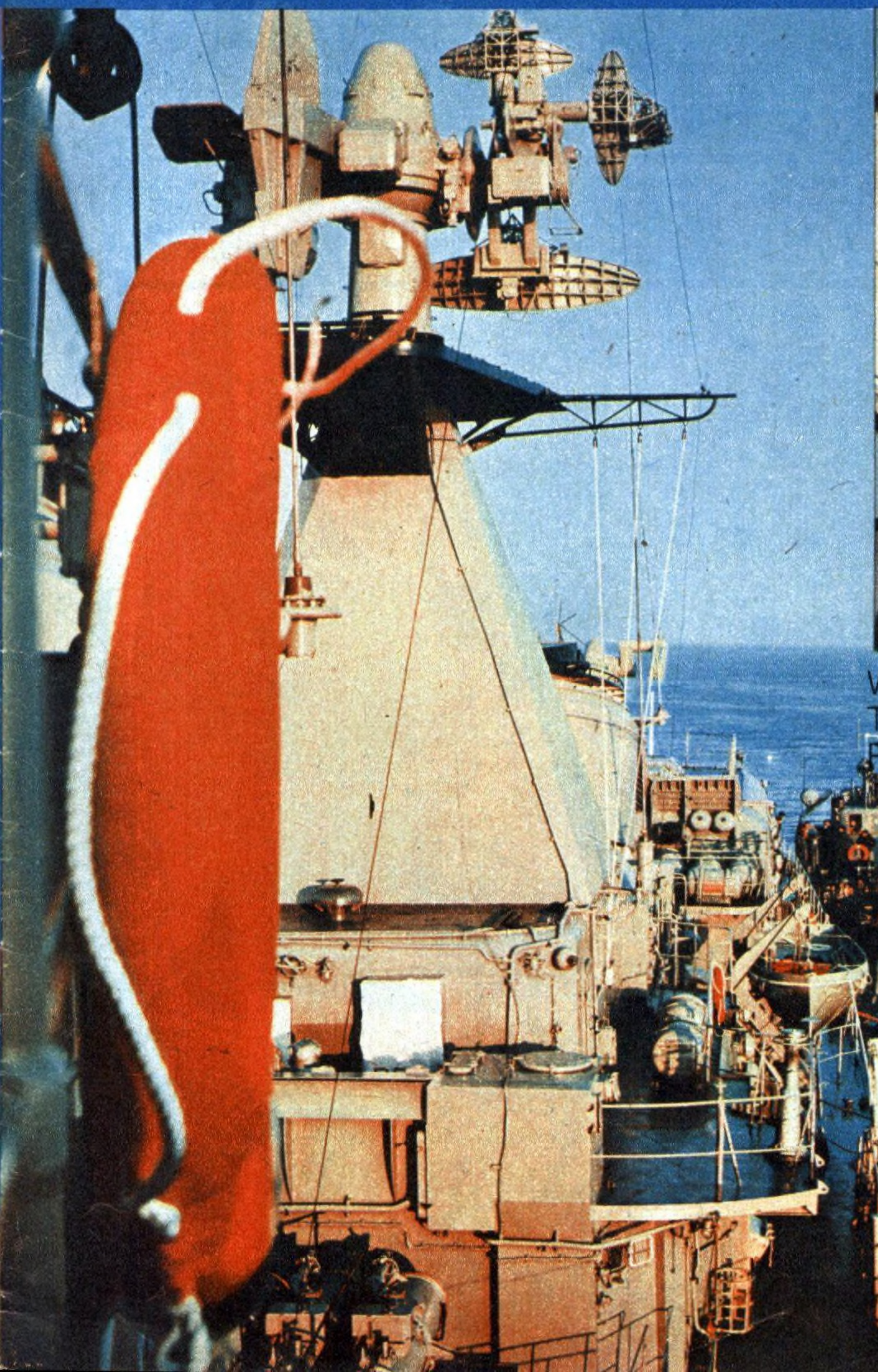
X 6 1987 (14)

INFORMATYKA KOMPUTERY SYSTEMY

Dodatek „Żołnierza Wolności" ISSN 0860-2794 Cena 50 zł



WOJNA O KOMPUTERY
TAJEMNICE SPECTRUM
PROGRAMY

W numerze:

UWAGA CZYTELNICY IKSa!

Następny IKS będzie nosił numer 7—8 i ukaże się na przełomie lipca i sierpnia. Podwójna będzie również cena — 100 zł, ale sądzimy, że poziom zawartości będzie co najmniej tak wysoki jak i ona.

I jeszcze o cenie. Od chwili, kiedy rozpoczęliśmy wydawać nasz miesięcznik (a to już 14 numerów) znacznie wzrosły koszty produkcji, podrożała poligrafia i papier. Stąd też rachunek ekonomiczny zmusza nas do zmiany ceny. Od września IKS będzie kosztował 80 złotych i choć jest to wzrost znaczny, pozostaniemy przecież najtańszym pismem mikrokomputerowym w kraju. Mamy też nadzieję, że dzięki pomocy Czytelników będziemy coraz lepsi — jesteśmy przecież dla Was. Chcemy być Waszym pismem.

Rozpoczynając przed rokiem wydawanie „IKSa" zdawaliśmy sobie sprawę z olbrzymiej popularności mikrokomputerów. Urządzenia te napływały do kraju wartkim strumieniem, a ich cena sprawiła, że dla wielu stały się osiągalne. W ten sposób bariera dzieląca człowieka (z reguły nie zagłębiającego się w tajniki informatyki) z wielkimi komputerami zniknęła. Komputer przestał być już potworem mrugającym małymi żarówkami w niezrozumiałym języku — po prostu stał się bliższy. Nie oznacza to oczywiście, że wszystkie przeszkody zostały pokonane.

Najpierw była euforia nowości. Za „jedne" sto tysięcy można było stać się właścicielem najprostszego Z 8(czy Spectrum. W ślad za nimi przyszły nieco lepsze i droższe typy. Na informatyce chcieli znać się wszyscy; o komputerze trzeba było wiedzieć wszystko. W sytuacji najlepszej znaleźli się ci, którzy na swoim biurku mogli postawić owe cudo.

