

I NFORMATYKA
K OMPUTERY
S YSTEMY



CENA — 50 zł

Dodatek „Żołnierza Wolności” nr 8/1986 ISSN 0860—2794



Za parę tygodni rozpoczniemy nowy rok bardziej dojrzały. Osiem wydanych numerów „IKS-a” zaświadcza, że wyszliśmy już z wieku niemowlęcego. Podsumowania jednak nie będzie, bowiem jaki jest nasz miesięcznik Czytelniczy wiedzą doskonale. Liczymy się z ich zdaniem, a pomysły, dobre oczywiście, starannie wykorzystujemy. Łamy „IKS-a” dla wszystkich, którzy mają coś ciekawego do powiedzenia, stoją otworem. Otrzymujemy wiele listów, szczególnie serdecznie staramy się przyjmować gości w naszej redakcji. Jesteśmy zapraszani na wiele mikrokomputerowych imprez.

Zainteresowanie mikrokomputerami nie słabnie. Sprzętu jest coraz więcej. Przybywa też wiadomości o jego możliwościach. Już dziś zdarza się, że kilkunastoletni młodzieńcy zawstydzają dojrzałych profesjonalistów. Nadal jednak jesteśmy na początku drogi. Informatyka porwała wielu, niestety dla znacznej części młodzieży stała się również przykrym obowiązkiem.

Niestety bywa i tak. Kiedy w Warszawie rozpoczynała się

wystawa Mikro Expo wokół Muzeum Techniki hulał mroźny, listopadowy wiatr. Nie przeszkadzało to jednak domorosłym informatykom w uformowaniu cierpliwie stojącej kolejki. Tak było w sobotę i niedzielę (8 i 9 listopada). Od wtorku na wystawę przychodziły szkoły. „I co my tu będziemy robić przez te pół godziny” pytali najbardziej znużeni. Prawda, co mieli robić, kiedy tylko nieliczni wiedzieli po co tu przyszli. W ten sposób już wkrótce okazać się może, że jesteśmy na dobrej drodze do zagłaskania informatyki, aż do obrzydzenia.

Jeszcze raz trzeba sobie zdać sprawę z tego, że mikrokomputer jest tylko narzędziem. Porównać go można do samochodu. Proszę bardzo: posiadanie prawa jazdy nie jest czymś nadzwyczajnym, trzeba znać zasady ruchu, umieć wykonać podstawowe czynności techniczne, ot chociażby zmienić koło — to wystarczy. Znakomita większość kierowców nie „grzebie” w silniku, poważniejsze naprawy pozostawiając specjalistom.

Z komputerem jest tak samo, dokładnie tak samo: trzeba poznać zasady dialogu z tym urzą-

dzeniem i umieć go uruchomić. A środek pozostawmy specjalistom. Pamiętajmy, że z komputerem trzeba się oswoić, bo większość użytkowników nie będzie w przyszłości pisać programów! Będzie je jedynie eksploatować. Tak, tak. Ucząc zatem podstaw nie wymagajmy gruntownej wiedzy technicznej. Przyszły użytkownik nie musi wiedzieć jak działa mikroprocesor, nie jest mu potrzebna wiedza o układach logicznych, tak jak kierowcy nie jest potrzebna znajomość na przykład ilości zębów w przekładni głównej — bo i po co!

Nie straszmy zatem informatyką, pokażmy jedynie jej możliwości. I zapamiętajmy, że tylko nieliczni będą genialnymi konstruktorami i wspaniałymi programistami — większość pójdzie tylko ich śladem.

Pokonywaniu tych właśnie barier, niechęci, niezrozumienia, niemalże przestrochu chcemy służyć i nadal liczymy na Waszą pomoc.

Wszystkim Czytelnikom Przyjaciółom i Sympatykom życzymy udanego startu w nowy rok.

Redakcja

Firma PHILIPS SUBSYSTEMS and PERIPHERALS z Nowego Jorku pracuje nad nowym typem optycznego dysku typu „pisz raz, czytaj wiele razy” (write once read many) o pojemności 500 Megabajtów. Będzie on kompatybilny z CD ROM (compact disk read only memory). Philips nazwał te nowe dyski CD-PROM (compact disk programmable read only memory).

Holenderska firma DOC-data planuje wykonać optyczny mechanizm taśmowy, który będzie rejestrował 1 Terabajt (1000 Gigabajtów danych). Urządzenie to nazwano DOCwheel, wykorzystuje ono 128 optycznych kaset (optical cassettes) o rozmiarach standardowych audio-kaset (4,4x2,2x0,6 cala). Każda kaseeta ma pojemność 8 Gigabajtów. To nadzwyczajne urządzenie będzie dostępne w 1987 roku.

Natomiast już dzisiaj można kupić BACKUP Tape System firmy IRWIN. Jest to pamięć taśmowa o pojemności 40 Megabajtów. Informacje rejestrowane są na kasecie typowej wielkości. Można ją schować do kieszeni marynarki. Dzięki autokorekcji błędów gwarantowana jest poprawność odczytu bez konieczności weryfikacji zapisanych

na kasecie informacji. Proponowane (oczywiste) zastosowanie — składowanie twardego dysku: 10, 20 i 40 Megabajtów.

KURZWELL zaprezentował foniczny terminal, który umożliwia wykorzystanie wypowiedzianych słów i fraz do sterowania, wprowadzania i obierania danych z mikrokomputerów. System ten automatycznie „przekłada” słowa komend i dane na postać wymaganą

NOWINKI

przez wejście w komputerze stosowane dla typowej klawiatury. Innymi słowy można wykorzystać klawiaturę lub mikrofon, ekran lub głośnik.

Firma MITSUBISHI proponuje tani (395 dolarów) digitizer o aktywnej powierzchni tablicy 21x30 cm. Digitizer umożliwia operowanie z szybkością 60 punktów na sekundę i rozdzielczością 0,1 mm. Tablica współpracuje z AutoCAD, P-CAD, PC-Paint itp. (Digitizer jest urządzeniem wejściowym mikro, które, podobnie jak myszka, zmienia położenie znacznika na ekranie moni-

tora. Miejsce stołu zajmuje wspomniana tablica, myszki — „pióro”).

HEWLETT-PACKARD wyprodukował maszynę kompatybilną z IBM PC/AT, która może pracować w języku angielskim lub jednym z czterech azjatyckich: japońskim, koreańskim, chińskim lub uproszczonym chińskim. Maszyny te pracują w oparciu o procesor 80286. Dołączona jest bogata oferta uzupełniająca (RAM, floppy 1,2 Mb, oprogramowanie). Wybierając wersję japońską (3.1) MS-DOS staje się systemem operacyjnym w języku japońskim. W ROM-ie zapisano około 7000 japońskich znaków. Znaki Kanji są wyświetlane na ekranie jako matryce 24x24 pixele.

Pojemność typowego optycznego dysku wynosi 1 Gigabajt. Jest to ekwiwalentem około 400 000 stron A4, 500 do 1000 obrazów czarno-białych, 2000 mikrofisz, 8000 m taśmy magnetycznej. FUJITSU twierdzi, że to dopiero początek, bowiem propozycja tej firmy to 12-calowy dysk 3 Gigabajty z możliwością zapisu i odczytu informacji. Póki co można kupić CM100 Philipsa 0,6 Giga (115x320x267 mm — 5 kg).

W.G.

POLSKIE ZNAKI NA ZX SPECTRUM

Wszystkie programy, które powstają z myślą o polskim użytkowniku powinny być wyposażone w cały zestaw liter polskiego alfabetu. Dotyczy to szczególnie programów dydaktycznych, programów przetwarzających teksty oraz programów, w których występuje lista nazwisk lub dowolnych innych nazw. Stosowanie liter polskiego alfabetu w programach podyktowane jest nie tylko wymaganiami dydaktycznymi czy formalnymi. To właśnie język przez długie lata był ostoją naszej narodowości.

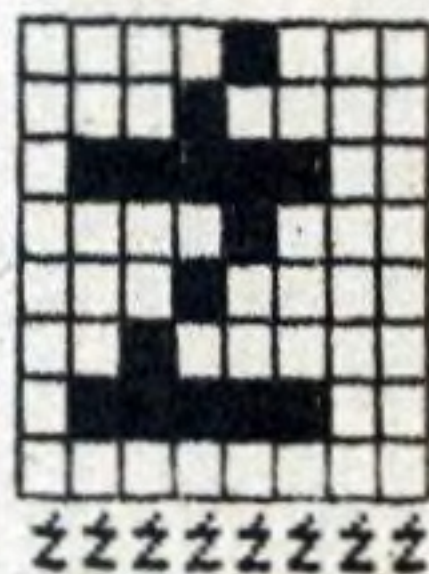
Żaden z zachodnich mikrokomputerów nie jest wyposażony we wszystkie litery polskiego alfabetu. Nie występują bowiem polskie litery diakrytyzowane, takie jak Ą, ą, Ć, ć, Ę, ę itd.

W mikrokomputerze ZX Spectrum można stosunkowo łatwo zdefiniować dowolny znak, w szczególności więc również i polskie litery diakrytyzowane. Łącznie zdefiniować można 21 znaków, które dostępne są w trybie graficznym na klawiszach A—U. Definiując znak, trzeba rozstrzygnąć dwie kwestie:

- kształt znaku
- przyporządkowanie znaku do jednego z klawiszy A—U.

Wszystkie znaki definiowane są w macierzy 8X8 pól (pikseli). Każdy rząd składający się z ośmiu pikseli może być zapisany w postaci ośmiobitowej liczby binarnej, gdzie 1 oznacza pole zaczerńnione (przy założeniu pisma pozytywnego, tj. trybu INVERSE) a 0 pole nie zaczerńnione. Liczby te wpisywane są następnie za pomocą instrukcji POKE w tzw. obszar UDG (user defined graphics) pod adresem wybranego klawisza A—U. Łatwo zauważyć, że jeden tak zdefiniowany znak zajmuje 8 bajtów pamięci, podczas gdy wszystkie znaki i słowa kluczowe dostępne wprost z klawiatury zajmują 1 bajt.

Poniżej przedstawiona została macierz znaku „ż”:



```

10 REM Definiowanie znaku
20 POKE USR "g",BIN 00001000
30 POKE USR "g"+1,BIN 00010000
40 POKE USR "g"+2,BIN 01111100
50 POKE USR "g"+3,BIN 00001000
60 POKE USR "g"+4,BIN 00010000
70 POKE USR "g"+5,BIN 00100000
80 POKE USR "g"+6,BIN 01111100
90 POKE USR "g"+7,BIN 0

```

Aby uniknąć powtarzania instrukcji POKE, można użyć pętli FOR, należy jednak przeliczyć liczby binarne na dziesiętne. Stosowny fragment programu definiujący znak „ż” wyglądałby wówczas następująco:

```

10 REM Definiowanie znaku
20 LET k#="g"
30 FOR i=0 TO 7
40 READ a
50 POKE USR k#+i,a
60 NEXT i
70 DATA 8,16,124,8,16,32,124,0

```

Tak zdefiniowany „ż” dostępny jest w trybie graficznym po naciśnięciu klawisza „g”.

W związku z dużym upowszechnieniem mikrokomputera ZX Spectrum celowe byłoby ujednoczenie przyporządkowania poszczególnych liter diakrytyzowanych do klawiszy A—U. O ile dla małych liter ą, ć, ę itd. panuje zgodność — gdyż przyporządkowuje się je do klawiszy odpowiednio A, C, E — o tyle dla dużych liter zgodności takiej nie ma. Ponadto pewien chaos wprowadzają znaki „ż”, „Ż”, „ź”, „Ź”, które nie mieszczą się w przedziale A—U. Najlepszym kryterium dla przyporządkowania znaków do klawiszy jest sąsiedztwo w alfabecie (porządek alfabetyczny) oraz — dla znaków „ż” i „ź” — skojarzenie z wymową francuską.

Należy podkreślić, że w ASCII kody znaków zdefiniowanych przez UDG są większe od kodów dużych i małych liter alfabetu angielskiego, co utrudnia leksykograficzne porządkowanie słów. Problem można rozwiązać stosując umiejętnie kodowanie, np. nazwisko „Ślusarczyk” powinno być zapisane w komputerze jako „S1lusarczyk”. Przy wprowadzaniu danych trzeba analizować pierwszy znak i — jeśli jest to znak z zakresu UDG — przeprowadza się kodowanie. Przy wyprowadzaniu danych (na ekran lub drukarkę) trzeba analizować drugi znak nazwy i — jeśli jest to znak użyty do kodowania (!), dwa pierwsze znaki nazwy zastępuje się odpowiednim znakiem z zakresu UDG. Kodowanie i dekodowanie powinno być realizowane przez odrębne podprogramy. Znak użyty do kodowania powinien być tak dobrany, aby jego kod był zawarty pomiędzy kodami dużych i małych liter. Jest to niezbędne dla zapewnienia prawidłowego sortowania. Warunek ten spełnia np. znak 1.

Alfabet polski zawiera 18 liter diakrytyzowanych (małych i dużych razem). Ponieważ obszar UDG pozwala na zdefiniowanie 21 znaków, pozostałe 3 znaki można wykorzystać do celów specjalnych (np. bardzo przydatna jest kreska do podkreśleń tekstu).

Znaki raz zdefiniowane zgodnie z przedstawionym opisem należy zapisać na taśmie za pomocą instrukcji

```
SAVE "Znaki-pol" CODE 65368,168
```

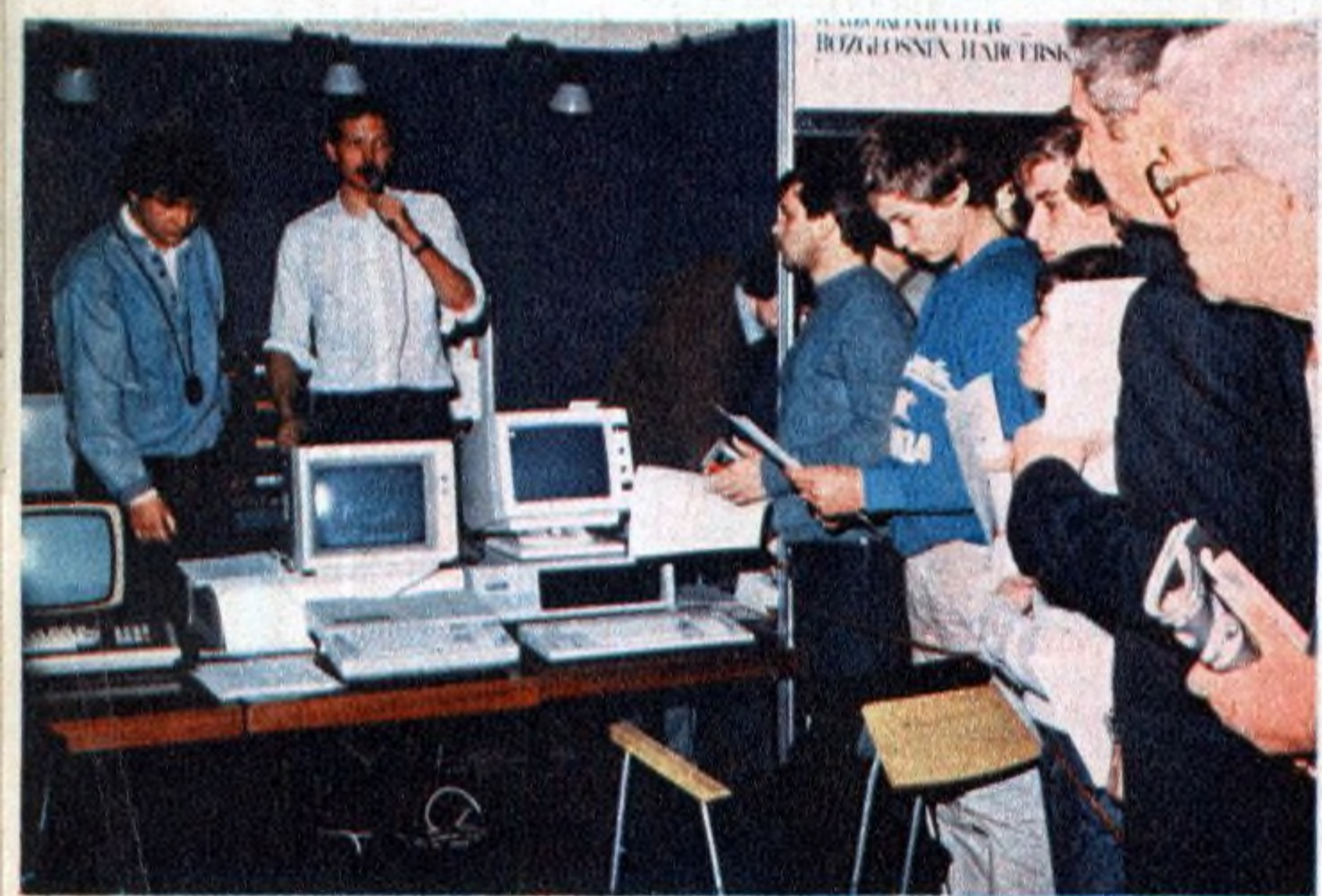
i używać w każdym programie, ładując je instrukcją

```
LOAD "Znaki-pol" CODE
```

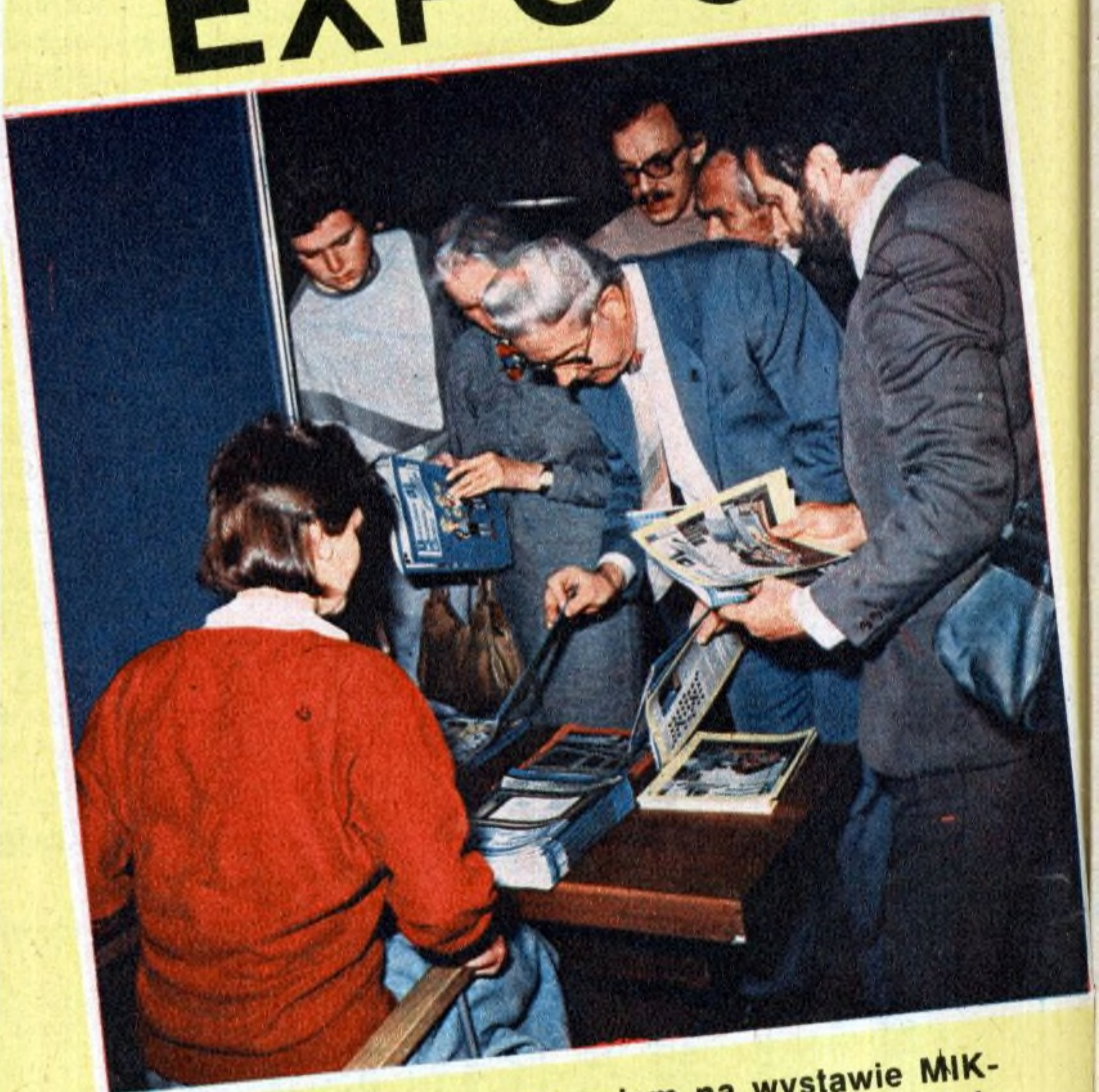
lub

```
LOAD "" CODE
```

dokończenie na str. 6



MIKRO EXPO'86



Niebywałym powodzeniem na wystawie MIKRO EXPO'86 cieszył się „IKS” (a szczególnie wcześniejsze jego numery). Było również na co popatrzeć...

Foto: Jan Zelman

PROGRAM

„ZGADYWANKA” (ATARI) jest komputerową wersją popularnej gry „Master mind”. Jest trudniejsza od pierwowzoru, polega bowiem na odgadnięciu liczby z przedziału od 0000 do 9999. Może w niej brać udział do 10 graczy, można zatem urządzać turniej logicznego myślenia.

Istnieją dwa poziomy gry; łatwy, dla początkujących, w którym każda cyfra liczby jest inna oraz trudny, dla zaawansowanych, w którym mogą wystąpić powtarzające się cyfry.

UWAGA: W linii 110 należy wpisać w cudzysłowach po B \$ i C \$ następujące znaki graficzne, które nie zostały wydrukowane, odpowiednio CTRL P oraz CTRL T. Te same znaki należy wpisać w nawiasy w linii 1450, w takiej samej kolejności.

```

90 REM << ZGADYWANKA >>
100 DIM T(10),C(10),Z(2,4),K(4)
105 DIM O$(1),B$(2),C$(2),S(4)
110 B$=" ";C$=" "
115 OPEN #1,4,0,"K:"
117 FOR I=1 TO 10:T(I)=0:NEXT I
120 GOSUB 1000:REM STRONA TYTULOWA
400 FOR P=1 TO 0
405 POKE 752,0
410 GRAPHICS 0:?" UWAGA GRACZ ";P;" WYBIERAM LICZBE.":? :? :?
420 FOR I=1 TO 6:C(I)=0:NEXT I
430 N=0:FOR I=1 TO 4
440 N1=INT(10*RND(0)):IF A9=1 THEN 470
450 IF C(N1+1)=1 THEN 440
460 C(N1+1)=1
470 N=N+(N1*(10^(I-1))):NEXT I
480 N1=N:T5=0
490 T1=INT(N/1000):N=N-(T1*1000)
500 H1=INT(N/100):N=N-(H1*100)
510 E1=INT(N/10):N=N-(E1*10)
520 Z(1,1)=T1:Z(1,2)=H1:Z(1,3)=E1:Z(1,4)=N
530 IF T5<>20 THEN 550
540 ? :? "ZBYT WIELE PROB, SZUKANA LICZBA: ";N1:GOTO 720
550 POSITION 1,22:?"PODAJ SWOJA LICZBE: "
560 FOR I=1 TO 4:S(I)=0:K(I)=0:NEXT I:X1=0:T5=T5+1
570 FOR KL=5 TO 11 STEP 2
580 GOSUB 950
590 IF KL=5 THEN T=C:POSITION K L,T5:?" T:Z(2,1)=T
594 IF KL=7 THEN H=C:POSITION K L,T5:?" H:Z(2,2)=H
598 IF KL=9 THEN E=C:POSITION K L,T5:?" E:Z(2,3)=E
600 IF KL=11 THEN O=C:POSITION KL,T5:?" O:Z(2,4)=O
605 NEXT KL
610 G=T*1000+H*100+E*10+O
615 POSITION 1,22:?"
620 POSITION 16,T5:POKE 752,1
622 FOR I=4 TO 1 STEP -1

```

```

624 IF Z(2,I)=Z(1,I) THEN ? B$;:K(I)=2
626 NEXT I
628 IF G=N1 THEN 710
630 X=INT(4*RND(0))+1:IF S(X)=1 THEN 630
640 S(X)=1:FOR L=1 TO 4:IF Z(2,X)<>Z(1,L) THEN 700
645 IF K(L)=2 THEN 700
650 IF K(L)=1 THEN 700
660 K(L)=1:IF X<>L THEN 680
670 POKE 752,1:?" B$;:GOTO 690
680 POKE 752,1:?" C$;
690 L=5
700 NEXT L:X1=X1+1:IF X1=4 THEN 530
705 GOTO 630
710 ? :? :? :? :? " BARDZO DOBRZE, ODGADLES LICZBE W ";T5;" RUCHACH."
715 FOR I=1 TO 600:NEXT I
720 ? :? :T(P)=T(P)+T5:NEXT P:R=R+1
730 ? :? " PO ZAKONCZENIU RUNDY ";R:?"
740 ? "GRACZ","RAZEM","SREDNIA","STOPIEN"
750 ? :FOR I=1 TO 0
760 ? " ";I;" ";T(I),:T9=INT((T(I)/R)*100)/100:?" ";T9,
770 GOSUB 830
780 NEXT I
790 ? :? :?
800 ? "NASTEPNA RUNDA? (T/N) ";:INPUT O$
810 IF O$="T" THEN 400
815 IF O$<>"T" AND O$<>"N" THEN 800
820 STOP
830 T9=T9+((A9-1)*4)
840 IF T9<=4 THEN ? "EKSPERT":RETURN
850 IF T9<=8 THEN ? "WYJATKOWO":RETURN
860 IF T9<=12 THEN ? "DOBRZE":RETURN
870 IF T9<=15 THEN ? "SREDNIO":RETURN
880 IF T9<=18 THEN ? "SLABO":RETURN
890 IF T9<=20 THEN ? "BARDZO SLABO":RETURN
900 ? "WYMAGA":?" ","","","TRENINGU":RETURN
950 REM PPROGRAM GET
960 GET #1,C:C=C-48
965 IF C=78 THEN KL=KL-2:POSITION ON KL,T5:?"
970 IF C<0 OR C>9 THEN 960
980 RETURN
1000 REM STRONA TYTULOWA
1010 DL=PEEK(560)+256*PEEK(561)
1020 GRAPHICS 0:SETCOLOR 2,0,0:POKE 580,1:POKE 752,2
1030 SETCOLOR 1,0,15:SETCOLOR 0,2,8:SETCOLOR 3,11,7:POKE 82,0
1040 POKE DL+9,7:POKE DL+13,7
1050 POSITION 5,4:?"zgadywanka"
1060 POSITION 3,15:?"OPTION" - POZIOM GRY:"
1070 POSITION 3,17:?"SELECT" - ILOSC GRACZY:"
1080 POSITION 3,19:?"START" - POCZATEK GRY"
1090 POSITION 3,21:?"HELP" -

