100
450 IF INKEY$="0" THEN RUN
460 GO TO 200
1000 RESTORE : FOR i=0 TO 5: PLOT i*32,175: DRAW 0,-160: NEXT i
1010 FOR i=0 TO 5: PLOT 0,i*32+15: DRAW 160,0: NEXT i
1020 FOR i=1 TO 17 STEP 4: FOR j=1 TO 17 STEP 4: READ a$: PRINT AT i,j,a$: NEXT
j: NEXT i
1030 PRINT AT 11,21;"WCISNIECIE ";AT 13,21;"0 POWODUJE";AT 15,21;"ROZPOCZECIE";A
T 17,21;"GRY OD PO-";AT 19,21;"CZATKU"
1200 RETURN
3000 LET a=0: LET b=3: LET c=0: LET d=3: GO SUB 6000: RETURN
3010 LET a=4: LET b=7: LET c=0: LET d=3: GO SUB 6000: RETURN
3020 LET a=8: LET b=11: LET c=0: LET d=3: GO SUB 6000: RETURN
3030 LET a=12: LET b=15: LET c=0: LET d=3: GO SUB 6000: RETURN
3040 LET a=16: LET b=19: LET c=0: LET d=3: GO SUB 6000: RETURN
3050 LET a=0: LET b=3: LET c=4: LET d=7: GO SUB 6000: RETURN
3060 LET a=4: LET b=7: LET c=4: LET d=7: GO SUB 6000: RETURN
3070 LET a=8: LET b=11: LET c=4: LET d=7: GO SUB 6000: RETURN
3080 LET a=12: LET b=15: LET c=4: LET d=7: GO SUB 6000: RETURN
3090 LET a=16: LET b=19: LET c=4: LET d=7: GO SUB 6000: RETURN
3100 LET a=0: LET b=3: LET c=8: LET d=11: GO SUB 6000: RETURN
3110 LET a=4: LET b=7: LET c=8: LET d=11: GO SUB 6000: RETURN
3120 LET a=8: LET b=11: LET c=8: LET d=11: GO SUB 6000: RETURN
3130 LET a=12: LET b=15: LET c=8: LET d=11: GO SUB 6000: RETURN
3140 LET a=16: LET b=19: LET c=8: LET d=11: GO SUB 6000: RETURN
3150 LET a=0: LET b=3: LET c=12: LET d=15: GO SUB 6000: RETURN
3160 LET a=4: LET b=7: LET c=12: LET d=15: GO SUB 6000: RETURN
3170 LET a=8: LET b=11: LET c=12: LET d=15: GO SUB 6000: RETURN
3180 LET a=12: LET b=15: LET c=12: LET d=15: GO SUB 6000: RETURN
3190 LET a=16: LET b=19: LET c=12: LET d=15: GO SUB 6000: RETURN
3200 LET a=0: LET b=3: LET c=16: LET d=19: GO SUB 6000: RETURN
3210 LET a=4: LET b=7: LET c=16: LET d=19: GO SUB 6000: RETURN
3220 LET a=8: LET b=11: LET c=16: LET d=19: GO SUB 6000: RETURN
3230 LET a=12: LET b=15: LET c=16: LET d=19: GO SUB 6000: RETURN
3240 LET a=16: LET b=19: LET c=16: LET d=19: GO SUB 6000: RETURN
6000 FOR j=a TO b: FOR i=c TO d: PRINT OVER 1;AT i,j;"■": NEXT i: NEXT j: RETU
RN
9000 DATA "A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q";"R
","S","T","U","W","X","Y","Z"

```

AMSTRAD

PROGRAM

6

Grafika „Wieża Hanoi”

Zamieszczony w poprzednim zeszycie programów komputerowych program WIEŻA HANOI wyświetlał wyniki w postaci tekstu na ekranie monitora. Wykorzystując podstawowe procedury tego programu realizujące przełożenie krążków na prętach oraz przeniesienie pojedynczego krążka można rozbudować program, który w sposób graficzny wyświetla wyniki.

W programie zastosowano elementy pakietu graficznego systemu TURBO PASCAL zaimplementowanego na komputerze AMSTRAD CPC6128.

Warto je przechować w bibliotece programów PASCALA. Komentarze zawierające znaki „+” ograniczają niezmiennicze fragmenty programu WIEŻA HANOI.

Istotne jest to, dla tych, którzy zechcą zmodyfikować po-



siadany program. Dodatkowo w programie wprowadzono podkład muzyczny dla grafiki kolorowej.

Andrzej

```
PROGRAM graficznawiezahanoi;
{$a-}
( ***** )
( GRAFIKA TURBO PASCALA )
( ***** )
```

```
const CPM3 = true;
```

```
const tagflag:boolean = false;
tagaddr:integer = 0;
indscr:integer = 0;
fct = 0.017453292520;
xerror:real = 0.0;
yerror:real = 0.0;
getmode:byte = 2;
getgrafmode:byte = 0;
```

```
var itg:integer;
btg:byte;
rd:array[1..131] of byte;
re:real;
```

```
procedure Initgrafik;
begin
write(0,0,0,0);
end;
```

```
procedure leavegrafik;
begin
write(0,0,0,0);
end;
```

```
procedure gotoxy(x,y:integer);
begin
write(0,0,chr(y+31),chr(x+31));
end;
```

```
procedure sound(q:byte;h,i:integer;j,k,l,m:byte);
var sd:array[0..8] of byte;
begin
sd[0] := q;
move(h,sd[3],2);
move(i,sd[7],2);
sd[1] := k;
sd[2] := l;
sd[5] := m;
sd[6] := j;
inline($21/sd/$cd/$5a/$fc/$aa/$bc);
end;
```

```
procedure soundreset;
begin
inline($cd/$5a/$fc/$a7/$bc);
end;
```

```
procedure draw(x,y:integer);
begin
inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$F9/$BB);
end;
```

```
procedure grafmove(x,y:integer);
begin
inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$C0/$BB);
end;
```

```
procedure grafmover(x,y:integer);
begin
inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$C3/$BB);
end;
```

```
procedure plot(x,y:integer);
begin
inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$EA/$BB);
end;
```

```
procedure grafwindow(x1,y1,x2,y2:integer);
begin
inline($2A/x2/$EB/$2A/x1/$cd/$5a/$fc/$CF/$BB/
$2A/y2/$EB/$2A/y1/$cd/$5a/$fc/$D2/$BB);
end;
```

```
procedure grafpen(fs:byte);
begin
inline($3A/fs/$cd/$5a/$fc/$DE/$BB);
end;
```

```
procedure grafpaper(fs:byte);
begin
inline($3A/fs/$cd/$5a/$fc/$E4/$BB);
end;
```

```
procedure clg(fs:byte);
begin
inline($cd/$5a/$fc/$DB/$BB);
grafpaper(fs);
end;
```

```
procedure ink(fs.f1,f2:byte);
begin
inline($3a/f1/$47/$3a/f2/$4f/$3a/fs/$cd/$5a/$fc/
$32/$bc);
end;
```

```
procedure paper(fs:byte);
begin
inline($3A/fs/$cd/$5a/$fc/$96/$BB);
end;
```

```
procedure pen(fs:byte);
begin
inline($3A/fs/$cd/$5a/$fc/$90/$BB);
end;
```

```
procedure mode(md:byte);
begin
if md in [0..2] then
begin
getmode := md;
inline($3a/md/$cd/$5a/$fc/$0e/$bc);
end;
end;
```

```
procedure border(v1,v2:byte);
begin
inline($3a/v1/$47/$3a/v2/$4f/$cd/$5a/$fc/$38/$bc);
end;
```

```
procedure grafout(ch:char);
begin
inline($3A/ch/$cd/$5a/$fc/$FC/$BB);
end;
```

```
procedure tag;
begin
if not tagflag then
begin
tagaddr := conoutptr;
tagflag := true;
conoutptr := addr(grafout);
end;
end;
```

```
procedure tagoff;
begin
if tagflag then
begin
conoutptr := tagaddr;
tagflag := false;
end;
end;
```

```
( ***** )
( DEKLARACJA TYPOW )
( DEFINICJA STALYCH I ZMIENNYCH )
( ***** )
```

```
CONST
(+++++)
maxilkr=5; (maksymalna ilosc krzaskow)
(+++++)
```

```
wysk=50; (wysokosc krzaska)
kolw=26; (kolor wiezy-bialy)
kolt=0; (kolor tla-czarny)
dpoz=150; (dlugosc pozioma wiezy)
wysw=-270; (wysokosc wiezy)
kolzn=16; (kolor znaku-rozowy)
pxzn=-16; (pozycjonowanie znaku)
pyzn=-25;
kolob=0; (kolor obrzeza)
```

```
TYPE
(+++++)
znak='a'..'c'; (nazwa preta)
wyswiez=1..maxilkr; (maksymalna wysokosc wiezy
mierzona iloscia krzaskow)
opiswiew=ARRAY [znak] OF (opis wiew)
RECORD
m:wyswiez; (wskaznik wierzchołka wiezy)
wiewa:ARRAY (wyswiez) OF wyswiez
END;
(+++++)
```

```
gopiswiew=ARRAY [znak] OF
RECORD
m:integer; (wspolrzedne poczatku)
m:integer; (wiezy)
m:integer; (wspolrzedne wierzchołka)
m:integer;
m:integer; (wspolrzedne wierzchołka)
m:integer; (wspolrzedne poczatku krzaska)
m:integer;
END;
```

```
VAR
END;
```

```

(+++++
ilkr:integer; (zadana ilosc krazkow)
start, (pret, z ktorego przesuwamy krazki)
cel, (pret, na ktory przesuwamy krazki)
posr:znak; (pret - magazyn pomocniczy)
opw:opiswiaz;
(+++++

x,obraz:char;
dlkraz:ARRAY [myswiaz] OF integer; (dlugosci krazka)
kolkr:ARRAY [myswiaz] OF integer; (kolory krazka)
ogw:gopiswiaz; (graficzny opis wiaz)

( *****
( PROCEDURY INICJOWANIA DANYCH
( *****

(+++++
PROCEDURE ustawstpcz;
(procedura ustawiajaca poczatkowy stan wiaz)
VAR
s:znak; (zmiennie robocze)
i:myswiaz;
BEGIN
WITH opw['a'] DO
BEGIN
w:=maxilkr-ilkr+1;
FOR i:=w TO maxilkr DO
wiaz[i]:=i-w+1
END;
opw['b'].w:=0;
opw['c'].w:=0
END;
(+++++

PROCEDURE ustawdgraf;
(procedura ustawiajaca poczatkowy stan)
(danych graficznych)
VAR
s:znak; (zmiennie robocze)
i:myswiaz;
j:integer;
BEGIN
soundreset;
grafwindow(0,10,640,400);
ink(8,0,0); (tekst)
ink(9,26,26); (tlo tekstu)

FOR i:=1 TO maxilkr DO
dlkraz[i]:=70+((i-1)*20);
kolkr[1]:=15; (pomarancz)
kolkr[2]:=17; (magneta)
kolkr[3]:=6; (czerwony)
kolkr[4]:=1; (niebieski)
kolkr[5]:=21; (zielony)

WITH ogw['a'] DO
BEGIN
wspx:=31;
wspy:=75;
mwx:=106;
mwy:=345;
FOR i:=1 TO maxilkr DO (poziom)
BEGIN
FOR j:=1 TO maxilkr DO (numer krazka)
parkr[i].wspx[j]:=71-((j-1)*10);
parkr[i].wspy:=275-((i-1)*50)
END;
END;

WITH ogw['b'] DO
BEGIN
wspx:=244;
wspy:=75;
mwx:=319;
mwy:=345;
FOR i:=1 TO maxilkr DO
BEGIN
FOR j:=1 TO maxilkr DO
parkr[i].wspx[j]:=284-((j-1)*10);
parkr[i].wspy:=275-((i-1)*50)
END;
END;

WITH ogw['c'] DO
BEGIN
wspx:=457;
wspy:=75;
mwx:=532;
mwy:=345;
FOR i:=1 TO maxilkr DO
BEGIN
FOR j:=1 TO maxilkr DO
parkr[i].wspx[j]:=497-((j-1)*10);
parkr[i].wspy:=275-((i-1)*50)
END;
END;
END;

( *****
( PROCEDURY KRESLENIA WIAZ
( *****

PROCEDURE pusta(o:znak);
(kreslenie wiaz bez krazkow)
VAR zn:char;
BEGIN
ink(6,kolw,kolw);
grafpen(6); (pioro wiaz)
WITH ogw[o] DO
BEGIN
sound(120,wspx,50,15,3,0,0);
sound(120,wspy,50,15,3,0,0);
sound(40,kolw*(1+random(150)),25,15,0,0,0);
plot(wspx,wspy);
draw(dpozw,0);

```

```

grafmove(mwx,mwy);
END;
draw(0,mwy);

CASE o OF
'a':zn:='A';
'b':zn:='B';
'c':zn:='C';
END;
ink(7,kolzn,kolzn);
tag; grafpen(7);
grafover(pxzn,pyzn);
grafout(zn);
tagoff;
END;

PROCEDURE niepusta(o:znak);
(kreslenie krazkow jednej wiaz)
VAR nrkr,i:myswiaz;
j:integer;
BEGIN
WITH opw[o] DO
WITH ogw[o] DO
FOR i:=maxilkr DOWNTO w BO
BEGIN
nrkr:=wiaz[i];
ink(nrkr,kolkr[nrkr],kolkr[nrkr]);
(pioro krazka)

grafpen(nrkr);
sound(100,parkr[i].wspy+200*i,7,12,0,0,0);
sound(200,parkr[i].wspy+200*i,7,12,0,0,0);
sound(240,nrkr*(1+random(150))+200*i,7,12,
0,0,0);
FOR j:=0 TO 48 DO
BEGIN
sound(1,parkr[i].wspx[nrkr],6,10,0,0,0);
sound(2,parkr[i].wspy+j,6,10,0,0,0);
sound(4,j*nrkr*(1+random(150)),6,10,0,0,0);
plot(parkr[i].wspx[nrkr],parkr[i].wspy+j);
draw(dlkraz[nrkr],0);
END;
END;
END;

PROCEDURE kreslenie;
(procedura kreslaca stany wiaz)
VAR
i:integer;
s:znak; (zmiennie robocze)
BEGIN
clr(0);
FOR s:='a' TO 'c' DO
WITH opw[s] DO
BEGIN
pusta(s); (kreslenie wiaz bez krazkow)
IF w>0 THEN niepusta(s);
END;
IF obraz='n' THEN
FOR i:=1 TO 31000 DO
s:=s; (opoznienie)
END;
( *****
( PROCEDURY PRZENOSZENIA KRAZKOW
( *****

PROCEDURE przen(start,cel:znak);
(zmodyfikowana procedura przenoszaca pojedynczy
krazek z preta "start" na pret "cel")
BEGIN
WITH opw[cel] DO
BEGIN
IF w=0 THEN w:=maxilkr
ELSE w:=w-1;
wiaz[w]:=opw[start].wiaz[opw[start].w]
END;
WITH opw[start] DO
IF w=maxilkr THEN w:=0

```

```

ELSE w:=w+1;
kreslenie;
IF obraz='t' THEN
BEGIN
gotoxy(1,25);
WRITE('Wcisnij klawisz');
REPEAT UNTIL keypressed;
READ(kbd,x);
END;
END;

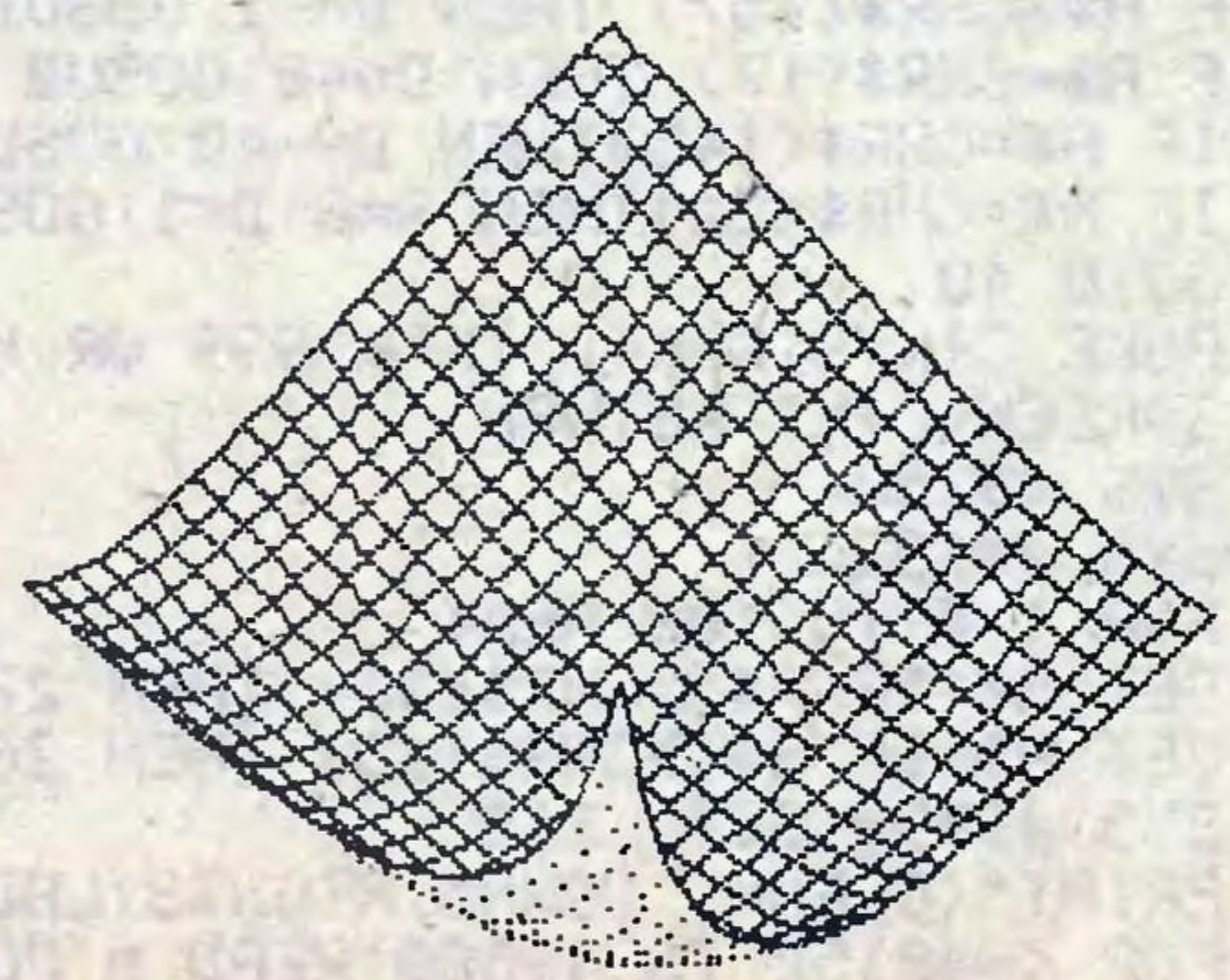
(+++++
PROCEDURE przel(n:myswiaz; VAR start,posr,cel:znak);
(procedura przesuwajaca "n" krazkow z preta
"start" na pret "cel" z wykorzystaniem preta
"posr" jako magazynu)
BEGIN
IF n=1
THEN
przen(start,cel)
ELSE
BEGIN
przel(n-1,start,cel,posr);
przen(start,cel);
przel(n-1,posr,start,cel)
END
END;
(+++++

( *****
( MODUL STERUJACY
( *****

BEGIN
REPEAT
WRITELN('podaj ilosc krazkow (<=',maxilkr);
READLN(ilkr);
WRITELN('Czy chcesz akceptowac-wcisnasc?');
WRITELN('dowolny klawisz-przeloznienie');
WRITELN('kazdego krazka (t/n)');
READLN(obraz);
UNTIL (ilkr>0) AND (ilkr=<=maxilkr) AND
((obraz='t') OR (obraz='n'));
clrscr;
BEGIN
ustawstpcz;
ustawdgraf;
initgrafik;
mode(0);
border(kolob,kolob);
ink(0,kolt,kolt);
grafpaper(0);
paper(9);
per(B);
kreslenie;
gotoxy(1,25);
WRITE('Poczatek-wcisnij kl. ');
REPEAT UNTIL keypressed;
READ(kbd,x);
END;
start:='a';
cel:='c';
posr:='b';
przel(ilkr,start,posr,cel);
gotoxy(1,25);
WRITE('Koniec-wcisnij kl. ');
REPEAT UNTIL keypressed;
READ(kbd,x);
leavegrafik;
END.

```

GRAFIKA



COMMODORE

PROGRAM

7

„Życie”

Gra symuluje powstawanie i zanikanie pokoleń organizmów w populacji. Oparta jest na trzech zasadach:

a) każdy organizm otoczony przez dwa lub trzy inne ocaleje,

b) cztery lub więcej organizmów sąsiednich lub ich brak powoduje zniszczenie organizmu na skutek przerostu populacji lub izolacji organizmu,

c) w pustej komórce otoczonej dokładnie trzema sąsiednimi organizmami powstaje nowe życie (nowy organizm).

Program został pierwotnie opracowany w języku BASIC, natomiast w niżej przedstawionym zastosowano w kilku miejscach język maszynowy. Dzięki temu gra przebiega około 100 razy szybciej. Prędkość przebiegu programu można zmienić przez zmianę końcowej wartości dla pętli określonej w linii 240. Działanie programu można zatrzymać w dowolnej chwili klawiszem SPACJA i ponownie uruchomić jego działanie naciskając RETURN. Usunięcie linii 240—260 powoduje zwiększenie szybkości działania programu, bez możliwości zatrzymania i ponownego uruchomienia w sposób wyżej opisany.

READY.