Po niedługim czasie sytuacja się wyjaśniła: informatyka nie jest dyscypliną trudną, ale na komputer nie każdy może sobie pozwolić. Po dużej fali zainteresowania przyszedł czas na refleksję. Z komputerów zrezygnowali ci, dla których były to jedynie zabawki. Sprzęt profesjonalny zaś jest jeszcze zbyt drogi, aby był powszechnie dostępny. Dlatego też tak ważne pozostają kluby i szkoły, skracające dystans do mikrokomputerów, bo jeśli nawet dziś nie stać nas na sprzęt, to możemy sobie zafundować wiedzę, a ta z pewnością się przyda.

Mamy zatem nadzieję, że Czytelnicy wybaczą nam, iż nie kokietujemy ich lawiną informacji o technicznych nowinkach — sądzimy, że ważniejsze jest ich wykorzystanie. Chcemy również ułatwić pierwszy kontakt z komputerem tym, którzy się go jeszcze lękają.

O wykorzystaniu mikrokomputerów i sprzętu wideo dużo mówiło się w czasie Infowideo '87, imprezy organizowanej przez Stowarzyszenie Mikrokomputerowe ABAKUS. I tu zaszły zmiany: w ubiegłym roku do Warszawskiego Ośrodka Kultury przy Elektoralnej — na Infowideo '86 — trudno było wejść — po roku zainteresowanie mimo wysiłków organizatorów było raczej mizerne. Jak będzie za rok?

REDAKCJA

— **Mnie proszę nie dziękować. Wytypował pana komputer.**

M * I * K * R * O * S * E * R * W * I * S

T U R B O B A S I C

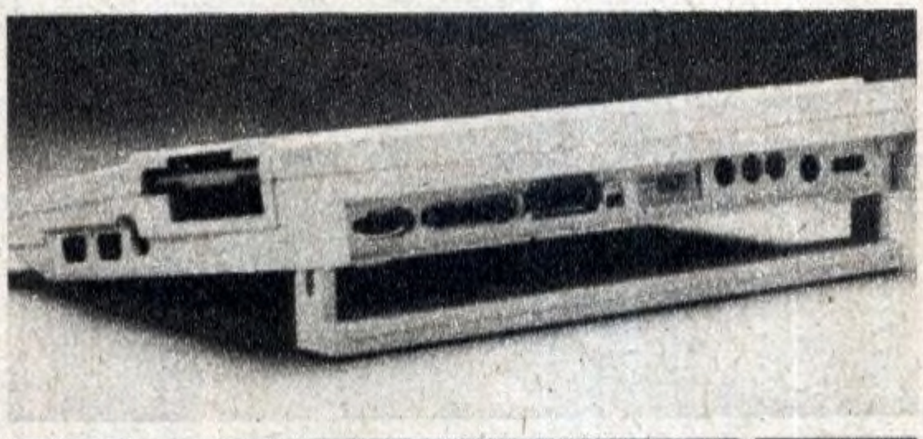
Firma Borland International wprowadziła na rynek nowy swój produkt - Turbo Basic (99.95 $) . Program kompiluje 12 .000 linii na minutę i tworzy zbiory typu EXE .Umożliwia on również kompilację w pamięci, posiada własny edytor ,program łączący oraz bibliotekę . Turbo Basic wykorzystuje instrukcje strukturalne IF ELSE IF , SELECT CASE , DO WHILE , DO UNTIL , LOOP WHILE , LOOP UNTIL . Program jest napisany w asemblerze i jest kompatybilny z IBM Advanced Basic oraz Microsoft GW-BASIC . Umożliwia obsługę karty graficznej EGA oraz koprocesora matematycznego 8087 / 80287 .Turbo Basic pozwala tworzyć programy , które przekraczają 64 KB ($ SEGMENT).

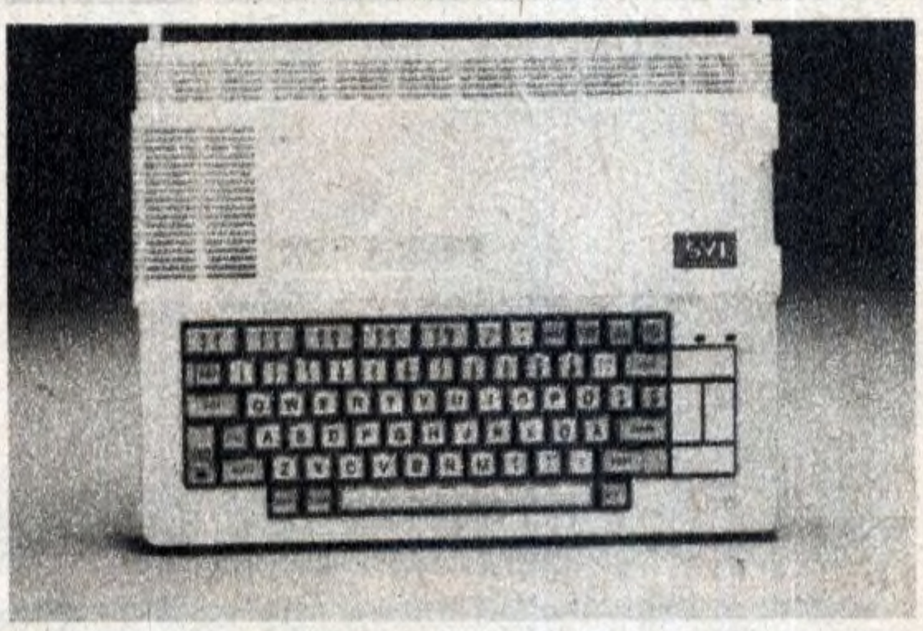
Turbo Prolog w medycynie

Pracownicy naukowi w Medical College of Ohio rozwijają system ekspertowy w Turbo Prologu , przeznaczony dla lekarzy z oddziału intensywnej terapii , określający czy pacjent rzeczywiście uległ atakowi serca . Program ekspertowy , będący częścią składową systemu Expert Scane Segmentation , ocenia dane zgromadzone przez skaner ultradźwiękowy . Dane przesyłane są do komputera ,który wykorzystuje algorytmy napisane w języku C