```

dokonczenie na str. 6

```

INSTRUKCJA"
1100 A9=1:GOSUB 1200
1110 Q=0:GOSUB 1300
1120 W=PEEK(53279)
1130 IF W=3 THEN GOSUB 1200
1140 IF W=5 THEN GOSUB 1300
1150 IF W=6 THEN RETURN
1160 IF PEEK(732)=17 THEN GOSUB
1400
1170 GOTO 1120
1200 REM POZIOM GRY
1210 A9= NOT A9
1215 FOR I=1 TO 50:NEXT I
1220 IF A9=0 THEN 1240
1230 POSITION 25,15:? "LATWY /
#öñãkP":RETURN
1240 POSITION 25,15:? "4ã#öP /
TRUDNY":RETURN
1300 REM ILOSC GRACZY
1305 FOR I=1 TO 50:NEXT I
1310 IF Q=10 THEN Q=0
1320 Q=Q+1:POSITION 30,17:? "
"
1330 POSITION 30,17:? STR$(Q)
1340 RETURN
1400 REM INSTRUKCJA
1410 POKE 732,0:GRAPHICS 0:POKE
82,2:POKE 83,39
1420 POSITION 15,1:? "ZASADY GR
Y":? " "
1430 ? " WYBIERAM LICZBE MIED
ZY 0000-9999. SPROBUJ ODGADNAC
TE LICZBE W MINIMAL- NEJ ILOSC
I PROB. ZA KAZDYM RAZ"
1440 ? "NEJ ILOSCI PROB. ZA KAZ
DYM RAZEM DAM CI WSKAZOWKI. JAK
BLISKI JESTES PO-"
1450 ? "PRAWNEJ ODPOWIEDZI. WSK
AZOWKAMI BEDA <CZARNE>() I/LUB
<BIALE>(). JESLI"
1460 ? "POWIEM <BIALE> TO ZNACZ
Y, ZE JEDNA Z CYFR JEST POPRAW
NA, LECZ NIE JEST NA"
1470 ? "SWOIM MIEJSCU. JESLI PO
WIEM <CZARNE>, WTEDY MASZ POPRA
WNA CYFRE NA SWOIM"
1480 ? "MIEJSCU. POZWALA CI TO
WNIOSKOWAC JAKIE CYFRY SA P
OPRAWNE."
1490 ? " " :? " " :? "Jesli skonc
zyles czytac, nacisnij dowol
ny klawisz." :? " " :POKE 764,25
5
1495 IF PEEK(764)=255 THEN 1495
1500 ? " " :? " " :? " " :? " " :?
" " :? " "
1510 ? " ISTNIEJA DWA POZIOMY
TRUDNOSCI GRY. WERSJA LATWA - D
LA POCZATKUJACYCH."
1520 ? "NIE BEDZIE MIEC TYCH SA
MYCH CYFR UZY- TYCH DWA RAZY W
LICZBIE. WERSJA DLA"
1530 ? "ZAAWANSOWANYCH MOZE MIE
C NAWET CZTERY TAKIE SAME CYFRY
"
1540 ? " MOZE GRAC OD 1 DO 10
GRACZY."
1550 ? " " :? " " :? " " :? "*****
***** POWODZENIA! *****"
1560 ? " " :? " " :? " " :? " " :? " Je
sli skonczyles czytac, nacisnij
dowolny klawisz."
1570 POKE 764,255
1580 IF PEEK(764)=255 THEN 1580
1590 POP :GOTO 1000

```

Jeżeli powyższe instrukcje ładowania będą umieszczone w programie jako pierwsze instrukcje wykonywalne, a znaki zapisane zostaną na taśmie bezpośrednio z programem, możliwe jest automatyczne ładowanie polskiego alfabetu.

Co zrobić gdy w programie potrzebne są dwa lub więcej alfabetów (np. słowniki)? Można zwiększyć obszar UDG korzystając ze zmiennej systemowej o tej samej nazwie (adresy 23676 i 23676) lub wykorzystać części wspólne znaków. Jeżeli np. zdefiniowaliśmy znak „ą”, to chcąc uzyskać znak niemieckiego alfabetu „ä”, wystarczy zlikwidować ogonek u dołu znaku i dopisać kropki u góry. Częścią wspólną, którą można wykorzystać dla obu liter jest tu znak „a”. Zamiana znaku „ą” na „ä” odbywa się dzięki

```

POKE USR "a" ,BIN 00101000
POKE USR "a"+7,BIN 0

```

lub w innym zapisie

```

POKE USR "a" ,40
POKE USR "a"+7,0

```

Przejsie odwrotne od „ä” do „ą” polegające na usunięciu kropek u góry znaku i dopisaniu ogonka u dołu wykonują instrukcje

```

POKE USR "a" +7,BIN 0
POKE USR "a" +7,BIN 00000010

```

lub

```

POKE USR "a" ,0
POKE USR "a" +7,2

```

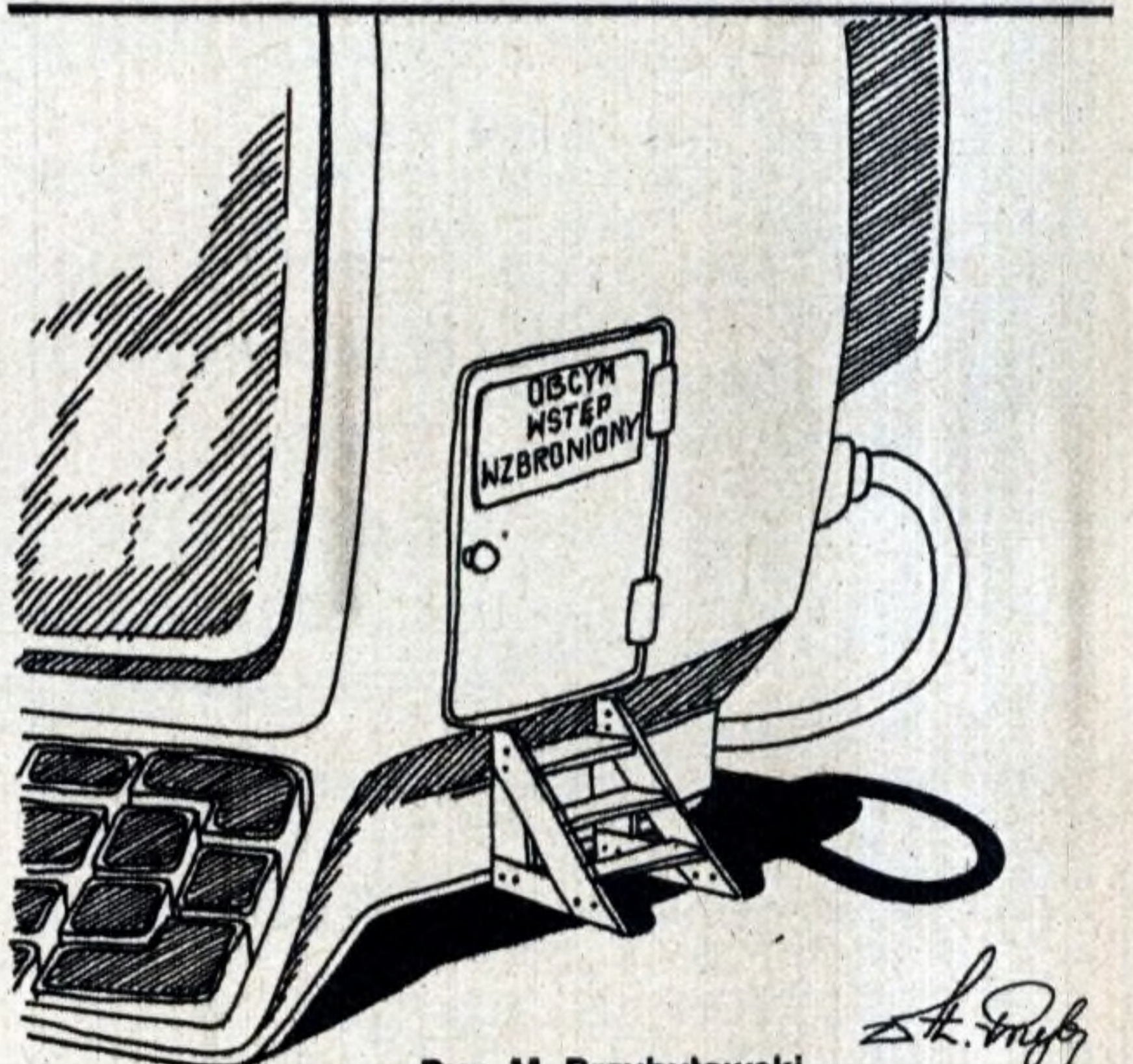
Oczywiście obydwa znaki „ą” i „a” dostępne są w trybie graficznym na klawiszu „a”, przy czym w danej chwili dostępny jest tylko jeden z nich. Powyższy sposób „przeładowania” alfabetów jest bardzo szybki i prosty. Przeładowanie całego alfabetu wymaga zaledwie kilku instrukcji POKE i może być zrealizowane przez podprogram.

Kończąc pragnę zachęcić Czytelników do korzystania w swoich programach ze wszystkich liter polskiego alfabetu. Pomni nawoływania Mikołaja Reja

*„niechaj narodowie wždy postronni znają
iż Polacy nie gęsi, iż swój język mają”*

dbajmy o poprawną polszczyznę we własnych programach, które wykonywane przez komputer — pisane są przecież — dla ludzi.

Janusz MORBITZER



Rys. M. Przybyłowski

CYKLOIDA

Krąg toczy się po prostej. Związany z nim punkt leżący na okręgu pozostawia „ślad” znacząc swą drogę. Efektem tej „pracy” punktu będzie krzywa okresowa o długości łuku równej dla jednego cyklu ośmiu promieniom okręgu. Wystartujmy program. Linie 190... 240 wczytują 107 bajtów tworzących trzy podprogramy w języku wewnętrznym Z80. Nie można bez nich marzyć o poprawnej animacji.

Menu pozwala wybrać 9 wykresów cykloidy w trzech wariantach. „Pracujący” punkt może leżeć na okręgu (cykloida zwyczajna), na zewnątrz (cykloida wydłużona) lub wewnątrz niego (cykloida skrócona). W każdym wariantcie możemy obejrzeć trzy charakterystyczne przebiegi podając liczbę pełnych obrotów okręgu.

Ekran podzielony jest na trzy części identycznie, jak to zrobiono firmowo. W dolnej części przesuwany jest okrąg, aby go tam umieścić (zaczyna o linie edycyjne) trzeba go narysować w części środkowej (linia 420) i przerzucić w dół (linia 450). Operację wykonuje część programu maszynowego nazwana żartobliwie „desantem okręgu”.

Od linii 460 rozpoczyna się główna pętla realizująca wykres. Położenie okręgu w dolnym ekranie jest zsynchronizowane z obliczonym położeniem promienia wodzącego rysowanym na górnym ekranie (linie 480... 510). Teraz obie części ekranu są logicznie sumowane przed kod „nakładanie ekranu” i wyświetlane w części środkowej dając efekt animacji. Okrąg przesuwa się (linie 550... 610) i cały cykl powtarza się. Przemieszczanie okręgu realizuje procedura „scroll prawy”. Proces zachodzi co jeden punkt w każdym przebiegu głównej pętli.

Kreślenie krzywej można zatrzymać w każdym momencie dowolnym klawiszem i podobnym wznowić (stopklatka).

Rezygnacja z opisów i wzorów jest świadoma — można je znaleźć w książkach.

K.M.

```

10 REM ** cykloida 86
20 OVER 0: INVERSE 0: FLASH 0: BRIGHT 0
30 CLEAR 59999
40 REM grafiki
50 LET c=16224
60 FOR a=0 TO 23
70 POKE USR "a"+a, PEEK (c+a)
80 IF a=7 THEN LET c=16240
90 IF a=15 THEN LET c=16320
100 NEXT a
110 POKE USR "a"+3,24
120 POKE USR "a"+8,16
130 POKE USR "a"+16,16
140 LET c$=" CYKLOIDA"
150 LET d$="zwyczajna"

```

```

160 LET e$="wyd"+ CHR$ 144+"u"+ CHR$ 146
+"ona"
170 LET f$="skr"+ CHR$ 145+"cona"
180 REM kod maszynowy
190 LET a=60000
200 READ r
210 IF r<0 THEN GOTO 250
220 POKE a,r
230 LET a=a+1
240 GOTO 200
245 REM menu
250 BORDER 5: PAPER 5: INK 0: CLS
260 PRINT "'c$'" 1 "+d$'" 2 "+e$'" 3
"+f$"
270 IF PEEK 23560=49 THEN LET r1=0: GOTO
310
280 IF PEEK 23560=50 THEN LET r1=5: GOTO
310
290 IF PEEK 23560=51 THEN LET r1=-5: GOT
D 310
300 GOTO 270
310 INPUT AT 22,0;AT 15,4;"liczba cykli
(1..3)"i
320 LET i=ABS (INT i)
330 IF i<1 OR i>3 THEN GOTO 310
340 REM cykloida
350 BORDER 1
360 PAPER 1: INK 1: CLS
370 FOR a=22784 TO 23039
380 POKE a,14
390 NEXT a
400 POKE 23624,9: INPUT ()
410 LET r=15
420 CIRCLE INK 8:PAPER 8;r,74,r
430 PLOT r,123: DRAW 225,0: DRAW -5,3
440 PLOT r,123: DRAW 0,50: DRAW 3,-5
450 RANDOMIZE USR 60067
460 FOR a=1 TO 225 STEP i
470 IF LEN INKEY$ THEN GOSUB 680
480 LET alfa=a*PI/95*i
490 LET x=-(r+r1)* SIN alfa
500 LET y=-(r+r1)* COS alfa
510 GOSUB 650
520 RANDOMIZE USR 60079
530 GOSUB 650
540 PLOT PEEK 23677,PEEK 23678
550 FOR b=1 TO i
560 POKE 23602,0
570 POKE 23603,80
580 POKE 23604,64
590 POKE 23605,32
600 RANDOMIZE USR 6E4
610 NEXT b
620 NEXT a

630 PRINT INK 9;AT 18,3;c$;" ";
631 IF NOT r1 THEN PRINT INK 9;d$
632 IF r1>0 THEN PRINT INK 9;e$

```

dokończenie na str. 8

```

633 IF r1<0 THEN PRINT INK 9;f$
634 IF NOT LEN INKEY$ THEN GOTO 634
635 GOTO 350
650 POKE 23677,r+a: POKE 23678,138
660 DRAW OVER 1;x,y
670 RETURN
680 IF LEN INKEY$ THEN GOTO 680
690 IF NOT LEN INKEY$ THEN GOTO 690
700 RETURN
710 DATA 042,050,092,237,075,052,092,167
,203,030,035,005,032,250,042,050,092,205
,131,234,034,050,092,237,075,052,092,013
,200,237,067,052,092,024,228,124,230,007
,254,007,040,002,036,201,125,230,224,254
,224,040,007,017,224,006,167,237,082,201
720 DATA 124,254,087,200,017,032,000,025
,201,033,000,072,017,000,080,001,000,008
,237,176,201,033,000,064,017,000,080,001
,000,008,221,033,000,072,026,182,221,119
,000,035,019,221,035,011,120,177,032,242
,201

```

```
730 DATA -1
```

```
5 ;scroll prawy
```

```

10 LD HL,(23602)
20 LD BC,(23604)
30 L1 AND A
40 L2 RR (HL)
50 INC HL
60 DEC B
70 JR NZ,L2
80 LD HL,(23602)
90 CALL CH
100 LD (23602),HL
110 LD BC,(23604)
120 DEC C
130 RET Z
140 LD (23604),BC
150 JR L1

```

```

160 CH LD A,H
170 AND #07
180 CP #07
190 JR Z,L3
200 INC H
210 RET
220 L3 LD A,L
230 AND #E0
240 CP #E0
250 JR Z,L4
260 LD DE,01760
270 AND A
280 SBC HL,DE
290 RET
300 L4 LD A,H
310 CP #57
320 RET Z
330 LD DE,00032
340 ADD HL,DE
350 RET
370 ;desant okregu
390 LD HL,18432
400 LD DE,20480
410 LD BC,2048
420 LDIR
430 RET
440 ;nakladanie ekranu
460 LD HL,16384
470 LD DE,20480
480 LD BC,2048
490 LD IX,18432
500 LOOP LD A,(DE)
510 OR (HL)
520 LD (IX),A
530 INC HL
540 INC DE
550 INC IX
560 DEC BC
570 LD A,B
580 OR C
590 JR NZ,LOOP
600 RET

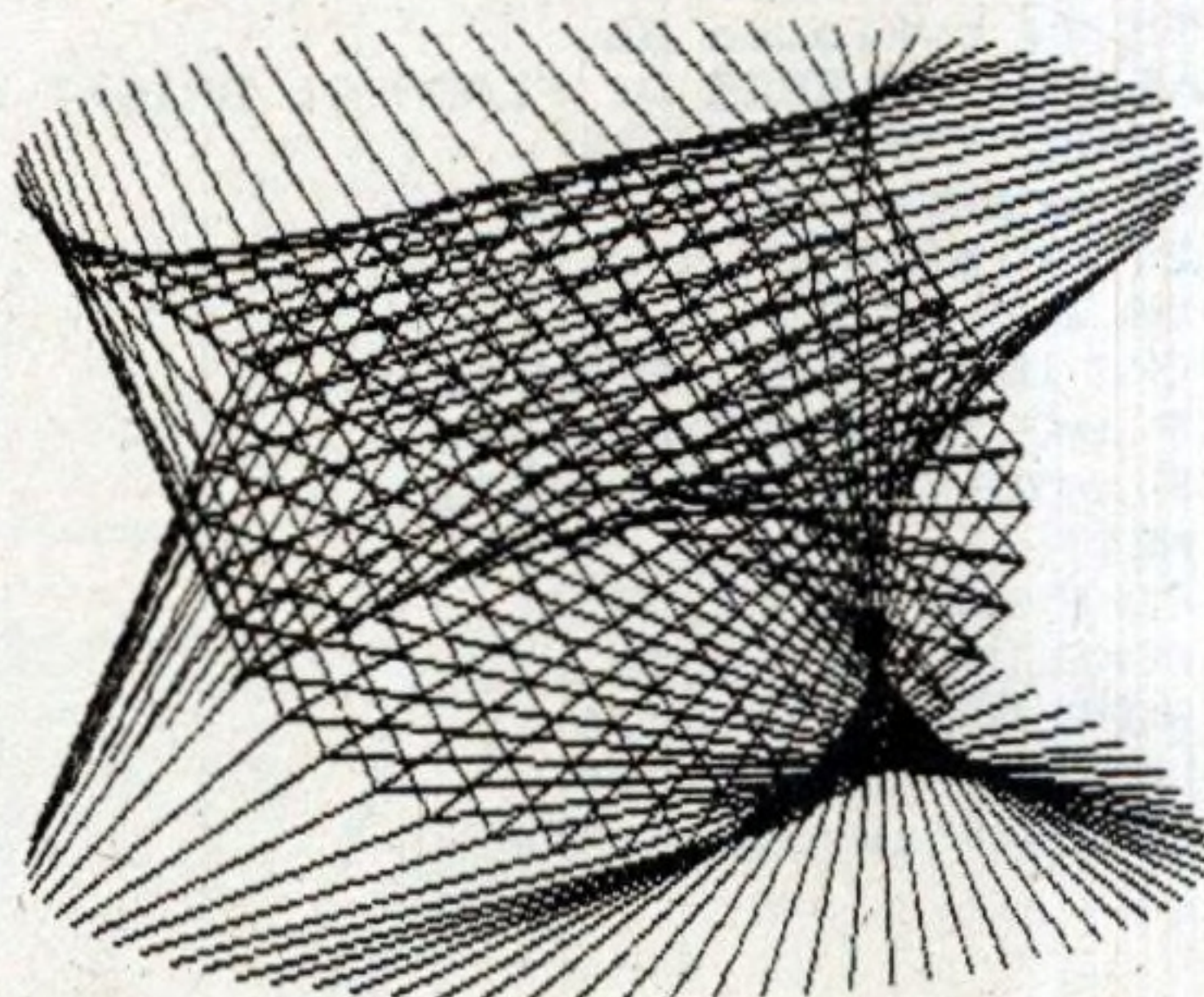
```

```

10 CLS
20 LET z=0
30 b%=RND*5+1
40 c%=RND*5+1
50 ORIGIN 320,200
60 FOR a=0 TO 100 STEP PI/20
70 x%=140*COS(a)
80 y%=140*SIN(a)
90 MOVE x%,y%
100 DRAW 200*SIN(a/b%),200*COS(a/c%)
110 LET z=z+1
120 IF z=600 GOTO 140
130 NEXT a
140 PRINT "koniec"

```

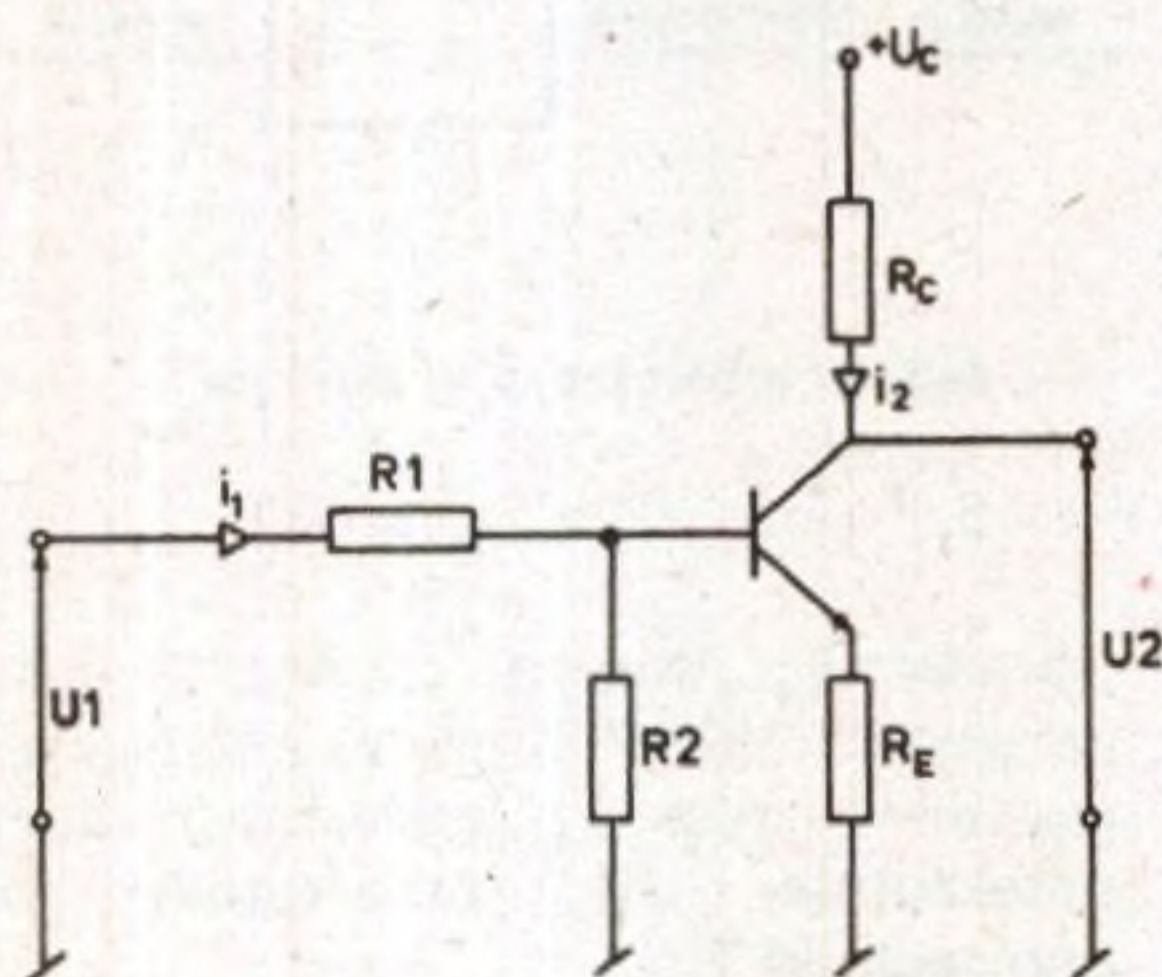
**Grafika
(AMSTRAD)**



Logika i arytmetyka w technice cyfrowej

Poprzedni odcinek tego cyklu zapoznał Was z architekturą mikrokomputera i mikroprocesora. Czas przejść do podstaw matematycznych, terminologii i pojęć, których znajomość jest niezbędna każdemu, kto pragnie poznać działanie elementów mikrokomputera.

Podstawowym elementem, używanym do budowy mikroprocesorów i pozostałych elementów systemów mikrokomputerowych jest tranzystor, elektroniczny element półprzewodnikowy o trzech wyprowadzeniach: bazie, emitażerze i kolektorze. Tranzystor, przedstawiony na rys. 3., może stabilnie i pewnie pracować w dwóch stanach.



Rys. 3. Tranzystor jako element dwustanowy

1° Gdy na wejście tranzystora, podane zostanie napięcie U_1 bliskie 0 V, wówczas prąd wyjściowy jest prawie równy zero i napięcie U_2 na wyjściu tranzystora bliskie jest napięciu U_c . Tranzystor nie przewodzi, jest rozarty.

2° Gdy na wejście podane zostanie napięcie bliskie U_c , wówczas prąd wyjściowy jest bardzo duży, a napięcie U_2 bliskie jest 0 V. Tranzystor jest nasycony, zwarty.

Dwa powyższe stany, oznaczone umownie 1 (**jedynka logiczna**) i 0 (**zero logiczne**) oraz to, że ludzie mają tendencję do używania zwrotów **prawda** (1) lub **falsz** (0) powoduje, iż naturalnym systemem liczenia w komputerach jest **dwójkowy system liczbowy**.