```
5 PRINT "ZYCIE"
10 W=51968:C=55296:POKE 53280,0:POKE53281,0
20 IF PEEK (49152)<>169 OR PEEK (49580)<>96 THEN GOSUB 300
30 SYS 49397:K=0:S=0
40 GET A$:IF A$="" THEN A=PEEK(C+K):POKEC+K,NOT(A AND 1)AND (A OR 1)
50 IF A$=CHR$(42)THEN S=1:D=1:GOSUB 150
60 IF A$=CHR$(13)THEN POKEC+K,S:GOTO210
70 IF A$=CHR$(29) THEN D=1:GOSUB 150
80 IF A$=CHR$(157) THEN D=-1:GOSUB 150
90 IF A$=CHR$(17) THEN D=40:GOSUB 150
100 IF A$=CHR$(145)THEN D=-40:GOSUB 150
110 IF A$=CHR$(32)THEN S=0:D=1:GOSUB150
120 GOTO 40
150 POKE C+K,S:K=K+D:IF K>999 OR K<0 THEN K=K-D
160 S=PEEK(C+K):RETURN
210 SYS 49250
220 SYS 49152
240 FOR I=0 TO 100:NEXT
250 GET A$:IF A$<>CHR$(32) THEN 270
260 GET A$:IF A$<>CHR$(13) THEN 260
270 SYS 49298:GOTO 210
300 PRINT"(CLEAR, DOWN5,RIGHT8)LADOWANIE JEZYKA MASZYNOWEGO"
310 FOR K=49152 TO 49580:READ M:POKE K,M:NEXT:RETURN
```



Dodanie linii: 230 SYS 49452, powoduje, że gra odbywa się na ekranie bez określonych granic — ekran jak gdyby zawija się z góry do dołu i z lewej strony na prawą. Aby grać w „Życie” należy program po prostu wpisać do pamięci komputera i uruchomić. Na czarnym ekranie pojawi się świecący jasno kursor w postaci piłki (kuli). Do zbudowania początkowego wzoru służą klawisze kursora (ustawianie piłki w żądanym miejscu) i klawisz mnożenia (załączenie komórki — powołanie do życia). Gdy wzór jest gotowy, naciśnięcie RETURN zapoczątkuje jego ewolucję. Naciśnięcie RUN/STOP przerywa działanie programu i kończy ewolucję wzoru. Gra może być zabawą graficzną, jak również ciekawą grą matematyczną:

Opracował: **Tadeusz CISEK**

```

1000 DATA 169,39,133,253,169,0,133,254
1010 DATA 169,216,133,255,169,0,141,36
1020 DATA 192,169,203,141,37,192,160,0
1030 DATA 177,254,41,15,240,11,160,7
1040 DATA 190,90,192,254,26,207,136,16
1050 DATA 247,238,36,192,208,3,238,37
1060 DATA 192,198,253,16,18,169,39,133
1070 DATA 253,24,173,36,192,105,2,141
1080 DATA 36,192,144,3,238,37,192,230
1090 DATA 254,208,2,230,255,165,254,201
1100 DATA 232,208,195,165,255,201,219,208
1110 DATA 189,96,0,1,2,42,44,84
1120 DATA 85,86,169,0,141,111,192,169
1130 DATA 203,141,112,192,162,0,142,110
1140 DATA 207,238,111,192,208,3,238,112
1150 DATA 192,173,111,192,201,110,208,238
1160 DATA 173,112,192,201,207,208,231,165
1170 DATA 251,201,15,208,2,134,251,230
1180 DATA 251,96,169,39,133,253,169,0
1190 DATA 133,254,169,216,133,255,169,43
1200 DATA 141,171,192,169,203,141,172,192
1210 DATA 160,0,173,69,207,201,2,240
1220 DATA 19,201,3,208,12,177,254,41
1230 DATA 15,208,9,165,251,145,254,208
1240 DATA 3,152,145,254,238,171,192,208
1250 DATA 3,238,172,192,198,253,16,18
1260 DATA 169,39,133,253,24,173,171,192
1270 DATA 105,2,141,171,192,144,3,238

```

```

1280 DATA 172,192,230,254,208,2,230,255
1290 DATA 165,254,201,232,208,188,165,255
1300 DATA 201,219,208,182,96,169,0,133
1310 DATA 254,169,216,133,255,169,0,133
1320 DATA 252,169,4,133,253,160,0,169
1330 DATA 0,145,254,169,81,145,252,230
1340 DATA 252,208,2,230,253,230,254,208
1350 DATA 2,230,255,165,254,201,232,208
1360 DATA 230,165,255,201,219,208,224,169
1370 DATA 1,133,251,96,160,1,24,185
1380 DATA 0,203,121,26,207,153,26,207
1390 DATA 24,185,69,207,121,42,203,153
1400 DATA 42,203,200,192,41,208,231,169
1410 DATA 25,133,253,169,42,133,254,169
1420 DATA 203,133,255,160,0,24,177,254
1430 DATA 160,40,113,254,145,254,160,41
1440 DATA 24,177,254,160,1,113,254,145
1450 DATA 254,24,165,254,105,42,133,254
1460 DATA 144,2,230,255,198,253,208,219
1470 DATA 24,173,0,203,160,40,121,26
1480 DATA 207,153,26,207,160,41,24,185
1490 DATA 0,203,160,1,121,26,207,153
1500 DATA 26,207,24,173,68,207,160,40
1510 DATA 121,42,203,153,42,203,160,41
1520 DATA 24,185,68,207,160,1,121,42
1530 DATA 203,153,42,203,96
1540 END

```

READY.



„ANDROMEDA” (Planetarium i Obserwatorium Astronomiczne w Łodzi) wyposażona w mikrokomputery Spectrum, Meritum, Commodore oraz Amstrad zajmuje się przede wszystkim tworzeniem biblioteki programów astronomicznych, popularyzacją informatyki oraz nauczaniem języków programowania.

Plany na przyszłość obejmują nie tylko „komputerowe loty” poza naszą galaktykę, ale znaczne rozszerzenie działalności poprzez organizację masowych imprez dla młodzieży.



Tworząc programy astronomiczne zgodnie z narzuconymi algorytmami można również wykazać dużo własnej inicjatywy.

DŻUNGLA

Ten program dedykujemy Czytelnikom z fantazją: proponujemy... wędrówkę przez dżunglę!

Oczywiście „wędrownik” jest uzbrojonym po zęby strzelcem wyborowym. Jako wytrawny podróżnik ma kompas, tyle tylko że... nie zna drogi!

A wokół rządzą *prawa dżungli!* Co krok jakaś *niespodzianka*: raz urwisko, innym razem pyton, udający gałąź lub... sprężona do ataku pantera!

Wędrownik ma tylko jeden cel: wyjść z dżungli. I to jak najszybciej!

Może on wybrać kierunek północny, południowy, wschodni lub zachodni. Tyle tylko że każdy kierunek jest jednakowo dobry (a raczej jednakowo zły), gdyż startuje on ze „środku” dżungli.

Teoretycznie można wyjść po 20 manewrach. Praktycznie: po 30, 40...

Gdy wędrownik zbyt długo błądzi (po 60 manewrach), wtedy zwycięża dżungla: pojawia się meldunek „zaginiony” i... komputer staje.

Rysunek podaje przykład udanego wyjścia z dżungli. Uwaga: nie znamy „sposobu na wyjście”, gdyż niespodzianki modelują dwa generatory liczb losowych: jeden ($R=RND4$) lokalizuje, zaś drugi ($T=RND8$) definiuje zdarzenie.

Program wywołuje etykieta „X” (RUN 10). Życzymy szybkiego wyjścia!

Janusz MILLER

```
10: "X": CLEAR :
    RANDOM : USING
    : WAIT 0: DIM P$(
    (8)
```

```
20: DATA "U R W I
    S K O", "L I A
    N Y", "M O C Z
    A R Y", "P O T
    O K"
```

```
30: DATA "P A N T
    E R A", "K O B
    R A", "G O R Y
    L", "P Y T O N"
```

```
40: RESTORE 20: FOR
    K=1 TO 4: READ P
    $(K): NEXT K
```

```
50: RESTORE 30: FOR
    K=5 TO 8: READ P
    $(K): NEXT K
```

```
60: I$=" ? T=1 N=0
    ":R$="U C I E
    K A J !!"
70: CLS :R=RND 4:T
    =RND 8:GOTO 90
80: N=0:S=0:W=0:E=
    0:PRINT P$(T):
    BEEP 3,25,500
90: WAIT 0:PRINT "
    POLNOC";I$;:
    INPUT "";N
100: CLS : IF N=0
    GOTO 130
110: IF R*N=10R R*N
    =2GOTO 80
120: Y=Y+1:Z=Z+1:
    GOTO 170
130: PRINT "POLUDNI
    E";I$;: INPUT "
    ";S
140: CLS : IF S=0
    GOTO 170
150: IF R*S=30R R*S
    =4GOTO 80
160: Y=Y-1:Z=Z+1
170: PRINT "WSCHOD"
    ;I$;: INPUT "";
    E
180: CLS : IF E=0
    GOTO 210
190: IF R*E=20R R*E
    =3GOTO 80
200: X=X+1:Z=Z+1:
    GOTO 240
```



```
210: PRINT "ZACHOD"
    ;I$;: INPUT "";
    W
220: CLS : IF W=0
    GOTO 240
230: X=X-1:Z=Z+1
240: IF (N+S+W+E)=0
    THEN PRINT R$:
    BEEP 10,25,50:
    GOTO 90
250: IF Z>=60THEN
    WAIT :PRINT "Z
    A G I N I O N
    Y":GOTO 290
260: N=0:S=0:W=0:E=
    0
270: IF ABS X<20AND
    ABS Y<20GOTO 7
    0
280: WAIT :PRINT "W
    YSZEDLES PO";Z
    ;" MANEWRACH"
290: WAIT 0:PRINT "
    OD NOWA";I$;:
    INPUT "";P
300: CLS : IF P=0
    THEN END
310: X=0:Y=0:Z=0:
    GOTO 70
```

SPECTRUM

```
5
0000 REM DZUNGLA
0001 LET C=0: LET Z=0: LET X=0
0002 LET R=0: LET S=0: LET W=0
0003 LET E=0: LET P=0: LET PR=0
0004 CLS : DIM P$(8,17)
0005 CIRCLE 125,85,2: PLOT 125,8
0006 DATA "U R W I S K O"
0007 DATA "L I A N Y"
0008 DATA "M O C Z A R Y"
0009 DATA "P O T O K"
0010 DATA "P A N T E R A"
0011 DATA "K O B R A"
0012 DATA "G O R Y L"
0013 DATA "P Y T O N"
0014 DATA "X C H O D"
0015 DATA "P O L N O C"
0016 DATA "P O L U D N I E"
0017 DATA "Z A C H O D"
0018 DATA "W S C H O D"
0019 DATA "Z A G I N I O N Y"
0020 DATA "M A N E W R A C H"
0021 DATA "O D N O W A"
0022 DATA "Z A C H O D"
0023 DATA "W S C H O D"
0024 DATA "P O L N O C"
0025 DATA "P O L U D N I E"
0026 DATA "Z A C H O D"
0027 DATA "W S C H O D"
0028 DATA "Z A G I N I O N Y"
0029 DATA "M A N E W R A C H"
0030 DATA "O D N O W A"
0031 DATA "Z A C H O D"
0032 DATA "W S C H O D"
0033 DATA "P O L N O C"
0034 DATA "P O L U D N I E"
0035 DATA "Z A C H O D"
0036 DATA "W S C H O D"
0037 DATA "Z A G I N I O N Y"
0038 DATA "M A N E W R A C H"
0039 DATA "O D N O W A"
0040 DATA "Z A C H O D"
0041 DATA "W S C H O D"
0042 DATA "P O L N O C"
0043 DATA "P O L U D N I E"
0044 DATA "Z A C H O D"
0045 DATA "W S C H O D"
0046 DATA "Z A G I N I O N Y"
0047 DATA "M A N E W R A C H"
0048 DATA "O D N O W A"
0049 DATA "Z A C H O D"
0050 DATA "W S C H O D"
0051 DATA "P O L N O C"
0052 DATA "P O L U D N I E"
0053 DATA "Z A C H O D"
0054 DATA "W S C H O D"
0055 DATA "Z A G I N I O N Y"
0056 DATA "M A N E W R A C H"
0057 DATA "O D N O W A"
0058 DATA "Z A C H O D"
0059 DATA "W S C H O D"
0060 DATA "P O L N O C"
0061 DATA "P O L U D N I E"
0062 DATA "Z A C H O D"
0063 DATA "W S C H O D"
0064 DATA "Z A G I N I O N Y"
0065 DATA "M A N E W R A C H"
0066 DATA "O D N O W A"
0067 DATA "Z A C H O D"
0068 DATA "W S C H O D"
0069 DATA "P O L N O C"
0070 DATA "P O L U D N I E"
0071 DATA "Z A C H O D"
0072 DATA "W S C H O D"
0073 DATA "Z A G I N I O N Y"
0074 DATA "M A N E W R A C H"
0075 DATA "O D N O W A"
0076 DATA "Z A C H O D"
0077 DATA "W S C H O D"
0078 DATA "P O L N O C"
0079 DATA "P O L U D N I E"
0080 DATA "Z A C H O D"
0081 DATA "W S C H O D"
0082 DATA "Z A G I N I O N Y"
0083 DATA "M A N E W R A C H"
0084 DATA "O D N O W A"
0085 DATA "Z A C H O D"
0086 DATA "W S C H O D"
0087 DATA "P O L N O C"
0088 DATA "P O L U D N I E"
0089 DATA "Z A C H O D"
0090 DATA "W S C H O D"
0091 DATA "Z A G I N I O N Y"
0092 DATA "M A N E W R A C H"
0093 DATA "O D N O W A"
0094 DATA "Z A C H O D"
0095 DATA "W S C H O D"
0096 DATA "P O L N O C"
0097 DATA "P O L U D N I E"
0098 DATA "Z A C H O D"
0099 DATA "W S C H O D"
0100 DATA "Z A G I N I O N Y"
0101 DATA "M A N E W R A C H"
0102 DATA "O D N O W A"
0103 DATA "Z A C H O D"
0104 DATA "W S C H O D"
0105 DATA "P O L N O C"
0106 DATA "P O L U D N I E"
0107 DATA "Z A C H O D"
0108 DATA "W S C H O D"
0109 DATA "Z A G I N I O N Y"
0110 DATA "M A N E W R A C H"
0111 DATA "O D N O W A"
0112 DATA "Z A C H O D"
0113 DATA "W S C H O D"
0114 DATA "P O L N O C"
0115 DATA "P O L U D N I E"
0116 DATA "Z A C H O D"
0117 DATA "W S C H O D"
0118 DATA "Z A G I N I O N Y"
0119 DATA "M A N E W R A C H"
0120 DATA "O D N O W A"
0121 DATA "Z A C H O D"
0122 DATA "W S C H O D"
0123 DATA "P O L N O C"
0124 DATA "P O L U D N I E"
0125 DATA "Z A C H O D"
0126 DATA "W S C H O D"
0127 DATA "Z A G I N I O N Y"
0128 DATA "M A N E W R A C H"
0129 DATA "O D N O W A"
0130 DATA "Z A C H O D"
0131 DATA "W S C H O D"
0132 DATA "P O L N O C"
0133 DATA "P O L U D N I E"
0134 DATA "Z A C H O D"
0135 DATA "W S C H O D"
0136 DATA "Z A G I N I O N Y"
0137 DATA "M A N E W R A C H"
0138 DATA "O D N O W A"
0139 DATA "Z A C H O D"
0140 DATA "W S C H O D"
0141 DATA "P O L N O C"
0142 DATA "P O L U D N I E"
0143 DATA "Z A C H O D"
0144 DATA "W S C H O D"
0145 DATA "Z A G I N I O N Y"
0146 DATA "M A N E W R A C H"
0147 DATA "O D N O W A"
0148 DATA "Z A C H O D"
0149 DATA "W S C H O D"
0150 DATA "P O L N O C"
0151 DATA "P O L U D N I E"
0152 DATA "Z A C H O D"
0153 DATA "W S C H O D"
0154 DATA "Z A G I N I O N Y"
0155 DATA "M A N E W R A C H"
0156 DATA "O D N O W A"
0157 DATA "Z A C H O D"
0158 DATA "W S C H O D"
0159 DATA "P O L N O C"
0160 DATA "P O L U D N I E"
0161 DATA "Z A C H O D"
0162 DATA "W S C H O D"
0163 DATA "Z A G I N I O N Y"
0164 DATA "M A N E W R A C H"
0165 DATA "O D N O W A"
0166 DATA "Z A C H O D"
0167 DATA "W S C H O D"
0168 DATA "P O L N O C"
0169 DATA "P O L U D N I E"
0170 DATA "Z A C H O D"
0171 DATA "W S C H O D"
0172 DATA "Z A G I N I O N Y"
0173 DATA "M A N E W R A C H"
0174 DATA "O D N O W A"
0175 DATA "Z A C H O D"
0176 DATA "W S C H O D"
0177 DATA "P O L N O C"
0178 DATA "P O L U D N I E"
0179 DATA "Z A C H O D"
0180 DATA "W S C H O D"
0181 DATA "Z A G I N I O N Y"
0182 DATA "M A N E W R A C H"
0183 DATA "O D N O W A"
0184 DATA "Z A C H O D"
0185 DATA "W S C H O D"
0186 DATA "P O L N O C"
0187 DATA "P O L U D N I E"
0188 DATA "Z A C H O D"
0189 DATA "W S C H O D"
0190 DATA "Z A G I N I O N Y"
0191 DATA "M A N E W R A C H"
0192 DATA "O D N O W A"
0193 DATA "Z A C H O D"
0194 DATA "W S C H O D"
0195 DATA "P O L N O C"
0196 DATA "P O L U D N I E"
0197 DATA "Z A C H O D"
0198 DATA "W S C H O D"
0199 DATA "Z A G I N I O N Y"
0200 DATA "M A N E W R A C H"
0201 DATA "O D N O W A"
0202 DATA "Z A C H O D"
0203 DATA "W S C H O D"
0204 DATA "P O L N O C"
0205 DATA "P O L U D N I E"
0206 DATA "Z A C H O D"
0207 DATA "W S C H O D"
0208 DATA "Z A G I N I O N Y"
0209 DATA "M A N E W R A C H"
0210 DATA "O D N O W A"
0211 DATA "Z A C H O D"
0212 DATA "W S C H O D"
0213 DATA "P O L N O C"
0214 DATA "P O L U D N I E"
0215 DATA "Z A C H O D"
0216 DATA "W S C H O D"
0217 DATA "Z A G I N I O N Y"
0218 DATA "M A N E W R A C H"
0219 DATA "O D N O W A"
0220 DATA "Z A C H O D"
0221 DATA "W S C H O D"
0222 DATA "P O L N O C"
0223 DATA "P O L U D N I E"
0224 DATA "Z A C H O D"
0225 DATA "W S C H O D"
0226 DATA "Z A G I N I O N Y"
0227 DATA "M A N E W R A C H"
0228 DATA "O D N O W A"
0229 DATA "Z A C H O D"
0230 DATA "W S C H O D"
0231 DATA "P O L N O C"
0232 DATA "P O L U D N I E"
0233 DATA "Z A C H O D"
0234 DATA "W S C H O D"
0235 DATA "Z A G I N I O N Y"
0236 DATA "M A N E W R A C H"
0237 DATA "O D N O W A"
0238 DATA "Z A C H O D"
0239 DATA "W S C H O D"
0240 DATA "P O L N O C"
0241 DATA "P O L U D N I E"
0242 DATA "Z A C H O D"
0243 DATA "W S C H O D"
0244 DATA "Z A G I N I O N Y"
0245 DATA "M A N E W R A C H"
0246 DATA "O D N O W A"
0247 DATA "Z A C H O D"
0248 DATA "W S C H O D"
0249 DATA "P O L N O C"
0250 DATA "P O L U D N I E"
0251 DATA "Z A C H O D"
0252 DATA "W S C H O D"
0253 DATA "Z A G I N I O N Y"
0254 DATA "M A N E W R A C H"
0255 DATA "O D N O W A"
0256 DATA "Z A C H O D"
0257 DATA "W S C H O D"
0258 DATA "P O L N O C"
0259 DATA "P O L U D N I E"
0260 DATA "Z A C H O D"
0261 DATA "W S C H O D"
0262 DATA "Z A G I N I O N Y"
0263 DATA "M A N E W R A C H"
0264 DATA "O D N O W A"
0265 DATA "Z A C H O D"
0266 DATA "W S C H O D"
0267 DATA "P O L N O C"
0268 DATA "P O L U D N I E"
0269 DATA "Z A C H O D"
0270 DATA "W S C H O D"
0271 DATA "Z A G I N I O N Y"
0272 DATA "M A N E W R A C H"
0273 DATA "O D N O W A"
0274 DATA "Z A C H O D"
0275 DATA "W S C H O D"
0276 DATA "P O L N O C"
0277 DATA "P O L U D N I E"
0278 DATA "Z A C H O D"
0279 DATA "W S C H O D"
0280 DATA "Z A G I N I O N Y"
0281 DATA "M A N E W R A C H"
0282 DATA "O D N O W A"
0283 DATA "Z A C H O D"
0284 DATA "W S C H O D"
0285 DATA "P O L N O C"
0286 DATA "P O L U D N I E"
0287 DATA "Z A C H O D"
0288 DATA "W S C H O D"
0289 DATA "Z A G I N I O N Y"
0290 DATA "M A N E W R A C H"
0291 DATA "O D N O W A"
0292 DATA "Z A C H O D"
0293 DATA "W S C H O D"
0294 DATA "P O L N O C"
0295 DATA "P O L U D N I E"
0296 DATA "Z A C H O D"
0297 DATA "W S C H O D"
0298 DATA "Z A G I N I O N Y"
0299 DATA "M A N E W R A C H"
0300 DATA "O D N O W A"
0301 DATA "Z A C H O D"
0302 DATA "W S C H O D"
0303 DATA "P O L N O C"
0304 DATA "P O L U D N I E"
0305 DATA "Z A C H O D"
0306 DATA "W S C H O D"
0307 DATA "Z A G I N I O N Y"
0308 DATA "M A N E W R A C H"
0309 DATA "O D N O W A"
0310 DATA "Z A C H O D"
0311 DATA "W S C H O D"
0312 DATA "P O L N O C"
0313 DATA "P O L U D N I E"
0314 DATA "Z A C H O D"
0315 DATA "W S C H O D"
0316 DATA "Z A G I N I O N Y"
0317 DATA "M A N E W R A C H"
0318 DATA "O D N O W A"
0319 DATA "Z A C H O D"
0320 DATA "W S C H O D"
0321 DATA "P O L N O C"
0322 DATA "P O L U D N I E"
0323 DATA "Z A C H O D"
0324 DATA "W S C H O D"
0325 DATA "Z A G I N I O N Y"
0326 DATA "M A N E W R A C H"
0327 DATA "O D N O W A"
0328 DATA "Z A C H O D"
0329 DATA "W S C H O D"
0330 DATA "P O L N O C"
0331 DATA "P O L U D N I E"
0332 DATA "Z A C H O D"
0333 DATA "W S C H O D"
0334 DATA "Z A G I N I O N Y"
0335 DATA "M A N E W R A C H"
0336 DATA "O D N O W A"
0337 DATA "Z A C H O D"
0338 DATA "W S C H O D"
0339 DATA "P O L N O C"
0340 DATA "P O L U D N I E"
0341 DATA "Z A C H O D"
0342 DATA "W S C H O D"
0343 DATA "Z A G I N I O N Y"
0344 DATA "M A N E W R A C H"
0345 DATA "O D N O W A"
0346 DATA "Z A C H O D"
0347 DATA "W S C H O D"
0348 DATA "P O L N O C"
0349 DATA "P O L U D N I E"
0350 DATA "Z A C H O D"
0351 DATA "W S C H O D"
0352 DATA "Z A G I N I O N Y"
0353 DATA "M A N E W R A C H"
0354 DATA "O D N O W A"
0355 DATA "Z A C H O D"
0356 DATA "W S C H O D"
0357 DATA "P O L N O C"
0358 DATA "P O L U D N I E"
0359 DATA "Z A C H O D"
0360 DATA "W S C H O D"
0361 DATA "Z A G I N I O N Y"
0362 DATA "M A N E W R A C H"
0363 DATA "O D N O W A"
0364 DATA "Z A C H O D"
0365 DATA "W S C H O D"
0366 DATA "P O L N O C"
0367 DATA "P O L U D N I E"
0368 DATA "Z A C H O D"
0369 DATA "W S C H O D"
0370 DATA "Z A G I N I O N Y"
0371 DATA "M A N E W R A C H"
0372 DATA "O D N O W A"
0373 DATA "Z A C H O D"
0374 DATA "W S C H O D"
0375 DATA "P O L N O C"
0376 DATA "P O L U D N I E"
0377 DATA "Z A C H O D"
0378 DATA "W S C H O D"
0379 DATA "Z A G I N I O N Y"
0380 DATA "M A N E W R A C H"
0381 DATA "O D N O W A"
0382 DATA "Z A C H O D"
0383 DATA "W S C H O D"
0384 DATA "P O L N O C"
0385 DATA "P O L U D N I E"
0386 DATA "Z A C H O D"
0387 DATA "W S C H O D"
0388 DATA "Z A G I N I O N Y"
0389 DATA "M A N E W R A C H"
0390 DATA "O D N O W A"
0391 DATA "Z A C H O D"
0392 DATA "W S C H O D"
0393 DATA "P O L N O C"
0394 DATA "P O L U D N I E"
0395 DATA "Z A C H O D"
0396 DATA "W S C H O D"
0397 DATA "Z A G I N I O N Y"
0398 DATA "M A N E W R A C H"
0399 DATA "O D N O W A"
0400 DATA "Z A C H O D"
0401 DATA "W S C H O D"
0402 DATA "P O L N O C"
0403 DATA "P O L U D N I E"
0404 DATA "Z A C H O D"
0405 DATA "W S C H O D"
0406 DATA "Z A G I N I O N Y"
0407 DATA "M A N E W R A C H"
0408 DATA "O D N O W A"
0409 DATA "Z A C H O D"
0410 DATA "W S C H O D"
0411 DATA "P O L N O C"
0412 DATA "P O L U D N I E"
0413 DATA "Z A C H O D"
0414 DATA "W S C H O D"
0415 DATA "Z A G I N I O N Y"
0416 DATA "M A N E W R A C H"
0417 DATA "O D N O W A"
0418 DATA "Z A C H O D"
0419 DATA "W S C H O D"
0420 DATA "P O L N O C"
0421 DATA "P O L U D N I E"
0422 DATA "Z A C H O D"
0423 DATA "W S C H O D"
0424 DATA "Z A G I N I O N Y"
0425 DATA "M A N E W R A C H"
0426 DATA "O D N O W A"
0427 DATA "Z A C H O D"
0428 DATA "W S C H O D"
0429 DATA "P O L N O C"
0430 DATA "P O L U D N I E"
0431 DATA "Z A C H O D"
0432 DATA "W S C H O D"
0433 DATA "Z A G I N I O N Y"
0434 DATA "M A N E W R A C H"
0435 DATA "O D N O W A"
0436 DATA "Z A C H O D"
0437 DATA "W S C H O D"
0438 DATA "P O L N O C"
0439 DATA "P O L U D N I E"
0440 DATA "Z A C H O D"
0441 DATA "W S C H O D"
0442 DATA "Z A G I N I O N Y"
0443 DATA "M A N E W R A C H"
0444 DATA "O D N O W A"
0445 DATA "Z A C H O D"
0446 DATA "W S C H O D"
0447 DATA "P O L N O C"
0448 DATA "P O L U D N I E"
0449 DATA "Z A C H O D"
0450 DATA "W S C H O D"
0451 DATA "Z A G I N I O N Y"
0452 DATA "M A N E W R A C H"
0453 DATA "O D N O W A"
0454 DATA "Z A C H O D"
0455 DATA "W S C H O D"
0456 DATA "P O L N O C"
0457 DATA "P O L U D N I E"
0458 DATA "Z A C H O D"
0459 DATA "W S C H O D"
0460 DATA "Z A G I N I O N Y"
0461 DATA "M A N E W R A C H"
0462 DATA "O D N O W A"
0463 DATA "Z A C H O D"
0464 DATA "W S C H O D"
0465 DATA "P O L N O C"
0466 DATA "P O L U D N I E"
0467 DATA "Z A C H O D"
0468 DATA "W S C H O D"
0469 DATA "Z A G I N I O N Y"
0470 DATA "M A N E W R A C H"
0471 DATA "O D N O W A"
0472 DATA "Z A C H O D"
0473 DATA "W S C H O D"
0474 DATA "P O L N O C"
0475 DATA "P O L U D N I E"
0476 DATA "Z A C H O D"
0477 DATA "W S C H O D"
0478 DATA "Z A G I N I O N Y"
0479 DATA "M A N E W R A C H"
0480 DATA "O D N O W A"
0481 DATA "Z A C H O D"
0482 DATA "W S C H O D"
0483 DATA "P O L N O C"
0484 DATA "P O L U D N I E"
0485 DATA "Z A C H O D"
0486 DATA "W S C H O D"
0487 DATA "Z A G I N I O N Y"
0488 DATA "M A N E W R A C H"
0489 DATA "O D N O W A"
0490 DATA "Z A C H O D"
0491 DATA "W S C H O D"
0492 DATA "P O L N O C"
0493 DATA "P O L U D N I E"
0494 DATA "Z A C H O D"
0495 DATA "W S C H O D"
0496 DATA "Z A G I N I O N Y"
0497 DATA "M A N E W R A C H"
0498 DATA "O D N O W A"
0499 DATA "Z A C H O D"
0500 DATA "W S C H O D"
0501 DATA "P O L N O C"
0502 DATA "P O L U D N I E"
0503 DATA "Z A C H O D"
0504 DATA "W S C H O D"
0505 DATA "Z A G I N I O N Y"
0506 DATA "M A N E W R A C H"
0507 DATA "O D N O W A"
0508 DATA "Z A C H O D"
0509 DATA "W S C H O D"
0510 DATA "P O L N O C"
0511 DATA "P O L U D N I E"
0512 DATA "Z A C H O D"
0513 DATA "W S C H O D"
0514 DATA "Z A G I N I O N Y"
0515 DATA "M A N E W R A C H"
0516 DATA "O D N O W A"
0517 DATA "Z A C H O D"
0518 DATA "W S C H O D"
0519 DATA "P O L N O C"
0520 DATA "P O L U D N I E"
0521 DATA "Z A C H O D"
0522 DATA "W S C H O D"
0523 DATA "Z A G I N I O N Y"
0524 DATA "M A N E W R A C H"
0525 DATA "O D N O W A"
0526 DATA "Z A C H O D"
0527 DATA "W S C H O D"
0528 DATA "P O L N O C"
0529 DATA "P O L U D N I E"
0530 DATA "Z A C H O D"
0531 DATA "W S C H O D"
0532 DATA "Z A G I N I O N Y"
0533 DATA "M A N E W R A C H"
0534 DATA "O D N O W A"
0535 DATA "Z A C H O D"
0536 DATA "W S C H O D"
0537 DATA "P O L N O C"
0538 DATA "P O L U D N I E"
0539 DATA "Z A C H O D"
0540 DATA "W S C H O D"
0541 DATA "Z A G I N I O N Y"
0542 DATA "M A N E W R A C H"
0543 DATA "O D N O W A"
0544 DATA "Z A C H O D"
0545 DATA "W S C H O D"
0546 DATA "P O L N O C"
0547 DATA "P O L U D N I E"
0548 DATA "Z A C H O D"
0549 DATA "W S C H O D"
0550 DATA "Z A G I N I O N Y"
0551 DATA "M A N E W R A C H"
0552 DATA "O D N O W A"
0553 DATA "Z A C H O D"
0554 DATA "W S C H O D"
0555 DATA "P O L N O C"
0556 DATA "P O L U D N I E"
0557 DATA "Z A C H O D"
0558 DATA "W S C H O D"
0559 DATA "Z A G I N I O N Y"
0560 DATA "M A N E W R A C H"
0561 DATA "O D N O W A"
0562 DATA "Z A C H O D"
0563 DATA "W S C H O D"
0564 DATA "P O L N O C"
0565 DATA "P O L U D N I E"
0566 DATA "Z A C H O D"
0567 DATA "W S C H O D"
0568 DATA "Z A G I N I O N Y"
0569 DATA "M A N E W R A C H"
0570 DATA "O D N O W A"
0571 DATA "Z A C H O D"
0572 DATA "W S C H O D"
0573 DATA "P O L N O C"
0574 DATA "P O L U D N I E"
0575 DATA "Z A C H O D"
0576 DATA "W S C H O D"
0577 DATA "Z A G I N I O N Y"
0578 DATA "M A N E W R A C H"
0579 DATA "O D N O W A"
0580 DATA "Z A C H O D"
0581 DATA "W S C H O D"
0582 DATA "P O L N O C"
0583 DATA "P O L U D N I E"
0584 DATA "Z A C H O D"
0585 DATA "W S C H O D"
0586 DATA "Z A G I N I O N Y"
0587 DATA "M A N E W R A C H"
0588 DATA "O D N O W A"
0589 DATA "Z A C H O D"
0590 DATA "W S C H O D"
0591 DATA "P O L N O C"
0592 DATA "P O L U D N I E"
0593 DATA "Z A C H O D"
0594 DATA "W S C H O D"
0595 DATA "Z A G I N I O N Y"
0596 DATA "M A N E W R A C H"
0597 DATA "O D N O W A"
0598 DATA "Z A C H O D"
0599 DATA "W S C H O D"
0600 DATA "P O L N O C"
0601 DATA "P O L U D N I E"
0602 DATA "Z A C H O D"
0603 DATA "W S C H O D"
0604 DATA "Z A G I N I O N Y"
0605 DATA "M A N E W R A C H"
0606 DATA "O D N O W A"
0607 DATA "Z A C H O D"
0608 DATA "W S C H O D"
0609 DATA "P O L N O C"
0610 DATA "P O L U D N I E"
0611 DATA "Z A C H O D"
0612 DATA "W S C H O D"
0613 DATA "Z A G I N I O N Y"
0614 DATA "M A N E W R A C H"
0615 DATA "O D N O W A"
0616 DATA "Z A C H O D"
0617 DATA "W S C H O D"
0618 DATA "P O L N O C"
0619 DATA "P O L U D N I E"
0620 DATA "Z A C H O D"
0621 DATA "W S C H O D"
0622 DATA "Z A G I N I O N Y"
0623 DATA "M A N E W R A C H"
0624 DATA "O D N O W A"
0625 DATA "Z A C H O D"
0626 DATA "W S C H O D"
0627 DATA "P O L N O C"
0628 DATA "P O L U D N I E"
0629 DATA "Z A C H O D"
0630 DATA "W S C H O D"
0631 DATA "Z A G I N I O N Y"
0632 DATA "M A N E W R A C H"
0633 DATA "O D N O W A"
0634 DATA "Z A C H O D"
0635 DATA "W S C H O D"
0636 DATA "P O L N O C"
0637 DATA "P O L U D N I E"
0638 DATA "Z A C H O D"
0639 DATA "W S C H O D"
0640 DATA "Z A G I N I O N Y"
0641 DATA "M A N E W R A C H"
0642 DATA "O D N O W A"
0643 DATA "Z A C H O D"
0644 DATA "W S C H O D"
0645 DATA "P O L N O C"
0646 DATA "P O L U D N I E"
0647 DATA "Z A C H O D"
0648 DATA "W S C H O D"
0649 DATA "Z A G I N I O N Y"
0650 DATA "M A N E W R A C H"
0651 DATA "O D N O W A"
0652 DATA "Z A C H O D"
0653 DATA "W S C H O D"
0654 DATA "P O L N O C"
0655 DATA "P O L U D N I E"
0656 DATA "Z A C H O D"
0657 DATA "W S C H O D"
0658 DATA "Z A G I N I O N Y"
0659 DATA "M A N E W R A C H"
0660 DATA "O D N O W A"
0661 DATA "Z A C H O D"
0662 DATA "W S C H O D"
0663 DATA "P O L N O C"
0664 DATA "P O L U D N I E"
0665 DATA "Z A C H O D"
0666 DATA "W S C H O D"
0667 DATA "Z A G I N I O N Y"
0668 DATA "M A N E W R A C H"
0669 DATA "O D N O W A"
0670 DATA "Z A C H O D"
0671 DATA "W S C H O D"
0672 DATA "P O L N O C"
0673 DATA "P O L U D N I E"
0674 DATA "Z A C H O D"
0675 DATA "W S C H O D"
0676 DATA "Z A G I N I O N Y"
0677 DATA "M A N E W R A C H"
0678 DATA "O D N O W A"
0679 DATA "Z A C H O D"
0680 DATA "W S C H O D"
0681 DATA "P O L N O C"
0682 DATA "P O L U D N I E"
0683 DATA "Z A C H O D"
0684 DATA "W S C H O D"
0685 DATA "Z A G I N I O N Y"
0686 DATA "M A N E W R A C H"
0687 DATA "O D N O W A"
```