Mikrokomputer domowy M S X X'press SVI.738

Mikrokomputer X'press nie jest tylko komputerem zabawowym , może korzystać z trzech systemów operacyjnych :CP/M, MSX-DOS i MSX Disk BASIC .Dzięki nim istnieje szeroki dostęp do biblioteki programów użytkowych . Mikrokomputer posiada pamięć zapisywalną RAM 80 KB. W komputerze wbudowano stację dysków 3 1/2 cala ,pozwalającą osiągnąć pojemność 360 KB . Video Display Processor umożliwia pracę monitora :
● w trybie alfanumerycznym 40 lub 80 znaków w wierszu;● w trybie graficznym w standardach małej , średniej i dużej rozdzielczości.Mikrokomputer posiada wyjścia RS-232C i Centronix. Klawiatura typu QWERTY składa się z 73 klawiszy ,w tym 10 funkcyjnych. Do mikrokomputera można dołączyć modem,drugą stację dysków, magnetofon, joystick,drukarkę.Mikrokomputer może pracować w lokalnej sieci komputerowej





□ □ ■ □□ M I K R O D E P E S Z E □□ ■ □ □

Wśród produktów azjatyckich można znaleźć mikrokomputer kompatybilny z IBM PC/AT ,wyposażony w procesor 80286 pracujący z częstotliwością 6 / 8 / 10 / 12 MHz i posiadający 4 MB pamięci RAM na płycie podstawowej . ■ □□ ■ □□ ■

opracowali: Mariusz Jarzębowski i Jacek Wojtala

Statystyka z ICL

Na pierwsze powojenne wyposażenie Głównego Urzędu Statystycznego składały się nieliczne maszyny małej mechanizacji, przeniesione z okupacyjnego urzędu statystycznego w Krakowie. Od 1946 roku rozpoczęto użytkowanie mocno już wyeksploatowanych maszyn systemu kart dziurkowanych, wydzierżawionych od Ministerstwa Przemysłu. Istotnym postępem było dokonanie w 1949 roku zakupu we francuskiej firmie Bull 16 zestawów maszyn liczącо-analitycznych, odznaczających się w owym czasie jednymi z najlepszych możliwościami eksploatacyjnymi. W 10 lat później — przed kolejnym Narodowym Spisem Powszechnym — zakupiono następną partię maszyn w firmie Bull, nowocześniejszych i wyposażonych w elektroniczne przystawki kalkulacyjne.

W 1963 roku dla zapewnienia zorganizowania warunków zakupu i uruchomienia elektronicznej techniki obliczeniowej w GUS, utworzono w Zakładzie Techniki Statystycznej Dział Studiów i Analiz. Stał się on zalążkiem przyszłego Ośrodka Elektronicznego. Prowadzone przez DSiA prace studialnobadawcze były ściśle powiązane z praktycznym uruchomieniem opracowań statystycznych na dostępnej wówczas w Polsce maszynie elektronicznej **ZAM 2**, zainstalowanej w Instytucie Maszyn Matematycznych oraz na maszynie **ICT-1300** pracującej w Centralnym Ośrodku Doskonalenia Kadr Kierowniczych.

W wyniku szczegółowej analizy parametrów techniczno-użytkowych, oprogramowania, cen, warunków dostaw itp., odnoszących się do różnych typów komputerów dostępnych na rynku, ostatecznie zdecydowano się zakupić dla GUS maszynę typu ICL 1905 w brytyjskiej firmie ICL (dawniej ICT). Komputer ten uruchomiony został w październiku 1967 i od pierwszych dni po odbiorze technicznym rozpoczęła się jego eksploatacja na dwie, a wkrótce na trzy zmiany.

Wybór tego typu komputera okazał się dla GUS szczęśliwy również z tego względu, że w kilka lat później polski przemysł komputerowy na podstawie porozumienia z firmą ICL zastosował ich system oprogramowania w polskich komputerach **Odra 1300**, dzięki czemu obie te rodziny maszyn zastosowały w pełni wymienialne oprogramowanie.

W 1970 roku nastąpiły kolejne komputerowe zakupy: **ICL 1902 A**, który już na miejscu przebudowano na większy model **1903 A** oraz polską **Odrę 1304** — pierwszy egzemplarz tej maszyny z produkcji seryjnej.

Obecnie Ośrodek Elektroniczny GUS ma 6 kompatybilnych komputerów **Odra 1305** i **ICL**. Poszczególne zestawy mogą pracować ze sobą, sprzęt jest wymienny, można łączyć odpowiednie moduły. Dyrektor ośrodka, Andrzej Karpiński, twierdzi, że jest to sprzęt dobry, w pełni odpowiadający stawianym wymaganiom. Firma jest znana, solidna, wywiązuje się z umów, rozporządza bogatym oprogramowaniem. W Warszawie znajduje się biuro ICL. Komputery cały czas są unowocześniane, ciągle dokupuje się nowe moduły. Jeśli pamięć pierwszego komputera wynosiła 32 tys. słów, to dzisiaj wynosi 512 tys. Dane rejestrowane są na taśmach magnetycznych. Można też robić to na kartach drukowanych, ale to mało ekonomiczna metoda. Do prowadzenia danych służą mikrokomputery **Mera 9150** z możliwością pracy równocześnie na 32 stanowiskach.

Jerzy RAJCH

(Foto: — Jan Zelman)

PROGRAM 12

Henryk Jasiński — MÓJ PROGRAM

„IKS" był, jest i zawsze będzie Wasz!
Nie wszystkowiedzących, zadufanych w sobie redaktorów, lecz *pismem Czytelników*.

Dlatego z przyjemnością publikujemy program, który opracował jeden z Was. CZEKAMY NA KOLEJNE!

Henryk Jasiński z Warszawy pisze:
„...Jesteście jedyną redakcją doceniającą fakt istnienia... tzw. Pocket Computers.

Pozostałe pisma, jeśli w ogóle je dostrzegają, to traktują komputery kieszonkowe jako ciekawostki, dodając obowiązkową uwagę, że nie można do nich dołączyć joysticka, jakby to decydowało o klasie komputera.