Cała informacja dla mikroprocesora oraz innych elementów systemu mikrokomputerowego przedstawiona jest jako informacja dwójkowa, za pomocą wielkości fizycznych, zwanych **sygnalami dwuwartościowymi (dwustanowymi)**. W danym systemie można

symbol graficzny	nazwa bramki	funkcja algebraiczna nazwa operacji	tablica funkcji															
	bufor	$X = A$	<table border="1"> <tr><td>A</td><td>X</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	A	X	0	0	1	1									
A	X																	
0	0																	
1	1																	
	inwerter NOT	$X = \bar{A}$ negacja (dopełnienie)	<table border="1"> <tr><td>A</td><td>X</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	
	AND	$X = A \cdot B$ iloczyn logiczny	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
	NAND	$X = \overline{A \cdot B}$ zanegowany iloczyn logiczny	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
	OR	$X = A + B$ suma logiczna	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
	NOR	$X = \overline{A + B}$ zanegowana suma logiczna	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
	EXCLUSIVE-OR	$X = A \oplus B$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
	EXCLUSIVE-NOR	$X = A \odot B$ równoważność	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

używać sygnału 3 V (z tolerancją 1 V) jako logicznej jedynki (1) i sygnału 0,5 V (z tolerancją 0,5 V) jako logicznego zera (0).

Systemy elektroniczne, zbudowane z elementów (układów) dwustanowych, nazywane są **systemami cyfrowymi**. Układy dwustanowe, inaczej cyfrowe, na swoich wejściach akceptują sygnały dwuwartościowe o określonej tolerancji. Na wyjściach tych układów pojawiają się również sygnały dwuwartościowe.

Obróbkę informacji dwójkowych można prowadzić za pomocą układów cyfrowych zwanych **bramkami logicznymi**. W tabeli 1 przedstawiono osiem bramek wykonujących operacje logiczne. Operacje negacji, sumy logicznej i iloczynu logicznego, trzy podstawowe operacje logiczne, łatwo można przeprowadzić za pomocą układów tranzystorowych. Do budowy układów scalonych, realizujących powyższe operacje, układów popularnie zwanych bramkami, wykorzystano właśnie tranzystory.

Operacjami logicznymi i zmiennymi dwójkowymi zajmuje się **algebra Boole'a**.

TABELA 1 BRAMKI LOGICZNE

le'a. Ze znanych wam już symboli operacji logicznych oraz ze zmiennych dwójkowych, nawiasów i znaku równości można utworzyć wyrażenie algebraiczne, funkcję boolowską, która dla danej wartości zmiennych przyjmuje wartość 1 lub 0.

Daną funkcję boolowską, inaczej **funkcję logiczną**, można przedstawić za pomocą układu złożonego z bramek logicznych.

dokończenie na str. 10

dokończenie ze str. 9

Dobra znajomość poniżej przedstawionych podstawowych zależności algebry Boole'a, pozwala na otrzymywanie prostszych funkcji logicznych, realizowanych za pomocą mniejszej ilości bramek.

TABLICA 1

PODSTAWOWE ZALEŻNOŚCI ALGEBRY BOOLE'A

- (1) $a \cdot 0 = 0$
- (2) $a \cdot 1 = a$
- (3) $a \cdot a = a$
- (4) $a \cdot \bar{a} = 0$ */
- (5) $a \cdot b = b \cdot a$
- (6) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- (7) $a \cdot (b + c) = a \cdot b + a \cdot c$
- (8) $a + 0 = a$
- (9) $a + 1 = 1$
- (10) $a + a = a$
- (11) $a + \bar{a} = 1$
- (12) $a + b = b + a$
- (13) $a + (b + c) = (a + b) + c$
- (14) $a + b \cdot c = (a + b) \cdot (a + c)$
- (15) $\bar{\bar{a}} = a$
- (16) $\overline{a \cdot b} = \bar{a} + \bar{b}$
- (17) $\overline{a + b} = \bar{a} \cdot \bar{b}$

*/ \bar{a} — „a” zanegowane

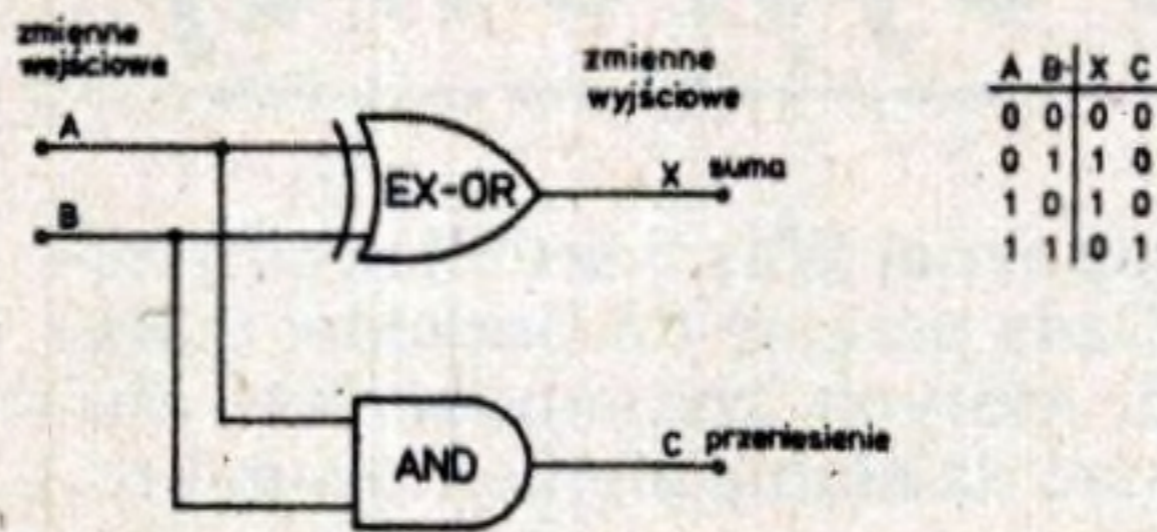
Dwie ostatnie zależności, oznaczone (16), (17), nazywane **prawami de Morgana**, są bardzo ważne przy działaniach na funkcjach NAND i NOR. Z praw tych wynika, że dowolną funkcję boolowską można zrealizować za pomocą bramek jednego z dwóch typów: NAND lub NOR. Nie jest to bez znaczenia, w związku m. in. z tym, podstawowymi, powszechnie wykorzystywanymi bramkami są NAND i NOR, umieszczone (po cztery danego typu) w układach 7400 i 7402.

Jedną z metod upraszczania postaci funkcji logicznych jest metoda **tablic Karnaugha**, znana też pod nazwą metody **wykresów Veitcha**. Metoda ta polega na minimalizacji funkcji przez łączenie w grupy jedynek i jest pomocna przy przechodzeniu od opisu słownego problemu do postaci funkcji boolowskiej.

Połączenie bramek logicznych tworzy **układ kombinacyjny**, w którym

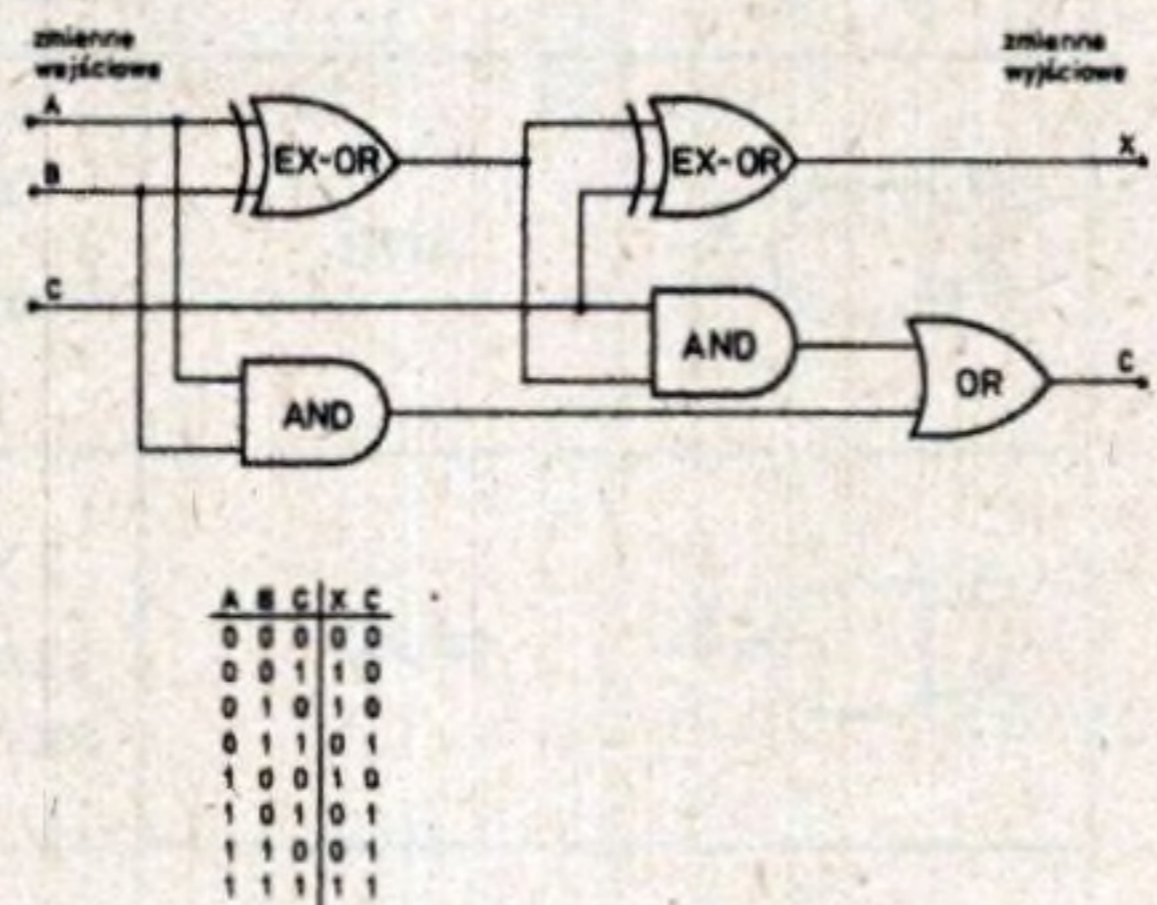
sygnały dwuwartościowe na wyjściach są jednoznacznie określonymi funkcjami kombinacji zer i jedynek, podanych w danej chwili na wejścia. Wśród przykładów układów kombinacyjnych można podać półsumator i sumator pełny.

Półsumator, przedstawiony na rys. 4,



Rys. 4. Układ półsumatora

umożliwia arytmetyczne dodawanie dwóch bitów. Natomiast sumator pełny, przedstawiony na rys. 5, umożliwia



Rys. 5. Układ sumatora pełnego

arytmetyczne dodawanie trzech bitów wejściowych. Na wejście C podawane jest przeniesienie z poprzedniej, mniej znaczącej pozycji.

Układ, w którym, oprócz bramek logicznych, znajdują się układy pamiętające, zwane przerzutnikami jest **układem sekwencyjnym**. W takim układzie sygnały dwuwartościowe na wyjściach są kombinacją zer i jedynek na wejściach, jak również zależą od stanów przerzutników, czyli od ostatnio zapamiętanych wartości sygnałów.

Układy sekwencyjne można podzielić na: **synchroniczne**, w których momenty działania poszczególnych elementów elektronicznych taktowane są **impulsami zegarowymi** i **asynchroniczne**, bez wyróżnionych impulsów zegarowych.

Do przechowywania informacji dwójkowych wykorzystywane są cyfrowe układy scalone zwane **rejestrami**. Rejestr zawiera w sobie **przerzutniki** — komórki logiczne do przechowywania jednego bitu informacji oraz może posiadać bramki kombinacyjne, ułatwiające przetwarzanie informacji.

Przerzutniki można podzielić na **asynchroniczne R-S** oraz **synchroniczne**, wykorzystywane w systemach

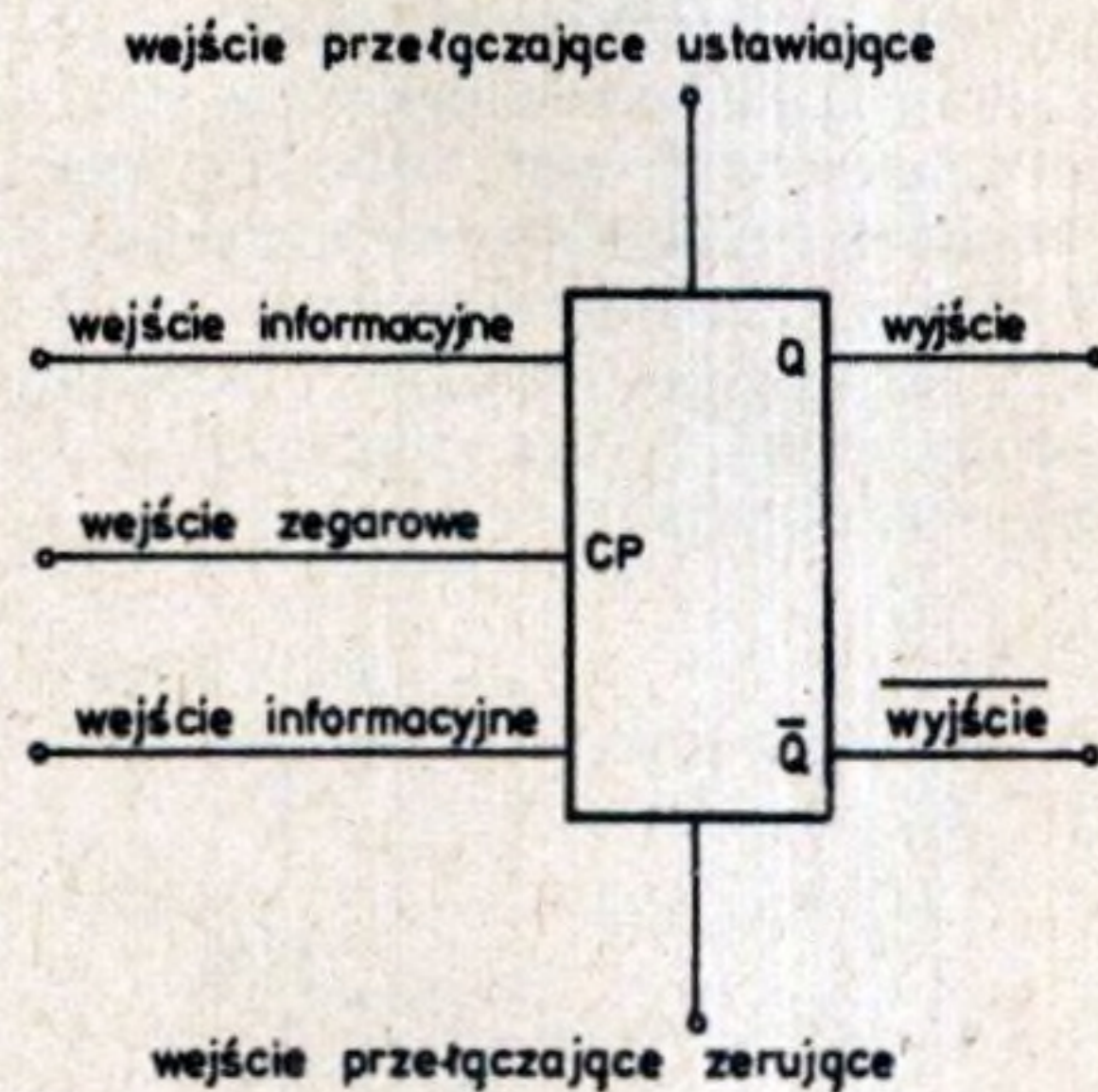
komputerowych. Przerzutniki synchroniczne można podzielić na:

— **wyzwalane zboczem**, zmieniające swój stan w chwili zmiany wartości sygnału zegarowego z 1 na 0 lub z 0 na 1;

— **wyzwalane poziomem** inaczej zatraskowe typu „latch”, przenoszące zmiany z wejścia na wyjście przy jedynce logicznej na wejściu zegarowym;

— **dwuzboczowe typu Master-Slave**, składające się z przerzutnika głównego i przerzutnika pomocniczego.

Symbol graficzny przerzutnika synchronicznego przedstawia rys. 6



Rys. 6. Przerzutnik synchroniczny

Upraszczając można stwierdzić, iż w zależności od posiadanych wejść przerzutniki można sklasyfikować jako: **przerzutniki JK**, **przerzutniki D**, **przerzutniki T**.

Rejestry można sklasyfikować w następujący sposób:

— **szeregowe**, do szeregowego (kolejnego) wprowadzania i wyprowadzania informacji, bit po bicie;

— **równoległe**, do równoległego (jednoczesnego) wprowadzania i wyprowadzania informacji;

— **szeregowo-równoległe**, do szeregowego wprowadzenia i równoległego wyprowadzenia informacji;

— **równoległo-szeregowe**, do równoległego wprowadzenia i szeregowego wyprowadzenia informacji.

Rejestry szeregowe mogą być **rejestrami jednokierunkowymi**, przesuwającymi informację w lewo lub w prawo albo **rejestrami dwukierunkowymi**, przesuwającymi informację w lewo i w prawo. Rejestry równoległe wykorzystywane są do budowania **pamięci buforowych**, przechowujących informację podawane w sposób równoległy.

Rejestr zmieniający stan w określonej kolejności nazywany jest **licznikiem**. Licznik służy do zliczenia i zapamiętania liczby impulsów podanych na jego wejścia zliczające. Licznik można podzielić m.in. na **jednokierunkowe** i **dwukierunkowe (rewersyjne)**

oraz na **asynchroniczne, synchroniczne, asynchroniczno-synchroniczne**. Liczniki zbudowane są, podobnie jak rejestry, z przerzutników i bramek kombinacyjnych.

Innymi układami, które spotkacie w systemie cyfrowym są układy do przetwarzania kodów. Układy służące do zmiany kodu, w czasie komunikacji człowiek - system, nazywane są **przetwornikami (konwerterami) kodów**. Do zamiany informacji dwójkowej na kod pierścieniowy „1 z n” służą **dekodery**, do odwrotnej zamiany zaś **enkodery**. Pozostałe przetworniki nazywa się **transkoderami**.

Do realizacji **transmisji (przesyłania) informacji** wykorzystywane są **multipleksery i demultipleksery** czyli przetworniki formatu słów z równoległego na szeregowy oraz do transmisji linią przesyłową z szeregowego na równoległy.

W części sterującej każdego systemu cyfrowego spotykane są **układy uzależnień czasowych**. Są wśród nich **przerzutniki monostabilne, generatory fali prostokątnej, rozdzielacze, selektory impulsów** itp.

Do wykonywania działań arytmetycznych w systemach cyfrowych wykorzystywane są **sumatory; subtraktory** (do odejmowania); sumatory uzupełnione o układy przesuwające, umożliwiające wykonywanie mnożenia i dzielenia; **komparatory**, porównujące wielkość kilku liczb; **multiplikatory**, do realizacji operacji mnożenia; **jednostki arytmetyczno-logiczne ALU**, wykonujące szereg operacji arytmetycznych; **układy kontroli parzystości** itp.

Kolejne grupy układów systemów cyfrowych to: **układy sprzęgające**, sterujące m.in. wskaźnikami cyfrowymi, współpracujące z klawiaturą oraz **układy transmisji**, do przekazywania informacji na duże odległości.

Tak w skrócie wygląda charakterystyka układów systemów cyfrowych.

Wróćmy do arytmetyki systemów cyfrowych. System liczbowy o **podstawie R** jest systemem, w którym każda z r cyfr oznaczona jest różnym **symbolem**. W **dziesiętnym systemie liczbowym**, systemie o podstawie r = 10, używa się dziesięciu symboli: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. W **dwójkowym systemie liczbowym** r = 2 i używane są dwa symbole: 0, 1. W **ósemkowym (oktalnym) systemie liczbowym** r = 8 i używanych jest osiem symboli: 0, 1, 2, 3, 4, 5, 6, 7. W **szesnastkowym (heksadecymalnym) systemie liczbowym** r = 16 i używa się szesnastu symboli: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

W systemach cyfrowych spotkacie się właśnie z tymi systemami liczbowymi. Zapiszmy przykładowo w każ-

dym z tych systemów liczbę 1024.

Liczba w systemie dziesiętnym zapisywana jest następująco:

(18)

$$L_{10} = \sum_{i=-m}^{n-1} a_i \cdot 10^i = a_{n-1} \cdot 10^{n-1} + a_{n-2} \cdot 10^{n-2} + \dots + a_3 \cdot 10^3 + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$$

1024 to:

$$1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$$

Liczba w systemie dwójkowym zapisywana jest następująco:

(19)

$$L_2 = \sum_{i=-m}^{n-1} a_i \cdot 2^i = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \dots + a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + \dots + a_{-m} \cdot 2^{-m}$$

1024₁₀ to:

$$1 \cdot 2^{10} + 0 \cdot 2^9 + 0 \cdot 2^8 + \dots + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$1024_{10} = 100\ 0000\ 0000_2$$

Liczba w systemie ósemkowym zapisywana jest następująco:

(20)

$$L_8 = \sum_{i=-m}^{n-1} a_i \cdot 8^i = a_{n-1} \cdot 8^{n-1} + a_{n-2} \cdot 8^{n-2} + \dots + a_3 \cdot 8^3 + a_2 \cdot 8^2 + a_1 \cdot 8^1 + a_0 \cdot 8^0 + a_{-1} \cdot 8^{-1} + \dots + a_{-m} \cdot 8^{-m}$$

1024₁₀ to:

$$2 \cdot 8^3 + 0 \cdot 8^2 + 0 \cdot 8^1 + 0 \cdot 8^0$$

$$1024_{10} = 2000_8$$

Liczba w systemie szesnastkowym zapisywana jest następująco:

(21)

$$L_{16} = \sum_{i=-m}^{n-1} a_i \cdot 16^i = a_{n-1} \cdot 16^{n-1} + a_{n-2} \cdot 16^{n-2} + \dots + a_3 \cdot 16^3 + a_2 \cdot 16^2 + a_1 \cdot 16^1 + a_0 \cdot 16^0 + a_{-1} \cdot 16^{-1} + \dots + a_{-m} \cdot 16^{-m}$$

1024₁₀ to:

$$4 \cdot 16^2 + 0 \cdot 16^1 + 0 \cdot 16^0$$

$$1024_{10} = 400_{16}$$

zestawiając:

$$1024_{10} = 100\ 0000\ 0000_2 = 2000_8 = 400_{16}$$

Przykłady liczb całkowitych, zapisanych w różnych systemach przedstawiono w tablicy 2.

Tablica 2

LICZBY W RÓŻNYCH SYSTEMACH LICZBOWYCH

System dziesiętny	System szesnastkowy
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
20	14
40	28
50	32
100	64

System ósemkowy	System dwójkowy
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
10	1000
11	1001
12	1010
13	1011
14	1100
15	1101
16	1110
17	1111
20	10000
24	10100
48	101000
62	110010
144	1100100

Od arytmetyki rozpoczniemy nasze następne spotkanie.

Jacek WOJTALA

IKS — 11

KOMPUTERY W LOTNICTWIE

Komputery zadomowiły się na dobre we współczesnym lotnictwie, przejmując wiele różnorodnych zadań obciążających do tej pory człowieka bądź wykonywanych przez rozbudowane i zawodne urządzenia elektromechaniczne. Pełnią one wiele funkcji usługowych w urządzeniach naziemnych, rewolucjonizując szkolenie pilotów i obsługę techniczną dzięki realistycznej symulacji warunków lotu i złożonych systemów samolotu, usprawniając naziemną obsługę techniczną i diagnostykę, pracując w systemach ochrony lotnisk, pomagając w kontroli ruchu powietrznego czy też w rezerwacji miejsc na rejsy pasażerskie.

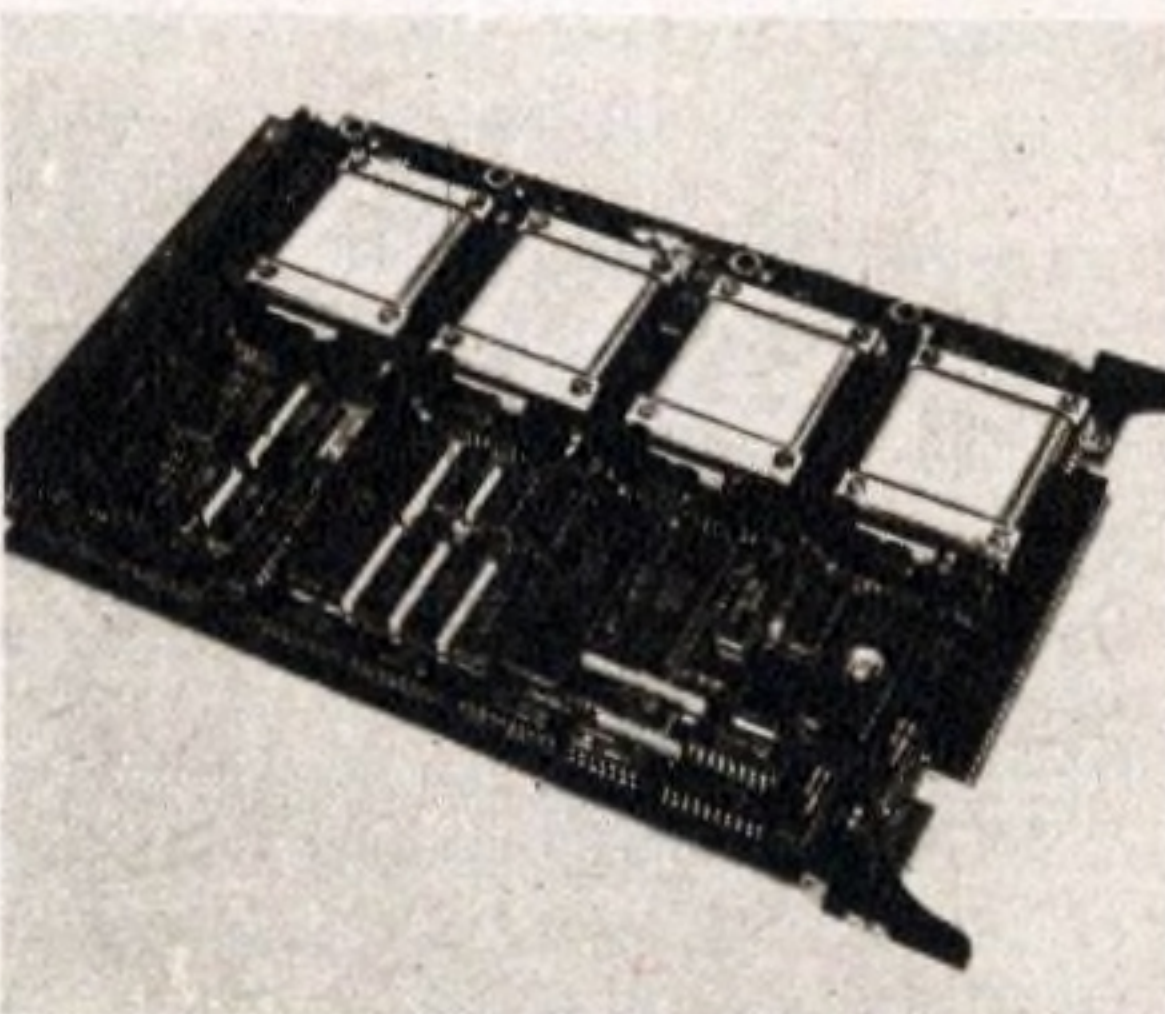
W dalszej części artykułu zajmiemy się jednak komputerami znajdującymi się na pokładzie samolotów. Mają one wiele zadań, z których najważniejsze, to:

- obliczenia nawigacyjne,
- pilotowanie samolotu,
- kontrola funkcjonowania elementów wyposażenia lotniczego,
- sterowanie pracą silników,
- sterowanie systemów dystrybucji paliwa,
- przetwarzanie informacji pilotażowo-informacyjnej na elektroniczne układy obrazowania (wskaźniki),
- obliczenia do kierowania ogniem i bombardowania (w samolotach wojskowych).