```

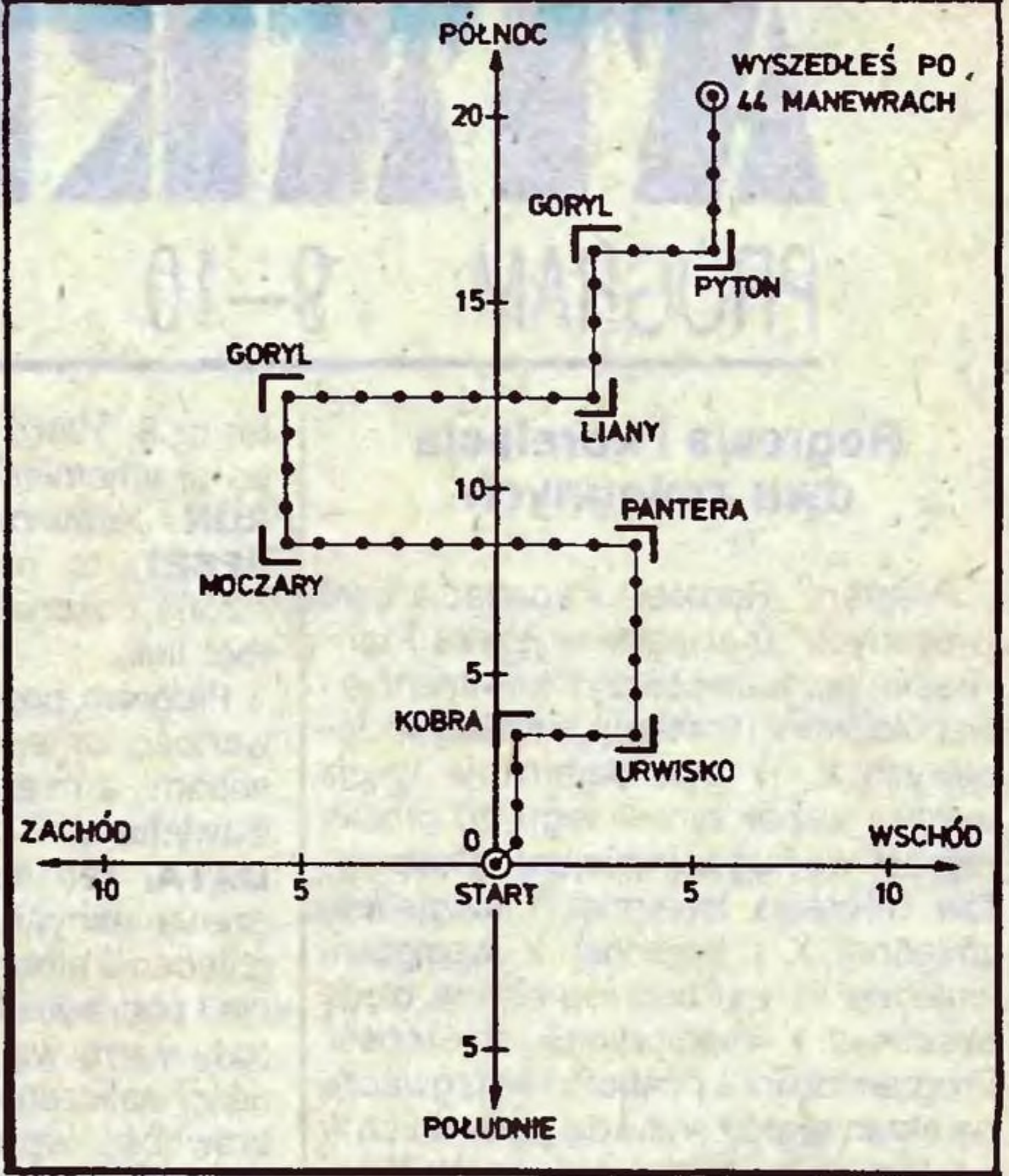
30 COSTA "D Y T O N"
40 FOR K=1 TO 8
44 READ P#(K,1 TO 17)
50 LET I#="" ? Tak = 1 Nie = 0
60 LET I#="" U C I E K A J !
70 PRINT (RND*4)
75 PRINT (RND*7) + 1
80 GO TO 90
90 LET S#0
100 LET E#0
110 PRINT AT 0,1: P#(1,1 TO 17)
120 PRINT AT 1,1: P#(1,1 TO 17)
130 PRINT AT 2,1: "POLNOC" ; ;
140 IF E#0 THEN GO TO 130
150 LET U#1
160 IF U#1 OR U#2 THEN GO TO 80
170 PRINT AT 0,3
180 LET Z#Z+1
190 GO TO 170
200 PRINT AT 1,1: "POLUDNIE"; ;
210 PRINT AT 2,1: "POLUDNIE"; ;
220 IF S#0 THEN GO TO 170
230 LET U#3
240 IF U#3 OR U#4 THEN GO TO 80
250 PRINT AT 0,3
260 LET Z#Z+1
270 PRINT AT 1,1: "WSCHOD" ; ;
280 PRINT AT 2,1: "WSCHOD" ; ;
290 IF E#0 THEN GO TO 210
300 LET U#1
310 IF U#1 OR U#2 THEN GO TO 80
320 PRINT AT 0,3
330 LET Z#Z+1
340 PRINT AT 1,1: "ZACHOD" ; ;
350 PRINT AT 2,1: "ZACHOD" ; ;
360 IF E#0 THEN GO TO 240
370 LET X#X-1: LET Z#Z+1
380 LET C#D+S+W+E

```

```

400 IF C#0 THEN PRINT AT 2,10; ;
410 PRINT AT 2,10; ;
420 GO TO 90
430 IF Z#8 THEN GO TO 250
440 PRINT AT 10,1; FLASH 1; "Z A
450 D Y T O N Y"
460 BEEP 0,0
470 GO TO 90
480 LET D#0: LET S#0
490 LET E#0: LET E#0
500 LET U#D+S: LET V#ABS U
510 IF U#0 AND V#0 THEN GO TO
520 PRINT AT 10,1; FLASH 1; "Wys
530 zedles po ; Z; " manewrach"
540 FOR J=1 TO 8: BEEP .1, J: NE
550 XT
560 INPUT "czy grasz dalej (tak
570 /nie) : ; ;
580 IF D#"nie" THEN STOP
590 LET X#0: LET Y#0: LET Z#0
600 CLS : GO TO 70

```



Rys. 1. Przykład podróży w dżungli



Firma „LASER” przywędrowała aż z USA. W rozmowie z Jadwigą Bieńkowską dowiedzieliśmy się, że jej bossowie nastawiają się na sprzedaż profesjonalnych komputerów dla zakładów pracy i instytucji. Oczywiście z pełnym oprogramowaniem. Niskie ceny, to główny atut, mający najbardziej przemawiać do wyobraźni ewentualnych nabywców. Tylko że... zakupy można realizować za twardą walutę poprzez wpłaty na konto w Berlinie Zachodnim.

ATARI

PROGRAM 9-10

Regresja i korelacja dwu zmiennych

Program „Regresja i korelacja dwu zmiennych” (napisany w języku Atari-Basic) pozwala obliczyć kowariancję i współczynnik korelacji zmiennych losowych X i Y oraz parametry (wyraz wolny i współczynnik regresji) prostej regresji według najmniejszych kwadratów (regresja zmiennej Y względem zmiennej X i zmiennej X względem zmiennej Y), wartości regresyjne, błędy szacunku i współczynnik zbieżności. Program oblicza ponadto i wyprowadza na ekran szereg wyników pośrednich w postaci różnych sum zmiennych X i Y. Tym samym program daje użytkownikowi pełen zestaw charakterystyk liczbowych potrzebnych do analizy współzależności dwóch zmiennych losowych. Może to okazać się szczególnie przydatne w procesie dydaktycznym, w trakcie demonstracji różnych wzorów (metod) służących do obliczania ww. parametrów (w tym również szacowania parametrów prostej regresji uproszczoną metodą najmniejszych kwadratów).

Aby podnieść walory dydaktyczne programu wyposażono go w generator polskich liter (wg W. Zientara, Kompu-

ter nr 8, 1986), który jest wczytywany po uruchomieniu programu instrukcją **RUN**. Jeżeli nie był używany klawisz **RESET**, to następnym uruchomieniu można dokonać instrukcją **GOTO 80** (bez linii).

Program pozwala na wprowadzenie wartości zmiennych X i Y dwoma sposobami, a mianowicie „ręcznym” (tj. z klawiatury) i za pomocą instrukcji **DATA**. Ten ostatni sposób wprowadzania danych jest szczególnie wart polecenia albowiem ułatwia sprawdzanie i poprawianie danych wejściowych (gdy mamy wątpliwość co do poprawności obliczeń możemy sprawdzić poprawność wprowadzonych danych i wnieść poprawki bez konieczności ponownego wprowadzania wszystkich wartości, co przy dużej liczbie prób jest bardzo uciążliwe), a także pozwala na wielokrotne powtarzanie obliczeń i wprowadzanie danych z taśmy (zapisanych instrukcją **LIST**). Wartość zmiennych X i Y wprowadza się instrukcjami **DATA** (w kolejności: X1, Y1, X2, Y2, ..., Xn, Yn) począwszy od linii 730 (program zajmuje linie od 20 do 720, a generator polskich liter od 29000 do 29058).

Przy wprowadzaniu danych z klawiatury program zabezpiecza proces wprowadzania przed:

```
20 ? CHR$(125):POKE 752,1
30 POSITION 3,10:?"REGRESJA I K
ORELACJA DWU ZMIENNYCH"
40 FOR F=1 TO 1500:NEXT F
50 ? CHR$(125):POSITION 1,10:?"
Poczekaj 21 sekund.Wprowadzam se
nerator"
60 POSITION 13,12:?"polskich li
ter":GOSUB 29000
70 POKE 752,0
80 CLR :DIM X(100),Y(100),X$(10)
,Y$(10),YR(100),Q$(1)
90 ? CHR$(125):?"Wprowadzanie d
anych r";CHR$(4);"czne czy za po
moc";CHR$(17);" instrukcji DATA?
R/D"
91 INPUT Q$
100 IF Q$(1)="D" THEN 210
110 N=1:?"CHR$(125):?"Podaj war
to";CHR$(23);"ci zmiennych X i Y
"
120 ? :?"Zako";CHR$(13);"czenie
wprowadzania danych:wpisz END i
naci";CHR$(23);"nij RETURN":?
130 ? "Je";CHR$(24);"eli ";CHR$(
7);"le wprowadzi";CHR$(123);"e";
CHR$(23);
134 ? "wsp";CHR$(16);CHR$(123);
"RZ";CHR$(4);"dne daneso punktu."
;
135 ? "naci";CHR$(23);"nij 1 i R
```

```
ETURN,je";CHR$(24);"eli dobrze -
naci";CHR$(23);"nij 2 i RETURN"
140 ? :?"Punkt";" ";N:INPUT X$:
TRAP 140:IF X$(1,1)="E" THEN N=N
-1:GOTO 300
150 INPUT Y$:TRAP 150:X(N)=VAL(X
$):Y(N)=VAL(Y$)
160 ? "X(";N;")=";X(N)
170 ? "Y(";N;")=";Y(N)
180 INPUT U:IF U=1 THEN 140
190 N=N+1:GOTO 140
200 REM Wprowadzenie danych inst
rkcjami DATA
210 ? CHR$(125):?"Liczba pr";CH
R$(16);"b N=";:INPUT N:FOR F=1 T
O N:X(F)=0:Y(F)=0:NEXT F:K=0
220 FOR F=1 TO N
230 FOR G=1 TO 2
240 READ L
250 IF G=1 THEN X(F)=L:?"X(";F;
")=";X(F):GOTO 270
260 Y(F)=L:?"Y(";F;")=";Y(F):K=
K+1:IF K=9 THEN K=0:?"c.d. p
o naci";CHR$(23);"ni";CHR$(4);"c
iu RETURN"
261 INPUT Q$
270 NEXT G
280 NEXT F
290 ? :?"c.d. po naci";CHR$(23)
;"ni";CHR$(4);"ciu RETURN":INPUT
Q$
```

- zbyt szybkim naciśnięciem klawisza **RETURN** (przed wprowadzeniem wartości X lub Y),
- błędym wprowadzeniem współrzędnych aktualnie zapisywanego punktu.

Po stwierdzeniu poprawności wpisanego współrzędnych kolejnego punktu należy to potwierdzić wpisaniem liczby 2 i naciśnięciem klawisza **RETURN**. Dopiero wówczas wartości współrzędnych zostaną zaakceptowane przez program i można przejść do zapisu współrzędnych następnego punktu. Po stwierdzeniu błędu należy wpisać 1 i nacisnąć klawisz **RETURN**. Wówczas program umożliwi powtórne wpisanie wartości współrzędnych danego punktu.

Liczność prób N nie może przekraczać 100. Można ją zmienić przez wprowadzenie zmian w deklaracji zmiennych X, Y, YR w linii 80.

W programie użyto określeń zaczerpniętych z książek: Oktaba W., Niedokos E.: Matematyka i podstawy statystyki matematycznej, PWN, Warszawa 1974 i Zajac K.: Zarys metod statystycznych, PWE, Warszawa 1971.

Prezentowany program różni się od innych programów tego typu przede wszystkim zakresem dostarczanej informacji, co czyni żeń narzędzie mogące zaspokoić wymagania również profesjonalnego użytkownika. Walory użytkowe programu znakomicie podnosi możliwość wprowadzania danych za pomocą instrukcji **DATA** a jego walory dydaktyczne zastosowanie polskiego alfabetu.

Andrzej GRZESIAK