Pozwalam sobie przysłać... program drukujący kalendarz na dowolny rok...".

Tyle nasz Czytelnik. A oto jego program. Wywołuje go etykieta „A" (RUN 100).

```
100:"A":CLEAR :GRAPH
110:DIM A$(8)*1,B$(13)*1,M$(11)*11,L
    D(11),NM(11)
120:DATA "0","1","2","3","4","5","6"
    ,"7","8","9"
130:DATA "STYCZEN",31,1,"MAJ",31,5,"
    WRZESIEN",30,8,"LUTY",28,2
140:DATA "CZERWIEC",30,6,"PAZDZIERNI
    K",31,10,"MARZEC",31,3,"LIPIEC",
    31,7
150:DATA "LISTOPAD",30,11,"KWIECIEN"
    ,30,4,"SIERPIEN",31,8,"GRUDZIEN"
    ,31,12
160:DATA "N","d","P","n","W","t","S"
    ,"r","C","z","P","t","S","b"
170:FOR I=0TO 8
180:READ A$(I)
190:NEXT I
200:FOR M=0TO 11
210:READ M$(M),LD(M),NM(M)
220:NEXT M
230:INPUT "Rok ";R$:R=VAL R$
240:IF R<>INT ROR R<>ABS R60TO 230
250:ROTATE 0:CSIZE 8:COLOR 1
260:GLCURSOR (0,-72):GORGN
270:FOR I=0TO 2
280:GLCURSOR (84I,84I):LPRINT R$
290:NEXT I
300:LINE (0,0)-(216,72),0,0,B
310:ROTATE 1:CSIZE 1
320:M=0:GLCURSOR (216,0):GORGN
330:FOR K=1TO 4
340:FOR L=1TO 3
350:M=M+1:B=NM(M-1):A=R
360:IF M<>460TO 410
370:X=0:IF A-INT (A/4)*4=0LET X=1
380:IF A-INT (A/100)*100<>0GOTO 400
390:X=0:IF A-INT (A/400)*400=0LET X=
    1
400:LD(M-1)=28+X
410:IF B-3<0LET A=R-1:B=B+13:GOTO 43
    0
420:B=B+1
430:C=INT (365.25*A)+INT (30.6*B)-
    INT (A/100)+INT (A/400)-427:D=C-
    INT (C/7)*7
440:GLCURSOR (-10,-4):LPRINT M$(M-1)
450:RESTORE 160
460:FOR I=0TO 13
470:READ B$(I)
480:NEXT I
490:E=-20:GLCURSOR (E,-4):GOSUB 700
500:FOR J=0TO 5
```

SHARP

```
520:FOR J=1TO 7
530:Y=7*I+J-D:Z=INT (Y/10):U=Y-Z*10
540:IF Y<1OR Y>LD(M-1)LET B$(2*J-2)=
    " ":B$(2*J-1)=" ":GOTO 580
550:B$(2*J-2)=A$(Z)
560:IF Z=0LET B$(2*J-2)=" "
570:B$(2*J-1)=A$(U)
580:NEXT J
590:E=-28-I*8:GLCURSOR (E,-4):GOSUB
    700
600:NEXT I
610:LINE (0,0)-(-72,-102),0,0,B
620:GLCURSOR (-72,0):GORGN
630:NEXT L
640:GLCURSOR (216,-102):GORGN
650:NEXT K
660:ROTATE 0:TEXT :LF 4:END
700:FOR N=0TO 12STEP 2
710:IF N=0COLOR 3:GOTO 740
720:IF N=2COLOR 1:GOTO 740
730:IF N=12COLOR 2
740:B$=B$(N)+B$(N+1)
750:GLCURSOR (E,-4-N*7):LPRINT B$
760:NEXT N
770:RETURN
```

1987

STYCZEN: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30 31

MAJ: Nd Pn Wt Sr Cz Pt Sb — 1 2 / 3 4 5 6 7 8 9 / 10 11 12 13 14 15 16 / 17 18 19 20 21 22 23 / 24 25 26 27 28 29 30 / 31

WRZESIEN: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 / 7 8 9 10 11 12 13 / 14 15 16 17 18 19 20 / 21 22 23 24 25 26 27 / 28 29 30

LUTY: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28

PAZDZIERNIK: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30 31

CZERWIEC: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 / 7 8 9 10 11 12 13 / 14 15 16 17 18 19 20 / 21 22 23 24 25 26 27 / 28 29 30

MARZEC: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30 31

LISTOPAD: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30

LIPIEC: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30 31

KWIECIEN: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30

GRUDZIEN: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 / 7 8 9 10 11 12 13 / 14 15 16 17 18 19 20 / 21 22 23 24 25 26 27 / 28 29 30 31

SIERPIEN: Nd Pn Wt Sr Cz Pt Sb — 1 2 3 4 5 6 7 / 8 9 10 11 12 13 14 / 15 16 17 18 19 20 21 / 22 23 24 25 26 27 28 / 29 30 31

POCKET COMPUTER SHARP PC-1501

Tryb graficzny 8 dostarcza obrazów o najwyższej rozdzielczości, jakie możemy wytworzyć za pomocą Atari, z 320 punktami w poziomie i 192 (lub 160 dla rozszczepionego ekranu) w pionie. Ma jednak podstawową wadę: zezwala na używanie tylko jednego koloru. Instrukcja SETCOLOR 2 określa kolor i jaskrawość tła, a SETCOLOR 1 określa jaskrawość punktu lub linii wykreślonej za pomocą rozkazów PLOT i DRAWTO. Jedyną inną instrukcją SETCOLOR, która działa w trybie 8, jest SETCOLOR 4, ustawiająca kolor i jaskrawość ramki.

W szponach Atari (5)

Wielokolorowa grafika w trybie 8

Trudno kwestionować powyższe informacje, zaczerpnięte z podręcznika języka BASIC dla komputerów Atari, lecz nie zaszkodzi je sprawdzić. Dlatego po włączeniu komputera wprowadźmy instrukcję GRAPHICS 8. Otrzymamy rozszczepiony ekran z oknem tekstowym na dole. Wprowadźmy następnie **COLOR** 1: **PLOT** 5, **80:DRAWTO 315,80.** Na ekranie pojawi się linia pozioma, której kolor możemy zmieniać za pomocą instrukcji SETCOLOR.