Zadania te komputery pokładowe wykonywać muszą niejednokrotnie w bardzo trudnych warunkach, wymaga się więc od nich wyjątkowej odporności na wpływy otaczającego środowiska, jak np. możliwości pracy w temperaturach od -55°C do $+80^{\circ}\text{C}$, odporności na zmiany ciśnienia, tj. poprawnej pracy zarówno na powierzchni Ziemi, jak też na wysokościach rzędu 15–25 km, odporności na przeciążenia w granicach 7–9 (przeciążenie — stosunek przyspieszeń występujących w locie po torze krzywoliniowym do przyspieszenia ziemskiego), wibracje w zakresie częstotliwości 10–1000 Hz, kurz, aerozole soli itp... a wszystko to w wilgotności względnej aż do 95%. Pomimo to komputery pokładowe pracują bezawaryjnie przez kilka tysięcy godzin ku zadowoleniu użytkowników.

Różnorodność zadań doprowadziła do pewnej specjalizacji i w chwili obecnej wyróżnić można komputery nawigacyjne, komputery danych powietrznych, komputery-przeliczniki kierowania bronią, komputery zarządzające urządzeniami awionicznymi i inne wąskospecjalizowane maszyny znajdujące się głównie na pokładzie samolotów bojowych, współpracujące z radarem, systemami kontroli podwieszanego uzbrojenia, radiokomunikacji, dystrybucji paliwa itp... Stało się tak jednak nie od razu...

Pierwsze próby wprowadzenia komputerów w lotnictwie sięgają lat pięćdziesiątych i dotyczą głównie maszyn analogowych. Wyjątkiem jest tutaj maszyna cyfrowa opracowana we wczesnych latach pięćdziesiątych, służąca jako przelicznik do kierowania ogniem rakiet odpalanych z myśliwca F-102. Była to konstrukcja oparta na lampach miniaturowych, funkcjonująca z szybkością setek operacji na sekundę. Jej program operacyjny złożony z kilkuset słów, zapisany był obwodowo na małym bębnie magnetycznym, w języku maszynowym.



Fot. 4 Magnetyczna pamięć pęcherykowa komputera systemu kierowania lotem

Burzliwy rozwój techniki komputerowej w latach sześćdziesiątych sprawił, że jednemu centralnemu komputerowi na pokładzie samolotu usiłowano podporządkować całość rozbudowanych urządzeń pokładowych, niejako wbrew usługowym funkcjom komputera, który jest tylko instrumentem pomocniczym, a nie celem samym w sobie. Obecnie komputery cyfrowe (procesory) rozmieszczone są w sposób zdecentralizowany, zwykle w pobliżu czujników bądź wskaźników, z którymi współpra-



Fot. 1 Komputery pomagają w kontroli ruchu powietrznego

czą, rozproszone w różnych miejscach samolotu, komunikując się wzajemnie poprzez magistrale danych integrujące wszystkie urządzenia awioniczne w jeden zwarty system. Dzięki nim możliwości pilotażowe rozszerzone zostały poza granice dotychczas niewyobrażalne. Przykładami mogą być tutaj systemy ślepego lądowania, pilot automatyczny, systemy nawigacji według profilu terenu czy też elektroniczne systemy sterowania (ang. „fly by wire”).

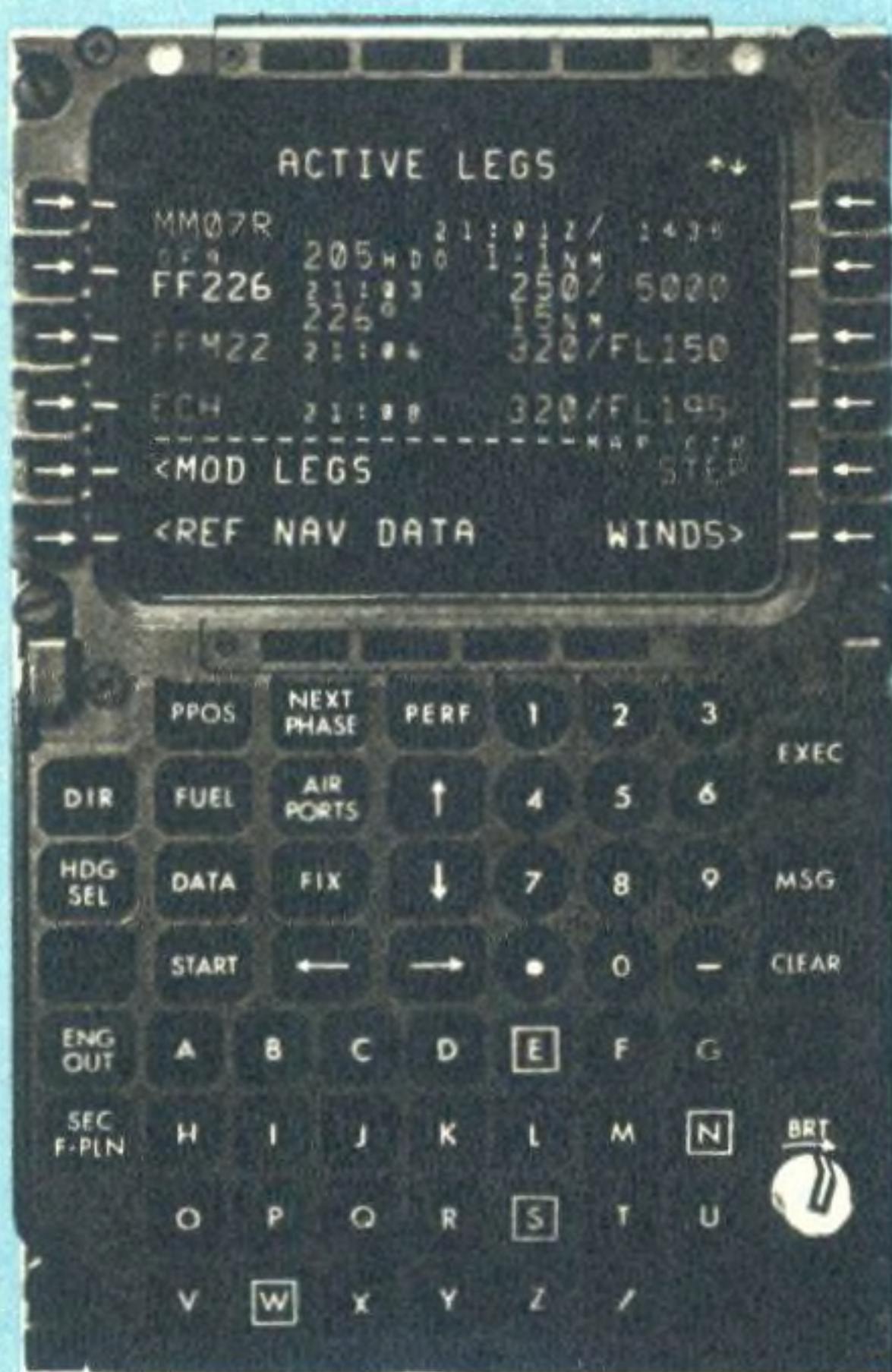
Właśnie opracowanie elektronicznego systemu sterowania samolotem było przełomem w wykorzystywaniu komputerów pokładowych. We wczesniejszych hydro-mechanicznych systemach sterowania projektanci zdani byli na łaskę i niełaskę charakterystyk aerodynamicznych samolotu. System „fly by wire” pozwala na nowatorskie rozwiązywanie zagadnień sterowania samolotem poprzez wprowadzanie sił sterujących dokładnie tam, gdzie trzeba i kiedy trzeba. Efektem jest samolot, który zachowuje się tak, jakby jego konfiguracja była idealnie dopasowana do warunków zastanych w danym momencie, zwany krótko CCV (ang. „control configured vehicle”), tj. samolot o konfigurowanej stateczności. Złożone problemy aerodynamiki lotu rozwiązywane są przez komputery za pomocą metod optymalizacji. Zaletami elektronicznego systemu sterowania są:

- duża zwrotność samolotu,
- mniejsze rozmiary powierzchni sterowych,
- niedopuszczanie do przeciągnięcia samolotu poprzez umieszczenie ograniczników w pętach sterowania.

Istota systemu „fly by wire” polega na usunięciu tradycyjnych mechanicznych połączeń pomiędzy pilotem a powierzchniami sterowymi i zastąpieniu ich sygnałem elektrycznym analogowym bądź cyfrowym, który na swej drodze do urządzeń wykonawczych przetwarzany jest przez specjalizowane komputery pokładowe sprzężone z

różnorodnymi czujnikami informującymi o parametrach lotu. Ponad połowa oprogramowania systemu przeznaczona jest do kontroli parametrów lotu i utrzymania odpowiednich wartości granicznych, tj. prędkości, współczynników obciążenia itp...

W celu zwiększenia niezawodności stosuje się potrójnie lub poczwórnie rezerwowane (rezerwowanie — stosowanie dodatkowych (rezerwowych) systemów lub elementów strukturalnych) systemy sterowania zapewniające zmniejszenie ryzyka przypadku utraty samolotu z powodu całkowitego uszkodzenia układu sterowania do 1 na 10 milionów tj. 10^{-7} milionów. Wyliczanie samolotów wyposażonych w elektroniczny system sterowania zajęłoby dużo miejsca, wspomnę więc tylko, że samolot pasażerski „Concorde” posiada poczwórnie rezerwowany analogowy system sterowania, samolot wielozadaniowy „Tornado” potrójnie rezerwowany system analogowy, a samolot myśliwski F-18 poczwórnie rezerwowany cyfrowy. Prom kosmiczny „Space Shuttle” zaopatrzony jest również w poczwórnie rezerwowany cyfrowy system sterowania.



Fot. 3 Moduł wprowadzania i zobrazowania informacji komputera systemu kierowania lotem

Wprowadzenie obwodów o dużej i bardzo dużej skali integracji doprowadziło do rewolucji w dziedzinie wykorzystywania systemów sterowania i informowania na pokładzie samolotu. Komputery (procesory) pozwoliły na realizację techniczną wielu nowych urządzeń i zmiany jakościowe urządzeń dotychczas istniejących, a mianowicie: wskaźników refleksyjnych (obraz wyświetlany na przedniej szybie kabiny), wielofunkcyjnych wskaźników

barwnych, wielofunkcyjnych radarów dopplerowskich, obserwacyjnych urządzeń termalnych (termowizyjnych), laserowych dalmierzy, oświetlaczy i termolokatorów, sterowanych mikroprocesorowo urządzeń walki radioelektronicznej, systemów nawigacji według profilu terenu i wspomnianych już elektronicznych układów sterowania samolotem. Te nowe rozwiązania zwiększyły znacznie bezpieczeństwo latania poprzez dokładne i szybkie informowanie pilota, poprawienie manewrowości i umożliwienie lotów w praktycznie każdych warunkach atmosferycznych. W samolotach wojskowych poprawiły istotnie bojową efektywność poprzez zwiększenie celności i skuteczności ognia oraz wprowadzenie skuteczniejszych urządzeń walki elektronicznej. Ten jakościowy skok technologiczny stał się możliwy dzięki przejęciu przez komputery co najmniej czterech zadań niemożliwych dotychczas do wykonania przez urządzenia. Należą do nich:

- łączenie w sposób komplementarny wskazań wielu czujników pokładowych pozwalających na uzyskanie jakościowo wyższych parametrów (wskazań), jak ma to miejsce w przypadku komputerów nawigacyjnych przetwarzających dane z układu nawigacji bezwładnościowej, radaru dopplerowskiego, radiokompasu, itp....
- zastąpienie złożonych urządzeń elektromechanicznych, jak ma to miejsce w systemach nawigacji bezwładnościowej, radarach dopplerowskich, systemach stabilizacji częstotliwości prądów i napięć zasilających, przelicznikach kierowania ogniem itp....
- przetwarzanie danych z czujników jak np. procesory radarowe, procesory danych powietrznych, procesory obrazowe wskaźników barwnych na lampach kineskopowych, itp....
- wykorzystanie nowych zjawisk fizycznych w konstrukcji urządzeń pokładowych, dotychczas niemożliwych do realizacji, jak np. żyroskop laserowy w systemie nawigacji bezwładnościowej.

Jednym z ciekawszych zastosowań komputerów w lotnictwie komunikacyjnym jest komputerowy system kierowania lotem. Integruje on funkcje systemów nawigacji, automatycznego sterowania silnikami i autopilota. Umożliwia on bezpieczne prowadzenie samolotu według zadanej trasy przelotu, oblicza optymalną prędkość lotu i wysokość, zużycie paliwa i czasy przylotu do wyznaczonych punktów trasy i lotnisk. Dokładny opis techniczny systemu jest trudny i niezrozumiały dla średnio przygotowanego czytelnika. Jego struktura oparta jest na kilku procesorach, co pozwala uzyskać większe

szybkości działania niż w przypadku jednego procesora centralnego. Są to dostępne komercyjnie procesory 16-bitowe typu Intel 8086, 8087 i 8088. Pamięć swobodnego dostępu RAM może być trójakiego rodzaju: magnetyczna rdzeniowa, magnetyczna na dyskach lub magnetyczna pęcherzykowa. Ta ostatnia stosowana jest w najnowocześniejszych rozwiązaniach ze względu na dużą gęstość upakowania danych i występowanie części ruchomych, co zwiększa jej niezawodność. Objętość programów komputera systemu kierowania lotem zawiera się w granicach 256 do 512 kilobajtów.

Najbardziej nasycone techniką mikrokomputerową są samoloty bojowe. Pilot współczesnego samolotu wielozadaniowego musi wykryć cele, zidentyfikować, śledzić i zniszczyć, w czym pomocne mu są najnowocześniejsze systemy pokładowe i broni. W toku wykonywania zadania komputery pokładowe wykonują miliony obliczeń w



Fot. 2 „Skomputeryzowana” kabina samolotu pasażerskiego najbliższej przyszłości A-320. Widoczne wielofunkcyjne wskaźniki barwne na lampach kineskopowych i w dolnej części dwa moduły wprowadzania i zobrazowania danych systemu kierowania lotem

ułamku sekundy i podejmują różnorodne decyzje. Pilot prowadząc samolot musi ciągle kontrolować przyrządy i interpretować ich wskazania aby wykonać zadanie bojowe. W celu odciążenia pilota wszelkie decyzje i funkcje, które można zautomatyzować zostały włączone w zintegrowane podsystemy komputerowe. Oczywiście operacje wykonywane przez nie są pod kontrolą pilota i w każdej chwili może wykonać je sam lub zdając się na podsystemy komputerowe realizować swoje podstawowe zadania, tj. obmyślać taktykę walki i obserwować wskazania przyrządów. Rozwiązania systemowe współczesnych samolotów bojowych pozwalają bowiem na w pełni zautomatyzowane prowadzenie samolotu, bez ingerencji w pracę poszczególnych układów. Samolot potrzebuje ręki pilota jedynie w momencie startu i lądowania,

dokończenie na str. 14

jak również podczas manewrów w rejonie działań bojowych (celu). Przelot według zadanej trasy do wybranych lotnisk i rejonów, jak również podejście do lądowania, do wysokości kilkudziesięciu metrów, odbywają się automatycznie.

System komputerowy samolotu bojowego składa się z 1—2 komputerów zadaniowych i pewnej liczby komputerów specjalizowanych współpracujących z czujnikami i wskaźnikami. Praktycznie wszystkie nowoczesne samoloty bojowe wyposażone są w tego typu systemy komputerowe lub przynajmniej w komputerowy system nawigacyjno-celowniczy. Wszelkie istotne dane pomocne do wykonania zadania wprowadzane są do pamięci komputera przed startem samolotu. Zwykle są to współrzędne geograficzne lotnisk, punktów pośrednich trasy przelotu, współrzędne rejonów celów oraz cała gama dodatkowych warunków związanych ze specyfiką wykonywanego zadania. Dane te wprowadzać można z kart perforowanych, taśm magnetycznych bądź za pomocą klawiatury alfanumerycznej.

Komputery pokładowe są znacznie zróżnicowane zależnie od typu samolotu i zadań przed nim stawianych. Obecnie na pokładzie samolotów występują komputery z procesorami 8-, 16-, 24- i 32-bitowymi adresującymi pamięci o pojemności od 32 do 512 kilobitów i szybkości działania w granicach 200 tys. do 800 tys. operacji na sekundę. Standardem w przypadku samolotów wielozadaniowych, myśliwsko-bombowych stał się już praktycznie komputer z procesorem 16-bitowym, jakkolwiek starsze modele tych samolotów mają procesor 8-bitowy. Procesory 16-bitowe wchodzi również w skład większości opracowywanych obecnie systemów minikomputerowych cywilnych samolotów pasażerskich. Natomiast samoloty specjalnego przeznaczenia służące np. do wczesnego ostrzegania powietrznego, wykrywania i zwalczania okrętów podwodnych oraz bombowce strategiczne wyposażone są w komputery o standardzie 32-bitowym, co wiąże się z ogromem informacji zbieranej z dużego obszaru przestrzeni i przetwarzanej z czasem rzeczywistym przez pokładowe systemy komputerowe.

Podobnie zróżnicowana jest wielkość oprogramowania komputerów pokładowych. Program samolotu F-16 składa się np. ze 128 kilobitów, z czego 32 kilobitów to program komputer-przelicznika kierowania bronią, o szybkości działania 500 tys. operacji na sekundę. Natomiast program samolotu F-18 złożony jest z 400 kilobitów i realizowany przez 13 komputerów pokładowych. Większe samoloty, jak np. samolot wczesnego ostrzegania marynarki wojennej USA, typu P-3, którego oprogramowanie ma objętość 700 ki-

lobitów, z czego 128 kilobitów w magnetycznej pamięci rdzeniowej, bądź samolot wczesnego ostrzegania E-3A „AWACS” mający oprogramowanie o objętości 512 kilobitów, to już niemal latające ośrodki obliczeniowe. Żeby utwierdzić się w tym przekonaniu dodam, że program wykorzystywany do celów diagnostyki i obsługi technicznej tego typu samolotów ma objętość nieco ponad 1 milion słów. Od 30% do 70% programów komputerów pokładowych napisanych jest w językach wyższego rzędu, takich jak: Coral, Jovial, Fortran lub Macro. Reszta zaś w językach maszynowych typu Assembler.

W dziedzinie technologii komputerów pokładowych o standardzie 32-bitowym dokonał się w ciągu ostatnich kilku lat ogromny postęp. Dowodem są dane porównawcze tego typu komputerów z lat 1980 i 1985, opublikowane przez amerykańską firmę IBM:

Do światowej czołówki w produkcji pokładowych komputerów lotniczych należą firmy amerykańskie, angielskie i francuskie. Komputery o wysokim standardzie produkowane są również w Szwecji, RFN, Izraelu i we Włoszech. W krajach socjalistycznych komputery lotnicze produkowane są głównie w ZSRR.

Dzień dzisiejszy przynosi bowiem ciągle nowe rozwiązania techniczne. Dopiero w chwili obecnej konstruktorzy systemów awionicznych zaczynają pojmować, jak różnorodne zadania spełniać mogą napływające w znacznej ilości komputery i w jaki sposób można je wykorzystać najbardziej efektywnie. Jednymi z nowszych pomysłów, w toku prób, są np. cyfrowe systemy sterowania głosem i fotel wyrzucany, którego katapultowaniem steruje układ mikroprocesorowy.

ZBIGNIEW KAROLCZYK

**1980
komputer syst. „AWACS”
typu CC-2**

szybkość działania 2 mln oper./s
pojemność pamięci 2,5 Mbitów
pobór mocy 6500 W
objętość 1036 dm³

**1985
komputer AN/UJK-41**

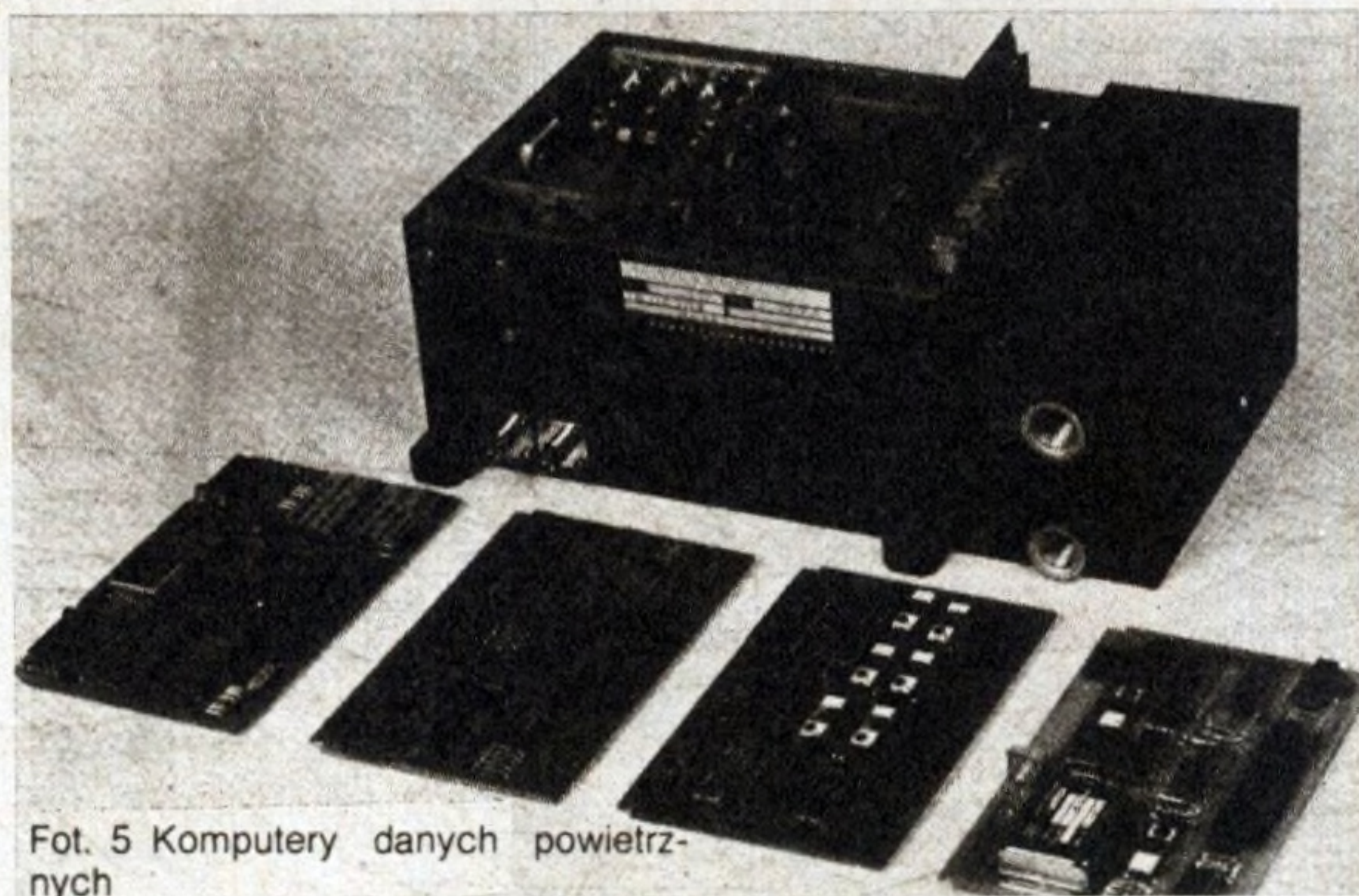
3 mln oper./s
2 Mbitów
100 W
17 dm³

**komputer nawig.-celow-
niczy bomb. B-52**

szybkość działania 400 tys. oper./s
pojemność pamięci 128 Kbitów
pobór mocy 450 W
objętość 25,2 dm³

**komputer pojedynczo-
-modułowy**

500 tys. oper./s
128 Kbitów
5 W
0,56 dm³ (pojed. karta)



Fot. 5 Komputery danych powietrznych

Trudne początki, ale...

Rozmowa z Tomaszem Okólskim — szefem klubu komputerowego przy ZW ZSMP we Wrocławiu.

● **Trudno się dodzwonić do klubu, ale za to chyba drzwi do niego są otwarte.**

— Telefonować trzeba przez centralę ZW ZSMP i na pewno uzyska się połączenie. Do niedawna rzeczywiście były trudności, gdyż organizowaliśmy zajęcia w terenie. Nikogo tu nie było. Mamy już jednak swój kąt. Drzwi, jak widać są otwarte.

● **Ale dopiero się tutaj urządza. Nie widać sprzętu.**

— To ostatnie przygotowania do otwarcia klubu komputerowego z prawdziwego zdarzenia. Jak widać pomieszczenie jest duże. Chcemy ruszyć ostro w listopadzie.

● **Co tu kiedyś było?**

— Bufet Zarządu Wojewódzkiego ZSMP.