```
300 S1=0:S2=0:S3=0:S4=0:S5=0
310 FOR F=1 TO N
320 S1=S1+X(F):S2=S2+Y(F):S3=S3+
X(F)*X(F):S4=S4+X(F)*Y(F):S5=S5+
Y(F)*Y(F)
330 NEXT F
340 XS=S1/N:YS=S2/N
350 ? CHR$(125)
360 ? "SIGMA X=";S1,"SIGMA Y=";S
2:?"SIGMA X^2=";S3,"SIGMA Y^2="
;S5:?"SIGMA X*Y=";S4
370 ? "X";CHR$(23);"red.=";XS,"Y
";CHR$(23);"red.=";YS
380 SXY=S4/N:XS*YS:?"Kowaria
ncja=";SXY:NSXY=N*SXY
390 NS2X=S3-(S1*S1)/N:NS2Y=S5-(S
2*S2)/N:RXY=NSXY/SQR(NS2X*NS2Y)
400 ? :?"SIGMA (X-X";CHR$(23);"
red)^2=";NS2X:?"SIGMA (Y-Y";CHR
$(23);"red)^2=";NS2Y
410 ? "SIGMA (X-X";CHR$(23);"red
)(Y-Y";CHR$(23);"red)=";NSXY
411 ? :?"Wsp";CHR$(16);CHR$(123)
);"czynnik korelacji=";RXY
420 BYX=NSXY/NS2X:WV=YS-BYX*XS
430 ? :?"R";CHR$(15);"WNANIE PR
OSTEJ REGRESJI"
440 ? :?"CHR$(217);"=";WV;"+";BY
X;CHR$(216)
450 SYWX=SQR((NS2Y/N)*(1-RXY^2))
460 ? :?"Odchylenie standardowe
=";SYWX
```

```

470 VYWX=N*(SYWX^2)/(N-2)
480 ? "B";CHR$(123);CHR$(17);"d
";CHR$(23);"redni=";SQR(VYWX)
490 BSWR=SQR(VYWX/NS2X)
500 POSITION 1,21:?"B";CHR$(123
);CHR$(17);"d standardowy wsp";C
HR$(16);CHR$(123);"czynnika regr
esji=";BSWR
510 ? "Ci";CHR$(17);"s dalszy po
naci";CHR$(23);"ni";CHR$(4);"ci
u [RETURN]";INPUT Q$
520 ? CHR$(125):?"Warto";CHR$(2
3);"ci regresyjne":?
530 K=0
540 FOR F=1 TO N
550 Y=WW+BYXX(F):YR(F)=Y
560 ? "Y";F;"=";YR(F)
570 K=K+1:IF K=15 THEN K=0:?"c.
d. wydruku po naci";CHR$(23);"ni
";CHR$(4);"ciu RETURN";INPUT Q$
580 NEXT F
590 REM Obliczanie współczynnika
zbieżności
600 FOR F=1 TO N
610 SUMA1=SUMA1+(Y(F)-YR(F))^2
620 SUMA2=SUMA2+(Y(F)-YS)^2
630 NEXT F
640 SUMA3=SUMA1/SUMA2
650 ? :?"Wsp";CHR$(16);CHR$(123

```

```

);"czynnika zbie";CHR$(24);"no";C
HR$(23);"ci=";SUMA3
660 ? :?"Czy chcesz obliczyć";CH
R$(22);" regresj";CHR$(4);" odwr
otn";CHR$(17);"? T/N":INPUT Q$
670 IF Q$="N" THEN 700
680 FOR F=1 TO N:Z1=Y(F):Z2=X(F)
:X(F)=Z1:Y(F)=Z2:NEXT F
690 GOTO 300
700 ? :?"Czy pracujemy dalej? T
/N":INPUT Q$
710 IF Q$="T" THEN 80
720 END
29000 CHB=(PEEK(106)-8)*256
29010 FOR I=0 TO 1023:POKE CHB+I
,PEEK(57344+I):NEXT I:RESTORE 29
040
29020 READ A:IF A=+1 THEN POKE 7
56,CHB/256:RETURN
29030 FOR I=0 TO 7:READ B:POKE C
HB+A*B+I,B:NEXT I:GOTO 29020
29040 DATA 65,0,24,60,102,102,12
5,102,12
29041 DATA 67,12,60,102,96,96,10
2,60,0
29042 DATA 68,0,0,60,102,126,96,
60,6
29043 DATA 69,0,126,96,124,96,96
,126,12

```

```

29044 DATA 70,12,24,126,12,24,48
,126,0
29045 DATA 71,12,24,0,126,12,48,
126,0
29046 DATA 76,0,96,120,112,224,9
6,126,0
29047 DATA 77,12,24,0,124,102,10
2,102,0
29048 DATA 78,24,102,118,126,126
,110,102,0
29049 DATA 79,12,60,102,102,102,
102,60,0
29050 DATA 80,12,24,0,60,102,102
,60,0
29051 DATA 81,0,0,60,6,62,102,62
,12
29052 DATA 83,12,60,96,60,6,6,60
,0
29053 DATA 86,12,24,0,60,96,96,6
0,0
29054 DATA 87,12,24,52,96,60,6,1
24,0
29055 DATA 88,0,24,0,126,12,48,1
26,9
29056 DATA 90,24,0,126,12,24,48,
126,0
29057 DATA 123,0,56,24,28,56,24,
60,0
29058 DATA 1,0,0,0,0,0,0,0,0

```

Korelacja wielu zmiennych

Program pozwala obliczyć współczynniki równania regresji, średni błąd szacunku (tzw. standardowy błąd oceny), wartości regresyjne, współczynniki korelacji prostej (wyrażającej związek danej pary cech obliczony bez względu na ewentualny związek z innymi cechami) oraz współczynniki korelacji wielokrotnej, dla dowolnej liczby P zmiennych i dowolnej liczby N prób (ograniczenia wynikają tylko z pojemności pamięci). Oprócz tego obliczane są i wyświetlane na monitorze wyniki pośrednie:

SIGMAX(P) — suma (po N) wartości zmiennej X z indeksem P,

SIGMAX2(P) — suma kwadratów zmiennej X,

XSRED(P) — wartość średnia zmiennej X,

NS2X(P) — suma kwadratów odchyłeń po jedynczych wyników od średniej,

SIGMAXX(P,P) — suma iloczynów mieszanych,

NSXX(P,P) — suma iloczynów mieszanych odchyłeń od ich średnich.

Wartości sum SIGMAX(P), SIGMAXX(P,P), NS2X(P) oraz NSXX(P,P) przechowywane są w pamięci (zadeklarowane jako zmienne tablicowe w linii 120), stąd przy ewentualnych przeróbkach programu mogą być bezpośrednio wykorzystane do obliczeń. Pozostałe sumy i wartości średnie należy w takim przypadku zadeklarować jako zmienne tablicowe i nadać im wartości (np. w linii 300).

Znajomość współczynników korelacji prostej pozwala na obliczenie współczynników korelacji cząstkowej.

Aby uczynić program jak najbardziej czytelnym, a tym samym ułatwić użytkownikom

dokonywanie przerobek, nadano zmiennym, wykorzystywanym w trakcie obliczeń, nazwy odpowiadające typowemu sposobowi zapisu stosowanemu w podręcznikach z zakresu statystyki (W. Oktaba, E. Niedokos: Matematyka i podstawy statystyki matematycznej, PWN, Warszawa 1974).

Dane, X(P,N), wprowadza się instrukcjami DATA (poczynając od linii 1400), wg narastającej wartości P (przy ustalonym N). Ilustruje to poniższy przykład, dla P = 2: 1400 DATA X(1,1), X(2,1), ..., X(1,N), X(2,N). Na pierwszym miejscu (X(1,1), ..., X(1,N), ...) wprowadza się zmienną niezależną. Wprowadzanie danych można przerwać (bez uszczerbku w procesie wprowadzania) w dowolnym momencie, naciskając jednocześnie klawisze **CONTROL** i **1**; wznowienie procesu wczytywania następuje po kolejnym naciśnięciu klawiszy. Operacja ta wymaga pewnej wprawy.

Program można wpisać bez linii 1110—1300, w których umieszczono informacje dotyczące programu (w tym przypadku również bez linii 60 i 70), jednak doświadczenie autora wskazuje, że trud związany z wprowadzeniem tej informacji opłaca się sownie przy dłuższej eksploatacji programu.

Metodę rozwijania równań liniowych, z których oblicza się w programie współczynniki równania regresji oraz wzory według których oblicza się błąd standardowy oceny i współczynniki korelacji wielokrotnej można znaleźć w książce: F. C. Mills: Statistical Methods..., Columbia University, third edition, N. York 1955 (wyd. w jęz. rosyjskim z 1958 r.).

Błędem standardowym nazywa się w programie błąd uzyskany w wyniku dzielenia, w którym mianownik równy jest N, a

błędem standardowym poprawionym — błąd uzyskany w wyniku dzielenia, w którym mianownik równy jest ((N - P).

```

10 REM Strona tytułowa
20 DIM Q$(3)
30 ? CHR$(125):GRAPHICS 2:SETCOL
OR 2,0,0:POSITION 6,2:?"#6;"KORE
LACJA "
40 POSITION 8,4:?"#6;"WIELU":POS
ITION 6,6:?"#6;"ZMIENNYCH"
50 FOR ZMP=1 TO 1000:NEXT ZMP
60 ? "Czy chcesz informacje o pr
ogramie? T/N":INPUT Q$
70 IF Q$="T" THEN GRAPHICS 0:GOS
UB 1120
80 ? CHR$(125)
90 ? :?"Podaj liczbę zmiennych
P=";INPUT P
100 ? :?"Podaj liczbę prob N=";
:INPUT N
110 GRAPHICS 0
120 DIM X(P,N),A(P,P),B(P),SIGMA
X(P),SIGMAXX(P,P),NS2X(P),NSXX(P
,P),Z(P)
130 REM Wprowadzanie danych
140 FOR F=1 TO N:FOR G=1 TO P
150 READ L:X(G,F)=L
160 ? "X(";G;",";F;")=";X(G,F)
170 NEXT G:NEXT F
180 ? :?"Koniec wprowadzania da
nych":?"c.d. po nac. ";CHR$(
210);CHR$(197);CHR$(212):INPUT Q
$
190 ? CHR$(125):?"Czy odczytuje
sz wartosci posrednie? TAK-1;N
IE-2":INPUT QWP
200 REM Podprogram: obliczanie w
artosci posrednich
210 REM Obliczanie SIGMAX, SIGMA
X2, NS2X i XSRED
220 FOR G=1 TO P:SIGMAX=0:SIGMAX
2=0
230 FOR F=1 TO N
240 SIGMAX=SIGMAX+X(G,F)

```



```

250 SIGMAX2=SIGMAX2+X(G,F)*X(G,F)
260 NEXT F
270 NS2X=SIGMAX2-(SIGMAX^2)/N
280 XSRED=SIGMAX/N
290 IF G=1 THEN B(G)=SIGMAX:ZD2=
SIGMAX2
300 A(G,G)=SIGMAX2:SIGMAX(G)=SIG
MAX:NS2X(G)=NS2X
310 IF G>1 THEN A(1,G)=SIGMAX:A
(G,1)=SIGMAX
320 ? CHR$(125)
330 ? :? "SIGMAX(";G;")=";SIGMAX
: ? :? "SIGMAX2(";G;")=";SIGMAX2
340 ? :? "NS2X(";G;")=";NS2X: ?
? "XSRED(";G;")=";XSRED
350 IF OWP=2 THEN 370
360 ? :? :? :? "c.d. po nac. ";C
HR$(210);CHR$(197);CHR$(212):INF
UT Q$
370 NEXT G
380 REM Obliczanie SIGMAXX
390 FOR G=1 TO P-1
400 FOR J=G+1 TO P:SIGMAXX=0
410 FOR F=1 TO N
420 SIGMAXX=SIGMAXX+X(G,F)*X(J,F)
430 NEXT F
440 NSXX=SIGMAXX-(SIGMAX(G)*SIGM
AXX(J))/N
450 SIGMAXX(G,J)=SIGMAXX:NSXX(G,
J)=NSXX:IF G=1 THEN B(J)=SIGMAXX
460 IF G>1 THEN A(G,J)=SIGMAXX:A
(J,G)=SIGMAXX
470 ? CHR$(125)
480 ? :? "SIGMAXX(";G;";";J;")="
;SIGMAXX
490 ? :? "NSXX(";G;";";J;")=";NS
XX
500 IF OWP=2 THEN 520
510 ? :? :? :? :? "c.d. po nac.
";CHR$(210);CHR$(197);CHR$(212):
INPUT Q$
520 NEXT J
530 NEXT G
540 A(1,1)=N
550 REM Podprogram: obliczanie ws
półczynnika równania regresji (
układ równań liniowych)
560 ZMP=1.0E-07
570 FOR F=1 TO P:ZMD=ZMP
580 FOR I=F TO P:G=A(I,F):IF ABS
(G)>ABS(ZMD) THEN ZMD=G:J=I:NEXT
I
590 IF ABS(ZMD)<=ZMP THEN ? "Ukl
ad równań nie ma jednoznacznieo
rozwiązania":END
600 FOR I=F TO P:G=A(J,I):A(J,I)
=A(F,I):A(F,I)=G:NEXT I
610 G=B(J):B(J)=B(F):B(F)=G:IF F
=P THEN GOTO 640
620 FOR I=F+1 TO P:G=A(I,F)/ZMD:
FOR J=F+1 TO P:A(I,J)=A(I,J)-G*A
(F,J):NEXT J:B(I)=B(I)-G*B(F):NE
XT I
630 NEXT F
640 Z(P)=B(P)/A(P,P)
650 FOR F=P-1 TO 1 STEP -1:Z(F)=
B(F)/A(F,F):FOR J=F+1 TO P:Z(F)=
Z(F)-A(F,J)/A(F,F)*Z(J):NEXT J:N
EXT F
660 REM Wydruk
670 ? CHR$(125)
680 POSITION 13,1: ? "RÓWNAŃ RE
GRESJI"
690 POSITION 3,3: ? "X1 = B11 + B
12*X2 + B13*X3 + B14*X4 +
700 POSITION 12,4: ? "+ B15*X5 +
..... + B1F*XP"
710 ? :? "sdzie dla P=";P: ?

```

```

720 LI=0
730 FOR I=1 TO P
740 PRINT " B1";I;"=";Z(I)
750 LI=LI+1:IF LI=8 THEN LI=0: ?
: ? "c.d. po nac. ";CHR$(210);CHR
$(197);CHR$(212):INPUT Q$
760 NEXT I
770 REM Podprogram: obliczanie w
spółczynnika korelacyjnych
780 REM Obliczanie błędu stand.
i wsp. kor. wielokrotnej
790 ZMP=0
800 FOR G=2 TO P:ZMP=ZMP+Z(G)*SI
GMAXX(1,G):NEXT G
810 ZMP=(ZD2-Z(1)*SIGMAX(1)-ZMP)
/N
820 ? :? "Bład standardowy=";SQR
(ZMP)
830 ZMP=ZMP*(N/(N-P))
840 ? :? "Bład stand. poprawiony
=";SQR(ZMP)
850 ZMP=0
860 FOR G=2 TO P:ZMP=ZMP+Z(G)*SI
GMAXX(1,G):NEXT G
870 K=N*((SIGMAX(1)/N)^2
880 ZMP=(Z(1)*SIGMAX(1)+ZMP-K)/(
ZD2-K)
890 ? :? "Współczynnik korelacji
=";SQR(ZMP)
900 ? :? "c.d. po nac. ";CHR$(21
0);CHR$(197);CHR$(212):INPUT Q$
910 REM Obliczanie wartości rear
esyjnych
920 ? CHR$(125):POSITION 12,2: ?
"WARTOŚCI REGRESYJNE": ? :FOR I=1
TO 500:NEXT I
930 LI=0
940 FOR F=1 TO N:ZMP=0
950 FOR G=2 TO P
960 ZMP=ZMP+Z(G)*X(G,F)
970 NEXT G
980 ZMP=ZMP+Z(1)
990 ? "Y(";F;")=";ZMP
1000 LI=LI+1:IF LI=15 THEN LI=0:
? "c.d. po nac. ";CHR$(210);CHR$
(197);CHR$(212):INPUT Q$
1010 NEXT F
1020 ? :? "c.d. po nac. ";CHR$(2
10);CHR$(197);CHR$(212):INPUT Q$
1030 REM Obliczanie wsp. kor. pr
ostejszy
1040 ? CHR$(125): ? "WSPÓLCZYNNIK
I KORELACJI PROSTEJ": ? :FOR ZMP=
1 TO 1000:NEXT ZMP:LI=1
1050 FOR F=1 TO P-1:FOR G=F+1 TO
P
1060 ZMP=NSXX(F,G)/SQR(NS2X(F)*N
S2X(G))
1070 ? "R";F;G;"=";ZMP
1080 LI=LI+1:IF LI=15 THEN LI=0:

```