Spróbujmy poważnie podejść do badań. Za pomocą **SETCOLOR 2, 0, 0** ustawiamy czarne tło, a za pomocą **SETCOLOR 1, 0, 12** wybieramy linie mocno kontrastowe. Teraz **PLOT 160, 150:DRAWTO 160,5** kreśli linię pionową w kolorze niebieskim.

Jeśli otrzymany kolor nie jest niebieski, to pamiętajmy, że kolory w naszym telewizorze zależą od jego regulacji i warunków pracy. Chociaż kolory, które widzicie, mogą się różnić od opisanych, zasady i techniki ich otrzymywania pozostają takie same.

Wprowadźmy teraz instrukcje **PLOT 101, 150:DRAWTO 101,5.** Zobaczymy pionową linię w kolorze brązowym. Dlaczego otrzymaliśmy dwie linie w różnych kolorach? Jest to zależne od wartości współrzędnej **X**, czy jest parzysta, czy nieparzysta. Wszystkie linie pionowe z nieparzystą wartością **X** będą w jednym kolorze, a wszystkie z wartością **X** parzystą — w innym. Jest to prawdziwe także dla wszystkich punktów zapalonych za pomocą **PLOT**. Jeśli wykreślimy linię pod dowolnym kątem poprzez zapalenie tylko tych punktów, które mają wartości współrzędnej **X** parzyste lub nieparzyste, otrzymamy linię niebieską lub brązową.

Jeśli wykreślimy inną linię za pomocą **PLOT 160, 80 : DRAWTO 240, 120**, będzie ona czerwona. Wprowadzenie **PLOT 101, 80 ; DRAWTO 180, 120** daje linię zieloną. Regułą jest, że wszystkie linie ukośne o kącie nachylenia określonym ułamkiem X/Y = 2/1 będą zielone lub czerwone, w zależności od współrzędnej X punktu początkowego.

Instrukcje **PLOT 10, 10 : DRAWTO 150, 150** dają linię szarą. Wszystkie linie wykreślone pod kątem X/Y = 1/1 będą szare.

Otrzymujemy zatem sześć podstawowych kolorów w trybie graficznym 8. Biały (linie poziome lub kilka linii jedna obok drugiej), brązowy i niebieski (linie pionowe), zielony i czerwony (linie ukośne) i szary. Pozostałe linie są mieszaniną tych podstawowych kolorów.

Jak to się ma zatem do informacji zawartych w podręczniku? Jaką wartość mogą mieć kolory, które otrzymuje się poprzez wykreślenie linii pod określonym kątem? Jak to się dzieje, że otrzymujemy tyle kolorów? Odpowiedzi na to należy szukać w strukturze kineskopu i sposobie tworzenia kolorów. Powierzchnia kineskopu pokryta jest szeregiem poziomych linii, które składają się z ciągów punktów niebieskich, zielonych i czerwonych. Punkty te zapalają się, jeśli trafi w nie strumień elektronów. Są one tak małe, że nie widzimy ich jako punkty, lecz tylko jako połączenie kolorów wielu punktów. Różnorodne barwy, jakie widzimy na ekranie telewizora uzyskuje się poprzez sterowanie intensywnością strumienia elektronów dla każdego punktu. Odpowiedni dobór niebieskich, zielonych i czerwonych punktów dostarcza pełnego zakresu kolorów.

W trybie graficznym 8 linia pionowa jest tak wąska, że nie może pokryć wszystkich trzech kolorów jednocześnie. Kiedy współrzędna X linii pionowej jest parzysta, trafia głównie w punkty niebieskie. Kiedy współrzędna X jest nieparzysta, trafia w punkty czerwone i zielone, dając linię brązową. Rozdzielczość trybu graficznego 8 jest prawie tak dobra jak rozdzielczość ekranu telewizora! Ponieważ tak wąskie linie nie mogą uaktywnić wszystkich trzech punktów koloru, barwa linii nie może być efektywnie wyregulowana poprzez zrównoważenie intensywności świecenia punktów, jak to jest robione w trybach graficznych o niższej rozdzielczości (linie poziome zawsze pokrywają wszystkie kolory).

Podsumujmy zatem otrzymane wyniki:

1. Kreślenie jasnego koloru (o dużej jaskrawości) na ciemnym tle lub ciemnego na jasnym tle wzmacnia otrzymany efekt.
2. Kolor elementarnego punktu nie jest zmieniany przez jego współrzędną Y.
3. Kolor elementarnego punktu zależy nie tylko od ustawionego koloru, lecz także od tego, czy jego współrzędna X jest parzysta, czy nie i od koloru bezpośrednich sąsiadów w poziomie.
4. Rozdzielczość pozioma ma praktyczną granicę raczej 160 niż 320. Dwa sąsiednie poziome punkty elementarne zlewają się w jeden punkt w standardowym kolorze.

Podstawowe efekty kreślenia białych punktów na czarnym tle podsumowane są w poniższej tablicy. N jest ilością sąsiednich białych punktów w poziomie. X jest współrzędną tych punktów w terminach „parzysty", „nieparzysty". Kolor jest w przybliżeniu aktualnym kolorem tych punktów, przy założeniu normalnej regulacji monitora.

N	X	kolor
1	parzysty	niebieski
1	nieparzysty	brązowy
2	parzysty—nieparzysty	czerwony
2	nieparzysty—parzysty	szary
3	—	prawie biały
4	—	biały

Program 1 ilustruje występowanie kolorów poprzez kreślenie dwóch ciągów prawie pionowych białych linii na czarnym tle. Kolorowe poziome wstęgi utworzone są zgodnie z regułami z powyższej tablicy. W tym przykładzie nie powstaje biały kolor, ponieważ istnieją najwyżej pary sąsiednich „białych" punktów. Zauważmy, że paski o stałym kolorze tworzone są albo dlatego, że wszystkie „parzyste" punkty dają kolor niebieski, albo wszystkie „nieparzyste" dają kolor brązowy.