● **Nikt nie miał do was pretensji, że zabraliście lokal na klub?**

— Przydzielono go nam i nie słyszałem żadnych głosów krytycznych pod naszym adresem. Wręcz przeciwnie. Inicjatywa z powołaniem klubu została przyjęta życzliwie. Działacze młodzieżowi pomagają nam w wielu sprawach.

● **Od kiedy faktycznie istnieje klub?**

— Od czerwca.

● **I odtąd nie ruszyliście jeszcze z miejsca?**

— Od tamtej pory działaliśmy, jak wspominałem, w terenie, prowadząc akcję pod nazwą „latający komputer”. Po prostu jeździliśmy po gminach województwa wrocławskiego ze sprzętem, propagując problematykę mikrokomputerową wśród tamtejszej młodzieży i władz terenowych. W czasie wakacji udało się nam zainteresować szerokie rzesze młodych ludzi tą dziedziną wiedzy, zachęcić do powoływania klubów.

● **I udało się?**

— Owszem. Zainteresowanie było bardzo duże. W gminach Sobótka i Kąty Wrocławskie już zakupiono sprzęt i powstały kluby. Ponadto uczestniczyliśmy w obozie młodzieżowym. Sprzęt otrzymaliśmy na ten czas z zakładów ELWRO. Również z tego zakładu oddelegowano specjalistę programowania, który prowadził zajęcia na wysokim poziomie metodycznym. Uczył programowania, obchodzenia się ze sprzętem, przekazywał wiele fachowych informacji. Ceniśmy sobie taką pomoc. Zresztą we Wrocławiu mamy wielu zwolenników. Utrzymujemy też ścisły kontakt z Elektronicznymi Zakładami Naukowymi. Najcenniejsze jest to, że

będziemy mogli korzystać z fachowego wsparcia specjalistów wysokiej klasy. Może także i sprzętu.

● **A jaki macie własny sprzęt?**

— Taki chyba sam jak pozostałe kluby wyposażone przez CB TMT. A więc trzy mikrokomputery UNIPOL-BRIT. Mamy też jeden SPECTRUM ufundowany przez ZW ZSMP. Ale obecnie jest uszkodzony i mamy kłopoty z naprawą.

● **Nie jest to wiele, tym bardziej że zainteresowanie młodzieży waszym klubem jest duże. Czym więc się będziecie zajmować na początek?**

— Przede wszystkim wymianą doświadczeń, ułatwieniem dostępu do literatury, oprogramowania dla wszystkich chętnych z całego województwa wrocławskiego. Zgłosiła się już duża grupa ze Strzelina. Mamy sygnały o zainteresowaniu grup młodzieży z wielu szkół średnich. Są także zgłoszenia indywidualne. Każdy może do nas przyjść. Jak to się mówi: prosto z ulicy.

● **Czy ci, którzy przyszli, są zadowoleni?**

— Różnie z tym bywa. Niektórzy odchodzą dość szybko, gdy dowiadują się, że nie prowadzimy gier i innych atrakcji. Po prostu traktujemy klub komputerowy nie jako salon gier, lecz

placówkę kształcącą młodych ludzi. Dlatego zostają ci, którzy traktują klub jako źródło zdobywania nowych wiadomości, poznawania tajników maszyny, programowania. Z nimi też wiążemy największe nadzieje.

● **Sądzę, że na początek macie już zarysowany jakiś program działania lub tematy, którymi zajmiecie się szczegółowo.**

— Jeszcze nie. Najpierw chcemy ludzi, a potem niech wszyscy ustalą czym powinniśmy się zajmować. Formuła jest otwarta. Na owoce naszej działalności będzie trzeba trochę poczekać.

● **Co z nazwą klubu?**

— Będzie również zależała od jego członków.

● **Jak zostałeś szefem?**

— Studiuję w Akademii Ekonomicznej i tam należę do koła naukowego informatyki. Jest to zresztą moja życiowa pasja. Właściwie przypadkowo trafiłem do tego klubu. Zaproponowano mi tę pracę, a wiadomo, że student nie pogardzi żadnym groszem. Ponadto będzie dla mnie dużą satysfakcją nauczanie innych obcowania z komputerem.

● **Czego oczekujesz od „IKS-a”?**

— Programy, które drukujecie, mnie osobiście już niewiele dają, choć są przydatne innym. Natomiast oczekuję na jeszcze więcej świeżych informacji o sprzęcie, na prezentację oprogramowania.

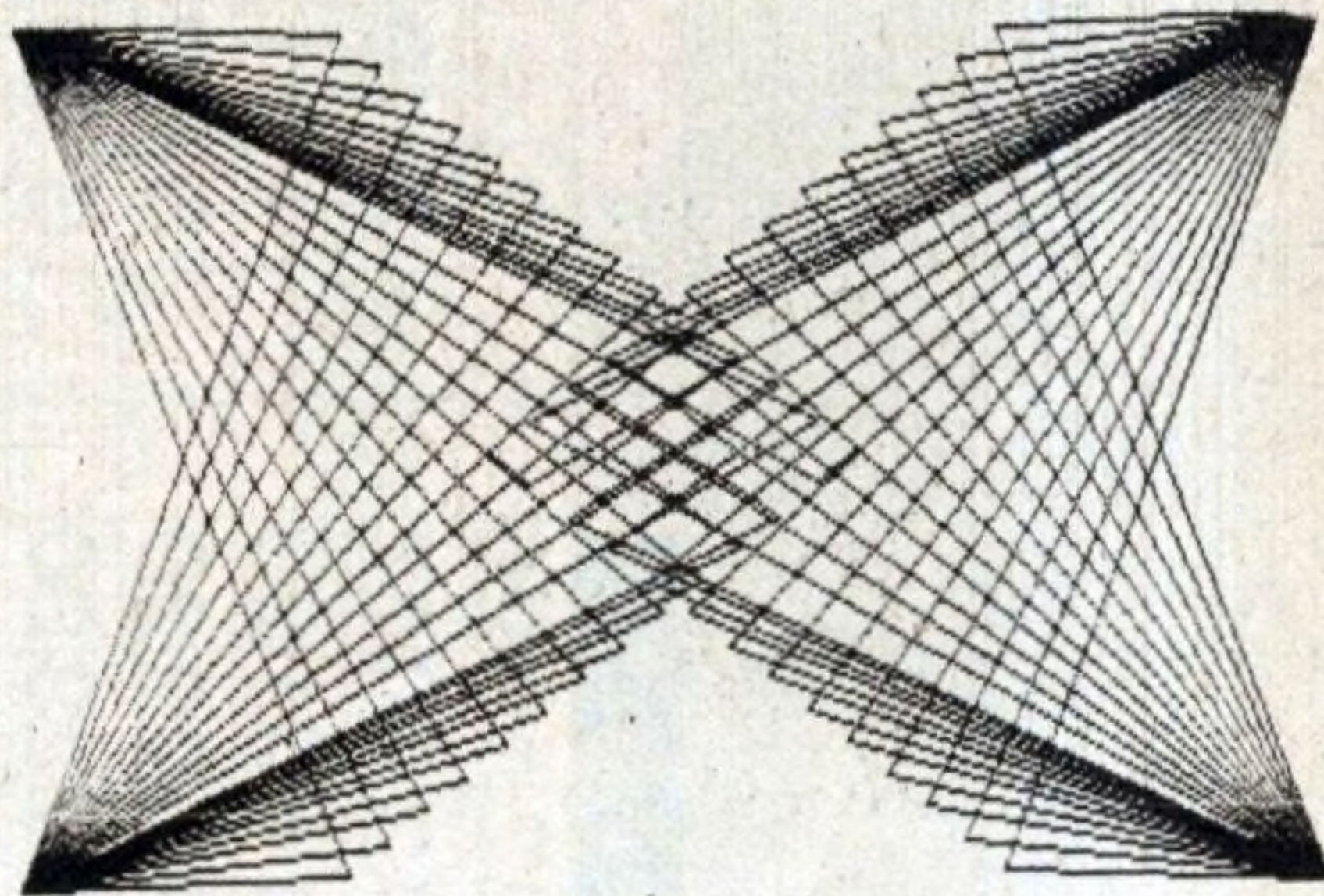
● **Mam nadzieję, że programy, które powstaną w waszym klubie trafią na łamy „IKS-a”.**

— Chętnie byśmy je tam widzieli. I skorzystamy z oferty.

● **Życzę więc udanego startu.**

rozmawiał: MAREK SIENIAWSKI

```
10 CLS
30 ORIGIN 250,200
40 FOR x=1 TO 13
60 DRAW 15*x,-15*x
65 DRAW 320,200
70 DRAW 15*x,15*x
80 DRAW 320,-200
90 NEXT x
100 ORIGIN 390,200
110 FOR y=1 TO 13
120 DRAW -15*y,-15*y
130 DRAW -320,200
150 DRAW -15*y,15*y
160 DRAW -320,-200
170 NEXT y
```



GRAFIKA (AMSTRAD)

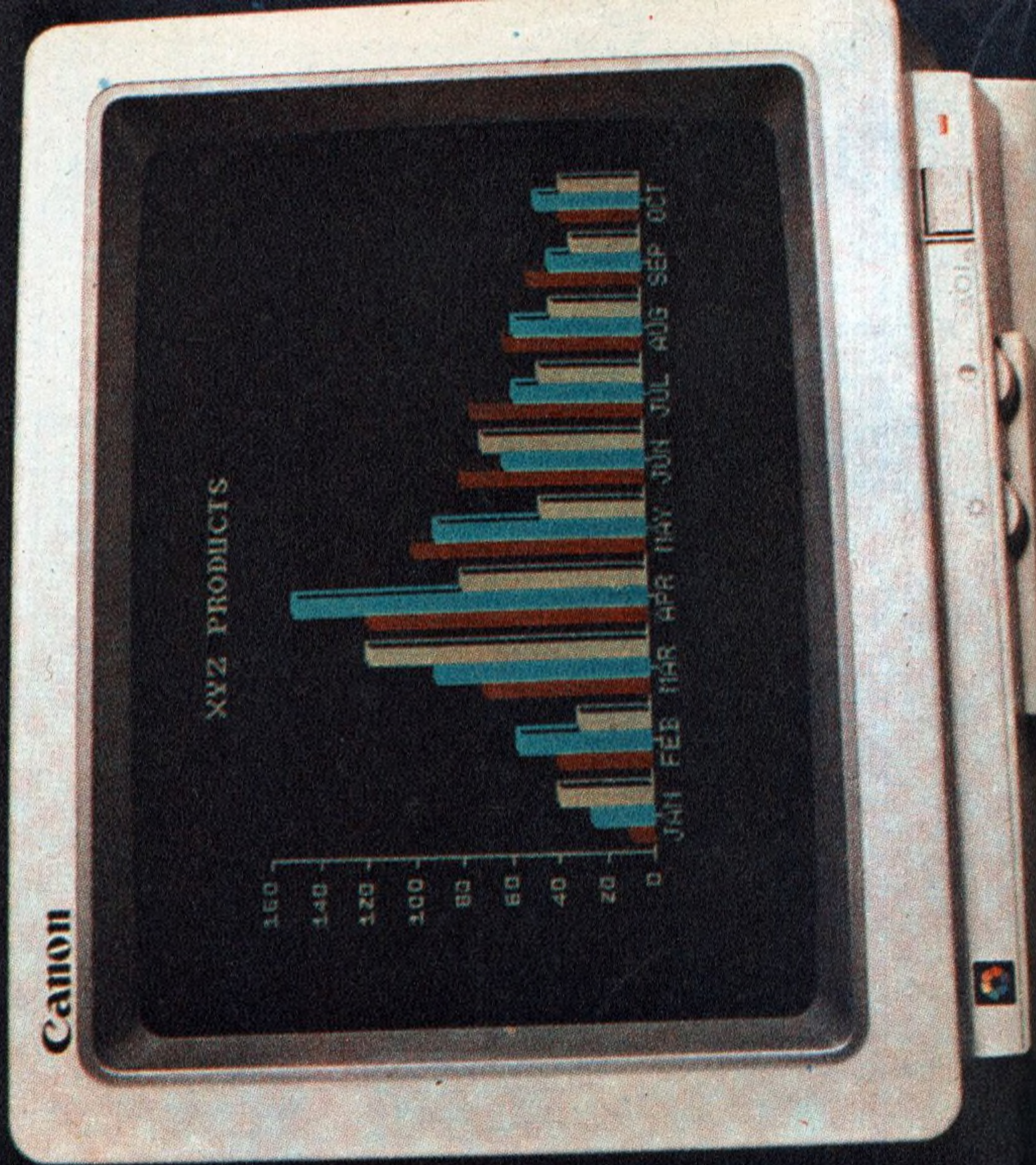
IKS — 15

IKS 1987

		I STYCZEŃ							II LUTY						
PN		5	12	19	26				2	9	16	23			
WT		6	13	20	27				3	10	17	24			
ŚR		7	14	21	28				4	11	18	25			
CZ	1	8	15	22	29				5	12	19	26			
PT	2	9	16	23	30				6	13	20	27			
SO	3	10	17	24	31				7	14	21	28			
N	4	11	18	25				1	8	15	22				

		III MARZEC							IV KWIECIEŃ						
PN		2	9	16	23	30				6	13	20	27		
WT		3	10	17	24	31				7	14	21	28		
ŚR		4	11	18	25				1	8	15	22	29		
CZ		5	12	19	26				2	9	16	23	30		
PT		6	13	20	27				3	10	17	24			
SO		7	14	21	28				4	11	18	25			
N	1	8	15	22	29				5	12	19	26			

		V MAJ							VI CZERWIEC						
PN		4	11	18	25				1	8	15	22	29		
WT		5	12	19	26				2	9	16	23	30		
ŚR		6	13	20	27				3	10	17	24			
CZ		7	14	21	28				4	11	18	25			
PT	1	8	15	22	29				5	12	19	26			



SO	2	9	16	23	30	6	13	20	27
N	3	10	17	24	31	7	14	21	28

VII LIPIEC										VIII SIERPIEŃ									
PN		6	13	20	27	3	10	17	24	31									
WT		7	14	21	28	4	11	18	25										
ŚR	1	8	15	22	29	5	12	19	26										
CZ	2	9	16	23	30	6	13	20	27										
PT	3	10	17	24	31	7	14	21	28										
SO	4	11	18	25		1	8	15	22	29									
N	5	12	19	26		2	9	16	23	30									

IX WRZESIEŃ										X PAŹDZIERNIK									
PN		7	14	21	28					5	12	19	26						
WT	1	8	15	22	29					6	13	20	27						
ŚR	2	9	16	23	30					7	14	21	28						
CZ	3	10	17	24					1	8	15	22	29						
PT	4	11	18	25					2	9	16	23	30						
SO	5	12	19	26					3	10	17	24	31						
N	6	13	20	27					4	11	18	25							

XI LISTOPAD										XII GRUDZIEŃ									
PN		2	9	16	23	30				7	14	21	28						
WT		3	10	17	24					1	8	15	22	29					
ŚR		4	11	18	25					2	9	16	23	30					
CZ		5	12	19	26					3	10	17	24	31					
PT		6	13	20	27					4	11	18	25						
SO		7	14	21	28					5	12	19	26						
N	1	8	15	22	29					6	13	20	27						

Dean Wspaniały

ANDRZEJ MONASTYRSKI

— Dean, kochanie, wstawaj. Dochodzi szósta trzydzieści...

Melodyjny kobiecy głos napętnił pokój. Wydawało się, że płynął zewsząd. Z góry, z dołu, od okien, ścian i drzwi. Jego natężenie stopniowo rosnęło. Od szeptu przechodził do unisono, by zakończyć się w ciepłym mezzoforte.

Dean leniwie przeciągnął się na wygodnym pneumatycznym tapczanie, przyjmującym przez cały czas kształt jego ciała. Okna powoli rozjaśniały się miłym, łagodnym blaskiem, tak różnym od natrętnych promieni słonecznych. Sprawiał to specjalny system filtrów optycznych. Leżący mężczyzna powoli uchylił powieki. Nie był ostatnio w najlepszym nastroju. Chyba że dwa razy w zeszłym miesiącu wydało mu się, że nie może porozumieć się bez kłopotu z najnowszym modelem omalizomatu X 10TW95, jego podstawowym narzędziem pracy. A gdyby to zauważył przypadkiem Warner — lub któryś z jego donosicieli, ot, choćby James — Dean nie miałby czego szukać w firmie. Więc dla poprawy samopoczucia zamówił sobie ten poranny program.

Na dużym ekranie ukazała się postać pięknej dziewczyny.

— Dean, przypominam ci, że już powinienes wstać. Dziś jest czwartek, piętnastego czerwca. Temperatura wynosi 18 stopni, będzie słonecznie, w południe może być do 25 stopni.

Jedna z najatrakcyjniejszych modelek znanej firmy Simson — Cloth uśmiechała się teraz tylko dla niego, ukazując nieskazitelną biel zębów i urocze dołeczki na buzi.

— Wanna już czeka. Dziś masz kąpiel jaśminową. No, rusz się wreszcie — pogroziła figlarnie paluszkami — bo ci się grzanki przypalą i kawa wystygnie! A do tego, jak będziesz marudził, czeka cię w windzie spotkanie z Jamesem...

Dean zmarszczył brwi. Nie cierpiał tego ulizanego cwaniaczka. Takiego, który zawsze i wszędzie musi być pierwszy, musi ze swym przypochebnym uśmiechem wejść od rana w oczy bossowi, przymilić się sekretarce, opowiedzieć najnowszy dowcip współpasażerom w windzie.

— Stary karierowicz — mruknął pod nosem.

— To stary karierowicz, obrzydliwy typ — powtórzyła jak echo złotoloka Miriam z ekranu. — Nie daj mu się dziś prześcignąć. Musisz być lepszy. Będziesz lepszy...

Dean Robinson podniósł się wreszcie z postania. Podeszedł do niewielkiej klawiatury i przebiegł palcami po przy-

ciskach. Zaprogramował sobie cały poranek. Miriam ponowiła zaproszenie do łazienki z wyjątkowo kuszącym uśmiechem.

— Dean, skarbie, pośpiesz się. Czekam...

Wszedł do sporego seledynowo-błękitnego pomieszczenia. Na ekranie zajmującym pół ściany widniała już Miriam. Stała pod prysznicem, a deszcz srebrzystych kropelek rozpryskiwał się dokoła.

— No, nie gap się tak — postyszał jej lekko strofujący głos — jakbyś mnie w życiu nie widział. Nie marudź, czas ucieka.

Odwrocił oczy, nieco skonfudowany. Wydawało mu się, że to realna Miriam, a nie jej ekranowa postać. Tymczasem ze ścian, sufitu i podłogi co chwila wysuwały się kolejne przyrządy gimnastyczne. Drażki, poręczce, kółka, sprężyny, worki treningowe. Robinson każdemu z nich poświęcał chwilę uwagi, by przejść do następnego. Kazał je zainstalować, gdy stwierdził, że mięśnie mu wiotczeją, brzuch zaczyna się zao kręlać, a sylwetka traci swój poprawny wygląd.

Wreszcie zmęczony wszedł do pachnącej kąpieli, poddając się łagodnemu podwodnemu masażowi. Odświeżony zabrał się z apetytem do śniadania. W tym czasie Miriam zabiwała go z ekranu.

— W dzisiejszych gazetach nie ma nic ciekawego. Notowania giełdy bez specjalnych zmian. W Europie spokój. W Meksyku wybuchł jakiś wulkan. Uważaj, nie poplam spodni kawą! Boss nie cierpi najmniejszej plamki na ubraniach pracowników, pamiętasz?

— Stary pedant — skrzywił się Dean.

— Stary, głupawy pedant — zaszczebotała Miriam — ale od niego wszystko zależy. I dzisiejszy dzień i twoja pozycja w pracy i nadzieja na awans. Więc nie marudź i uważaj. Zostaw już te kanapki! Nie zauważyłeś, że ci przybyło na wadze? A boss nie znosi otyłych. Sam chudy jak szczapa nie cierpi grubasów.

Dean cofnął rękę. Szkoda mu było świeżutkich grzanek zaserwowanych przez automat, ale musiał pohamować apetyt. To prawda, Ted Warner bez

skrupułów pozbywał się każdego, kto przekroczył 70 kilogramów. I diabli wiedzieli, skąd brał te informacje? Podobno Ted kazał umieścić wagę w wycieraczkę, ale nikt tego nie potrafił sprawdzić. Za to w firmie pracowali wyłącznie szczupli, wysportowani panowie, jednakowo ubrani i jednakowo pachnący lawendą. Wyłącznie lawendą.

Robinson poczuł narastającą złość i niechęć do pracy, szefa i świata. Codziennie ten sam rytuał, ten sam strój, ten sam zapach. Ale boss był twardy. Wymagał, płacił i nie tolerował najmniejszych odstępstw od tego swojego regulaminu.

— Gorzej, jak w wojsku — warknął. — Tam mogłem odpiąć guzik, podwinąć rękawy, rozluźnić pas, a nawet wylać sobie na twój garniec jakichkolwiek perfum i nikt nie miał pretensji...

— Nie marudź, Dean — postyszał znów strofujący głos. — Warner jest pedantyczny, ale dobrze płaci. Inni bossowie bywają gorsi.

— Racja — pomyślał.

— Kończ śniadanie. Pośpiesz się. Masz dziś szansę. Ted wkrótce będzie konferował z facetami z Chemical Industry. Będzie chyba chciał mieć tam swego człowieka. Może właśnie ciebie tam wsadzi? Zrób dziś na nim dobre wrażenie. Bądź miły, nie daj się wyprzedzić Jamesowi. Zabłyśnij czymś, postaraj się. Zrób coś, co go do ciebie przekona. No, pomyśl o czymś takim, zadziałaj dziś mądrze...

— Głupia, wydekoltowana papugo — warknął Dean. — Sam dobrze to wszystko wiem. Powtarzasz mi ciągle tylko to, co ci kazałem nagrać. A bądź taka mądra i powiedz wreszcie jak i czym tu błysnąć?

— Postaraj się Dean. Kochany, zrób coś. Przecież jesteś inteligentny, mądry, pracowity. Może być trudny dzień, ale łatwo dasz sobie radę. Tylko nie zapomnij zagłuszyć ten jaśmin lawendą!

— Jesteś głupia gęś! Stale gęgasz to samo! Mam już dosyć twoich wspaniałych rad! Chyba będę musiał ciebie wymienić. O, już wiem! Zamówię sobie Kate. Ona jest lepsza. Ma zgrabniejsze nogi, wiesz?

dokończenie na str. 23



Rys. M. Przybyłowski

PASCAL (2)

Materiał zaprezentowany w części pierwszej został tak dobrany, aby po jego opanowaniu możliwe było pisanie i uruchamianie prostych programów. W kolejnych częściach, na niezbyt skomplikowanych przykładach, będziemy omawiali wybrane elementy języka. W tej części zajmiemy się w głównej mierze instrukcjami umożliwiającymi zapisywanie w języku PASCAL czynności wykonywanych powtarzalnie. Zgodnie z zapowiedzią zamieszczamy również opis komend edytora, niezbędnych do poprawiania programów źródłowych oraz zadania do samodzielnego rozwiązania.

1. Instrukcje iteracyjne (powtarzania).

Rozpatrzmy następujący problem:

„Wyznaczyć maksymalną liczbę spośród zadanych N liczb rzeczywistych”.

Najprostszym rozwiązaniem jest uznanie pierwszej liczby za maksymalną i i porównanie jej z liczbą następną. Liczbę większą, z porównywanych, traktuje się jako maksymalną i i porównuje ją z kolejną liczbą. Zatem po $N-1$ porównaniach uzyskujemy liczbę największą spośród zadanych. Program wykonujący przedstawione zadanie przedstawiono w przykładzie 5.

```
PROGRAM prog5;
VAR n,
    i:integer;
    x,
    max:real;
    (ilosc liczb rzeczywistych)
    (licznik powtorzen)
    (kolejna liczba rzeczywista)
    (liczba maksymalna)
BEGIN
  READLN(n);
  i:=1;
  READLN(x);
  max:=x;
  WHILE i <= n-1 DO
  BEGIN
    READLN(x);
    i:=i+1;
    IF max < x THEN max:=x;
  END;
  WRITELN;
  WRITELN('maksymalna liczba spozrod wprowadzonych to ',max);
END.
```

Do opisu powtarzających się czynności wykorzystano znaną już instrukcję WHILE. Nietrudno zauważyć, że w tym wypadku stosowanie tej instrukcji nie jest zbyt wygodne. Konieczne jest bowiem ustalenie liczby powtórzeń operacji wykonanych zgodnie z instrukcją WHILE oraz określenie warunku kończącego jej wykonanie. Warunek powinien być spełniony wtedy, gdy osiągnięto wymaganą liczbę powtórzeń. Nieprecyzyjne określenie warunku prowadzi do niepoprawnej liczby powtórzeń (najczęściej o 1 mniejsza lub o 1 większa od wymaganej).

Bardziej naturalną konstrukcją do opisu czynności powtarzanych określoną ilość razy jest instrukcja o postaci:

```
FOR i:=e1 TO e2 DO
  instr
```

gdzie:

- i — zmienna kontrolna (licznik powtórzeń),
- e_1 — początkowa wartość zmiennej kontrolnej,
- e_2 — końcowa wartość zmiennej kontrolnej,
- instr — dowolna instrukcja.

Zapis ten oznacza, że instrukcja „instr” wykonywana jest dla kolejnych wartości zmiennej kontrolnej „ i ” od wartości początkowej „ e_1 ” z krokiem 1 do wartości końcowej „ e_2 ”. Wartość zmiennej kontrolnej jest zmieniana automatycznie, zatem zmiana taka nie jest dopuszczalna w ramach instrukcji „instr”. Zmienna kontrolna może być natomiast wykorzystywana jako argument w wyrażeniach, jeżeli takie występują w instrukcji „instr”. Przyjmuje się, że po wykonaniu instrukcji FOR wartość zmiennej kontrolnej jest określona.