```

? :? "c.d. po nac. ";CHR$(210);C
HR$(197);CHR$(212):INPUT Q$
1090 NEXT G:NEXT F
1100 END
1110 REM Blok informacji o progr
amie
1120 ? CHR$(125):GRAPHICS 0
1130 ? "Program oblicza współczy
nniki równania regresji, bład st
andardowy, współczynnik korelac
ji ";
1140 ? "wielokrotnej, wartości r
egresyjne oraz współczynniki kor
elacji prostej";
1150 ? " ";: ? "dla dowolnej liczy
by P zmiennych (w tym zm. nie
zależna)";
1160 ? "i dowolnej liczby N prob
(wielkości P i N ograniczone
są tylko pojemnością pamięci).
"
1170 ? "Dane X(P,N) wprowadza się
instrukcjami DATA (poczynając o
d linii 1400) wg narastającej war
tości indeksu P."
1180 ? "Przykład wpisywania dany
ch (dla P=2):"
1190 POSITION 1,14: ? "1400 DATA
";CHR$(216);CHR$(168);CHR$(177);
CHR$(172);CHR$(177);CHR$(169);
1191 ? ",X(2,1),,,";CHR$(216);CH
R$(168);CHR$(177);CHR$(172);CHR$
(206);CHR$(169);",X(2,N)"
1200 ? "W inverse video zapisana
jest zmienna niezależna."
1210 ? :? "c.d. po nac. ";CHR$(2
10);CHR$(197);CHR$(212):INPUT Q$
1220 ? CHR$(125):POSITION 14,1: ?
"OZNACZENIA"
1230 ? :? "SIGMAX(P) - suma (po
N) wartości zmienn
nej
X z indeksem P"
1240 ? :? "SIGMAX2(P) - suma kwa
dratów zmiennej X"
1250 ? :? "XSRED(P) - wartość sre
dnia zmiennej X"
1260 ? :? "NS2X(P) - suma kwadra
tów odchylen po-
jedyn
czych wyników od sred-
niej"
1270 ? :? "SIGMAXX(P,P) - suma il
oczynów miesza-
nych"
1280 ? :? "NSXX(P,P) - suma iloc
zynów mieszanych
odc
hylen"
1290 ? :? "c.d. po nac. ";CHR$(2
10);CHR$(197);CHR$(212):INPUT Q$
1300 RETURN

```

SPROSTOWANIE

W opublikowanym zeszycie nr 1 programów komputerowych „IKS-a” złośliwy chochlik drukarski „poobcinał” niektóre linie listingu programu MINI-ASSEMBLER. Oto ich wykaz:

linia 100 — jest ...POP:GOTO, powinno być ...POP:GOTO START.

linia 3130 — jest ...DEC = N2., a powinno być ...DEC = D2...

linia 30010 — ostatnią zmienną w instrukcji READ jest START a nie STAR jak na listingu.

linia 30110 — w linii brakuje trzech znaków, oto one SR”

linia 30180 — na końcu linii powinno być ...MENU.PRINT a nie ...MENU”.P.

Za powstałe z tego powodu kłopoty z uruchomieniem programu wszystkich czytelników serdecznie przepraszamy.

REDAKCJA



AMSTRAD

PROGRAM

11

Technologia przenoszenia programów ze Spectrum na Amstrada

Prezentowane rozwiązanie pozwala na przenoszenie programów czy zbiorów danych napisanych w języku Basic z mikrokomputera Spectrum na mikrokomputer Amstrad. Każdy program (zbiór), który można wylistować na Spectrum komendami **LLIST** lub **LPRINT** może być przesyłany, przy użyciu taśmy kasetowej, do mikrokomputera Amstrad. Użyta może być dowolna wersja Spectrum (najstarsza 48 k. Spectrum + Spectrum 128 lub najnowszy Spectrum 128+2).

Z wersji mikrokomputerów Amstrad można zastosować CPC 464, CPC 664 lub CPC 6128. Bardziej dogodna w użyciu jest dyskowa wersja Amstrada (CPC 664 lub CPC 6128), nie mniej w ostateczności można wykorzystać wersję taśmową CPC 464. Całość operacji realizują trzy z prezentowanych czterech programów. Pierwszy z nich (wydruk 1) rezyduje w górnej pamięci. Spectrum i dostarcza kody dla komendy **SAVE**, dzięki której poprzez instrukcje **LLIST** lub **LPRINT** następuje zapis przesyłanego programu (zbioru) danych na taśmę kasetową. Jednocześnie przesyłany program (zbiór) jest zapisywany w taki sposób, że może być z łatwością odczytany na Amstradzie. Pozostałe trzy programy są związane z mikrokomputerem Amstrad. Pierwszy z nich (wydruk 2), to program w kodzie maszynowym, którego działanie wzorowane jest na instrukcji **LOAD** używanej w mikrokomputerze Spectrum. Postać binarną, **ZX.BIN** pierwszego programu (wydruk 2) wykorzystuje napisany w Basic'u program kontrolny (wydruk 3 lub wydruk 4).

Powyższe programy służą do przesyłania innych programów lub zbiorów danych, zapisanych w języku **Basic** — całej operacji nie należy jednak utożsamiać z translacją, nie mamy tu także do czynienia z kompatybilnością mikrokomputerów. Mówiąc prościej, chociaż omawiane cztery programy zapewniają przeniesienie innych programów (zbiorów danych) to jedynie najprostszą ich wersję nadają się natychmiast do uruchomienia na mikrokomputerze Amstrad. Języka Basic na mikrokomputer Amstrad nie należy utożsamiać z językiem Basic na Spectrum. W Basic'u na Spectrum jest wiele elementów, które będą sygnalizowane na Amstradzie jako błędy składniowe.

Przed operacją przesyłania należy program przejrzeć i dokonać ewentualnych poprawek. Basic na Amstradzie dopuszcza w jednej linii programowej maksimum 255 znaków i dlatego zbyt długie linie programowe na Spectrum należy skrócić, wstawiając nowe linie. Wychwycenie zbyt długich linii programowych jest bardzo ważne, gdyż w innym razie ładowanie na Amstrada programu przeniesionego ze Spectrum, zakończy się błędem, a całość operacji trzeba będzie powtórzyć.

Poprawianie sygnalizowanych błędów składniowych można dokonywać po przeniesieniu programu na Amstrada, wykorzystując metodę kursora kopiującego, czy korzystając ze specjalizowanego oprogramowania mikrokomputerowego. Programy, posiadające bogatą grafikę będą przesyłane dłużej niż pozostałe. Programy, zawierające większą ilość tekstów i danych w kodzie ASCII, np. gry przygodowe powinny się dać łatwo przenieść.

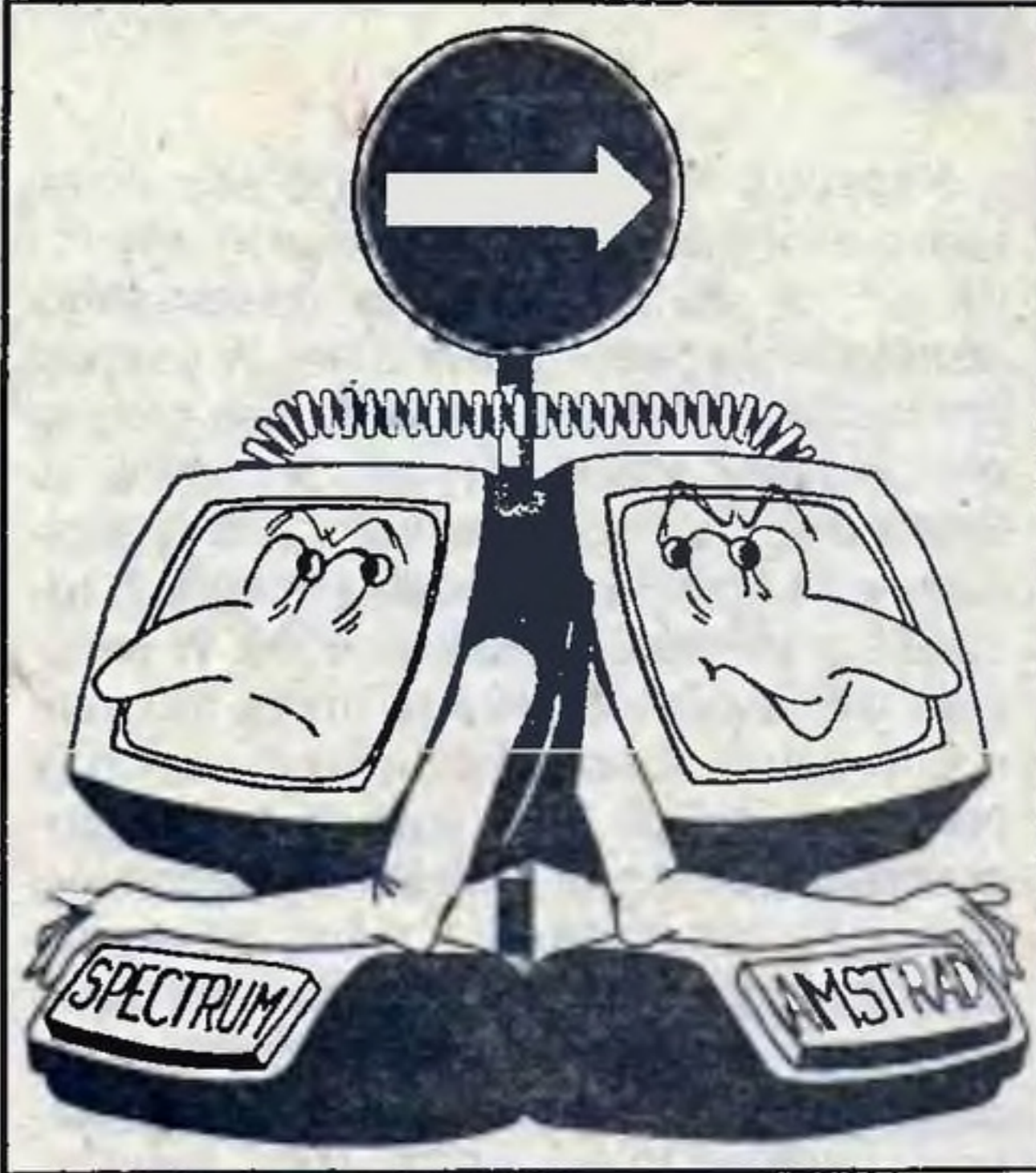
Programy, w których występuje duża ilość instrukcji **PEEK** i **POKE** właściwie nie nadają się do przeniesienia. W celu przesyłania programu ze Spectrum na Amstrada używa się taśmy kasetowej. Zapis programu na kasetę umożliwia program pokazany na wydruku pierwszym oraz instrukcja **LLIST** (przy przesyłaniu danych lepiej wykorzystać instrukcję **LPRINT**). Aby oznaczyć koniec przesyłanego programu należy jako ostatnią linię programową, wprowadzić instrukcję **REM @ X**, gdzie symbol @ oznacza **COPYRIGHT**.

Przesyłany program wprowadzany jest najpierw do buforu RAM, a potem wysyłany na taśmę. Bezpośrednie przesyłanie programu na kasetę instrukcją **LLIST** bajt po bajcie nie jest możliwe. Program w kodzie maszynowym (wydruk 1) zajmuje górne adresy pamięci od F000H do F1DFH, natomiast przesyłany program w Basic'u może zająć pozostałą jej część. Program przesyłający (wydruk 1) nie jest uruchamiany automatycznie i dlatego przed użyciem instrukcji **LLIST** lub **LPRINT** należy podać:

RANDOMIZE USR 61440.

W tym momencie program binarny pozostaje w stanie oczekiwania na instrukcję **LLIST** lub **LPRINT**. Przesyłany program jest zapisywany na taśmie kasetowej z bufora pamięci, służącego do przechowywania obrazu ekranu tzw. video RAM (VRAM), zajmujący obszar od adresu 4000H do adresu 5B00H. Omawiany bufor służy do przechowywania danych wyjściowych skierowanych tam zamiast na drukarkę. Zapis na taśmę w postaci zbioru binarnego odbywa się kiedy bufor VRAM będzie zapelniony lub, gdy wcześniej zakończy się wykonywanie instrukcji **LLIST**. Należy zauważyć, że chociaż zapisywany jest zbiór binarny to przenoszona treść jest w swojej naturze czysto tekstowa.

Większość programów na Spectrum może być przesyłana w jednym cyklu, ale niektóre szczególnie długie programy będą musiały być podzielone i przesyłane w dwóch lub więcej częściach. Wydaje się to dziwne tym bardziej że oba mikrokomputery mogą mieć taką samą wielkość pamięci użytkowej RAM. Powyższa rozbieżność wynika stąd, że program w Basic'u na Spectrum jest zwarty, gdy rezyduje w pamięci RAM, ale rozszerza się w trakcie listowania. Na przykład słowo kluczowe **RANDOMIZE** jest jednobajtowym symbolem w pamięci RAM, ale w trakcie listowania wyrażone będzie znakami zajmującymi 9 bajtów. Z drugiej strony liczby na



Spectrum zajmują w czasie listowania mniej miejsca. O ile dany program się rozszerzy decyduje jego rodzaj. „Symbol graficzny”, „Kształt” zdefiniowany przez użytkownika zajmuje jeden bajt pamięci RAM, ale na Amstrad jest przesyłany jako trzycyfrowy kod w klamrach np. [133], zajmując aż cztery bajty. Przechodząc do praktycznych rozważań w pierwszej kolejności powinniśmy się dowiedzieć w ilu częściach będziemy musieli przesyłać program i z którego programu kontrolnego należy skorzystać. W tym celu po załadowaniu na Spectrum, określamy górny adres, zajmowany przez interesujący nas program, podając:

PRINT PEEK (23628) * 256 + PEEK (23627).

- Jeśli uzyskana wartość jest mniejsza od 39 000 to przesyłanie programu powinno się zrealizować w jednym cyklu. Na Amstradzie wykorzystujemy program kontrolny z wydruku trzeciego.
- Jeśli uzyskana wartość jest zawarta pomiędzy 39 000 i 50 000 to przesyłanie programu będzie zrealizowane także w jednym cyklu ale na Amstradzie powinien być wykorzystany program kontrolny z wydruku czwartego.
- Jeśli uzyskana wartość jest większa od 50 000 to oznacza, że program powinien zostać podzielony na dwie części i przesyłany w dwóch cyklach. Na Amstradzie należy wykorzystać program kontrolny z wydruku czwartego.

Po uruchomieniu jednego z dwóch programów kontrolnych pojawi się na monitorze pytanie:

PODAJ ILOŚĆ ŁADOWANYCH SEGMENTÓW

Aby wpisać prawidłową ilość segmentów należy pamiętać, aby po zakończeniu instrukcji **LLIST** wprowadzić:

PRINT PEEK (61506) — 64

kłóra spowoduje wypisanie liczby, określającej ilość segmentów. W celu przywrócenia użytych instrukcjom **LLIST** lub **LPRINT** ich normalnego działania należy podać:

RANDOMIZE USR 61490

Po zapisaniu na taśmę kasetową przesyłanego programu nie można stwierdzić czy treść programu na taśmie jest poprawna. Aby to sprawdzić wgrujemy program z taśmy do pamięci Spectrum. Każdy błąd będzie sygnalizowany jako błąd ładowania taśmy.





Nadszedł wreszcie moment, aby wziąć taśmę z poprawnie nagrany programem i włożyć ją do odtwarzacza kasetowego, współpracującego z Amstradem. W pamięci Amstrada powinny rezydować dwa programy związane z przesyłaniem programu ze Spectrum. Pierwszy o nazwie ZX.BIN zapisany w kodzie maszynowym (wydruk 2) rezyduje w pamięci od adresu 8000. W wersjach dyskowych Amstrada adres ten zaliczany jest do obszaru pamięci górnej. Drugi program (wydruk 3 lub wydruk 4) jest programem kontrolnym i mieści się w pamięci dolnej. Po załadowaniu z taśmy kasetowej przesyłanego programu do pamięci RAM Amstrada, program kontrolny przepisze ten załadowany program na dyskietkę lub taśmę w postaci zbioru ASCII. Cała operacja jest realizowana w Basic'u i dlatego użytkownicy, wykorzystujący pamięć dyskową, zauważą nietypowe opóźnienia, przy realizowanym etapami zapisie programu na dyskietkę. W wyniku działania programu kontrolnego na dyskietce (taśmie) zostanie utworzony zbiór X.ASC, gdzie X jest podaną przez użytkownika nazwą przesyłanego programu, a ASC określa typ zbioru i jest dopisywany automatycznie przez program kontrolny.

Po zakończeniu pracy program kontrolny pozostawia MEMORY ustawione na wartość 3FFF i dlatego należy podać:

MEMORY & 7FFF

gdyż z chwilą ładowania programu do pamięci może pojawić się komunikat błędu przekroczenia pamięci. Po poprawnym załadowaniu do pamięci można rozpocząć edycję (poprawienie) przesyłanego programu. Ładowanie zakończy się komunikatem błędu, jeśli linia programu zawiera więcej niż 255 znaków.

ABY PRZESŁAĆ PROGRAM W BASIC'U Z MIKROKOMPUTERA SPECTRUM NA MIKROKOMPUTER AMSTRAD NALEŻY POSTĘPOWAĆ WEDŁUG NASTĘPUJĄCEGO ALGORYTMU:

1. Ładujemy do Spectrum program z wydruku „1” komendą **LOAD**.

2. Uruchamiamy program z wydruku „1” komendą **RUN**. W wyniku działania, program będzie chciał zapisać swoją postać binarną na taśmie (włączanie magnetofonu nie jest konieczne).

3. Ładujemy do Spectrum komendą **LOAD** program w BASIC'u, który ma być przesyłany na mikrokomputer Amstrad.

4. Należy sprawdzić czy program nie zawiera zbyt długich linii. Jeśli ilość znaków w liniach programowych przekracza liczbę 255, należy utworzyć nowe linie.

5. Wprowadzamy do programu

jako ostatnią linię wyrażenie:
REM @ X

6. Określamy górny adres pamięci przesyłanego programu podając:

PRINT PEEK (23628) * 256 + PEEK (23627)

— Jeśli uzyskana wartość jest mniejsza od 39 000, to przesyłanie programu realizujemy w jednym cyklu. Na Amstradzie powinien być wykorzystany program kontrolny z wydruku trzeciego.

— Jeśli uzyskana wartość jest zawarta pomiędzy 39 000 i 50 000, to przesyłanie programu będzie zrealizowane także w jednym cyklu, ale na Amstradzie powinien być wykorzystany program kontrolny z wydruku czwartego.

— Jeśli uzyskana wartość wynosi powyżej 50 000 to oznacza, że program powinien zostać podzielony na dwie części i przesyłany w dwóch cyklach. Na Amstradzie powinien być wykorzystany program kontrolny z wydruku czwartego.

7. Wprowadzamy rozkaz:
RANDOMIZE USR 61440

8. Wkładamy do magnetofonu kasetę z taśmą w celu nagrania przesyłanego programu i włączamy magnetofon na ZAPIS.

9. Wprowadzamy rozkaz:
LLIST

10. Po zakończeniu wykonywania instrukcji **LLIST** zatrzymujemy magnetofon. Sprawdzamy z ilu segmentów składa się przesyłany przez nas program podając:

PRINT PEEK (61506) — 64

11. Ładujemy do Amstrada komendą **LOAD** program z wydruku drugiego.

12. Uruchamiamy załadowany program komendą **RUN**, w wyniku czego na dyskietce (taśmie) zostanie zapisany program binarny **ZX.BIN**.

13. W zależności od wartości liczbowej otrzymanej w p. 6 ładujemy do Amstrada komendą **LOAD** program kontrolny z wydruku trzeciego lub wydruku czwartego.

14. Podłączamy do Amstrada magnetofon z zapisanym na taśmie kasetowej programem ze Spectrum.

15. Uruchamiamy na Amstradzie komendą **RUN** program kontrolny, podając w trakcie działania: ilość segmentów przesyłanego programu (patrz punkt 10) i jego nazwę (maksimum 6 znaków). Z chwilą pojawienia się komunikatu związanego z ładowaniem segmentu, włączamy magnetofon, umożliwiając tym samym zapis przesyłanego programu do pamięci RAM Amstrada.

16. Po załadowaniu do pamięci RAM przygotowujemy dyskietkę (taśmę) do zapisu przesyłanego ze Spectrum programu i naciskamy dowolny klawisz.

17. Po zapisaniu programu na dyskietkę (taśmę) ściągamy go do pamięci i dokonujemy niezbędnych poprawek po czym uruchamiamy komendą **RUN** poprawiony program.

UWAGA:

Jeśli przesyłamy program w dwóch częściach, to łączymy obie na Amstradzie instrukcją **MERGE**.

Życzymy owocnej pracy!

Na podstawie „YUOR COMPUTER” nr 1/87, 3/87 opracował:

Jacek SZANIAWSKI
Marian MARCINIAK

Wydruk nr 1

```
10 REM PROGRAM PRZENOSZĄCY ZBIORY Z M/K SPECTRUM NA M/K AMSTRAD
20 REM PROGRAM NA M/K SPECTRUM
30 CLEAR 61439: RESTORE 160
40 LET a=61440: LET d=499
50 LET s=0: LET d=d+1
60 FOR r=1 TO 8
70 READ v
80 LET s=s+v: POKE a,v: LET a=a+1
90 NEXT r
100 READ chk: IF chk<>s THEN PRINT "WIERSZ O NR ";d;" JEST NIEPRAWIDŁOWY":STOP
110 IF a>61919 THEN GOTO 130
120 GOTO 50
130 PRINT "ŁADOWANIE"
140 PRINT "ZACHOWANIE ZBIORU NA TASMIE..."
150 SAVE "transfer" code 61440,488: STOP
160 DATA 42,79,92,17,15,0,25,94,364
170 DATA 35,86,0,0,0,0,17,37,175
180 DATA 241,43,115,35,114,221,33,82,884
190 DATA 240,221,54,0,0,221,54,1,791
200 DATA 0,221,54,2,0,221,54,3,555
210 DATA 255,221,54,4,63,221,54,240,1112
220 DATA 64,201,42,79,92,17,15,0,510
230 DATA 25,237,91,87,240,115,35,114,944
240 DATA 201,3,64,32,32,32,32,428
250 DATA 32,32,32,32,210,1,0,64,403
260 DATA 0,0,1,0,0,147,65,244,457
270 DATA 9,254,1,18,0,174,16,39,511
280 DATA 232,3,100,0,10,0,1,0,346
290 DATA 0,0,0,0,0,197,58,66,321
300 DATA 240,60,50,66,240,221,33,82,992
```

```

310 DATA 240,221,115,250,221,114,251,221,1633
320 DATA 33,65,240,213,17,17,0,175,760
330 DATA 205,194,4,209,6,50,118,16,802
340 DATA 253,221,33,0,64,62,255,205,1093
350 DATA 194,4,193,201,197,33,0,64,886
360 DATA 54,13,17,1,64,1,255,26,431
370 DATA 237,176,193,201,229,213,197,79,1525
380 DATA 58,82,240,254,0,32,8,205,879
390 DATA 156,240,62,1,50,82,240,58,889
400 DATA 83,240,254,1,40,54,42,85,799
410 DATA 240,35,34,85,240,124,254,91,1103
420 DATA 32,20,125,254,0,32,15,17,495
430 DATA 0,27,205,109,240,205,156,240,1182
440 DATA 33,0,64,34,85,240,121,254,831
450 DATA 127,40,54,119,58,84,240,254,976
460 DATA 1,40,20,121,254,127,40,8,611
470 DATA 175,50,84,240,193,209,225,261,1377
480 DATA 62,1,50,84,240,24,245,121,827
490 DATA 254,88,32,231,17,255,63,237,1177
500 DATA 82,235,205,109,240,62,1,50,984
510 DATA 83,240,6,24,205,68,14,24,664
520 DATA 219,62,99,24,198,245,254,13,1114
530 DATA 40,86,254,32,56,8,254,128,858
540 DATA 56,78,254,165,48,74,71,58,804
550 DATA 92,240,183,120,32,61,38,0,766
560 DATA 111,17,104,240,221,33,94,240,1060
570 DATA 62,47,221,78,0,221,70,1,700
580 DATA 198,1,237,66,48,250,9,18,827
590 DATA 221,35,221,35,19,13,32,232,808
600 DATA 62,91,205,152,241,58,106,240,1155
610 DATA 205,152,241,58,107,240,205,152,1360
620 DATA 241,58,108,240,205,152,241,62,1307
630 DATA 93,24,24,58,93,240,24,19,575
640 DATA 254,255,40,8,254,165,56,11,1043
650 DATA 254,255,48,7,214,165,205,95,1243

```

```

560 DATA 11,241,201,205,152,241,241,201,1493
570 DATA 229,111,58,89,240,60,103,58,948
580 DATA 91,240,60,50,91,240,188,32,992
590 DATA 14,205,198,241,175,50,91,240,1214
700 DATA 125,254,13,40,15,24,227,125,823
710 DATA 254,13,32,5,175,50,91,240,860
720 DATA 125,205,204,241,225,201,62,13,1276
730 DATA 205,204,241,201,245,205,172,240,1717
740 DATA 254,13,32,10,58,90,240,32,729
750 DATA 5,62,10,205,172,240,241,201,1136
760 DATA 0,0,0,0,0,0,0,0,0

```

Wydruk nr 2

```

10 REM PROGRAM LADUJACY NA M/K AMSTRAD
20 MEMORY &3FFF:RESTORE 150
30 A=&8000:D=499
40 S=0:D=D+1
50 FOR R=1 TO 8
60 READ V$:V=VAL("&"+V$)
70 S=S+V:POKE A,V:A=A+1
80 NEXT R
90 READ V$:CHK=VAL("&"+V$)
100 IF CHK<>S THEN GOTO 130
110 IF A>&812F THEN SAVE "ZX.BIN",B,&8000,
304:STOP
120 GOTO 40
130 PRINT "WIERSZ O NR ";D;" JEST BLEDNY"
140 STOP
150 DATA cd,09,80,c9,00,40,00,00,025f
160 DATA 00,cd,a7,bc,cd,6e,bc,f3,051a
170 DATA af,32,06,80,08,d9,f5,d5,0412
180 DATA 16,10,d9,01,0e,f4,ed,49,0338
190 DATA 01,d0,f6,ed,49,0e,10,ed,0408
200 DATA 49,01,92,f7,ed,49,01,58,0362
210 DATA f6,ed,49,dd,21,2b,81,dd,04b3
220 DATA e5,11,11,00,af,cd,7f,80,0382

```

```

230 DATA dd,e1,30,ef,3a,2b,81,fe,04c1
240 DATA 03,20,e8,3e,ff,ed,5b,36,03c6
250 DATA 81,ed,53,07,80,dd,2a,04,0353
260 DATA 80,cd,7f,80,08,d9,d1,f1,04ef
270 DATA d9,08,01,82,f7,ed,49,01,0392
280 DATA 10,f6,ed,49,dd,22,04,80,03bf
290 DATA fb,f5,cd,71,bc,f1,38,01,0514
300 DATA c9,3e,01,32,06,80,c9,14,029d
310 DATA 08,15,3e,f5,db,00,e6,80,0391
320 DATA d9,1e,4c,d9,4f,bf,c0,cd,04b7
330 DATA 02,81,30,fa,21,15,04,10,01f7
340 DATA fe,2b,7c,b5,20,f9,cd,fe,053e
350 DATA 80,30,eb,06,9c,cd,fe,80,0488
360 DATA 30,e4,3e,c0,b8,30,e0,24,03fe
370 DATA 20,f1,06,c9,cd,02,81,30,0360
380 DATA d5,78,fe,d4,30,f4,cd,02,0512
390 DATA 81,d0,d9,1e,4a,d9,26,00,0391
400 DATA 06,b0,18,18,08,20,05,dd,01f0
410 DATA 75,00,18,0a,cb,19,ad,c0,02e8
420 DATA 79,17,4f,13,18,02,dd,23,020c
430 DATA 1b,08,06,b2,2e,01,cd,fe,02d5
440 DATA 80,d0,3e,bf,b8,cb,15,06,03eb
450 DATA b0,30,f3,7c,ad,67,7a,b3,0490
460 DATA 20,d2,7c,fe,01,c9,cd,02,0405
470 DATA 81,d0,3e,16,3d,20,fd,a7,03ab
480 DATA 04,c8,3e,f4,db,00,e6,04,03c3
490 DATA ee,04,c0,3e,f5,db,00,a9,0469
500 DATA e6,80,28,ec,79,2f,4f,d9,044a
510 DATA 7b,ee,1f,5f,ed,51,ed,59,046b
520 DATA d9,37,c9,00,00,00,00,00,01d9

```

Wydruk nr 3