PROGRAM NR 1

```
ZS 110 GRAPHICS 8:COLOR 1
RF 120 SETCOLOR 1,0,14:SETCOLOR 2
   ,0,0
JW 130 FOR X=0 TO 318 STEP 4
UF 140 PLOT X,0:DRAWTO X+1,159:NE
   XT X
JJ 150 FOR X=0 TO 308 STEP 4
BL 160 PLOT X,0:DRAWTO X+9,159:NE
   XT X
AV 199 REM ZMIANA KOLOROW
PB 200 OPEN #1,4,0,"K:"
NO 210 ? :? "WYBRANYM KOLOREM JES
   T TERAZ ";H
YH 220 ? "NACISNIJ KLAWISZ H DLA
   ZMIANY KOLORU"
VO 230 GET #1,X:IF X<>72 THEN 230
QX 240 H=H+1:IF H=16 THEN H=0
OH 250 SETCOLOR 2,H,0:GOTO 210
```

Aby umożliwić łatwą zmianę koloru, dodano linie 199 do 250, zachowując ustawienie zerowej jaskrawości dla tła i jaskrawości 14 dla kreślonych linii. Kolor tła może być prawie niewidoczny, lecz kolory poziomych wstęg mogą być zmieniane w znacznym stopniu. Najlepsze wyniki uzyskamy po wyregulowaniu jaskrawości i kontrastu telewizora na niską lub minimalną wartość.

PROGRAM NR 2

```
QF 10 GRAPHICS 24
QA 20 SETCOLOR 1,0,14:SETCOLOR 2,
   0,0
EZ 30 FOR X=0 TO 318 STEP 3
LE 40 COLOR 1:PLOT 159,0:DRAWTO X
   ,191
NZ 60 NEXT X
TQ 70 GOTO 70
```

Program 2 wykorzystuje zmianę kolorów dla tworzenia wzorów mory. Aby zbadać nieco inny wzór z bardziej jednolitym rozkładem światła i zwiększonym kontrastem, dodajmy następującą linię:

50 COLOR 0:PLOT 159,0:DRAWTO X + 1, 191

Program krok po kroku kreśli czarną linię, która kasuje połowę (lub więcej) „białych" punktów, które były wykreślone w poprzednim kroku, w linii 40. Pomysł ten, po pewnych modyfikacjach, wykorzystany został w programie 3. Jest on tak zaprojektowany, aby osoby mało wprawione w posługiwaniu się klawiaturą nie musiały wprowadzać całego programu, aby zobaczyć co on robi.

PROGRAM NR 3

```
GC 100 GRAPHICS 0:POKE 82,5:POSIT
   ION 9,2:? "***  PIRAMIDA  ***"
   :PRINT
RL 112 ? "USTAW KONTRAST SWOJEGO
   ODBIORNIKA"
GZ 113 ? "I JASKRAWOSC NA MINIMUM
   ."
LK 120 ? "CO WYBIERASZ:"
IU 130 ? "   0 PARAMETRY LOSOWE"
AX 140 ? "   1 PARAMETRY USTAWIAN
   E"
FM 150 TRAP 150:INPUT WYB:IF WYB=
   0 THEN GRAPHICS 24:GOTO 500
YK 160 IF WYB<>1 THEN 150
RM 170 ? :? "KRESLENIE TLA LINIAM
   I PIONOWYMI:"
FO 180 ? "   0 LINIE PARZYSTE"
SY 190 ? "   1 LINIE NIEPARZYSTE"
XA 200 ? "   2 BEZ TLA"
NA 210 TRAP 210:INPUT JASN:IF JAS
   N=2 THEN 230
UN 220 IF JASN<>0 AND JASN<>1 THE
   N 210
AP 230 ? :? "SPOSOB KRESLENIA PIR
   AMIDY:"
MD 240 ? "   0 OD SRODKA"
EK 250 ? "   1 OD LEWEJ NA PRAWO"
FG 260 TRAP 260:INPUT TRYB
XN 270 IF TRYB<>0 AND TRYB<>1 THE
   N 260
CM 280 ? :? "DLA WIERZCHOLKA PIRA
   MIDY UZYJ:"
TT 290 ? "   0 JEDNEGO PUNKTU"
JT 300 ? "   1 DWOCH PUNKTOW"
AN 310 TRAP 310:INPUT WIERZ
IZ 320 IF WIERZ<>0 AND WIERZ<>1 T
   HEN 310
JD 330 ? :? "ODSTEP PROMIENI OD W
   IERZCHOLKA:"
CR 340 ? "(LICZBA CALKOWITA: 2 DO
    6)"
TC 350 TRAP 350:INPUT ODST:IF ODS
   T<1 THEN 350
NS 360 ? :? "KRESLIC CIEMNE PIONO
   WE LINIE?"
FO 370 ? "   0 LINIE PARZYSTE"
SY 380 ? "   1 LINIE NIEPARZYSTE"
NP 390 ? "   2 BEZ LINII"
NG 400 TRAP 400:INPUT CIEM:IF CIE
   M=2 THEN 420
EL 410 IF CIEM<>1 AND CIEM<>0 THE
   N 400
LT 420 ? :? "JESLI KONIEC"
IX 430 ? "   0 PARAMETRY LOSOWE"
BO 440 ? "   1 UTRZYMAC WZORZEC"
XH 450 TRAP 450:INPUT UTRZ
QL 455 IF UTRZ=0 THEN GRAPHICS 24
   :GOTO 600
YX 460 IF UTRZ<>0 AND UTRZ<>1 THE
   N 450
SQ 470 POKE 82,2:? :? "WPROWADZAN
   IE WZORCA ZAKONCZONE"
PH 475 ? "NACISNIJ H ABY ZMIENIC
   KOLOR"
FB 480 ? :? "GOTOWE? NACISNIJ STA
   RT"
ET 490 IF PEEK(53279)<>6 THEN 490
VO 495 TRAP 40000:GRAPHICS 24:GOT
   O 600
SC 497 REM
CR 498 REM *** PARAMETRY LOSOWE *
   **
SI 499 REM
TI 500 JASN=INT(RND(0)*8)
SI 510 TRYB=INT(RND(0)*2)
BJ 520 WIERZ=INT(RND(0)*2)
ND 530 ODST=2+INT(RND(0)*5)
LD 535 IF TRYB=M AND WIERZ=A AND
   ODST=5 THEN 510
QB 540 CIEM=INT(RND(0)*8)
FO 545 KOL=INT(RND(0)*16)
UX 550 POKE 77,0:REM ZEROWANIE TF
   YBU ZWRACANIA UWAGI
SD 597 REM
TF 598 REM *** WYKONANIE PROGRAMU
    ***
SJ 599 REM
HM 600 SETCOLOR 2,KOL,0:SETCOLOR
   1,0,14
DE 620 IF JASN>1 THEN 640
ZJ 630 COLOR 1:B=JASN:GOSUB 1000
DA 640 IF TRYB=0 THEN GOSUB 2000
EP 650 IF TRYB=1 THEN GOSUB 3000
RD 660 REM
UG 670 FOR K=191 TO 1 STEP -ODST
JN 680 COLOR 1:PLOT 159,WIERZ:DRA
   WTO 318,K:PLOT 159,WIERZ:DRAWT
   O 0,K
BO 690 COLOR 0:PLOT 159,0:DRAWTO
   318,K-1:PLOT 159,0:DRAWTO 0,K-
   1:NEXT K
XA 700 IF CIEM>1 THEN 720
PW 710 COLOR 0:B=CIEM:GOSUB 1000
DC 720 IF UTRZ=1 THEN 4000
XU 730 IF RND(0)<0.2 THEN GRAPHIC
   S 24
LU 740 M=TRYB:A=WIERZ:S=ODST
NK 750 GOTO 500
SH 997 REM
JZ 998 REM *** PODPROGRAM LINII P
   IONOWYCH
SN 999 REM
RI 1000 FOR K=B TO 319 STEP 2
HZ 1010 PLOT K,0:DRAWTO K,191:NEX
   T K
AF 1020 RETURN
KV 1998 REM
KZ 1999 REM
GM 2000 FOR K=0 TO 158 STEP ODST
ZX 2010 COLOR 1:PLOT 159,0:DRAWTO
    159+K,191:PLOT 159,0:DRAWTO 1
   59-K,191
UZ 2020 COLOR 0:PLOT 159,WIERZ:DR
   AWTO 160+K,191:PLOT 159,WIERZ:
   DRAWTO 158-K,191
HD 2030 NEXT K:RETURN
KW 2998 REM
LA 2999 REM
FB 3000 FOR K=0 TO 318 STEP ODST
JC 3010 COLOR 1:PLOT 159,WIERZ:DR
   AWTO K,191
NE 3020 COLOR 0:PLOT 159,0:DRAWTO
    K+1,191:NEXT K:RETURN
HZ 4000 REM
ID 4001 REM
ET 4010 OPEN #1,4,0,"K:"
BJ 4020 GET #1,X:IF X=72 THEN KOL
   =KOL+1
EQ 4030 IF KOL=16 THEN KOL=0
PI 4040 SETCOLOR 2,KOL,0:GOTO 402
   0
```