Program wyznaczania maksymalnej liczby spośród zadanych N liczb rzeczywistych z wykorzystaniem omówionej instrukcji FOR zamieszczono w przykładzie 6.

```
PROGRAM prog6;
VAR n,
    i:integer;
    x,
    max:real;
    (ilosc liczb rzeczywistych)
    (zmienna kontrolna)
    (kolejna liczba rzeczywista)
    (liczba maksymalna)
BEGIN
  READLN(n);
  READLN(x);
  max:=x;
  FOR i:=1 TO n-1 DO
  BEGIN
    READLN(x);
    IF max < x THEN max:=x;
  END;
  WRITELN;
  WRITELN('maksymalna liczba spozrod wprowadzonych to ',max);
END.
```

Program „prog. 5” i „prog. 6” uruchomiono dla $N = 7$ oraz liczb rzeczywistych 2,57; — 12; 26,48; — 1,31; 5,126 i 7,33 uzyskując identyczny wynik o postaci:

**maksymalna liczba spozrod
wprowadzonych to 2.6480000000E+01**

W obu programach wykorzystano instrukcję warunkową IF warunek THEN instr

gdzie:

- warunek — wyrażenie logiczne,
- instr — dowolna instrukcja.

Jeżeli warunek jest spełniony, to zostaje wykonana instrukcja „instr”, a jej zakończenie jest uważane za zrealizowanie całej instrukcji warunkowej. W przypadku przeciwnym instrukcja „instr” nie jest wykonywana i uważa się, że cała instrukcja warunkowa została zrealizowana.

Po tym koniecznym wyjaśnieniu powróćmy ponownie do własności języka PASCAL w zakresie opisu czynności wykonywanych powtarzalnie.

Rozważmy problem obliczania wartości szeregu

$$1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \dots$$

z zadaną dokładnością. Można go rozwiązać przez dodawanie do siebie kolejno obliczanych wartości wyrazów szeregu. Należy przy tym zauważyć, że:

— wartości kolejnych wyrazów szeregu można wyznaczyć z zależności

$$a_{n+1} = a_n \cdot \frac{X}{n+1}$$

przyjmując, że do $a_0 = 1$ (a_n — n -ty wyraz szeregu), — o dokładności obliczeń decyduje wartość ostatniego obliczonego wyrazu szeregu.

Czytelnicy, którzy rozwiązali zadanie nr 3, bez trudu napiszą program obliczania wartości podanego szeregu, wykorzystując znaną już instrukcję WHILE.

dokończenie na str. 21

AMSTRAD

Amstrad PC1512

W Londynie, we wrześniu tego roku, firma AMSTRAD zaprezentowała swój najnowszy model, komputer osobisty PC 1512, kompatybilny z IBM PC. Porównując PC 1512 z niewiele starszymi, dobrze znanymi w naszym kraju, modelami CPC 6128 i JOYCE PCW 8256, należy stwierdzić, że to duży krok do przodu. Można też sądzić, iż model ten będzie cieszyć się olbrzymią popularnością nie tylko w W. Brytanii.

W AMSTRADZIE PC 1512 zastosowano 16-bitowy mikroprocesor 8086, pracujący z częstotliwością zegara 8 MHz. Mikroprocesor adresuje 512 KB pamięci RAM, którą można zwiększyć o 128 KB. Szybkość działania mikroprocesora może być zwiększona po dołączeniu do niego koprocessora arytmetycznego 8087, dla którego wykonano miejsce w płycie systemu. Nie wykonano natomiast złączek do rozszerzenia pamięci operacyjnej ponad pojemność 640 KB. W systemie zastosowano kwarcowy zegar czasu rzeczywistego z baterią, działającą po wyłączeniu zasilania.

PC 1512 przetłumaczył lansowany dotąd przez firmę AMSTRAD standard dyskietek 3-calowych. W modelu tym zastosowano dwustronne mechanizmy dyskowe, pozwalające na dyskach elastycznych 5 1/4 cala osiągnąć gęstość zapisu 360 KB, dokładniej 368 640 bajtów. Jako pamięć masową zastosowano również dysk stały o pojemności 10 MB lub 20 MB, co jest nowością w modelach firmy.

AMSTRAD PC jest wyposażony w interfejs szeregowy RS 232 C i równoległy typu Centronics. Ten drugi służyć może do podłączenia bardzo dobrych drukarek matrycowych DMP 2000 lub DMP 3000. Trzy zastosowane złączki umożliwiają dołączenie do konfiguracji podstawowej wspomnianego dysku stałego, modemu, przetwornika a/c lub c/a, dodatkowego portu dla drukarki. Istnieje również złączka dla pióra świetlnego.

PC 1512 ma klawiaturę typu QWERTY z 85 klawiszami, która ma wbudowaną złączkę dla drążka sterowego. W skład wyposażenia podstawowego wchodzi również „myszka”.

Z najnowszym AMSTRADEM mogą pracować monitor kolorowy PC CM lub

monochromatyczny PC MM. Pozwalają one na organizację ekranu w standardzie dużej rozdzielczości (high resolution) w modzie graficznym: w 16 lub



2 kolorach z 640 x 200 punktami bezpośrednio adresowanymi.

W standardzie średniej rozdzielczości (medium resolution) mogą być wykorzystywane 3 palety czterokolorowe z 320 x 200 punktami bezpośrednio adresowanymi. W modzie alfanumerycznym organizacja ekranu może być następująca: 16 kolorów w 25 liniach po 80 znaków lub 25 liniach po 40 znaków.

Waga poszczególnych elementów systemu AMSTRAD PC 1512 jest następująca: jednostka centralna z dwiema stacjami dysków 7,75 kg; klawiatura i, 17 kg; PC CM 11,6 kg; PC MM 7,4 kg; DMP 3000 4,2 kg.

Przedstawione parametry AMSTRAD-a PC nie są rewelacyjne, niemniej czynią go atrakcyjnym na rynku europejskim, tym bardziej że upodobać go do IBM PC/XT.

Teraz krótka charakterystyka oprogramowania. W PC 1512 zastosowano dyskowy system operacyjny wersję MS DOS3.20 oraz DOS Plus firmy DIGITAL RESEARCH. Firma ta opracowała również GEM, ułatwiający pracę w wyżej wymienionych modach graficznych. Firma LOCOMOTIVE SOFTWARE opracowała nową wersję języka Basic — BASIC 2.

W pamięci stałej ROM AMSTRADA PC umieszczono ROM BIOS (Basic Input/ Output System), pracujący bezpośrednio ze wszystkimi układami elektronicznymi, w tym peryferyjnymi, stanowiącymi system PC 1512.



Wśród programów użytkowych, oferowanych przez firmę a znanych na pewno dobrze użytkownikom IBM-ów, są: program do redagowania tekstów WordStar 1512; program dla dyrektorów i ekonomistów SuperCalc 3, ze swoimi pięć chartami i bar chartami, oraz inne, w tym: LOTUS 1-2-3; SIDE-KICK; PFS: FILE; DBMS; REFLEX.

Nie zapomniano o miłośnikach gier. Przygotowano dla nich szachy, golf, wyścigi samochodowe, zawody zimowe.

Na koniec ceny:

— 1 x360 KB, PC MM	£ 399;
— 2 x360 KB, PC MM	£ 499;
— 1 x360 KB, PC CM	£ 599;
— 2 x360 KB, PC CM	£ 699;
— 1 x360 KB, 10 MB, PC MM	£ 699;
— 1 x360 KB, 10 MB, PC CM	£ 849;
— 1 x360 KB, 20 MB, PC MM	£ 799;
— 1 x360 KB, 20 MB, PC CM	£ 949.

Rozszerzenie RAM o 128 KB zwiększa cenę o £44.

Jacek WOJTALA

Program, dający w wyniku wykonania taki sam efekt, zamieszczono w przykładzie 7. Zamiast instrukcji WHILE, do opisu czynności wykonywanych powtarzalnie wykorzystano instrukcję REPEAT o postaci:

```

REPEAT
  i1
  i2
  .
  .
  in
UNTIL wyrlog
    
```

gdzie: i_1, i_2, \dots, i_n — dowolne instrukcje,
wyrlog — wyrażenie logiczne

Zapis ten oznacza, że ciąg instrukcji zawarty pomiędzy słowami kluczowymi REPEAT—UNTIL wykonywany jest tak długo (przynajmniej raz — porównaj z instrukcją WHILE), aż wartość wyrażenia logicznego „wyrlog” będzie równa TRUE. Należy przy tym zdawać sobie sprawę z tego, że jeżeli w żadnym z powtórzeń wykonania ciągu instrukcji wyrażenie logiczne „wyrlog” nie osiągnie wartości TRUE, to wykonanie instrukcji REPEAT nigdy się nie zakończy.

```

PROGRAM prog7;
VAR n:integer;    {numer wyrazu szeregu}
    s,            {wartosc szeregu}
    p,            {wartosc kolejnego wyrazu}
    eps,          {dokladnosc obliczen}
    x:real;
BEGIN
  s:=0;
  p:=1;
  n:=0;
  READ(x,eps);
  REPEAT
    s:=s+p;
    n:=n+1;
    p:=p*x/(n+1)
  UNTIL ABS(p) <= eps;
  WRITELN;
  WRITELN('wartosc szeregu dla x =',x);
  WRITELN('obliczona z dokladnoscia eps =',eps);
  WRITELN('wynosi s =',s)
END.
    
```

Program „Prog 7” wykonano dla $X = 1$ i $\text{eps} = 10^{-10}$ (eps można wprowadzić w postaci $1e-10$ lub 0.0000000001) uzyskując wynik:

```

wartosc szeregu dla x = 1.0000000000E+00
obliczona z dokladnoscia eps = 1.0000000000E-10
wynosi s = 1.7182818284E+00
    
```

W programie wykorzystano standardową funkcję arytmetyczną ABS, obliczającą wartość absolutną argumentu „P” („P” zależy od „X”, a więc może przyjmować wartości mniejsze od zera). W języku PASCAL dostępne są jeszcze inne standardowe funkcje, np. obliczające wartość sinusa kąta czy logarytmu naturalnego z danej liczby. Funkcje standardowe zostaną omówione w jednej z kolejnych części.

Aby zamknąć listę instrukcji iteracyjnych języka PASCAL, opiszemy drugą postać instrukcji FOR (FOR—DOWNTO)

Jest ona następująca:

```

FOR i := e1 DOWNTO e2 DO
  instr
    
```

gdzie:

$i, e_1, e_2, \text{instr}$ — mają identyczne znaczenie, jak w już opisanej instrukcji FOR (FOR—TO).

Zapis ten oznacza, że instrukcja „instr” wykonywana jest dla kolejnych wartości zmiennej kontrolnej „i” od wartości początkowej „ e_1 ”, przy czym wartość zmiennej kontrolnej jest zmniejszona o 1 (a nie zwiększona, jak w instrukcji FOR—TO) po każdym wykonaniu instrukcji „instr”. Przykłady wykorzystania instrukcji FOR—DOWNTO podamy w odpowiednim czasie. Biorąc pod uwagę dotychczas poznane własności języka PASCAL, trudno jest znaleźć problem, w rozwiązaniu którego naturalne byłoby zastosowanie instrukcji FOR—DOWNTO.

2. Wybrane komendy edytora

Pierwsze próby uruchomienia programu z reguły kończą się niepowodzeniem. Przyczyny są następujące:

- w programie źródłowym wystąpił zapis niezgodny z regułami języka PASCAL (sygnalizowany jest błąd komplikacji),
- podczas wykonania programu wystąpił błąd, np. próba wczytania danej o postaci niedopuszczalnej dla typu zmiennej, której ma być przypisana (sygnalizowany jest błąd wykonania),
- w programie wystąpiły błędy logiczne (program wykonał się, lecz wyniki nie są zgodne z oczekiwanymi). W każdym z tych wypadków należy wywołać edytor przez wciśnięcie klawisza:
- ESC — po sygnalizacji błędu komplikacji lub wykonania; na ekranie monitora zostanie wyświetlony program źródłowy, a kursor wskaże miejsce, w którym wystąpił błąd,
- E — gdy stwierdziliśmy błędy logiczne; na ekranie monitora zostanie wyświetlony program źródłowy, a kursor będzie wskazywał pierwszy znak w pierwszej linii programu.

Wykorzystując komendy edytora programu można poprawić i uruchomić go ponownie. Edytor dostarcza 45 komend, z których opiszemy na razie kilka, naszym zdaniem, niezbędnych.

Komendy edytora składają się z jednego lub dwóch znaków. Będziemy je zapisywać symbolicznie w następujący sposób:

- <CTRL> — znak 1 (w przypadku komend jednoznakowych),
- <CTRL> — znak 1 — znak 2 (w przypadku komend dwuznakowych).

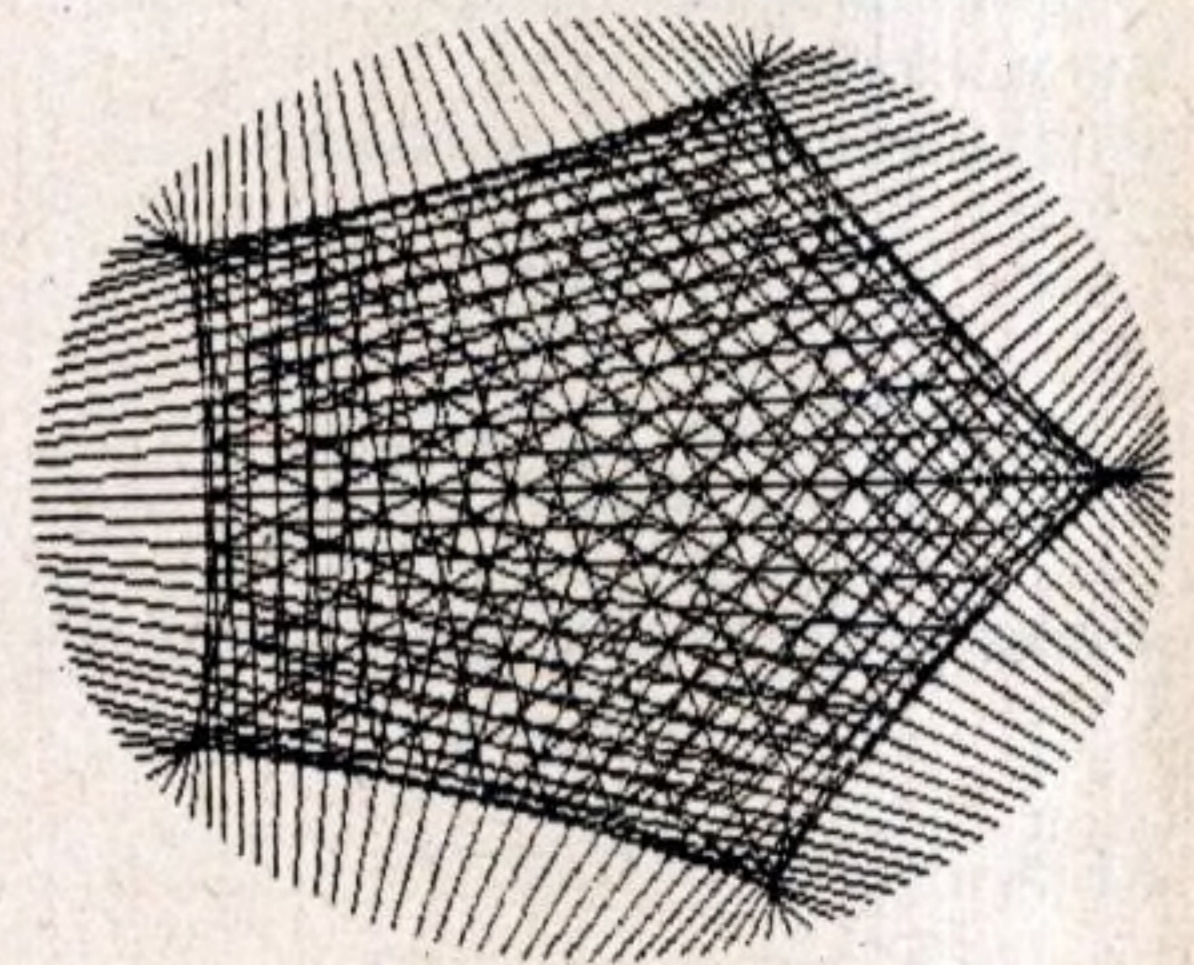
Zapisy te oznaczają, że przy wciśniętym klawiszu CONTROL należy nacisnąć klawisze ze znakiem „znak 1”, a dla komend dwuznakowych, należy nacisnąć następnie klawisz ze znakiem „znak 2”.

Poprawienie programu może polegać m. in. na:

- wstawieniu nowych znaków w wierszu,

dokończenie na str. 22

**GRAFIKA
(AMSTRAD)**



- skasowaniu określonych znaków w wierszu,
- wstawieniu nowego wiersza do programu,
- skasowaniu określonego wiersza programu.

Edytor ma dwa tryby wstawiania znaków w wierszu: INSERT oraz OVERWRITE. W trybie INSERT nowy znak zostaje wprowadzony na pozycję kursora, a pozostała część wiersza (od kursora w prawo do końca wiersza) jest przesuwana w prawo. W trybie OVERWRITE nowy znak zostaje wprowadzony na miejsce znaku, wskazywanego przez kursor. Zmianę trybu wstawienia znaków powoduje komenda <CTRL> — O—V. Do skasowania znaku znajdującego się przed kursorem wykorzystywany jest klawisz DEL. Do przesunięcia kursora można wykorzystać jedną z komend:

<CTRL> — S — przesunięcie kursora w prawo o jeden znak,

<CTRL> — D — przesunięcie kursora w lewo o jeden znak,

<CTRL> — E — przesunięcie kursora o jeden wiersz w górę,

<CTRL> — X — przesunięcie kursora o jeden wiersz w dół,

<CTRL> — W — tzw. przewinięcie w górę (przesunięcie tekstu wyświetlonego na ekranie o jeden wiersz w dół),

<CTRL> — Z — tzw. przewinięcie w dół (przesunięcie tekstu wyświetlonego na ekranie o jeden wiersz w górę).

Wstawienie nowego wiersza umożliwia komenda <CTRL> — N. Powoduje ona wstawienie nowego wiersza za wierszem, w którym znajduje się kursor. Znaki, od pozycji kursora w prawo do końca wiersza, zostają przeniesione do nowego wiersza. Pozycja kursora się nie zmienia. Do kasowania wiersza, w którym znajduje się kursor, służy komenda <CTRL> — Y. Po skasowaniu wiersza tekst znajdujący się poniżej kasowanego wiersza jest przesuwany o jeden wiersz w górę. Kursor przesuwa się na pozycję pierwszego znaku. Przy stosowaniu tej komendy należy zachować ostrożność, gdyż nie ma możliwości odzyskania skasowanego wiersza.

3. Zadania

Zadanie 4

Napisać program obliczający wartość funkcji

$$F(k) = K!$$

dla zadanej liczby naturalnej K.

Zadanie 5

Napisać program, który wczytuje dwie liczby całkowite i sprawdza, czy są one względnie pierwsze. Dwie liczby są względnie pierwsze, jeżeli nie mają wspólnych dzielników różnych od 1 i -1.

Zadanie 6

Napisać program obliczający cosinus kąta między dwoma wektorami $\{a_i\}$ i $\{b_i\}$ o n współrzędnych z zależności:

$$\cos \alpha = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2 \cdot \sum_{i=1}^n b_i^2}}$$

Uwaga: współrzędne wektorów wczytywać parami

$$a_i, b_i \quad (i = 1, 2, \dots, n).$$

Stefan ROZMUS

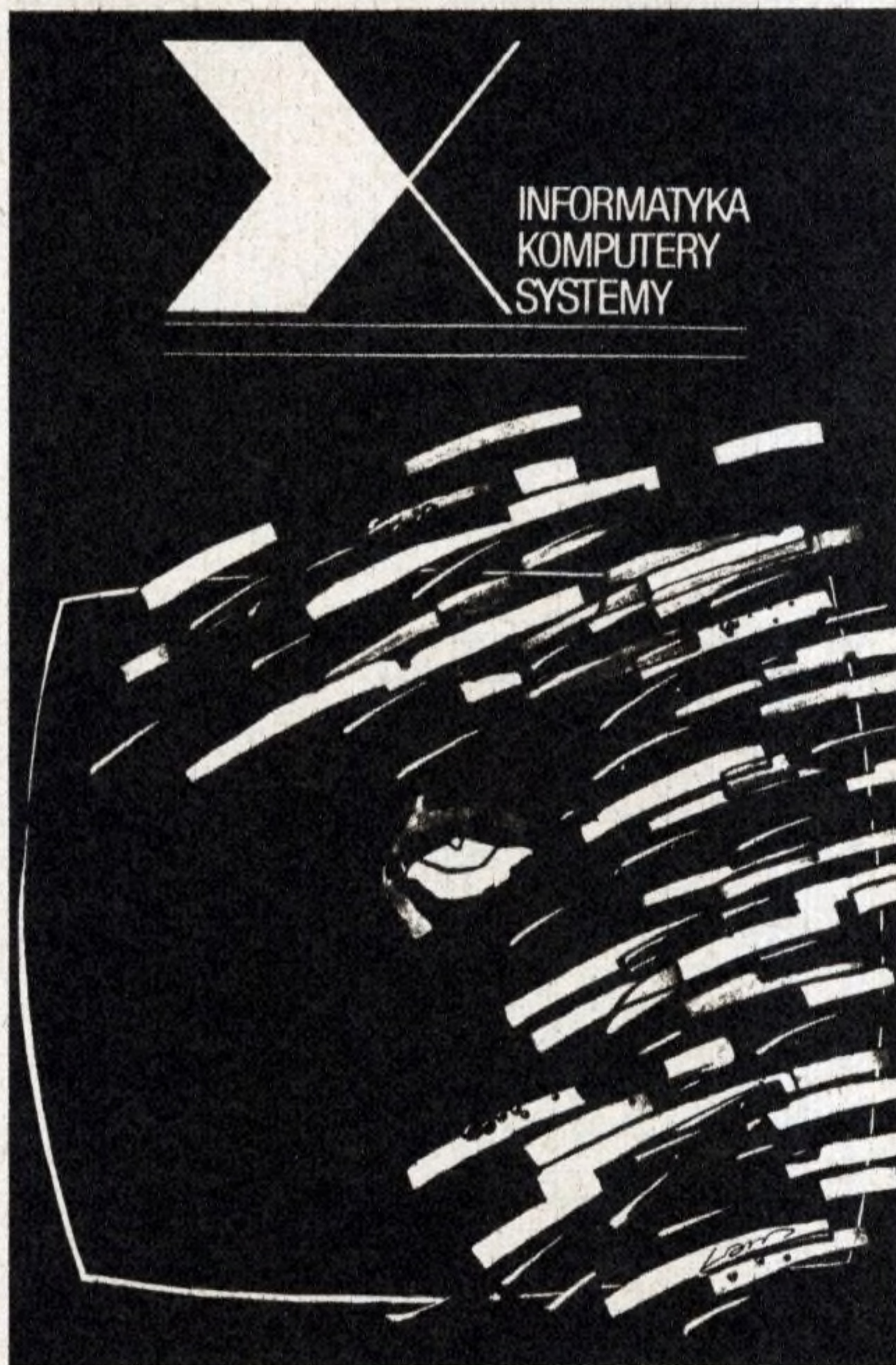
NATIONAL SEMICONDUCTOR 32332

Jest to procesor pracujący z częstotliwością zegara 10,12 lub 15 MHz. Magistrala danych może być zaprojektowana jako 8-, 16- lub 32-bitowa. Mikroprocesor 32332 adresuje 16-Megabajtową przestrzeń adresową, która po dodaniu 32-bitowego rejestru może być zwiększona do 4 Gigabajtów.

CP/M — 8000

Firmy DIGITAL RESEARCH i ZILOG opracowały system operacyjny CP/M dla 16-bitowego mikroprocesora Z-8000 (i jego kolejnych wersji Z-8001, Z-8002, Z-8003, Z-8004).

System wymaga pamięci operacyjnej o pojemności minimum 176 KB, zalecana pojemność 256 KB. CP/M — 8000 działa z kompilatorami języków: BASIC, C, FORTRAN, PASCAL z ASSEMBLEREM i różnymi programami użytkowymi. System może być umieszczony na 8-calowych dyskach elastycznych SS/SD lub 5 1/4 calowych dyskach DS/DD.



„IKS” od stycznia w nowej szacie...

„Zegar” na MERITUM

„Zegar” to podprogram obsługi przerwania NMI mikroprocesora Z-80. Umożliwia stały odczyt czasu podczas pracy z mikrokomputerem.

Wadą programu jest konieczność zatrzymywania „Zegara” przed zapisem i odczytem programów z taśmy. „Zegar” zatrzymuje się instrukcją OUT 250,0 i ponownie uruchamia instrukcją OUT 250,0. Czas uaktualnia się poprzez wpisywanie do odpowiednich komórek pamięci wartości: godzin, minut i sekund. Przykład zapisu nowego czasu do zegara

POKE 16500,12' sekundy
POKE 16501,59' minuty
POKE 16502,10' godziny

Wartości adresów są stałe. Operacje związane z zatrzymaniem, uruchomieniem i zmianą czasu w zegarze wykonuje się w trybie kalkulatorowym. W przypadku wyzerowania systemu należy kolejno wykonać instrukcje:

POKE 16526,17: POKE 16527,126: PRINT USR(0)

Spowodują one wykonanie programu maszynowego (od adresu 32273), który wznowi pracę „Zegara”. Po wpisaniu i uruchomieniu programu można wykonać rozkaz NEW.

PIOTR BIEDRZYCKI
LVI LO — Warszawa

```
8000 FOR A=32144 TO 32297
8005 READ A%
8010 N=ASC(LEFT$(A%,1))-48
8015 N=N-INT(N/12)*7
8020 M=ASC(RIGHT$(A%,1))-48
```

```
8025 M=M-INT(M/12)*7
8030 POKE A,N*16+M
8035 NEXT A
8040 ' KOD MASZYNOWY ZEGARA
8045 DATA F5,3E,B0,D3,FB,3E,C2,D3
8050 DATA FA,3E,D8,D3,FA,E5,C5,CD
8055 DATA AA,7D,CD,DC,7D,C1,E1,F1
8060 DATA ED,45,21,73,40,34,3E,1E
8065 DATA BE,C0,36,00,3C,D3,FE,3D
8070 DATA D3,FE,23,34,3E,3C,BE,C0
8075 DATA 36,00,23,34,BE,C0,36,00
8080 DATA 3E,FF,06,35,D3,FE,10,FE
8085 DATA 3D,20,F7,3E,18,23,34,BE
8090 DATA C0,36,00,C9,D5,11,74,40
8095 DATA 21,3F,3C,06,03,CD,F4,7D
8100 DATA 2B,36,3A,2B,13,10,F6,23
8105 DATA 36,20,D1,C9,1A,CD,05,7E
8110 DATA E6,0F,F6,30,77,79,E6,0F
8115 DATA F6,30,2B,77,C9,0E,FF,0C
8120 DATA D6,0A,F2,07,7E,A7,C6,0A
8125 DATA C9,00,21,90,7D,22,10,40
8130 DATA 3E,0F,D3,F7,3E,B0,D3,FB
8135 DATA 3E,00,D3,FA,D3,FA,00,C3
8140 DATA CC,06
8145 CLS:PRINT@ 520," ";
8150 INPUT"PODAJ GODZ ,MIN ,SEK "
      IG,M,S
8155 POKE 16500,S:POKE 16501,M :
      POKE 16502,G
8160 POKE 32294,201:
      POKE 16526,17:POKE 16527,126
      :U=USR(0) :POKE 32294,0
8165 GOTO 8165
```

dokończenie ze str. 18

Ale Miriam nie reagowała i kontynuowała swój monolog.