```

10 REM PROGRAM KONTROLNY NR 3 (BASIC-AMSTRAD)
20 :DISC:::BORDER 13:INK 0,23:INK 1,0:INK 2,0:INK 3,11
30 KEY 9,CHR$(174):MODE 2:MEMORY &3FFF:LOAD "zx.bin":TAPE.IN
40 POKE &8004,0:POKE &8005,&40 : 'PRZEZNACZA SIE ADRES &4000
50 TL=0:S=1:INPUT "PODAJ ILOSC LADOWANYCH SEGMENTOW ";N
60 PRINT "PODAJ NAZWE PRZESYLANEGO ZBIORU (MAX 6 ZNAKOW)"
70 PRINT "PODANA NAZWA NIE MOZE ZAWIERAC ROZSZERZENIA"
80 INPUT "NAZWA ZBIORU ";A$:A$=UPPER$(A$):A$=A$+".ASC":PRINT
90 PRINT "WLACZ KASETE ZE ZBIOREM DO PRZEGRANIA":PRINT
100 :
110 FOR R=1 TO N
120 PRINT "LADOWANIE SEGMENTU O NR ";S;" ZBIORU ZE SPEKTRUM":CALL &8000
130 F=PEEK(&8006):IF F=0 THEN PRINT "BLEDNE LADOWANIE ZBIORU":STOP
140 L=((PEEK(&8008))*256)+PEEK(&8007):TL=TL+L
150 PRINT "WIELKOSC ZALADOWANEGO DO PAMIECI SEGMENTU O NR ";S;" WYNOСИ ";HEX$(L,4);"H"
160 S=S+1:PRINT:NEXT R
170 :
180 PRINT "WCISNIJ DOWOLNY KLAWISZ GDY DYSKIETKA(TASMA) JEST PRZYKOTOWANA DO ZAPISU"
190 PRINT "ZBIOR O NAZWIE ";A$
200 IF INKEY$="" THEN GOTO 200
210 PRINT:PRINT "ZAPIS ZBIORU - PROSZE CZEKAC":OPENOUT A$
220 FOR A=&4000 TO &4000+TL-3
230 V$=CHR$(PEEK(A)):PRINT #9,V$;:NEXT A
240 CLOSEOUT:DISC:STOP

```

Wydruk nr 4

```

10 REM PROGRAM KONTROLNY NR 4 (BASIC-AMSTRAD)
20 :DISC:::BORDER 13:INK 0,23:INK 1,0:INK 2,0:INK 3,11
30 KEY 9,CHR$(174):MODE 2:MEMORY &3FFF:LOAD "zx.bin":TAPE.IN
40 INPUT "PODAJ ILOSC LADOWANYCH SEGMENTOW ";N:T=64:S=1
50 PRINT "PODAJ NAZWE PRZESYLANEGO ZBIORU (MAX 6 ZNAKOW)"
60 PRINT "PODANA NAZWA NIE MOZE ZAWIERAC ROZSZERZENIA"
70 INPUT "NAZWA ZBIORU ";A$:A$=UPPER$(A$):CLS
80 IF LEN(A$)>6 THEN A$=LEFT$(A$,6)
90 :
100 FOR R=1 TO N:CLS
110 PRINT "ZALDZ TASME ZE ZBIOREM DO PRZEGRANIA"
120 PRINT "WCISNIJ DOWOLNY KLAWISZ"
130 IF INKEY$="" THEN GOTO 130
140 POKE &8004,0:POKE &8005,&40:T=T+1
150 PRINT "LADOWANIE SEGMENTU O NR ";S;" Z TASY ZE SPEKTRUM"
160 CALL &8000:F=PEEK(&8006):IF F=0 THEN PRINT "BLEDNE LADOWANIE":STOP
170 L=((PEEK(&8008))*256)+PEEK(&8007):B$=A$+"-"+CHR$(T)+".ASC"
180 S=S+1:PRINT "WLOZ DYSKIETKE(TASME) DO ZAPISU ZBIORU"
190 PRINT "ZAPIS ZBIORU <<";B$;">> - WCISNIJ DOWOLNY KLAWISZ"
200 IF INKEY$="" THEN GOTO 200
210 PRINT:PRINT "ZAPIS ZBIORU - PROSZE CZEKAC":OPENOUT B$
220 FOR A=&4000 TO &4000+L-2
230 V$=CHR$(PEEK(A)):PRINT #9,V$;:NEXT A
240 CLOSEOUT:NEXT R
250 :DISC:STOP

```

„REZYSTOR”

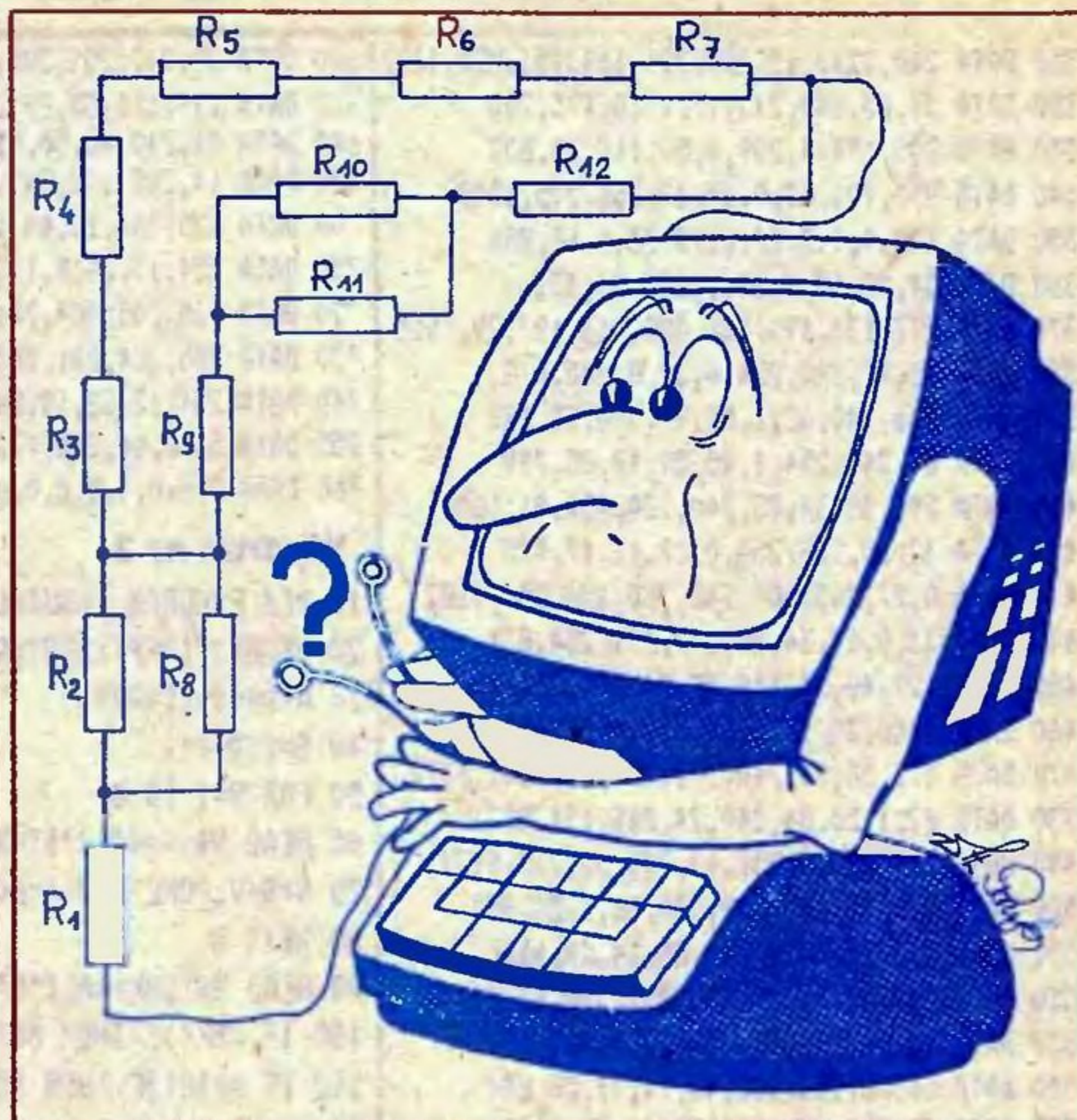
Po załadowaniu i uruchomieniu programu komendą RUN na ekranie monitora pojawi się opis czterech czynności. Chcąc wybrać jedną z czynności przyciskamy klawisz 1,2,3 lub 4. Pierwsze dwie czynności przedstawiają szeregowe i równoległe połączenie rezystorów. Obok rysunków typowych schematów pokazane są wzory na rezystancję zastępczą szeregowo lub równoległe połączonych rezystorów. Kolejne dwie czynności przedstawiają rysunki oraz umożliwiają obliczenie rezystancji obwodu przy przejściu z połączenia rezystorów w „trójkąt” w połączenie w „gwiazdę” i odwrotnie.

Program ten okaże się pomocny w nauce „Teorii obwodów”.

```

10 MODE 1
20 WINDOW #1,10,27,23,25
30 PRINT:PRINT
40 PRINT "1. POLACZENIE SZEREGOWE REZYSTOROW"
50 PRINT:PRINT
60 PRINT "2. POLACZENIE ROWNOLEGLE REZYSTOROW"
70 PRINT:PRINT
80 PRINT "3. OBLICZANIE REZYSTANCJI ZASTEPCZEGO"
90 PRINT "  POLACZENIA TYPU 'TROJKAT' PRZY"
100 PRINT "  PRZEJSCIU Z POLACZENIA TYPU 'GWIAZDA'  W POLACZENI
E TYPU 'TROJKAT'"
110 PRINT:PRINT
120 PRINT "4. OBLICZANIE REZYSTANCJI ZASTEPCZEGO"
130 PRINT "  POLACZENIA TYPU 'GWIAZDA' PRZY"
140 PRINT "  PRZEJSCIU Z POLACZENIA TYPU 'TROJKAT'  W POLACZENI
E TYPU 'GWIAZDA'"
150 INK 1,24:PEN #1,0,0:PAPER #1,13
160 LOCATE #1,11,24
170 PRINT #1," WYBIERZ CZYNNOSC"
180 #=#INKEY#
190 IF #=# THEN 180
200 IF #=#1 THEN 250
210 IF #=#2 THEN 630
220 IF #=#3 THEN 1090
230 IF #=#4 THEN 1810
240 GOTO 180
250 REM *****
260 REM #
270 REM #  POLACZENIE SZEREGOWE REZYSTOROW #
280 REM #
290 REM *****
300 MODE 1
310 PRINT SPC(4);"POLACZENIE SZEREGOWE REZYSTOROW"
320 LOCATE 7,7:PRINT CHR$(79)
330 LOCATE 5,9:PRINT "U"
340 MOVE 103,233:DRAWR 0,50:DRAWR -5,-10:DRAWR 5,10:DRAWR 5,-10
350 MOVE 114,297:DRAWR 26,0:DRAWR 0,8:DRAWR 34,0:DRAWR 0,-17:DRAW
R -34,0:DRAWR 0,8:DRAWR 0,-8:DRAWR 34,0:DRAWR 0,8:DRAWR 34,0
360 DRAWR 0,-22:DRAWR 8,0:DRAWR 0,-34:DRAWR -17,0:DRAWR 0,34:DRAW
R 8,0:DRAWR -8,0:DRAWR 0,-34:DRAWR 9,0

```



```

370 DRAWR 0,-22:DRAWR -34,0:DRAWR 0,8:DRAWR 0,-17:DRAWR -34,0:DR
AWR 0,17:DRAWR 34,0:DRAWR -34,0:DRAWR 0,-9:DRAWR -28,0
380 LOCATE 7,12:PRINT CHR$(79)
390 LOCATE 10,4:PRINT "R":LOCATE 11,5:PRINT "1"
400 LOCATE 15,9:PRINT "R":LOCATE 16,10:PRINT "2"
410 LOCATE 10,14:PRINT "R":LOCATE 11,15:PRINT "3"
420 MOVE 441,297:DRAWR 56,0:DRAWR 0,-22:DRAWR 8,0:DRAWR 0,-34:DR
AWR -17,0:DRAWR 0,34:DRAWR 8,0:DRAWR -8,0:DRAWR 0,-34:DRAWR 9,0
430 DRAWR 0,-22:DRAWR -56,0
440 LOCATE 33,9:PRINT "R"
450 LOCATE 34,10:PRINT "2"
460 LOCATE 28,7:PRINT CHR$(79)
470 LOCATE 28,12:PRINT CHR$(79)
480 LOCATE 26,9:PRINT "U"
490 MOVE 438,233:DRAWR 0,50:DRAWR -5,-10:DRAWR 5,10:DRAWR 5,-10
500 MOVE 297,250:DRAWR 40,0
510 MOVE 297,270:DRAWR 40,0
520 MOVE 357,260:DRAWR -20,20:DRAWR 20,-20:DRAWR -20,-20
530 LOCATE 6,18:PRINT "REZYSTANCJE ZASTEPCZA OBWODU"
540 LOCATE 6,19:PRINT "WYZNACZAMY ZE WZORU:"
550 LOCATE 10,21:PRINT "R = R + R + R"
560 LOCATE 11,22:PRINT "Z  1  2  3"
570 WINDOW #2,2,39,24,25
580 INK 1,24:PEN #2,0,0:PAPER #2,13
590 LOCATE #2,3,25
600 PRINT #2,"POWROT DO OPISU - [DOWOLNY KLAWISZ]"
610 IF INKEY#="" THEN 610
620 GOTO 10
630 REM *****
640 REM #
650 REM #  POLACZENIE ROWNOLEGLE REZYSTOROW #
660 REM #
670 REM *****
680 MODE 1
690 PRINT SPC(4);"POLACZENIE ROWNOLEGLE REZYSTOROW"
700 MOVE 90,326:DRAWR 154,0:DRAWR 0,-22:DRAWR 8,0:DRAWR 0,-34:DR
AWR -17,0:DRAWR 0,34:DRAWR 8,0:DRAWR -8,0:DRAWR 0,-34:DRAWR 9,0
710 DRAWR 0,-22:DRAWR -164,0
720 MOVE 180,326:DRAWR 0,-22:DRAWR 8,0:DRAWR 0,-34:DRAWR -17,0:DR
AWR 0,34:DRAWR 8,0:DRAWR -8,0:DRAWR 0,-34:DRAWR 9,0:DRAWR 0,-22
730 MOVE 116,326:DRAWR 0,-22:DRAWR 8,0:DRAWR 0,-34:DRAWR -17,0:DR
AWR 0,34:DRAWR 8,0:DRAWR -8,0:DRAWR 0,-34:DRAWR 9,0:DRAWR 0,-22
740 LOCATE 5,5:PRINT CHR$(79)
750 LOCATE 5,10:PRINT CHR$(79)
760 LOCATE 3,7:PRINT "U"

```

```

770 MOVE 70, 262: DRAW 0, 50: DRAW -5, -10: DRAW 5, 10: DRAW 5, -10
780 MOVE 80, 326: DRAW 10, 0
790 LOCATE 9, 7: PRINT "R"
800 LOCATE 10, 8: PRINT "1"
810 LOCATE 13, 7: PRINT "R"
820 LOCATE 14, 8: PRINT "2"
830 LOCATE 17, 7: PRINT "R"
840 LOCATE 18, 8: PRINT "3"
850 MOVE 461, 326: DRAW 56, 0: DRAW 0, -22: DRAW 8, 0: DRAW 0, -34: DRAW
-17, 0: DRAW 0, 34: DRAW 8, 0: DRAW -8, 0: DRAW 0, -34: DRAW 9, 0: DR
AW 0, -22: DRAW -56, 0
860 LOCATE 29, 5: PRINT CHR$(79)
870 LOCATE 29, 10: PRINT CHR$(79)
880 LOCATE 34, 7: PRINT "R"
890 LOCATE 35, 8: PRINT "Z"
900 MOVE 455, 262: DRAW 0, 50: DRAW -5, -10: DRAW 5, 10: DRAW 5, -10
910 LOCATE 27, 7: PRINT "U"
920 MOVE 320, 280: DRAW 40, 0
930 MOVE 320, 300: DRAW 40, 0
940 MOVE 380, 290: DRAW -20, 20: DRAW 20, -20: DRAW -20, -20
950 LOCATE 5, 14: PRINT "REZYSTANCJE ZASTĘPCZA OBWODU"
960 LOCATE 5, 15: PRINT "WYZNACZAMY ZE WZORU:"
970 LOCATE 27, 17: PRINT "R R R"
980 LOCATE 2, 18: PRINT "1 1 1 1": LOCATE 28, 18: PRINT "1 2 3"
990 LOCATE 2, 19: PRINT "- - - + - + - R ="
1000 LOCATE 23, 19: PRINT "-----"
1010 LOCATE 2, 20: PRINT "R R R R": SPC(5); "Z R R + R R + R R"
"
1020 LOCATE 3, 21: PRINT "Z 1 2 3": SPC(8); "1 2 1 3 2 3"
1030 WINDOW #2, 2, 39, 24, 25
1040 INK 1, 24: PEN #2, 0, 0: PAPER #2, 13
1050 LOCATE #2, 3, 25
1060 PRINT #2, "POWROT DO OPISU - (DOWOLNY KŁAWISZ)"
1070 IF INKEY$="" THEN 1070
1080 GOTO 10
1090 REM *****
1100 REM #
1110 REM # PRZEJSCIE Z POLACZENIA REZYSTOROW W 'TROJKAT' #
1120 REM # W POLACZENIE REZYSTOROW W 'GWIAZDE' #
1130 REM #
1140 REM *****
1150 MODE 2
1160 PRINT SPC(4); "OBLICZANIE REZYSTANCJI ZASTĘPCZEGO POLACZENIA
TYPU 'TROJKAT'"
1170 PRINT SPC(4); "PRZY PRZEJSCIU Z POLACZENIA TYPU 'GWIAZDA' W P
OLACZENIE TYPU 'TROJKAT'"
1180 LOCATE 76, 5: PRINT "R R"
1190 LOCATE 77, 6: PRINT "A C"
1200 LOCATE 63, 7: PRINT "R = R + R + ----"

1210 LOCATE 64, 8: PRINT "1 A C R"
1220 LOCATE 78, 9: PRINT "B"
1230 LOCATE 76, 12: PRINT "R R"
1240 LOCATE 77, 13: PRINT "A B"
1250 LOCATE 63, 14: PRINT "R = R + R + ----"
1260 LOCATE 64, 15: PRINT "2 A B R"
1270 LOCATE 78, 16: PRINT "C"
1280 LOCATE 76, 19: PRINT "R R"
1290 LOCATE 77, 20: PRINT "B C"
1300 LOCATE 63, 21: PRINT "R = R + R + ----"
1310 LOCATE 64, 22: PRINT "3 B C R"
1320 LOCATE 78, 23: PRINT "A"
1330 MOVE 20, 220: DRAW 10, 10: DRAW -3, 3: DRAW 20, 20: DRAW 7, -7: DR
AW -20, -20: DRAW -4, 4
1340 MOVE 51, 250: DRAW 10, 10: DRAW 0, 16: DRAW -4, 0: DRAW 0, 28: DR
AW 9, 0: DRAW 0, -28: DRAW -6, 0: MOVE 61, 304: DRAW 0, 16

```

```

1350 MOVE 61, 260: DRAW 10, -10: DRAW -3, -3: DRAW 20, -20: DRAW 7, 7:
DRAW -20, 20: DRAW -4, -4: MOVE 92, 230: DRAW 10, -10
1360 LOCATE 1, 12: PRINT "A"
1370 LOCATE 8, 4: PRINT "C"
1380 LOCATE 16, 12: PRINT "B"
1390 LOCATE 3, 9: PRINT "R": LOCATE 4, 10: PRINT "A"
1400 LOCATE 5, 6: PRINT "R": LOCATE 6, 7: PRINT "C"
1410 LOCATE 12, 9: PRINT "R": LOCATE 13, 10: PRINT "B"
1420 MOVE 240, 220: DRAW 40, 40: DRAW -4, 4: DRAW 20, 20: DRAW 9, -9: D
RAW -20, -20: DRAW -5, 5: MOVE 300, 280: DRAW 40, 40
1430 MOVE 340, 320: DRAW 40, -40: DRAW 4, 4: DRAW 20, -20: DRAW -9, -9
: DRAW -20, 20: DRAW 5, 5: MOVE 440, 220: DRAW -40, 40
1440 MOVE 440, 220: DRAW -84, 0: DRAW 0, 6: DRAW -32, 0: DRAW 0, -13: D
RAW 32, 0: DRAW 0, 7: MOVE 240, 220: DRAW 84, 0
1450 LOCATE 29, 12: PRINT "A": LOCATE 58, 12: PRINT "B": LOCATE 43, 4: PR
INT "C"
1460 LOCATE 34, 7: PRINT "R": LOCATE 35, 8: PRINT "1"
1470 LOCATE 50, 7: PRINT "R": LOCATE 51, 8: PRINT "3"
1480 LOCATE 43, 9: PRINT "R": LOCATE 44, 10: PRINT "2"
1490 MOVE 150, 270: DRAW 30, 0: MOVE 150, 286: DRAW 30, 0: MOVE 192, 278
: DRAW -15, 15: MOVE 192, 278: DRAW -15, -15
1500 LOCATE 2, 15: PRINT " PODAJ REZYSTANCJE "
1510 LOCATE 24, 15: PRINT " REZYSTANCJE UKŁADU W TROJKAT : "
1520 LOCATE 28, 17: PRINT "R =": LOCATE 29, 18: PRINT "1"
1530 LOCATE 28, 19: PRINT "R =": LOCATE 29, 20: PRINT "2"
1540 LOCATE 28, 21: PRINT "R =": LOCATE 29, 22: PRINT "3"
1550 LOCATE 3, 17: PRINT "R =": LOCATE 4, 18: PRINT "A"
1560 LOCATE 8, 17: INPUT ra
1570 IF ra<=0 OR ra>10000 THEN GOTO 1740
1580 LOCATE 3, 19: PRINT "R =": LOCATE 4, 20: PRINT "B"
1590 LOCATE 8, 19: INPUT rb
1600 IF rb<=0 OR rb>10000 THEN GOTO 1760
1610 LOCATE 3, 21: PRINT "R =": LOCATE 4, 22: PRINT "C"
1620 LOCATE 8, 21: INPUT rc
1630 IF rc<=0 OR rc>10000 THEN GOTO 1780
1640 r1=ra+rc+(ra*rc)/rb
1650 r2=ra+rb+(ra*rb)/rc
1660 r3=rb+rc+(rb*rc)/ra
1670 LOCATE 33, 17: PRINT r1: LOCATE 33, 19: PRINT r2: LOCATE 33, 21: PRI
NT r3
1680 GOTO 570
1690 WINDOW #5, 14, 65, 24, 25
1700 INK 1, 24: PEN #5, 0, 0: PAPER #5, 13
1710 LOCATE #5, 16, 25: PRINT #5, " PODAJ WARTOSC REZYSTANCJI Z PRZE
DZIAŁU <1,10000>"
1720 FOR i=1 TO 3000: NEXT: LOCATE 14, 24: PRINT SPC(55): LOCATE 14, 25
: PRINT SPC(55)
1730 RETURN
1740 GOSUB 1690
1750 LOCATE 6, 17: PRINT SPC(12): GOTO 1560
1760 GOSUB 1690
1770 LOCATE 6, 19: PRINT SPC(12): GOTO 1590
1780 GOSUB 1690
1790 LOCATE 6, 21: PRINT SPC(12): GOTO 1620
1800 GOTO 580

1810 REM *****
1820 REM #
1830 REM # PRZEJSCIE Z POLACZENIA REZYSTOROW W 'GWIAZDE' #
1840 REM # W POLACZENIE REZYSTOROW W 'TROJKAT' #
1850 REM #
1860 REM *****
1870 MODE 2
1880 PRINT SPC(4); "OBLICZANIE REZYSTANCJI ZASTĘPCZEGO POLACZENIA
TYPU 'GWIAZDA'"
1890 PRINT SPC(4); "PRZY PRZEJSCIU Z POLACZENIA TYPU 'TROJKAT' W P
OLACZENIE TYPU 'GWIAZDA'"

```