Część programu (linie 100 do 470) prawie w całości służy do ustawiania parametrów wzoru przez użytkownika, aby lepiej poznać w jaki sposób osiągane są różne efekty. W celu zmniejszenia liczby wprowadzanych linii, proszę podmienić pierwszą część programu pojedynczą linią: 100 GRAPHICS 24. Wprowadzenie rozpoczynamy od linii 500.

Program w zasadzie nie wymaga dodatkowych wyjaśnień. Najważniejsze informacje są w nim zawarte. Należy jednak podkreślić kilka rzeczy. Linie 500 do 545 wybierają zestaw parametrów losowych dla wzoru, który ma być kreślony. Zmienne JASN i CIEM skojarzone są z podprogramem kreślenia zestawu linii pionowych w wierszu 1000 programu. Są to zmienne o podwójnym przeznaczeniu: jeśli są równe 0 lub 1, wtedy kreślony będzie zestaw linii „parzystych" lub „nieparzystych", lecz jeśli większe niż 1, podprogram nie będzie wywołany. Zatem prawdopodobieństwo, że JASN spowoduje wywołanie podprogramu wynosi 0, 25, ponieważ losujemy liczbę całkowitą z przedziału 0 do 7. To samo dotyczy zmiennej CIEM.

Zmienna JASN ustala kolorowe tło dla wzoru, lecz daje nieco inny efekt, jeśli wzór nie został zniszczony przez linię 730. Zmienna CIEM powoduje wykasowanie wszystkich kolorów za wyjątkiem czarnego zanim program ponownie przejdzie do wyboru nowego zestawu parametrów losowych. Linia 535 wraz z 740 umożliwiają upewnienie się, że nowe wartości zmiennych TRYB, WIERZ i ODST są inne niż stare. Linia 550 chroni przed trybem zwracania uwagi (attract mode) dopóki program działa na nowych wartościach zmiennych. Za wyjątkiem możliwości zatrzymania wzoru przez użytkownika (linie 450 i 720), nie ma żadnych opóźnień czasowych. Jeden cykl programu trwa około minuty, musi to być wystarczające dla zaobserwowania aktualnego wzoru. Jeśli chcemy przyjrzeć się dokładniej, działanie programu może być wstrzymane i wznowione za pomocą klawisza **CTRL 1**.