— Dean, jesteś znakomity! Doskonale dziś wyglądasz! I nie przejmuj się, że masz trudny dzień. Jesteś w znakomitej formie, widziałam to podczas ćwiczeń. Wykonałeś je lepiej, niż wcześniej, daję słowo. Zastąpiłeś na dużą buźkę! Ale już zbieraj się, bo najwyższy czas.

W odpowiedzi wzruszył ramionami. Widział w łazienkowym lustrze swe odbicie, a i ćwiczenia wcale nie poszły mu najlepiej.

— Pamiętaj — ciągnęła Miriam — wkrótce stary Bob Andersson zrezygnuje z roboty. Może właśnie dziś Tedowi przyjdzie myśl, żeby cię dać na jego miejsce? Zarządzałbyś całym pierwszym zespołem analiz magazynu-

nowych! To by było wspaniałe! Dean, mógłbyś nareszcie zmienić mieszkanie, kupić nowy samochód, ożenić się... No, chłopie, nie zmarnuj szansy! Dean, jesteś jednym z najinteligentniejszych facetów spośród wszystkich, jakich znam...

— Kretynko! — wrzasnął dziko Dean. — Zamknij ten swój uśmiechnięty dziób! Przestań powtarzać mi do znudzenia to wszystko!

Cholera wie, czy stary Bob tak szybko odejdzie na emeryturę! Cholera wie, czy Ted zaproponuje mi jego stanowisko, czy nie wybierze tego gnojka Jamesa! Cholera wie, czy zechce mnie ulokować w Chemical Industry! Cholera wie, czy to wszystko prawda, co sobie wymyśliłem! A teraz ty mi to w kółko powtarzasz!

Uśmiechnięta twarz Miriam ani na moment nie zmieniła swego wyrazu.

Kwadrofonia nadal przekazywała jej ciepły, łagodny głos.

— Dean, skarbie, więcej pewności siebie. Dziś bez wątplenia dokonasz czegoś, czym zapewnisz sobie przychylność Warnera. Nie przejmuj się niczym. Wal do roboty i wracaj szybko do domu. Będę czekała na ciebie. Śmiało Dean, odwagi, chłopie. Pamiętaj, czekam! Pa, kochany. A teraz do roboty. No, wal...

Dean z ponurą miną chwycił ze stołu ciężki szklany klosz od sera i z całej siły z rozmachem walnął nim w uroczu uśmiechającą się z ekranu buzię. Zanim jednak krucha powłoka ekranu i szkło klosza rozmięły się na tysiące okruchów, posłyszał jeszcze zachwycony głos dziewczyny:

— Dean, jesteś wspa...

ANDRZEJ MONASTYRSKI

BASIC

(Podstawy programowania)

PAMIĘĆ KASETOWA: INSTRUKCJE, SYSTEMOWE LOAD, SAVE, VERIFY, MERGE (wiadomości podstawowe)

ZX Spectrum współpracuje z magnetofonem kasetowym. Taśmy magnetyczne (kasety) pełnią rolę pamięci zewnętrznej (pamięci pomocniczej). Na kasety nagrywamy programy przeznaczone do wielokrotnej realizacji.

Do nagrywania programów postępujemy się instrukcją postaci:

SAVE „nazwa”

— nazwa — jest to nazwa identyfikująca dany program na taśmie (jest to innymi słowy etykieta programu) zaczyna się od litery i jest ciągiem liter i cyfr, liczba znaków w nazwie ≤ 10

Załóżmy, że wprowadziliśmy program do komputera i chcemy go zapamiętać na taśmie magnetycznej z etykietą (nazwą) „rysunek”.

W tym celu musimy wykonać następujące czynności:

1. podłączyć magnetofon do komputera (wejście „MIC”) i ustawić taśmę na miejscu pustym (do zapisu)
2. napisać instrukcję:

SAVE „rysunek” (i ENTER)

3. komputer wypisze na ekranie:

Start tape then press any key

4. włączyć magnetofon (do zapisu)
5. nacisnąć dowolny klawisz w ZX Spectrum
6. na ekranie pokazują się różnej wielkości paski poprzeczne, poprawny zapis jest sygnalizowany przez komputer napisem w lewym dolnym rogu postaci: Ø OK

Zapisany program na taśmie należałoby sprawdzić, czy dobrze się nagrał. Instrukcja VERIFY porównuje program z taśmą z zawartością pamięci komputera i ma postać:

VERIFY „nazwa”

— nazwa — jest to nazwa identyfikująca dany program na taśmie (etykieta programu).

Sprawdźmy nagrany przez nas program. W tym celu musimy wykonać następujące czynności:

1. przewinąć taśmę tak, aby ustawić ją (mniej więcej) przed początkiem sprawdzanego programu
2. włączyć przewód łączący wejścia: słuchawki w magnetofonie a w komputerze „EAR”
3. napisać instrukcję:

VERIFY „rysunek” (i ENTER)

4. włączyć magnetofon (do odczytu)
5. następuje szukanie na taśmie programu o etykiecie (nazwie) rysunek. W trakcie szukania programu komputer wypisuje na ekranie nazwy każdej informacji na taśmie na jaką po drodze natrafi. W momencie znalezienia programu o etykiecie rysunek pojawi się na ekranie napis:

Program: rysunek

Następnie na ekranie pokazywać się będą różnej szerokości poprzeczne paski. Poprawne zweryfikowanie (sprawdzenie) programu sygnalizowane jest przez komputer w lewym dolnym rogu ekranu napisem postaci:

Ø OK, Ø:1

Do wczytywania programów zapisanych na taśmie służy instrukcja postaci:

LOAD „nazwa”

— nazwa — jest to nazwa identyfikująca dany program na taśmie (etykieta programu)

Chcemy wczytać uprzednio nagrany program o etykie-

cie (nazwie) rysunek.

W celu wczytania programu musimy wykonać następujące czynności:

1. włożyć do magnetofonu odpowiednią kasetę tzn. taką, na której zapisany jest program, który chcemy wczytać
2. ustawić taśmę w miejscu (przybliżonym) naszego programu
3. włączyć przewód łączący wejścia: słuchawki w magnetofonie, a w komputerze „EAR”
4. napisać instrukcję

LOAD „rysunek” (i ENTER)

5. włączyć magnetofon (do odczytu)
6. następuje szukanie na taśmie programu o etykiecie (nazwie) rysunek. W trakcie szukania programu komputer wypisuje na ekranie nazwę każdej informacji na taśmie na jaką po drodze natrafi. W momencie znalezienia programu o etykiecie rysunek pojawi się na ekranie napis:

Program: rysunek

Następnie na ekranie pokazywać się będą różnej szerokości poprzeczne paski. Poprawne wczytanie programu sygnalizowane jest przez komputer w lewym dolnym rogu ekranu napisem postaci:

Ø OK, Ø:1

Możemy wtedy program wykonywać (**RUN** i **ENTER** itd.) Instrukcja **LOAD** usuwa stary program i zmienne z pamięci komputera przed wczytaniem nowych z taśmy. Istnieje inna instrukcja — **MERGE** — która tego nie robi. **MERGE** usuwa tylko te linie i zmienne, które zastępuje nowymi, o tym samym numerze lub nazwie.

Instrukcja **MERGE** ma postać:

MERGE „nazwa”

— nazwa — jest to nazwa identyfikująca dany program na taśmie (etykieta programu)

Czynności, które wykonujemy są takie same jak w **LOAD**.

Instrukcję **MERGE** stosujemy wtedy, gdy „składamy” dwa programy (lub kilka programów) w jeden.

Przypuśćmy, że mamy dane dwa programy:

10 REM program 1

20

.

.

100

110

.

.

300

10 REM program 2

20

.

.

100

310

.

.

500

Chcemy z tych programów złożyć jeden program postaci:

10 REM program 2

20

linie z programu 2

.

.

100

110

linie z programu 1

.

.

300

500

Programy nagrane są na taśmę z etykietami (nazwami) program 1 i program 2. Wykonujemy następujące czynności:

LOAD „program 1” (i ENTER)

po wczytaniu tego programu (Ø OK) piszemy

MERGE „program 2” (i ENTER)

po wczytaniu tego programu (Ø) naciśniemy **LIST** i sprawdzimy czy otrzymaliśmy w programie te linie, które nas interesują.

Linie programu 1 o numerach 10+100 będą zastąpione liniami programu 2 o tych samych numerach. Linie programu 1 o nr 110+300 pozostaną bez zmian, natomiast linie o numerach 310+500 są liniami programu 2.

W instrukcjach **VERIFY**, **LOAD** i **MERGE** nazwa programu może być tekstem pustym tzn. możemy napisać:

VERIFY "LOAD" "MERGE"

wówczas komputer nie dba o nazwę programu i bierze pierwszy program z taśmy, na który natrafi.

Uwaga: w nazwach programów (w tych instrukcjach) różniące są duże i małe litery, np. RYSUNEK i rysunek nie oznaczają tej samej nazwy.

Jeżeli podczas czytania z taśmy lub sprawdzania (weryfikowania) programu komputer wypisze:

R Tape loading error

oznacza to, że wystąpił błąd we wczytywaniu (program na taśmie został odnaleziony, ale z jakiegoś powodu nie mógł zostać wczytany lub zweryfikowany).

W przypadku weryfikowania, najczęściej należy dany program nagrać od nowa (**SAVE**) i ponownie weryfikować (**VERIFY**).

W przypadku odczytywania (**LOAD** lub **MERGE**) należy zmieniać ustawienie głośności i tonów w magnetofonie. Jeżeli po kilku próbach nie uda się wczytać programu, należy zaniechać wczytywania (szczególnie wtedy, gdy taśma była nagrywana na innym magnetofonie).

Z.W.

Jak to zrobić

(2)

Wychodząc z założenia, że nawet najprostszy przykład wyjaśni więcej, niż wielogodzinny wykład chcę podbudować poprzednie rozważania o player'ach programem wykorzystującym tę technikę.

Jest to bardzo prosty program, pomagający w przyswojeniu czterech podstawowych działań arytmetycznych. Są w nim kolejno losowo wybierane działania i generowane ich składniki (czynniki). Pojawiają się one na monitorze w postaci równania z jedną niewiadomą, np:

$$9 : x = 3$$

a osoba „ucząca” się powinna wprowadzić prawidłową, jej zdaniem, wartość zmiennej x.

Cały proces przebiega w kilku częściach o zmiennym stopniu trudności, osiąganym przez zmianę zakresu zgadywanych liczb.

W programie efekt ten otrzymujemy przez zwiększenie liczby wartości możliwych do generowania za pomocą następującej instrukcji:

$$I = \text{INT}(\text{RMD}(\emptyset) * \text{Poziom})$$

oraz przesunięcie wartości bazowej wykorzystywanej do tworzenia zgadywanych liczb

$$x = \text{BAZA} + I$$

Pięknie, ale gdzie są nasze *player'y*. Otóż spójrzmy na ekran.

Prezentują one kolejne zmiany przebiegające po udzieleniu prawidłowej odpowiedzi.

W programie technika *player-missile* wykorzystana została do zobrazowania ruchu „ludzika”. Jego przesuwanie się jest uzależnione od poprawności odpowiedzi i ilustruje moment zbliżenia się do oceny bardzo dobrej lub sygnalizuje niebezpieczeństwo otrzymania dwójki. Dodatkowo, w czasie gdy grający myśli, jakiej udzielić odpowiedzi, ludzik ze zniecierpliwieniem tupie nogą.

W tekście programu oznaczone zostały odpowiednie fragmenty realizujące „odpowiedzialne” za opisane zachowanie ludzika. By nie odebrać przyjemności rozszyfrowania, jak one działają, nie chcę ich dogłębnie analizować.

Zyczę dobrej zabawy.

GRZEŚ

```

10 OPEN #2,4,Ø,"K:"
15 DIM A$(20),OP$(1),K$(1)
20 A$=""
25 PL=Ø:GOSUB 2205:REM CZYNNOSCI          POZATKOWE I DEFINIOWANIE
   PLAYERA Ø
30 FOR PL=1 TO 3:REM TWORZENIE            POZOSTALYCH PLAYEROW
35 GOSUB 2235:NEXT PL
40 MS=90:REM MIEJSCE WYSWIETLANIA        PLAYERA NA POZATKU GRY
45 L=5:REM USTAWIENIE LUDZIKA
50 P=4:REM (P-3) OKRESLA POZIOM         TRUDNOSCI
55 D=Ø:REM PRZESUNIECIE ZAKRESU        LICZB
60 REM S T A R T
65 IF P>8 THEN GOTO 900:REM GRA          MA PIECI POZIOMOW
70 REM ORGANIZACJA EKRANU
75 POSITION 1,8
80 PRINT #6;"2-----5"
85 POKE 53248,MS:REM WYSWIETLENIE       PLAYERA Ø - STOJACY LUDZIK
90 POSITION 11,Ø
95 PRINT #6;"czesc ";P-3
100 REM GENEROWANIE WYRAZENIA
105 J=D+INT(RND(Ø)*P)+1
110 I=D+INT(RND(Ø)*P)+1
115 X=D+INT(RND(Ø)*P)+1
120 C=INT(RND(Ø)*8)+1
125 ON C GOTO 130,130,135,140,145,145,150,155
130 I=J+X:OP$="+":GOTO 200+50*(C-1)
135 I=J-X:OP$="-":GOTO 200
140 I=X-J:OP$="*":GOTO 250
145 I=J*X:OP$="*":GOTO 200+50*(C-5)
150 J=I*X:OP$="*":GOTO 200
155 X=J*I:OP$="*":GOTO 250
195 REM WYPISANIE DZIALANIE
200 POSITION Ø,2:PRINT #6;A$:POSITION 1,2
205 PRINT #6;J;" ";OP$;" X = ";I
210 C=1:REM WSKAZNIK TYPU DZIALANIA
215 GOTO 300
250 POSITION Ø,2:PRINT #6;A$:POSITION 1,2
255 PRINT #6;"X ";OP$;" ";J;" = ";I
260 C=2
295 REM POBRANIE WYNIKU
300 POSITION Ø,10
305 PRINT #6;"podaj wartosc x ";
310 LK=Ø:IC=Ø:POSITION 17,10
315 IF PEEK(764)>255 THEN 330
320 REM OCZEKIWANIE NA WCISNIECIE        KLAWISZA
325 GOSUB 2320:GOTO 315
330 GET #2,K:IF K=155 THEN 380
335 REM KONTROLA POPRAWNOSCI
340 IC=IC+1:IF IC>3 THEN 300
345 IF K<48 OR K>57 THEN 300
350 PRINT #6;CHR$(K)
355 REM OBLICZANIE WARTOSCI LICZBY
360 LK=10*LK+K-48
370 GOTO 315
375 REM ANALIZA ODPOWIEDZI
380 POSITION Ø,10:PRINT #6;A$

```

dokończenie na str. 26

dokończenie ze str. 25

```
385 POSITION 1,2:PRINT #6;A$
390 POSITION 1,2:IF C=2 THEN 405
395 PRINT #6;J;" ";OP$;" ";X;" = ";I
400 GOTO 415
405 PRINT #6;X;" ";OP$;" ";J;" = ";I
410 REM SPRAWDZENIE POPRAWNOSCI
415 IF LK=X THEN 500
420 REM ODPOWIEDZ NIEPRAWIDLOWA
425 L=L-1:IF L<1 THEN 460
430 POKE 712,53:POKE 708,250
435 GOSUB 2500:REM RUCH W LEWO NA          ZMIENIONYM TLE
440 FOR N=1 TO 5:GOSUB 2300:NEXT N
445 POKE 712,250:POKE 708,50
450 GOTO 105
455 REM D W O J A KONIEC GRY
460 POSITION 1,8:PRINT #6;"2"
465 GOSUB 2300:PRINT #6;CHR$(125)
470 POSITION 1,3
475 PRINT #6;"MUSISZ JESZCZE"
480 POSITION 3,5
485 PRINT #6;"POPRAWOWAC"
490 GOTO 1000
495 REM ODPOWIEDZ POPRAWNA
500 L=L+1:IF L>19 THEN 520
505 GOSUB 2400:REM RUCH W PRAWO
510 GOTO 105:REM NASTEPNE DZIALANIE
515 REM OSIAGNIETA PIATKA
520 POSITION 18,8:PRINT #6;"5"
525 GOSUB 2300:GOSUB 1150
530 PRINT #6;CHR$(125)
535 POSITION 2,3:PRINT #6;"B R A W O"
540 FOR N=1 TO 3:GOSUB 2300:NEXT N
545 PRINT #6;CHR$(125)
550 REM USTAWIENIE PARAMETROW          NASTEPNEJ CZESCI
555 P=P+1:D=D+2:L=5:MS=90
560 GOTO 65
895 REM P I A T K A KONIEC GRY
900 PRINT #6;CHR$(125)
905 POSITION 5,3
910 PRINT #6;"WSPANIALE ZNASZ"
915 POSITION 2,5
920 PRINT #6;" M A T E M A T Y K E"
1000 GOTO 1000
1145 REM SYGNAL DZWIEKOWY
1150 FOR X=1 TO 3
1155 SOUND 0,82,10,8:GOSUB 2300
1160 SOUND 0,0,0,0:GOSUB 2301
1165 NEXT X
1170 SOUND 0,103,10,8
1175 GOSUB 2300:GOSUB 2300
1180 SOUND 0,0,0,0
1185 FOR N=1 TO 5:GOSUB 2300:NEXT N
1190 RETURN
2200 REM POCZATEK PROGRAMU
2205 M=PEEK(106)-16:GRAPHICS 18
2210 POKE 712,250:POKE 708,50
2215 POKE 709,176
2220 POKE 559,62:POKE 54279,M
```

```

2225 POKE 106,M:POKE 53277,3
2230 REM DEFINIOWANIE PLAYERA PL
2235 POKE 704+PL,160:POKE 53256+PL,2
2240 S=M*256+1024+PL*256
2245 REM TWORZENIE OBRAZU PLAYERA
2250 RESTORE 2275+PL
2255 FOR L=150 TO 170
2260 READ B:POKE S+L,B
2265 NEXT L
2270 RETURN
2275 DATA 16,56,56,56,16,124,254,186,186,186,186,16,40,40,40,40,40,108,0,0,0
2276 DATA 8,28,28,28,8,62,93,93,93,89,158,8,20,36,68,36,100,12,0,0,0
2277 DATA 16,56,56,56,16,56,124,186,170,186,121,16,40,36,34,36,38,48,0,0,0
2278 DATA 8,28,28,28,8,28,28,28,28,28,28,28,8,8,8,8,8,0,0,0
2300 FOR T=0 TO 100:NEXT T:RETURN
2301 FOR T=0 TO 15:NEXT T:RETURN
2315 REM TUPANIE NOGA
2320 S=M*256+1024+166
2325 POKE S,PEEK(S)+128
2330 GOSUB 2301:GOSUB 2301
2335 POKE S,PEEK(S)-128
2340 GOSUB 2301
2345 RETURN
2390 REM R U C H
2395 REM RUCH W PRAWO
2400 POKE 53248,1:GOSUB 2301:ILK=0
2405 ILK=ILK+1:IF ILK=8 THEN 2435
2410 MS=MS+1
2415 POKE 53251,1:POKE 53250,MS
2420 GOSUB 2301
2425 POKE 53251,MS:POKE 53250,1
2430 GOSUB 2301:GOTO 2405
2435 POKE 53251,1:POKE 53248,MS
2440 RETURN
2495 REM RUCH W LEWO
2500 POKE 53248,1:GOSUB 2301:ILK=0
2505 ILK=ILK+1:IF ILK=8 THEN 2535
2510 MS=MS-1
2515 POKE 53251,1:POKE 53249,MS
2520 GOSUB 2301
2525 POKE 53251,MS:POKE 53249,1
2530 GOSUB 2301:GOTO 2505
2535 POKE 53251,1:POKE 53248,MS
2540 RETURN

```

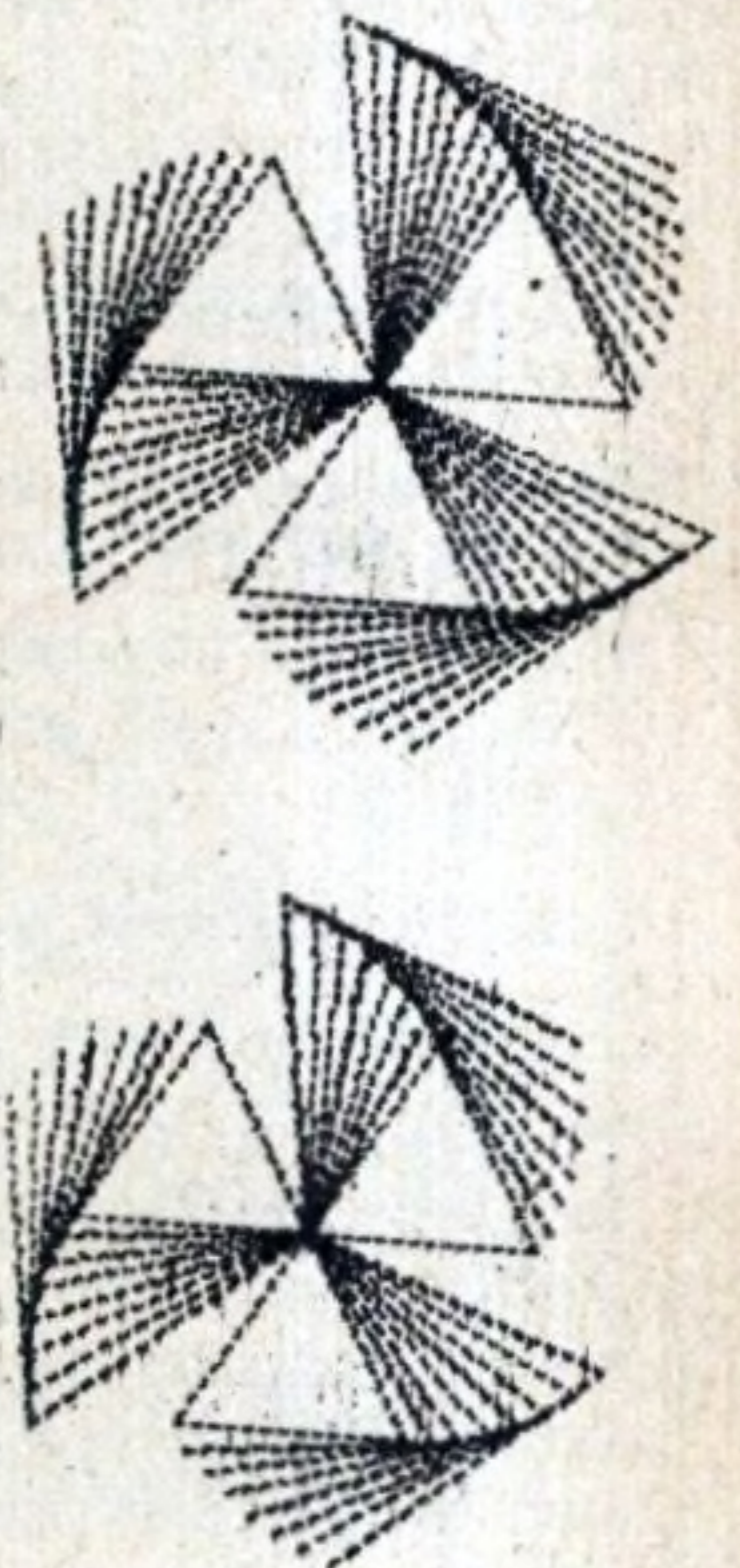
ZMIANA WARTOSCI W OBRAZIE PLAYERA

POWSTALY PRZEZ WYMIANE PLAYEROW

```

JANUSZ MILLER
SHARP PC-1500
WIATRAKI
10: BEEP 10,100,10
20: "A": CLEAR :
   USING : RADIAN
30: INPUT "ILE WIA
   TRAKOW ? N="; N
   : N=ABS N: IF NK
   3GOTO 10
40: INPUT "KOLOR ?
   U="; U: U=ABS U
   : U=INT U: IF U>
   3GOTO 10
50: INPUT "MNOZNIK
   ? M="; M: M=ABS
   M: IF M=0GOTO 1
   0
60: GRAPH : LINE -(
   0,0)-(100,0),9
   : SORGN
70: R=M: IF R>60LET
   M=M-1: GOTO 70
80: S=PI/(4*N): D=R/
   (2*N): DIM W(6,
   2)
90: U=U-1: IF U<1
   LET U=1
100: FOR L=1 TO N
110: T=0: U=U+1: IF U
   >3LET U=U-3
120: FOR K=1 TO 6
130: W(K,1)=R*COS (
   T+F)
140: W(K,2)=R*SIN (
   T+F): T=T+PI/3
150: NEXT K: COLOR U
160: FOR K=1 TO 6
170: IF K=2OR K=4OR
   K=6GOTO 230
180: X=W(K,1): Y=W(K
   ,2)
185: IF L>1THEN
   LINE -(0,0)-(X
   ,Y),9: GOTO 200
190: LINE -(0,0)-(X
   ,Y),1
200: P=K+1: IF P>6
   LET P=P-6
210: A=W(P,1): B=W(P
   ,2): LINE -(X,Y
   )-(A,B),1
220: LINE -(A,B)-(0
   ,0),1
230: NEXT K: F=F+S: R
   =R+D
240: NEXT L: TEXT :
   END

```



EKRAN GRAFICZNY

COORD 23677/8

Obszar ekranu zajmuje w Spectrum 6912 bajtów (adresy 16384.. 23295), przy czym ostatnie 768 bajtów jest przeznaczone na atrybuty. W trybie graficznym ekran tworzy sieć punktów o architekturze 192 linie na 256 kolumn, lecz dla BASIC'a dostępnych jest 176 linii, gdyż pominięto część edycyjną ekranu. Pomimo dobrych algorytmów procedury graficzne nie są pozbawione drobnych, lecz utrudniających życie elementów. Szczególną krytyką darzą wszyscy sposób przekazywania parametrów do instrukcji DRAW.