```

1900 LOCATE 70,5:PRINT "R R"
1910 LOCATE 71,6:PRINT "1 2"
1920 LOCATE 62,7:PRINT "R = -----"
1930 LOCATE 63,8:PRINT "A R + R + R"
1940 LOCATE 68,9:PRINT "1 2 3"
1950 LOCATE 70,12:PRINT "R R"
1960 LOCATE 71,13:PRINT "2 3"
1970 LOCATE 62,14:PRINT "R = -----"
1980 LOCATE 63,15:PRINT "B R + R + R"
1990 LOCATE 68,16:PRINT "1 2 3"
2000 LOCATE 70,19:PRINT "R R"
2010 LOCATE 71,20:PRINT "1 3"
2020 LOCATE 62,21:PRINT "R = -----"
2030 LOCATE 63,22:PRINT "C R + R + R"
2040 LOCATE 68,23:PRINT "1 2 3"
2050 MOVE 20,210:DRAWR 40,40:DRAWR -4,4:DRAWR 20,20:DRAWR 9,-9:DR
AWR -20,-20:DRAWR -5,5:MOVE 80,270:DRAWR 40,40
2060 MOVE 120,310:DRAWR 40,-40:DRAWR 4,4:DRAWR 20,-20:DRAWR -9,-9
:DRAWR -20,20:DRAWR 5,5:MOVE 220,210:DRAWR -40,40
2070 MOVE 220,210:DRAWR -84,0:DRAWR 0,6:DRAWR -32,0:DRAWR 0,-13:D
RAWR 32,0:DRAWR 0,7:MOVE 20,210:DRAWR 84,0
2080 LOCATE 2,13:PRINT "A":LOCATE 29,13:PRINT "B":LOCATE 15,5:PRI
NT "C"
2090 LOCATE 7,7:PRINT "R":LOCATE 8,8:PRINT "1":LOCATE 22,7:PRINT
"R":LOCATE 23,8:PRINT "3":LOCATE 15,10:PRINT "R":LOCATE 16,11:PRI
NT "2"
2100 MOVE 340,210:DRAWR 10,10:DRAWR -3,3:DRAWR 20,20:DRAWR 7,-7:D
RAWR -20,-20:DRAWR -4,4
2110 MOVE 371,240:DRAWR 10,10:DRAWR 0,16:DRAWR -4,0:DRAWR 0,28:DR
AWR 9,0:DRAWR 0,-29:DRAWR -6,0:MOVE 381,294:DRAWR 0,16
2120 MOVE 381,250:DRAWR 10,-10:DRAWR -3,-3:DRAWR 20,-20:DRAWR 7,7
:DRAWR -20,20:DRAWR -4,-4:MOVE 412,220:DRAWR 10,-10

```

```

2130 LOCATE 45,7:PRINT "R":LOCATE 46,8:PRINT "C":LOCATE 42,10:PRI
NT "R":LOCATE 43,11:PRINT "A":LOCATE 52,10:PRINT "R":LOCATE 53,11
:PRINT "B"
2140 LOCATE 42,13:PRINT "A":LOCATE 54,13:PRINT "B":LOCATE 48,5:PR
INT "C"
2150 MOVE 250,260:DRAWR 30,0:MOVE 250,276:DRAWR 30,0:MOVE 292,268
:DRAWR -15,15:MOVE 292,268:DRAWR -15,-15
2160 LOCATE 2,15:PRINT " PDDAJ REZYSTANCJE "
2170 LOCATE 24,15:PRINT " REZYSTANCJE UKLADU W GWIAZDE : "
2180 LOCATE 28,17:PRINT "R =":LOCATE 29,18:PRINT "A"
2190 LOCATE 28,19:PRINT "R =":LOCATE 29,20:PRINT "B"
2200 LOCATE 28,21:PRINT "R =":LOCATE 29,22:PRINT "C"
2210 LOCATE 3,17:PRINT "R =":LOCATE 4,18:PRINT "1"
2220 LOCATE 8,17:INPUT r1
2230 IF r1<=0 OR r1>10000 THEN GOTO 2350
2240 LOCATE 3,19:PRINT "R =":LOCATE 4,20:PRINT "2"
2250 LOCATE 8,19:INPUT r2
2260 IF r2<=0 OR r2>10000 THEN GOTO 2370
2270 LOCATE 3,21:PRINT "R =":LOCATE 4,22:PRINT "3"
2280 LOCATE 8,21:INPUT r3
2290 IF r3<=0 OR r3>10000 THEN GOTO 2390
2300 ra=r1*r2/(r1+r2+r3)
2310 rb=r2*r3/(r1+r2+r3)
2320 rc=r1*r3/(r1+r2+r3)
2330 LOCATE 33,17:PRINT ra:LOCATE 33,19:PRINT rb:LOCATE 33,21:PRI
NT rc
2340 GOTO 570
2350 GOSUB 1690
2360 LOCATE 6,17:PRINT SPC(12):GOTO 2220
2370 GOSUB 1690
2380 LOCATE 6,19:PRINT SPC(12):GOTO 2250
2390 GOSUB 1690
2400 LOCATE 6,21:PRINT SPC(12):GOTO 2280

```



„Polin” znany jest na tere-
nie całego województwa
szczecińskiego. Tylko w
1986 roku zorganizował 37
różnorodnych pokazów i ak-
cji. Z jego inicjatywy powsta-
ły kluby mikrokomputerowe
w Stargardzie Szczecińskim,
Golczewie, Gryficach, a na-
wet w Politechnice Szczeciń-
skiej. Klub uczestniczył rów-
nież w Targach Młodych Mi-
strzów Jutra w Rostocku
(NRD), prezentując różno-
rodne wykorzystanie sprzętu
komputerowego. W codzien-
nej działalności dominuje
praca szkoleniowa. Systema-
tycznie organizowane są
kursy programowania oraz
pokazy i wykłady na temat
zastosowania mikrokompute-
rów.



W roli nauczyciela uczeń szkoły podstawowej — słuchacze to: Bożena Janowicz (z zawodu informatyk) oraz jej klubowi koledzy: Andrzej Kaczmarek i Zbigniew Ignaszak.

AMSTRAD

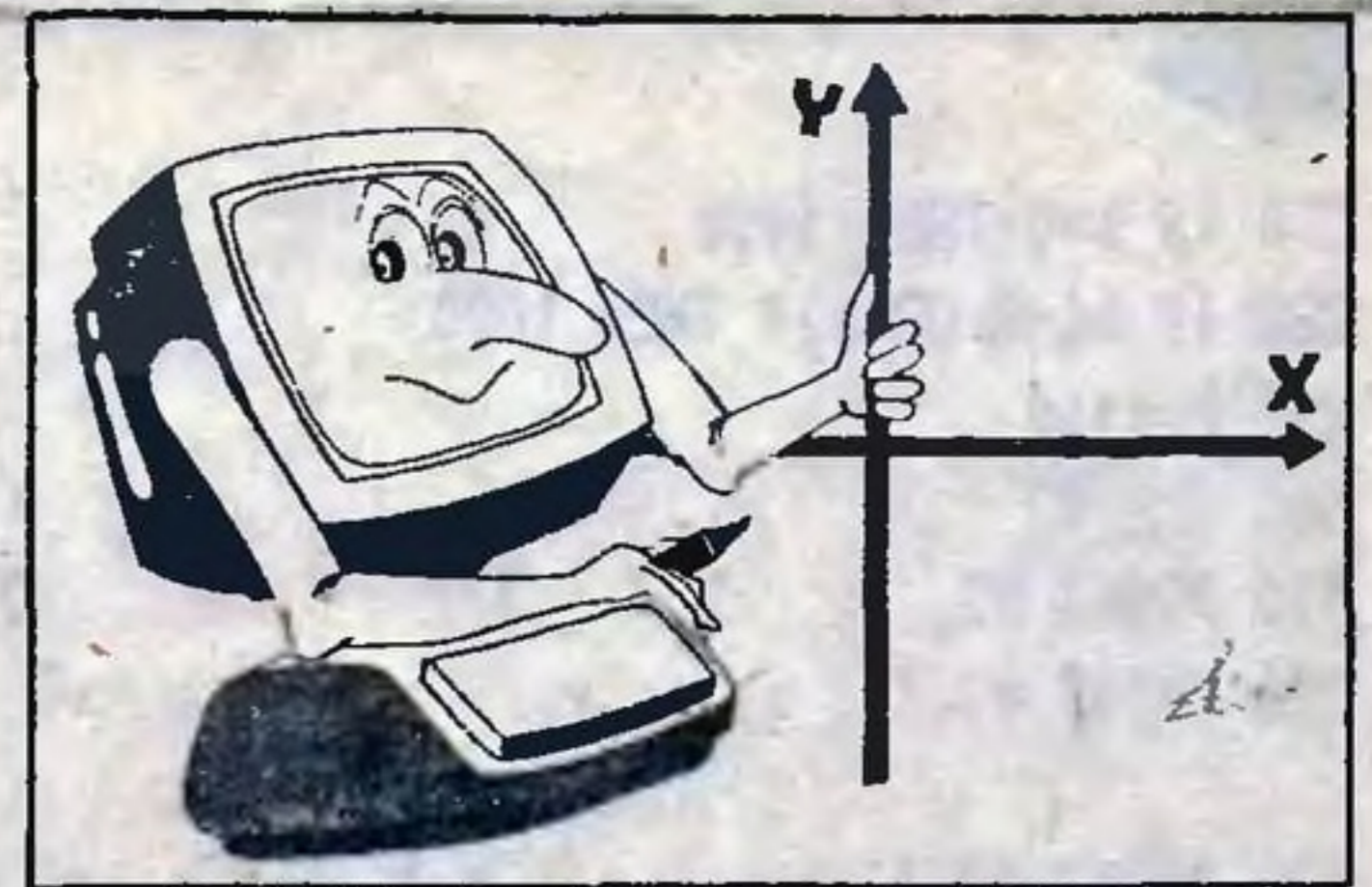
PROGRAM 13-14

HIPERBOLE RÓWNOOSIOWE

Powyższe programy przedstawiają jedną z krzywych stożkowych — hiperbolę w postaci hiperboli sprzężonych oraz hiperboli równoosiowych. Programy te stanowią kontynuację prezentowanych w pierwszym Dodatku Specjalnym „IKS-a” programów na krzywe stożkowe.

```
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
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=l+1
470 PRINT #3," ";l
480 IF l=2 THEN 550
490 IF l>=3 THEN 580
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
```

HIPERBOLE SPRZĘŻONE

```
10 CLS
20 GOSUB 860
30 CLS: BORDER 3
40 WINDOW #2,6,37,21,25
50 PAPER #2,13: PEN #2,0: CLS #2
60 WINDOW #3,35,40,1,1
70 PAPER #3,13: PEN #3,0: CLS #3
80 PRINT #3, " 0"
90 WINDOW #1,1,9,1,4
100 BORDER 9
110 PAPER #1,13: PEN #1,0: CLS #1
120 GOSUB 720
130 PLOT 100,240
140 FOR s=100 TO 600
150 PLOT s,240: NEXT: PLOT 598,242: PLOT 598,239: PL
      OT 596,244: PLOT 596,237
160 PLOT 350,400
170 FOR s=80 TO 390
180 PLOT 350,s: NEXT: PLOT 348,388: PLOT 346,386: PL
      OT 352,388: PLOT 354,386
190 LOCATE 38,12: PRINT "x"
200 LOCATE 21,1: PRINT "y"
210 l=0
220 INPUT #1,"a=";a
230 IF a=0 THEN 1020
240 IF a<-9 OR a>9 THEN 1080
250 a=a*10
260 INPUT #1,"b=";b
```

```

270 IF b=0 THEN 1050
280 IF b<-9 OR b>9 THEN 1090
290 b=b*10
300 CLS #2
310 PRINT #2, " 2 2 2 2 "
320 PRINT #2, " x y y x "

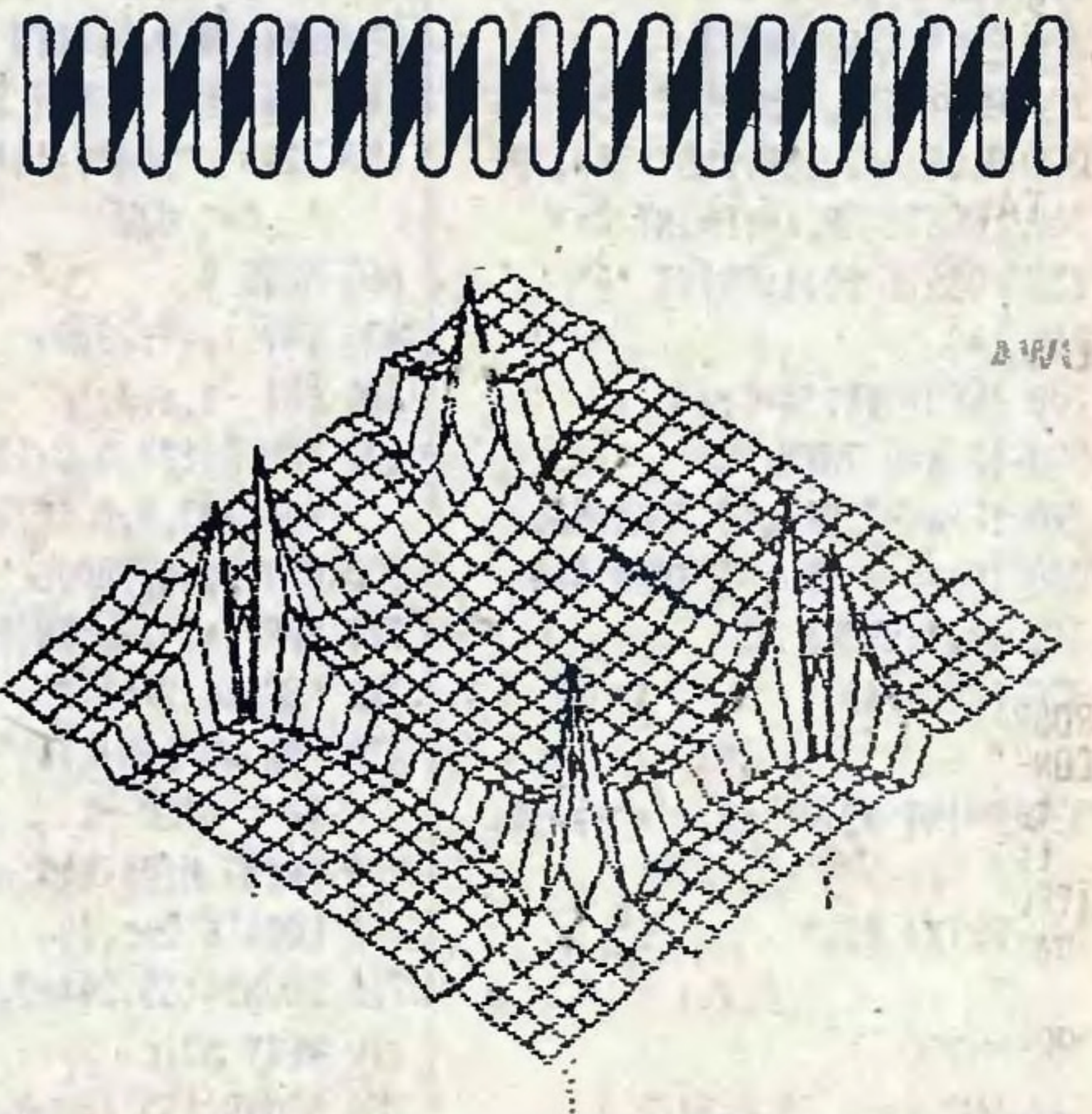
330 PRINT #2, " --- = 1 --- = 1 "
340 PRINT #2, " 2 2 2 2 "
350 PRINT #2, " ";a/10;" ";b/10;" ";b/10;"
" ";a/10;"
360 n=200
370 FOR x=0 TO -n STEP -1
380 k=(x-a)*(x+a):IF k<0 THEN 430
390 y=(b/a)*SQR(k):IF y>158 THEN 410
400 PLOT 350+x,240+y
410 y1=-(b/a)*SQR(k):IF y1<-158 THEN 440
420 PLOT 350+x,240+y1
430 NEXT
440 FOR z=0 TO n
450 k1=(z-a)*(z+a):IF k1<0 THEN 500
460 y2=(b/a)*SQR(k1):IF y2>158 THEN 480
470 PLOT 350+z,240+y2
480 y3=-(b/a)*SQR(k1):IF y3<-158 THEN 510
490 PLOT 350+z,240+y3
500 NEXT
510 FOR x=0 TO -n STEP -1
520 k=(x^2)+(a^2)
530 y=(b/a)*SQR(k):IF y>158 THEN 550
540 PLOT 350+x,240+y
550 y1=-(b/a)*SQR(k):IF y1<-158 THEN 580
560 PLOT 350+x,240+y1
570 NEXT
580 FOR z=0 TO n
590 k1=(z^2)+(a^2)
600 y2=(b/a)*SQR(k1):IF y2>158 THEN 620
610 PLOT 350+z,240+y2
620 y3=-(b/a)*SQR(k1):IF y3<-158 THEN 650
630 PLOT 350+z,240+y3
640 NEXT
650 CLS #1
660 CLS #3
670 l=1+1
680 PRINT #3, " ";l
690 IF l=1 THEN 780
700 IF l>=2 THEN 800
710 GOTO 220
720 PRINT #2, " 2 2 2 2 "
730 PRINT #2, " x y y x "
740 PRINT #2, " --- = 1 --- = 1 "
750 PRINT #2, " 2 2 2 2 "
760 PRINT #2, " a b b a "
770 RETURN
780 ENV 3,=9,9000
790 SOUND 1,300,0,13,3:GOTO 220

```

```

800 LOCATE 9,21
810 PRINT " koniec - [DOWOLNY KLAWISZ]"
820 ENV 3,=9,9000
830 SOUND 1,100,0,13,3
840 IF INKEY$="" THEN 840
850 ORIGIN 0,0:GOTO 30
860 t1$="H I P E R B O L E"
870 T2$=CHR$(32)+CHR$(164)+CHR$(32)+CHR$(32)+" C
JS "
880 MODE 0
890 ENV 1,=9,2000:ENT -1,6,3,1
900 ENV 2,127,0,0,127,0,0,127,0,0,127,0,0:ENV 3,
=9,9000
910 FOR c=1 TO LEN(t1$)
920 LOCATE 2+c,10
930 FOR n2=LEN(t1$) TO c STEP -1
940 PRINT MID$(t1$,n2,1)
950 LOCATE 2+c,10
960 SOUND 135,20*n2,5,12,2,1
970 NEXT n2,c
980 SOUND 135,100,0,13,3,1,20
990 LOCATE 5,15:PRINT t2$
1000 FOR c=1 TO 2000:NEXT
1010 CLS:MODE 1:RETURN
1020 LOCATE 11,21:PRINT "WPISZ a RÓZNE OD ZERA"
1030 FOR w=1 TO 2000:NEXT
1040 GOSUB 720:GOTO 220
1050 LOCATE 11,21:PRINT "WPISZ b RÓZNE OD ZERA"
1060 FOR w=1 TO 2000:NEXT
1070 GOSUB 720:GOTO 260
1080 LOCATE 11,21:PRINT " PODAJ a";CHR$(180);"
<-9,9> " :FOR w=1 TO 3000:NEXT:GOSUB 720:GOTO
220
1090 LOCATE 11,21:PRINT " PODAJ b";CHR$(180);"
<-9,9> " :FOR w=1 TO 3000:NEXT:GOSUB 720:GOTO
260

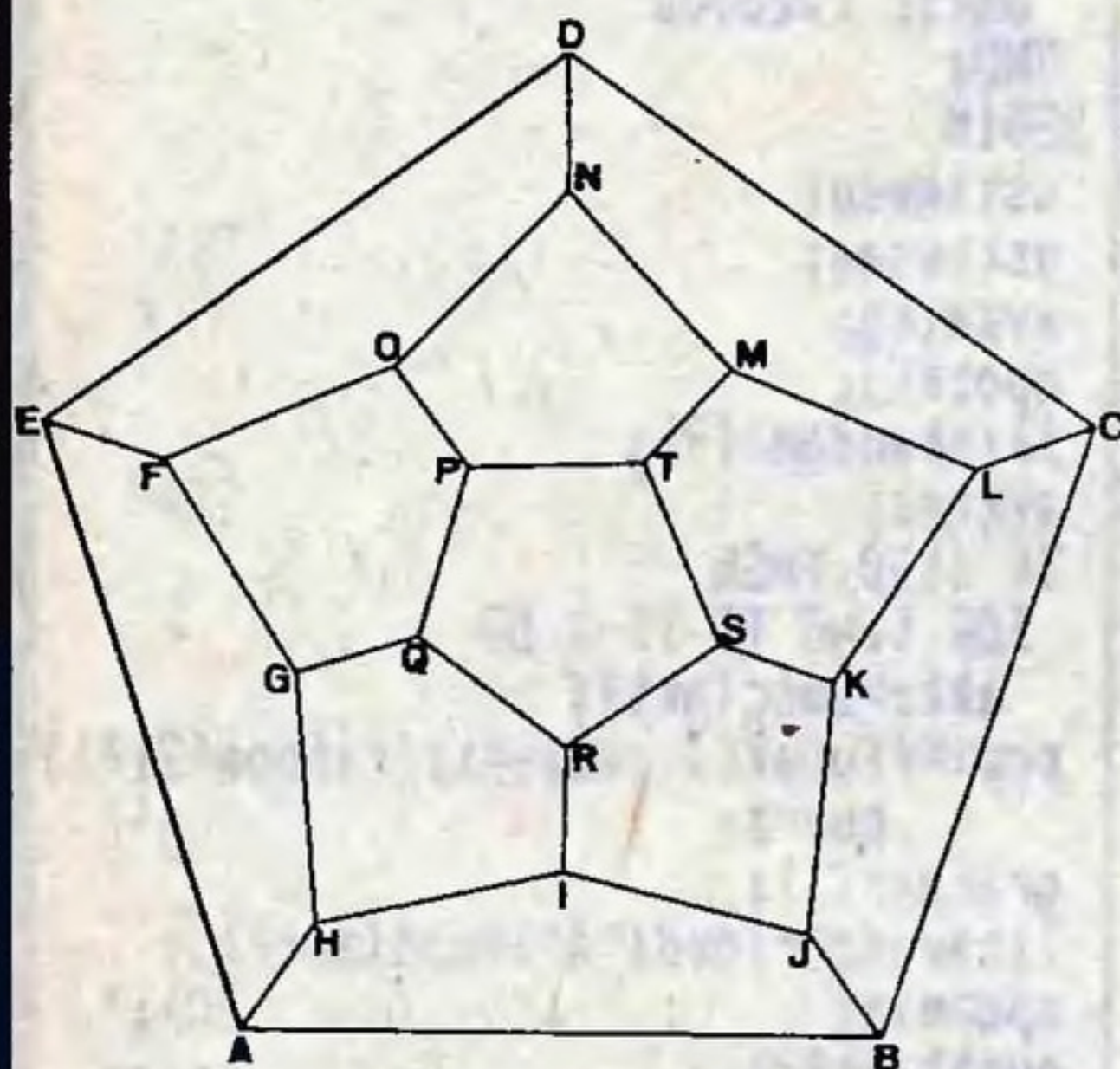
```



Gra Hamiltona

William Hamilton, matematyk i astronom angielski z pierwszej połowy XIX wieku, obmyślił ciekawą rozrywkę zwaną „podróżą po dodekaedrze” (dwunastościanie). W „podróży” tej chodzi o jednorazowe odwiedzić wszystkich naroży dwunastościanu z powrotem do naroża, z którego się wyszło.

Program umożliwia „podróż” lub demonstrować jej przebieg, przy czym dwunastościan został zastąpiony następującą figurą płaską:



Klawisze funkcyjne:

- 0 — wyjście z programu,
- 1 — powrót do „menu”,
- 2 — demonstracja gry,
- 3 — gra,
- A ÷ T — wybór odpowiedniego punktu figury,
- TAB — cofnięcie ruchu.

UWAGA: program kończy się, gdy następuje powrót do punktu początkowego; wygrywa ten, kto przejdzie przez wszystkie punkty zgodnie podanymi zasadami.