Z chwilą włączenia komputera oraz po rozkazach RUN, CLEAR, NEW i CLS punkt kursora graficznego ma współrzędne (0, 0). Spróbujmy stworzyć funkcję ustawiania pozycji tego kursora podobnie, jak to ma miejsce w Amstrad — BASIC (rozkaz MOVE). Wykorzystamy niestety jedyne, graficzne zmienne systemowe COORD. Adres 23677 trzyma kolumnę ostatnio postawionego punktu (współrzędna x), zaś 23678 linię (współrzędna y).

```
200 POKE 23677,x: POKE 23678,y
: RETURN
```

Aby przesunąć kursor do pozycji np. (80, 100) wczytujemy:

```
LET x = 80: LET y = 100: GOSUB 200
```

Oczywiście punkt ten nie będzie „zapalony” w przeciwieństwie do instrukcji PLOT 80, 100.

Rysowanie linii szczególnie początkującym nastrocza wiele kłopotów. Lepiej przecież podać współrzędne punktu, w którym linia będzie zakończona.

```
300 DRAW x- PEEK 23677,y- PEEK
23678: RETURN
```

Przykładowo, aby narysować linię od punktu (10, 15) do punktu (200, 75) wczytujemy:

```
10 LET x=10: LET y=15: GOSUB 200
20 LET x=200: LET y=75: GOSUB 300
```

Innym sposobem rysowania linii jest wykorzystanie współrzędnych biegunowych, jak to ma miejsce w LOGO. Należy ustalić azymut w stopniach i podać długość linii w punktach. Oto podprogram:

```
400 IF ABS azymut>360 THEN RET
URN
410 PLOT PEEK 23677, PEEK 23678
420 DRAW np* SIN (azymut* PI / 180),np* COS (azymut* PI / 180)
430 RETURN
```

Parametry podajemy następująco:
LET azymut = 45: LET np. = 100:
GOSUB 400

Symulacja komend LOGO: prawo i lewo, nie powinna teraz nastroczać trudności.

Powróćmy jeszcze do problemu braku możliwości stawiania punktów w dolnych liniach ekranu i nauczmy komputer. Zakładamy, że numery dodatkowych linii graficznych będą od 0 do 15, przy czym punkt (0, 0) będzie w lewym, górnym rogu prostokąta edycji. Stawianie punktów umożliwi nam procedura w języku wewnętrznym 280 o identyfikatorze EDIT—PLOT zapisana pod assemblerem GENS3M21:

```
5 edit-plot 'B6
10 ORG 61000
20 LD HL,(23677)
30 LD A,H
40 CP 16
50 JR C,NOERR
60 RST 8
70 DEFB 4
80 NOERR
90 AND 7
100 ADD A,80
110 LD D,A
120 LD A,H
130 CP 8
140 JR C,LINIA
150 SET 5,E
160 LINIA LD A,L
170 AND 248
180 RRCA
190 RRCA
```

```
200 RRCA
210 ADD A,E
220 LD E,A
230 LD A,L
240 AND 7
250 LD B,A
260 INC B
270 XOR A
280 SCF
290 NRBIT RRA
300 DJNZ NRBIT
310 EX DE,HL
320 BIT 1,(IY+87)
330 JR NZ,OVER1
360 OR (HL)
365 LD (HL),A
370 RET
380 OVER1 XOR (HL)
385 LD (HL),A
390 RET
```

linia 20 — wydobywanie współrzędnych punktów z COORD;

linie 30—40 — współrzędna y nie może przekroczyć 15;

linie 50—70 — błąd, jeśli przekroczyła; linie 80—220 — wyszukanie bajtu na ekranie, w którym punkt będzie postawiony;

linie 230—310 — wyszukanie numeru ustawionego bitu;

linie 320—330 — czy tryb OVER 1 — skok, jeśli tak;

linie 360—370 — wydruk punktu w trybie normalnym;

linie 380—390 — wydruk punktu w trybie OVER 1;

Aby skorzystać z procedury zastosujmy podprogram:

```
500 POKE 23677,x: POKE 23678,y
510 IF USR 61000 THEN
520 RETURN
```

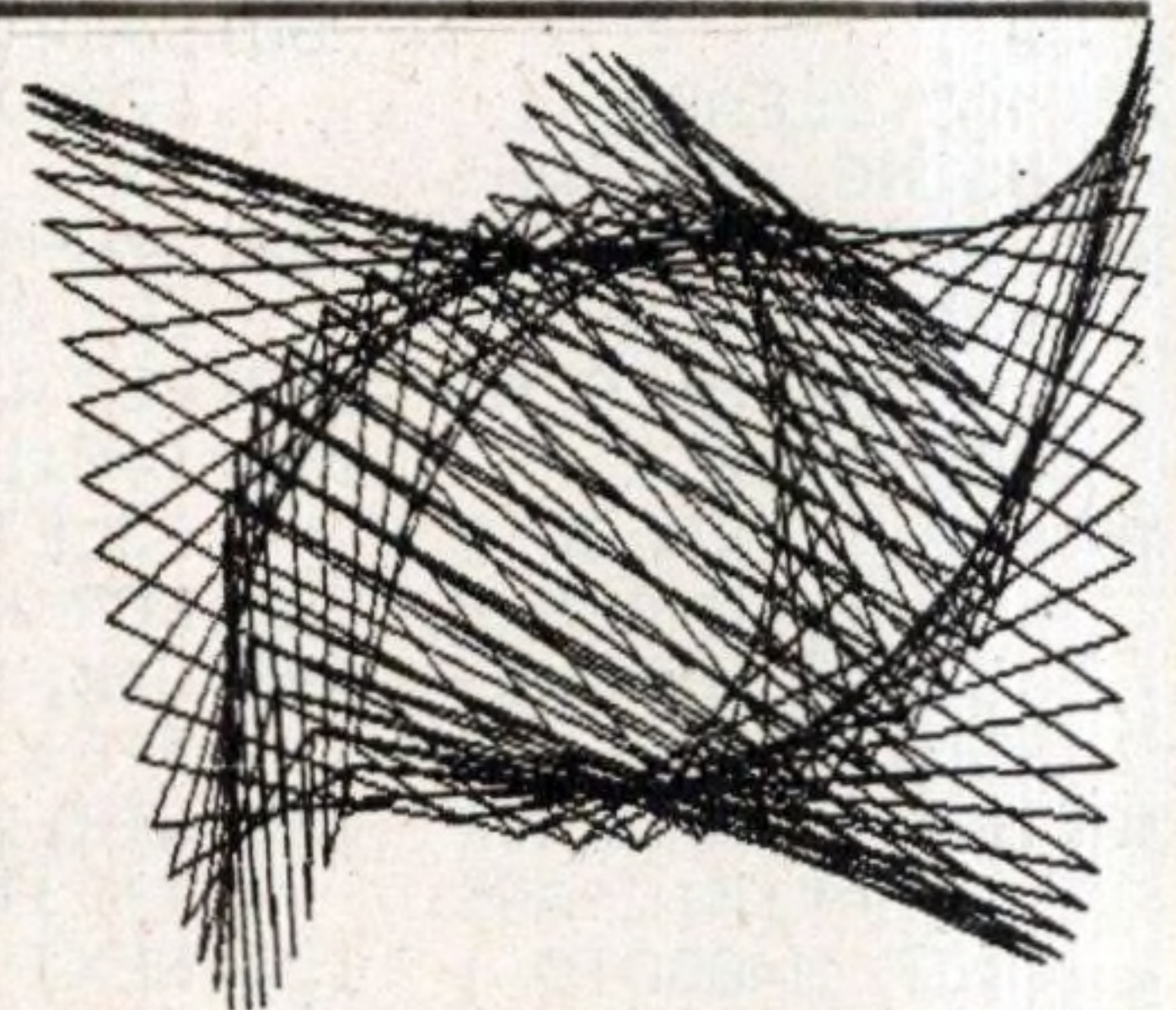
Narysowanie linii prostej wygląda następująco:

```
10 FOR a=0 TO 255
20 LET x=a: LET y=15
30 GOSUB 500
40 NEXT a
50 GOTO 50: REM obrona przed r
aportem
```

Procedura EDIT—PLOT uwzględnia tryb OVER, jeśli go włączymy. W następnym numerze jeszcze o ekranie...

Krzysztof MAMCARZ

**GRAFIKA
(AMSTRAD)**



Dysponując chwilowo wolnym czasem oraz komputerem, zespół robotów, kierowany przez pracownego Spektrusa, postanowił założyć komputerowy system matrymonialny, żeby ułatwić ludziom wybór partnera lub partnerki. Roboty zachwycone były tym pomysłem do tego stopnia, że nie brały pod uwagę ewentualności, iż kiedykolwiek ktoś może mieć z tego powodu do nich pretensje.

Przezorny Spektrus od początku miał jednak wątpliwości. Koncepcja, że maszyna dostarczać będzie żonę lub męża jego zdaniem zniechęcała do małżeństwa w ogóle, skoro pretensje będzie można mieć tylko do mikroprocesorów i programu. A jak się coś nypnie? A jak zabraknie części zamiennych? A jak wyłączyć prąd? A jak coś nawali? Trzeba będzie przeproszać, a co gorsza przez lata wysłuchiwać wyzwisk: ty tranzystorze skubany!

Mimo Spektrusowych wątpliwości system ostatecznie uruchomiono, a pierwszym efektem jego pracy było skojarzenie 72-letniego kawalera, który chciał udowodnić, że stary człowiek może z 80-letnią panią, która zapomniała już, o co mu chodzi, a ponadto była głucha.

Odpowiednia reklama zapewniała przedsiębiorstwu popularność i powodzenie, jednak do komputerowej firmy matrymonialnej zaczęły napływać spostrzeżenia, propozycje i sugestie, wśród których przewijały się koncepcje całowania w różne miejsca i jeszcze bardziej wyrafinowanych pieśczoł.

W tej sytuacji Spektrus, widzący jasno swą powinność polecił powołać bezkomputerową, wielolicydową firmę rozwodową, „żeby była jakaś całość”.

W komputerlandzie znowu wszyscy mają pełne ręce roboty...
Podglądał: **EUGENIUSZ MLECZAK**



UŻYTKOWNICY PAMIĘCI EPROM!

ZPAE „INCO” Wrocław oferuje:

- programator pamięci EPROM typ PPE-1
- kasownik pamięci EPROM typ KPE-1.



Zalety programatora:

- przyłączany do mikrokomputera ZX Spectrum służy do programowania pamięci EPROM typ 2716/2732/2732A/2764/27128/27256
- umożliwia programowanie pamięci, czytanie zawartości, kontrolę zaprogramowania i kasowania oraz łatwe przeniesienie danych z jednego EPROM-u do innego (również różnych typów).
- program obsługujący programator napisany jest w języku maszynowym mikroprocesora Z80 i umieszczony w pamięci stałej programatora
- dane do zaprogramowania można przestać z zewnątrz w sposób równoległy lub szeregowy

Na życzenie programator może być przystosowany do innego typu mikrokomputera (Armstrad, Meritum, Commodore).

Zalety kasownika:

- szybki czas kasowania
- możliwość jednoczesnego kasowania 6 szt. EPROM-ów
- nastawny czas kasowania 0—60 min.

Cena programatora: 79 500 zł

Cena kasownika: 39 000 zł

Zamówienia przyjmuje:



REDAGUJ RAZEM Z NAMI!

Z pewnością każdy z Was przeszedł dziesiątki labiryntów, stoczył sporo walk z kosmitami, czy też pędził po niejednym torze wyścigowym. Prędko też zauważyliście, że wiele gier jest do siebie podobnych i szybko nudzą. Zatem proponujemy byćście razem z nami reda-

gowali namiastkę listy przebojów, a właściwie tych programów, które najchętniej ładujemy do naszego „Atari”. Ponieważ niewielu z nas ma stację dysków — umawiamy się tylko na programy kasetowe. Uwzględniamy tylko gry komputerowe i programy użytkowe.

Pozycja	Nazwa i rodzaj programu (producent i data)	Grafika	Dźwięk	Atrakcyjność	Szczególne wartości zabawowe, edukacyjne i inne	Ogólna ocena
1	„Spy vs Spy” — cz. II „Walka szpiegów” — cz. II gra przygodowa	10	8	10	Gra na 2 joysticki. Wymagane logiczne myślenie i zdolność podzielnej uwagi oraz szybkiego kojarzenia szczegółów. Dodatkową atrakcją piękna grafika trójwymiarowa. Dużo emocji.	9
2	„The Last V-8” „Ostatnie V-8” gra zręcznościowa Mastertronic 1985	9	10	7	Dojechanie samochodem do bazy przed upływem czasu, wymaga mistrzowskiego opanowania joysticka. Doskonali podkład muzyczny. Wyraźna refleks. — 6 pkt.	9
3	„Sega's Spy Hunter” gra przygodowa U.S. Gold	8	8	8	Kto zna scenarię i dynamikę przygód Jamesa Bonda — ten wie już o co chodzi. Szybka akcja z dreszczykiem, samochodami, motocyklami, motorówką etc. Wyświetlane są komunikaty. — 6 pkt.	8
4	„Montezuma's Revenge” (Zemsta Montezumy) gra labiryntowo-przygodowa Utopia Software 83	9	6	10	Pasjonująca gra, w której stawką są zaklęte skarby wodza Azteków. Sceneria odpowiada z wyobrażeniem o azteckich labiryntach, czyhających w nich niespodziankach. — 8 pkt.	8
5	„Hacker” (Komputerowy gangster) Activision 1985 gra strategiczno-przygodowa	6	4	10	Sensacyjna przygoda ze skompletowaniem planu „Magma”. Aktywny udział w grze, wymagana podstawowa znajomość angielskiego i geografii stref czasowych. Uczy logicznego myślenia.	7

Odpadają w przedbiegach te, które zamiast bawić i uczyć, drażnią swą agresywnością lub noszą znamiona hazardu.

Proponujemy skalę ocen od 1—10, przy czym ocena końcowa nie jest średnią arytmetyczną. Decydują też walory szczególne programów. A więc, rozpoczynamy zabawę od typów Pawła listopad'86. Zapewne macie lepsze. Czekamy!

O ile gier na „Atari” jest już około 500 — gorzej z programami użytkowymi. Posiadacze DOS-ów mają większe szanse. Ja polecam na razie „Statystykę” i „Elektrownię atomową” w wydaniu polskim, choć szkoda, że za wygórowaną cenę — studio komputerowe nie zdobyło się na oryginalność i programy te przypominają do złudzenia pewne starocie, jak chociażby „SCRAM” — Symulator Reaktora Jądrowego. Ale, lepsze to niż nic.

Nadal dobrą zabawą jest „Magic Painter” z bogatym menu — pozwalającym na komputerowe malarstwo — prawie w całym tego słowa znaczeniu. Życzę przyjemnej zabawy.

NA 32 KOLUMNACH

Poniższy program pozwala na wprowadzenie istotnych modyfikacji w obszarze ekranu i Display List — zarządzających generacją obrazu na monitorze. Jason Grant ułożył „pięciolinię”, która daje Ci nowe możliwości w porównaniu z edytorem tekstu. Trochę „żonglerki” POKE-ami, a efekt widoczny będzie w postaci 32-kolumnowego edytora ekranu w zerowym trybie grafiki. Przekonaj się sam:

```
10 POKE 82,0:POKE 83,31:GRAPHICS 0:DS=
PEEK(88)+PEEK(89)*256:DL=1539
20 FOR I=1536 TO 1538:POKE I,112:NEXT
I:FOR I=1 TO 24:POKE DL,66
30 POKE DL+2,INT(DS/256):POKE DL+1,DS-
256*PEEK(DL+2):DS=DS+48
40 DL=DL+3:NEXT I:POKE 559,33:POKE DL+
2,6
50 POKE DLL,65:POKE DL+1,0:POKE 560,0:
POKE 561,6
```

NOWE URZĄDZENIE

Jeżeli posiadasz już „Atari” oraz współpracującą z nim drukarkę — pomyśl o elektronicznej maszynie do pisania Nowy wydatek? — ależ nie. Wystarczy pięcioliniowy program by Twój komputerowy keyboard spełniał funkcję

Pocztowa giełda

Rozwiązanie krzyżówki z „IKS-a” nr 6

Hasło brzmi: „PRZYPOMINAMY O PRENUMERACIE IKS-a”. Bony pieniężne (1000 zł) wylosowali: Paweł Dąbrowski — Łódź, Łukasz Foltyn — Radom, Danuta Jagiełła — Wrocław, Maciej Śliwiński — Konin, Bogusław Trzeciak — Milicz.

Nagrody książkowe otrzymują: Ewa Cofur — Będzin, Janusz Hełpa — Wolsztyn, Krzysztof Grabacki — Kraków, Wojciech Piłski — Gdańsk, Wiesława Kurczak — Waliszew, Antoni Rudź — Sokółka, Barbara Bańska — Ostrzeszów, Waldemar Jaróg — Nowy Targ, Paweł Hamerlik — Dąbrowa Górna, Adam Darda — Świętochłowice.

Prosimy o podanie dokładnego adresu Piotra Burka z Kamiennej Góry, który wylosował nagrodę książkową za prawidłowe rozwiązanie krzyżówki z nr. 2 „IKS-a”.

PILNE! Zakupię następujące materiały firmy SHARP:

1. pisaki czarne nr EA-850 B;
2. pisaki barwne nr EA-850 C;
3. papier (rolki) nr EA-1500 P.

Wiadomość: pisemnie pod adresem redakcji. Telefonicznie: 22-65-93 (Warszawa) Janusz Miller

Kupię — klawiaturę lub folię pod klawiaturę do ZX 81. Michał Praceł, ul. Bosmańska 28/12. 81-116 Gdynia

Kupię schemat ideowy mikrokomputera ZX Spectrum 48 K. Jacek Zieliński, ul. Bieruta 8/7 11-100 Lidzbark Warmiński woj. olsztyńskie

Sprzedam ZX Spectrum 48 K. Joanna Piotrowska, ul. Kniewskiego 25/5, Poznań

Kupię Commodore VIC 20. Adam Rusin, ul. Reja 11/36, 43-300 Bielsko-Biała

Odpowiedzi na zadania „Ligi Myślących” z 5 numeru „IKS-a”

Zadanie 1:

Niech pierwszy robotnik kopie rów w czasie t_1 , a drugi w czasie t_2 . Z treści zadania wynika, że cały rów robotnicy wykopali w ciągu 12 godzin, stąd w ciągu godziny wykopali następującą część rowu:

$$\frac{1}{t_1} + \frac{1}{t_2} = \frac{1}{12}$$

Gdyby pierwszy robotnik sam wykonał połowę pracy, a drugi resztę to wykopaliby rów w czasie 25 godzin, stąd

$$\frac{t_1}{2} + \frac{t_2}{2} = 25$$

A więc otrzymujemy układ równań:

$$\begin{cases} \frac{1}{t_1} + \frac{1}{t_2} = \frac{1}{12} \\ t_1 + t_2 = 50 \end{cases}$$

Rozwiązując powyższy układ równań otrzymujemy dwa rozwiązania:

$$\begin{array}{l} t_1 = 30 \quad \text{lub} \quad t_1 = 20 \\ \text{a stąd: } t_2 = 20 \quad \text{lub} \quad t_2 = 30 \end{array}$$

Odpowiedź: Wykonując pracę samodzielnie, jeden z robotników kopałby rów przez 20 godzin, a drugi przez 30 godzin.

Zadanie 2

W zadaniu tym należało obliczyć ilość lat matematyka greckiego, nazwanego w średniowieczu „ojcem algebry” Diofantosa. Oznaczmy przez x — długość życia Diofantosa.

Z treści zadania możemy ułożyć następujące równanie:

$$\frac{1}{6}x + \frac{1}{12}x + \frac{1}{7}x + 5 + \frac{1}{2}x + 4 = x$$

stąd: $x = 84$

Odpowiedź: Diofantos zmarł mając 84 lata.

dokończenie ze str. 30

klawiatury nowoczesnej maszyny do pisania. Spróbuj!

Gdyby w pisaniu tekstów przeszkadzał Ci cursor, możesz „schować” go; wpisując instrukcję: POKE 752, I.

Teraz już możesz napisać do nas na swojej nowoczesnej maszynie. Powodzenia!

```
10 PRINT CHR$(125):OPEN #1,4,0,"K:"
20 OPEN #2,8,0,"P:"
30 OPEN #3,8,0,"S:"
40 FOR GO=0 TO 1 STEP 0:GET #1,K
50 PUT #2,K:PUT #3,K:GOTO 40
```

I jeszcze ciekawostka. Jak podaje „Atari User” — który program opublikował — autorami tej praktycznej „pięciolinii” są uczniowie 5 klasy szkoły podstawowej. A może okażesz się lepszy od nich... „Pogłótkuj”!

Kolumnę opracował:

Paweł Kowalewski uczeń XIV LO im. K. Gottwalda w Warszawie. Wykorzystano teksty z „Atari User” nr 4/86



Rys. M. Przybyłowski

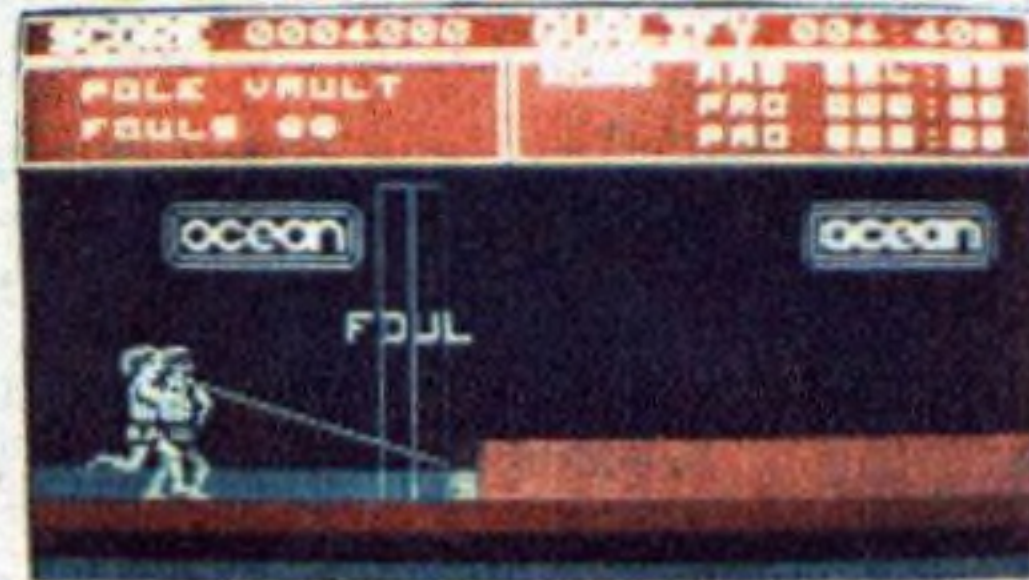
KRZYŻÓWKA NOWOROCZNA

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	
1	I	N	T	U	I	C	J	A				E	L	E	K	T	R	O	N
2	N			N		U		L		R		Ł							
3	F			I		D		A		O		K							
4	O	R	K	A	N				A	K	T				L				O
5	R													U					
6	M	O	T	O	R									N					
7	A													A					
8	T													P	E	G	A	Z	
9	Y													A		O			
10	K													R	A	D	O	M	
11	A													K		Ł			
12			K	R	A											B	O	K	
13														K					
14														R		E			
15														A					
16				O	S	O								A		O	M		
17				Ś		I				R		G		H					
18				C		R				A		A		I					
19	R	A	K	I	E	T	A			E	M	U		O					

PRZEWODNIK

po ZX

SPECTRUM



PREZENT NOWOROCZNY?!

Wydawnictwa Komunikacji i Łączności zapowiadają PRZEWODNIK PO ZX SPECTRUM. Podobno jeszcze w grudniu.

• Nie czytaliśmy. PODOBA nam się wysoki nakład (plan 200 000 egz.) NIE PODOBA nam się wysoka cena (około 500 złotych).

POZIOMO: 1-A komputer jej nie ma, 1-L krąży wokół jądra, 4-A synonim nawałnicy, 4-I część przedstawienia, 4-N rodzaj nadwozia samochodu osobowego. 6-A tygodnik zmotoryzowanych, 6-N namiestnik, 8-A architekt japoński — realizował „Expo-70”, 8-N skrzydlaty koń 10-A rodzina chrząszczy, 10-N miasto woj. nad. rz. Mleczna, 12-B. lód w kawałkach, 12-O jeden z trzech w trójkącie, 14-A gra zręcznościowa, 14-N miasto we Włoszech, 16-A aparatura do sondowania jonosfery, 16-K często kończy się bólem brzucha, 19-A Nazwa pierwszego parowozu; 19-I strus..., 19-L polonijne przedsiębiorstwo komputerowe.

PIONOWO: 1-A nauka o przetwarzaniu informacji, 1-D zjednoczenie, 1-F podobno się zdarza, 1-H imię żeńskie, 1-L miasto w woj. suwalskim, 1-M miasto w Holandii, 1-O świątynia buddyjska w Japonii, 1-S, gwiazdy o największych rozmiarach i jasności, 2-J 1987, 4-E państwo w Europie, 4-N park rozrywki, 8-C chorwacki pisarz (1876-1949), 8-P na tarczy herbowej, 13-A srebrna moneta 13-E, produkt węgla, 13-N stragan, 13-S imię męskie, 16-D szpilkowe kości, 16-F Międzynarodowa Organizacja Radio i Telewizji, 16-H zanieczyszcza atmosferę, 16-L tytuł oficerski w dawnej Turcji, 16-M Stan w USA, 16-O lubi imponować, 17-J pamięć w komputerze.

DOSIĘGO ROKU

Hasło: 3-F, 4-S, 16-E, 17-F, 14-P, 17-L, 19-M, 14-N, 4-A, 12-R, 1-D

Rozwiązania (tylko hasło) należy przysyłać pod adresem redakcji do końca stycznia 1987 r. Wśród czytelników rozlosujemy bony pieniężne i nagrody książkowe.

„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogolek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Michał Przybyłowski, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotoskład i druk rotograniowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 8325. Nr ind. 382809