```
PROGRAM grahamiltona;
CONST
  tagflag:boolean = false;
  tagaddr:integer = 0;
TYPE
  nazwypunktow=(a,b,c,d,e,f,g,h,i,j,k,
                l,m,n,o,p,q,r,s,t);
  opispunktu=RECORD
    x,y:integer;
```

```
wskp:boolean;
sasiedzi:ARRAY[1..3] OF
RECORD
  poprz,
  lewy,
  prawy:nazwypunktow
END;
```

```
END;
dodekaedr=ARRAY[nazwypunktow] OF
  opispunktu;
```

```
lzn:string[40];
```

```
VAR
  fig:dodekaedr;
  opocz:nazwypunktow;
  ch:char;
  pozlit:ARRAY['A'..'T'] OF
  RECORD
    x,
    y:integer
  END;
```

```
wykgr:boolean;
```

```
(*****
($ procedury pomocnicze do manipulacji $)
($ zobrazowanien graficznych $)
($ *****)
```

```
PROCEDURE InitGrafik;
```

```
  BEGIN
    WRITE(##1b,'0',##1b,'y');
```

```
  END;
PROCEDURE leaveGrafik;
```

```
  BEGIN
    WRITE(##1b,'1',##1b,'x');
```

```
  END;
PROCEDURE draw(x,y:integer);
```

```
  BEGIN
    INLINE($2A/x/$EB/$2A/y/$cd/$5a/
    $fc/$Fb/$BB);
```

```
  END;
PROCEDURE plot(x,y:integer);
```

```
  BEGIN
    INLINE($2A/x/$EB/$2A/y/$cd/
    $5a/$fc/$EA/$BB);
```

```
  END;
PROCEDURE grafpen(fs:byte);
```

```
  BEGIN
    INLINE($3A/fs/$cd/$5a/$fc/$DE/$BB);
```

```
  END;
PROCEDURE grafpaper(fs:byte);
```

```
  BEGIN
    INLINE($3A/fs/$cd/$5a/$fc/$E4/$BB);
```

```
  END;
PROCEDURE clg(fs:byte);
```

```
  BEGIN
    INLINE($cd/$5a/$fc/$DB/$BB);
    grafpaper(fs);
```

```
  END;
PROCEDURE ink(fs,f1,f2:byte);
```

```
  BEGIN
    INLINE($3a/f1/$47/$3a/f2/$4f/
    $3a/fs/$cd/$5a/$fc/$32/$bc);
```

```
  END;
PROCEDURE paper(fs:byte);
  BEGIN
```



```
  INLINE($3A/fs/$cd/$5a/$fc/$9b/$BB);
  END;
PROCEDURE mode(md:byte);
  BEGIN
    INLINE($3a/md/$cd/$5a/$fc/$0e/$bc);
  END;
PROCEDURE border(v1,v2:byte);
  BEGIN
    INLINE($3a/v1/$47/$3a/v2/$4f/
    $cd/$5a/$fc/$38/$bc);
  END;
PROCEDURE grafout(ch:char);
  BEGIN
```

```
  INLINE($3a/ch/$cd/$5a/$fc/$FC/$BB);
  END;
```

```
PROCEDURE tag;
```

```
  BEGIN
    if not tagflag then
      BEGIN
        tagaddr := conoutptr;
        tagflag := true;
        conoutptr := addr(grafout);
      END;
```

```
  END;
PROCEDURE tagoff;
```

```
  BEGIN
    if tagflag then
      BEGIN
        conoutptr := tagaddr;
        tagflag := false;
      END;
```

```
  END;
PROCEDURE punkt(nazp:nazwypunktow);
```

```
  BEGIN
    plot(fig[nazp].x,fig[nazp].y)
```

```
  END;
PROCEDURE linia(nazp:nazwypunktow);
```

```
  BEGIN
    draw(fig[nazp].x,fig[nazp].y)
```

```
  END;
PROCEDURE litera(lit:char);
```

```
  BEGIN
    tag;
    plot(pozlit[lit].x,pozlit[lit].y);
    grafout(lit);
    tagoff
```

```
  END;
PROCEDURE napis(l:lzn;nrlin,
  nrkol:integer);
```

```
  VAR
    i,j,k:integer;
```

```
  BEGIN
    i:=nrkol*16-16;
```



```

j:=400-nrln#16;
tag;
FOR k:=1 TO length(l) DO
  BEGIN
    plot(i,j);
    grafout(l[k]);
    i:=i+16
  END;
tagoff
END;

(*****
(*****
(*      ustawienie wspolrzednych      *)
(*      *)
PROCEDURE ustawmsp;
PROCEDURE wspp(nazp:nazwypunktow;
                xp,yp:integer);
  BEGIN
    WITH fig[nazp] DO
      BEGIN x:=xp; y:=yp; wskp:=false END
    END;
  BEGIN
    wspp(a,200,50); wspp(b,400,50);
    wspp(c,464,240); wspp(d,300,356);
    wspp(e,136,240); wspp(f,176,226);
    wspp(g,216,162); wspp(h,222,84);
    wspp(i,300,100); wspp(j,378,84);
    wspp(k,384,162); wspp(l,424,226);
    wspp(m,354,258); wspp(n,300,314);
    wspp(o,246,258); wspp(p,270,228);
    wspp(q,254,174); wspp(r,300,140);
    wspp(s,346,174); wspp(t,330,228)
  END;

(*****
(*      opisy wierzchołkow sasiednich      *)
(*      *)
PROCEDURE ustawsas;
PROCEDURE sd(nazp,p1,p2,
             p3:nazwypunktow);
  BEGIN
    WITH fig[nazp] DO
      BEGIN
        WITH sasiedzi[1] DO
          BEGIN
            poprz:=p1; lewy:=p2; prawy:=p3
          END;
        WITH sasiedzi[2] DO
          BEGIN
            poprz:=p2; lewy:=p3; prawy:=p1
          END;
        WITH sasiedzi[3] DO
          BEGIN
            poprz:=p3; lewy:=p1; prawy:=p2
          END
        END
      END;
    END;
  BEGIN
    sd(a,e,h,b); sd(b,a,j,c); sd(c,b,l,d);
    sd(d,c,n,e); sd(e,d,f,a); sd(f,e,o,g);
    sd(g,f,q,h); sd(h,g,i,a); sd(i,h,r,j);
    sd(j,i,k,b); sd(k,j,s,l); sd(l,k,m,c);
    sd(m,l,t,n); sd(n,m,o,d); sd(o,n,p,f);
    sd(p,o,t,q); sd(q,p,r,g); sd(r,q,s,i);
    sd(s,r,t,k); sd(t,s,p,m)
  END;

(*****
(*      wypełnienie tablicy opisujacej      *)
(*      *)
(*      polozenie nazw wierzchołkow      *)
(*      *)

```

```

PROCEDURE nazwym;
PROCEDURE wp(l:char;x,y:integer);
  BEGIN
    pozlit[l].x:=x;
    pozlit[l].y:=y
  END;
  BEGIN
    wp('A',195,40); wp('B',395,40);
    wp('C',475,245); wp('D',295,376);
    wp('E',115,245); wp('F',174,246);
    wp('G',195,160); wp('H',225,78);
    wp('I',295,90); wp('J',363,78);
    wp('K',392,160); wp('L',420,246);
    wp('M',361,276); wp('N',278,329);
    wp('O',231,276); wp('P',250,230);
    wp('T',317,247); wp('Q',244,164);
    wp('S',350,188); wp('R',307,139)
  END;

(*****
(*      wyswietlenie figury      *)
(*      *)
PROCEDURE wyswfig;
  VAR
    i,j:nazwypunktow;
    l:char;
  BEGIN
    border(0,0); grafpaper(3);
    clg(3); grafpen(0);
    punkt(a);
    FOR i:=a TO d DO
      linia(succ(i));
    linia(a);
    punkt(f);
    FOR i:=g TO o DO
      linia(i);
    linia(f);
    linia(e);
    j:=h;
    FOR i:=a TO d DO
      BEGIN
        punkt(i);
        linia(j);
        j:=succ(succ(j))
      END;
    punkt(p);
    FOR i:=q TO t DO
      linia(i);
    linia(p);
    linia(o);
    j:=q;
    FOR i:=q TO t DO
      BEGIN
        punkt(i);
        linia(j);
        j:=succ(succ(j))
      END;
    nazwym;
    grafpen(2);
    FOR l:='A' TO 'T' DO
      litera(l)
    END;

(*****
(*****
(*      pokaz wyznaczenia drogi hamiltona      *)
(*      *)
PROCEDURE pokaz;
  VAR
    akt,pop,pom:nazwypunktow;
    il,i,l1,l2:integer;
  FUNCTION nre:integer;
    VAR zn:boolean;
        i:integer;
  BEGIN
    zn:=false;

```

```

i:=1;
REPEAT
  WITH fig[akt] DO
    IF sasiedzi[i].poprz=pop THEN
      BEGIN nre:=i; zn:=true END;
    i:=i+1
  UNTIL zn
END;
PROCEDURE wlewo;
  BEGIN
    pom:=akt;
    akt:=fig[akt].sasiedzi[nre].lewy;
    pop:=pom;
    linia(akt)
  END;
PROCEDURE wpravo;
  BEGIN
    pom:=akt;
    akt:=fig[akt].sasiedzi[nre].prawy;
    pop:=pom;
    linia(akt)
  END;
PROCEDURE opoznij;
  VAR i:integer;
  BEGIN
    i:=1;
    REPEAT
      i:=i+1
    UNTIL i>20000
  END;
  BEGIN
    ustawmsp;
    ustawsas;
    wyswfig;
    opoznij;
    il:=random(19);
    akt:=a;
    IF il>0 THEN
      FOR i:=0 TO il-1 DO
        akt:=succ(akt);
      pop:=fig[akt].sasiedzi[random(2)+1]
        .poprz;
      grafpen(1);
      litera(chr(ord('A')+ord(akt)));
      opoznij;
      punkt(akt);
      FOR l1:=1 TO 2 DO
        BEGIN
          FOR l2:=1 TO 3 DO
            BEGIN wlewo; opoznij END;
          FOR l2:=1 TO 3 DO
            BEGIN wpravo; opoznij END;
          wlewo; opoznij;
          wpravo; opoznij;
          wlewo; opoznij;
          wpravo; opoznij;
        END;
      napis('KONIEC POKAZU',24,1);
      wykgr:=false
    END;

(*****
(*****
(*      menu      *)
(*      *)
PROCEDURE menu;
  BEGIN
    border(0,0); grafpaper(3);
    clg(3); grafpen(2);
    napis('GRA HAMILTONA',2,5);
    napis('MENU:',4,5);
    napis('0 - WYJSCIE Z PROGRAMU',6,10);
    napis('1 - POWROT DO MENU',8,10);
    napis('2 - POKAZ',10,10);
    napis('3 - GRA',12,10);
    wykgr:=false
  END;

```

```

(*****)
(*****)
(      gra      )
(      )
PROCEDURE gra;
VAR
  koniec:boolean;
  ruchy:ARRAY[1..20] OF nazwypunktow;
  ppocz,pnast:nazwypunktow;
  nrakt:integer;
FUNCTION nazp(ch:char):nazwypunktow;
VAR
  i,j:integer;
  p:nazwypunktow;
BEGIN
  IF (ch)='A' AND (ch<='T') THEN
    i:=ord(ch)-ord('A')
  ELSE
    i:=ord(ch)-ord('a');
  p:=a;
  FOR j:=1 TO i DO
    p:=succ(p);
  nazp:=p
END;
PROCEDURE wygaskom;
VAR
  lpom:ln;
  i:integer;
BEGIN
  grafpen(3);
  FOR i:=1 TO 40 DO
    lpom[i]:=' ';
  napis(lpom,24,1);
  grafpen(1)
END;
FUNCTION rpopr(pnast:nazwypunktow):
  boolean;
BEGIN
  WITH fig[pnast] DO
    IF (NOT wskp) AND
      ((sasiadzi[1].poprz=ruchy[nrakt]) OR
      (sasiadzi[2].poprz=ruchy[nrakt]) OR
      (sasiadzi[3].poprz=ruchy[nrakt]))
    THEN rpopr:=true
    ELSE rpopr:=false
  END;
BEGIN
  ustawmsp;
  ustawsas;
  wyswfig;
  wykgr:=true;
  koniec:=false;
  grafpen(1);
  napis('PODAJ PUNKT POCZATKOWY',24,1);
  REPEAT
    REPEAT UNTIL keypressed;
    READ(kbd,ch);
  UNTIL ((ch)='A' AND (ch<='T')) OR
    ((ch)='a' AND (ch<='t'));
  ppocz:=nazp(ch);
  IF (ch)='a' AND (ch<='t') THEN
    ch:=chr(ord(ch)-32);
  litera(ch);
  nrakt:=1;
  ruchy[1]:=ppocz;
  wygaskom;
  napis('PODAJ NASTEPNY PUNKT',24,1);
  punkt(ppocz);
  WHILE NOT ((ch=#48) OR (ch=#49) OR
    (ch=#50) OR (ch=#51) OR koniec) DO
    BEGIN
      REPEAT UNTIL keypressed;
      READ(kbd,ch);
      IF ch=#9 THEN
        BEGIN
          IF nrakt>1 THEN
            BEGIN
              grafpen(0);
              linia(ruchy[nrakt-1]);
              fig[ruchy[nrakt]].wskp:=false;
              nrakt:=nrakt-1;

```

```

  wygaskom;
  napis('PODAJ NASTEPNY PUNKT',24,1);
  punkt(ruchy[nrakt]);
  END
END
ELSE
  IF ((ch)='A') AND (ch<='T') OR
    ((ch)='a') AND (ch<='t') THEN
    BEGIN
      pnast:=nazp(ch);
      IF rpopr(pnast) THEN
        BEGIN
          linia(pnast);
          IF pnast=ppocz THEN koniec:=true
        ELSE
          BEGIN
            nrakt:=nrakt+1;
            ruchy[nrakt]:=pnast;
            fig[pnast].wskp:=true;
            wygaskom;
            napis('PODAJ NASTEPNY PUNKT',
              24,1);
            punkt(ruchy[nrakt]);
          END
        END
      ELSE
        BEGIN
          wygaskom;
          napis(
            'ZLY RUCH-PODAJ NASTEPNY PUNKT'
            24,1);
          punkt(ruchy[nrakt])
        END
      END;
END;
END;

```

```

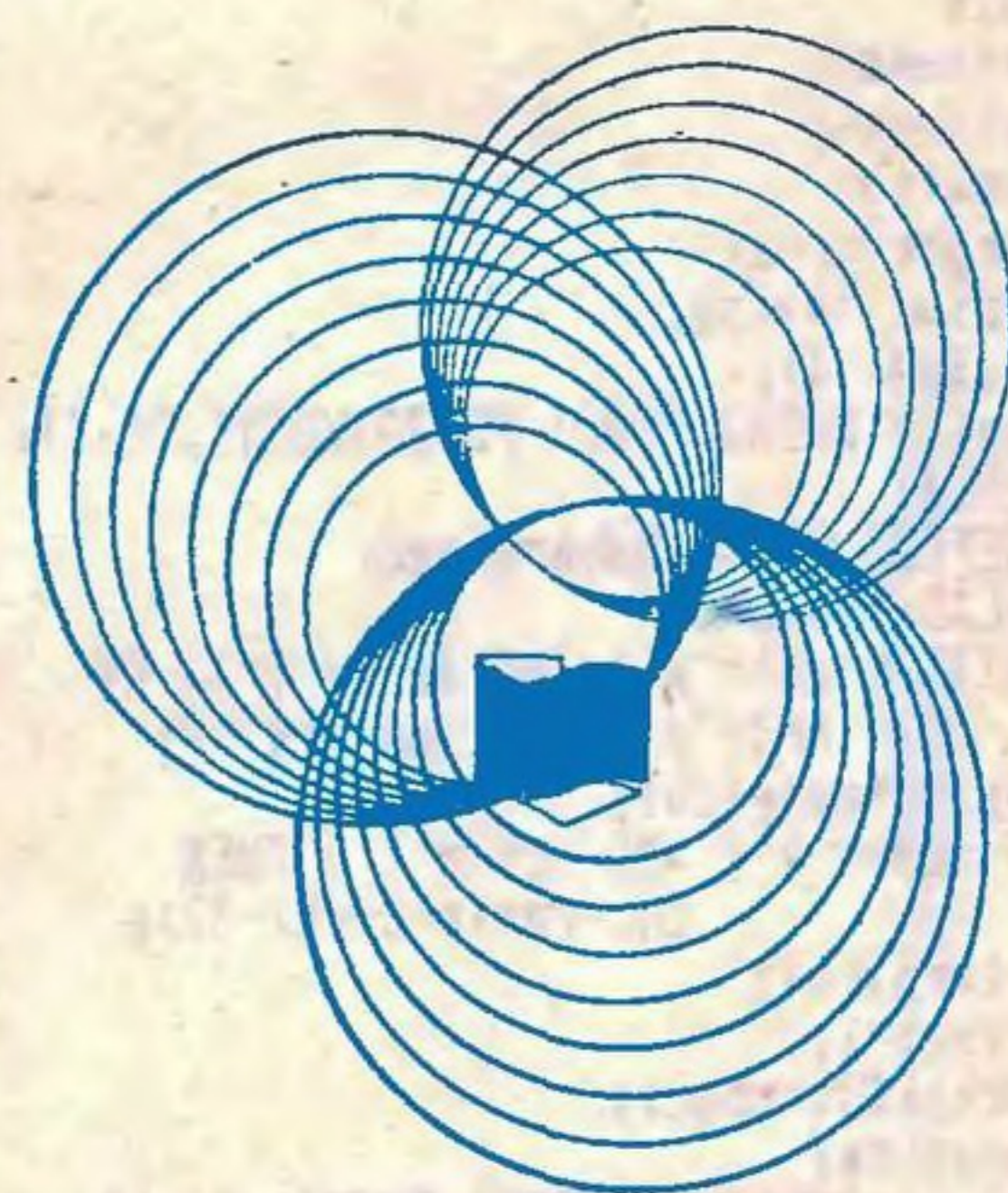
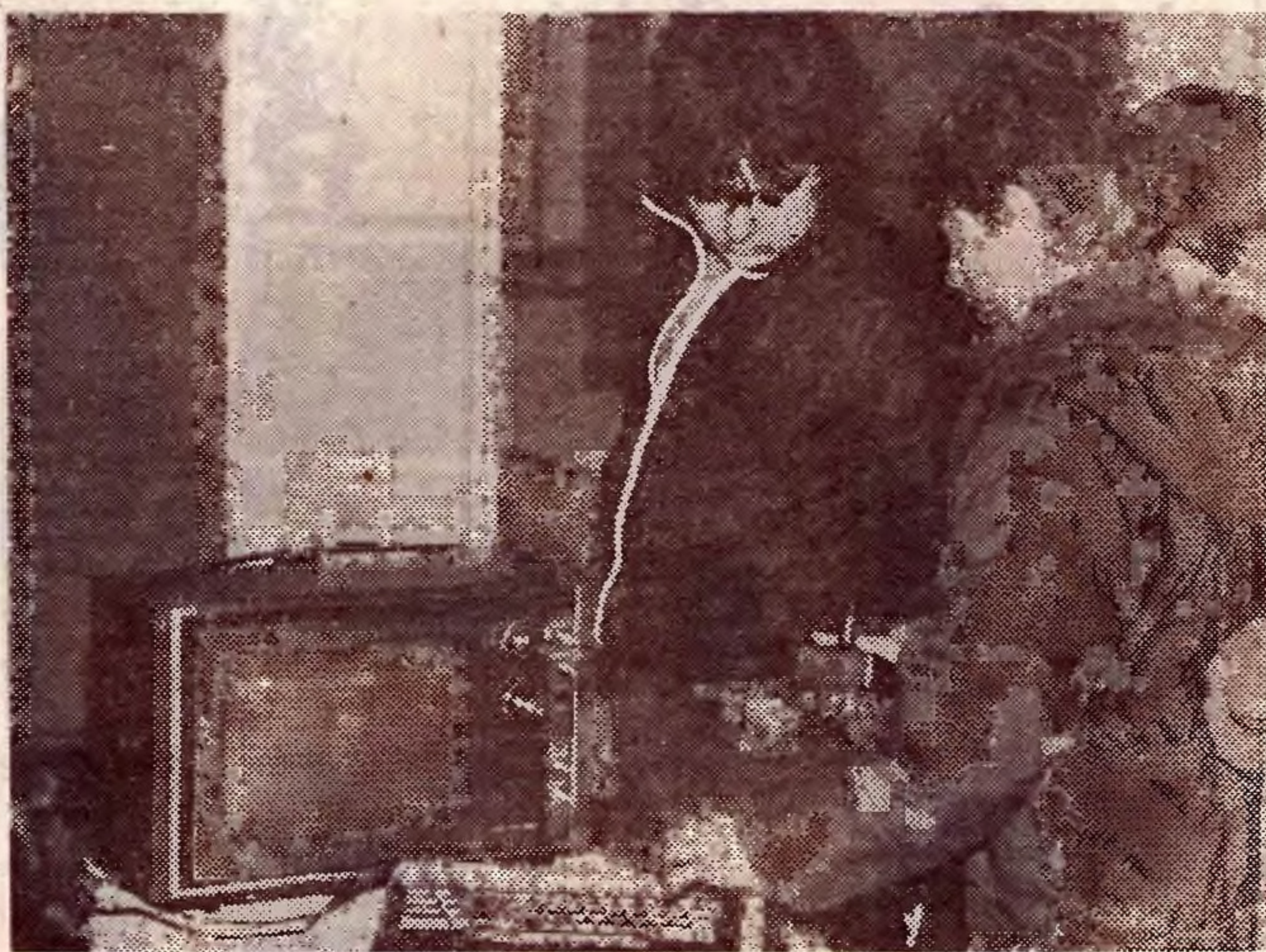
  IF koniec THEN
    BEGIN
      wygaskom;
      IF nrakt=20 THEN
        napis(
          'KONIEC GRY. WYGRALES-GRATULACJE',
          24,1)
      ELSE
        napis(
          'KONIEC GRY. NIESTETY PRZEGRALES',
          24,1);
      wykgr:=false
    END
  END;
(*****)
BEGIN
  crtinit;
  initgrafik;
  mode(1);
  ink(0,26,26);
  ink(1,15,15);
  ink(2,9,9);
  ink(3,0,0);
  menu;
  REPEAT
    IF NOT wykgr THEN
      BEGIN
        REPEAT UNTIL keypressed;
        READ(kbd,ch)
      END;
    CASE ch OF
      #49: menu;
      #50: pokaz;
      #51: gra
    END;
    UNTIL ch=#48;
    mode(2);
    leavegrafik;
  END.

```

Stefan Rozmus



— Widzę, że nie lubi pani komputera.



KLUBY KOMPUTEROWE
MŁODYCH MISTRZÓW TECHNIKI
ZSMP

„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stall współpracownicy: Włodzimierz Gogołek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szanławski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. 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. 8684. Nr. ind. 361682.