PROGRAM NR 4

```
FC 50 GRAPHICS 0:POSITION 16,2:?
   "KLEJNOT":? :?
AV 55 ? "PO ZAKONCZENIU KRESLENIA
   --":?
PX 60 ? "WYLACZ SWIATLO W POKOJU,
    ABY"
YB 70 ? "UZYSKAC EFEKT ZABRUDZONE
   GO   "
TR 80 ? "OKNA.":?
UH 90 ? "NACISNIJ KLAWISZ H DLA Z
   MIANY"
BY 95 ? "KOLORU I/LUB WYREGULUJ N
   ASYCENIE":? "KOLOROW.":?
EG 100 ? "BOK KWADRATU?"
GO 110 ? "(MAKSIMUM 4*47--TRY 4*4
   6)"
QQ 120 TRAP 150:? "4*";:INPUT BOK
   :BOK=4*BOK
YL 130 IF BOK<1 OR BOK>188 THEN 1
   50
MN 140 TRAP 40000:GOTO 300
DE 150 PRINT CHR$(253):GOTO 50
YO 300 REM *** DEFINIOWANIE ROGOW
   KWADRATU
HD 310 XDO=INT(159.5-BOK/2):XGO=X
   DO+BOK
GK 320 YDO=INT(95.5-BOK/2):YGO=YD
   O+BOK
RX 349 REM
UQ 350 REM *** KRESLENIE TLA ***
RN 355 REM
HN 360 GRAPHICS 24:SETCOLOR 2,0,0
   :SETCOLOR 1,0,14:COLOR 1
ES 410 FOR X=XDO TO XGO
IR 420 PLOT X,YDO:DRAWTO X,YGO:NE
   XT X
RY 449 REM
KU 450 REM *** KRESLENIE KLEJNOTU
    ***
RO 455 REM
VJ 460 COLOR 0:FOR K=0 TO BOK STE
   P 4
DJ 470 PLOT XDO,YDO:DRAWTO XGO,YG
   O-K
GT 480 PLOT XDO,YDO:DRAWTO XGO-K,
   YGO
BU 490 PLOT XGO,YDO:DRAWTO XDO,YG
   O-K
CH 500 PLOT XGO,YDO:DRAWTO XDO+K,
   YGO
XA 510 PLOT XGO,YGO:DRAWTO XDO,YD
   O+K
AM 520 PLOT XGO,YGO:DRAWTO XDO+K,
   YDO
YX 530 PLOT XDO,YGO:DRAWTO XGO,YD
   O+K
EN 540 PLOT XDO,YGO:DRAWTO XGO-K,
   YDO
GZ 550 NEXT K
SJ 599 REM
KG 600 REM *** ZMIANA KOLORU:NACI
   SNIJ KLAWISZ H ***
RG 605 REM
PH 610 OPEN #1,4,0,"K:"
FS 620 GET #1,X:IF X=72 THEN KOLO
   R=KOLOR+1
RM 630 IF KOLOR=16 THEN KOLOR=0
HR 640 SETCOLOR 2,KOLOR,0:GOTO 62
   0
MO 1000 GOTO 1000
```

Program 4 ilustruje zmiany kolorów za pomocą gęstego wzoru ciemnych linii kreślonych na jasnym tle. Równoodległe promienie kreślone są z każdego z czterech rogów kwadratu do dwóch przeciwległych boków. Zapalone punkty, pozostające po zakończeniu tego procesu tworzą figurę składającą się z czterech segmentów w kształcie trąby. Obraz staje się interesujący, jeżeli najpierw wyregulujemy telewizor na przyjemną kombinację kolorów, a następnie zgasimy światło w pokoju.

Uwaga: Litery z lewej strony numerów linii nie wchodzą w skład programu. Są to sumy kontrolne dla Edytora Basic'u.

Atari Writer — edytor tekstów

Szacuje się, że najbardziej uniwersalnym i najczęściej wykorzystywanym profesjonalnym zastosowaniem mikrokomputerów jest ich używanie do pisania i redagowania tekstów. Służące do tego programy noszą nazwę edytorów tekstowych (ang. text editors) lub programów przetwarzania tekstów (ang. word processors). Do typowych zastosowań takich programów należą: pisanie artykułów, referatów, pism urzędowych, tłumaczeń, itp.

Typowy program przetwarzania tekstów przekształca komputer w specyficzną maszynę do pisania. Tekst wprowadzamy do pamięci komputera za pomocą klawiatury, obserwując na ekranie czy nie ma w nim pomyłek. Gotowy tekst przeglądamy na ekranie, a po naniesieniu ewentualnych poprawek oraz sprawdzeniu poprawności układu (stronice, tytuły, wcięcia, odstępy itp.) przenosimy na papier za pomocą drukarki. Możemy go również przechowywać w pamięci zewnętrznej (głównie dyskowej) w celu sporządzenia nowej wersji lub też jako część większej całości.

Programy przetwarzania tekstów oferują dodatkowe możliwości, znacznie ułatwiające pracę:

— Program wykonuje za nas pewne czynności redakcyjne — dotyczy to głównie decyzji o przejściu do nowego wiersza, wyrównanie lewego i prawego marginesu, itp.
— Po wprowadzeniu zmian (poprawieniu błędów, zmianie układu zdań, przeniesieniu fragmentów tekstu w inne miejsca, itp.) nie przepisuje się całego tekstu od nowa. Wystarczy podać odpowiednie rozkazy oraz wpisać w określone miejsca nowe fragmenty tekstu.
— Ze zgromadzonych tekstów można tworzyć różne kombinacje, na przykład przed wydrukowaniem zmienić tę część, która dotyczy konkretnego adresata.
— Wydruk kolejnej wersji tekstu uzyskuje się bardzo szybko — jest to niezwykle istotne przy projektowaniu wyglądu zewnętrznego poszczególnych stron. Dodatkowych możliwości dostarczają drukarki mozaikowe: różne kroje czcionek, druk wytłuszczony, rozszerzony i inne.

Do dobrego tonu należy, aby każdy komputer posiadał własny edytor tekstów (najczęściej unikalny, niepodobny do innych). Dzięki temu również większość mikrokomputerów domowych wyposażona jest w takie programy. Są one zwykle mniej wyrafinowane niż te do zastosowań profesjonalnych, lecz realizują podobny zestaw podstawowych funkcji. Dla komputerów Atari 800 XL, 130 XE I 65 XE takie funkcje wykonuje Atari Writer. Nie jest on jedynym edytorem tekstów dla mikrokomputerów tej firmy. Jest jednym z prostszych. Dodatkową zaletą, ważną dla większości posiadaczy tych popularnych komputerów, jest możliwość wykorzystania pamięci taśmowej do przechowywania programu oraz zbiorów tekstowych.

W dalszej części spróbujemy przedstawić funkcje realizowane przez program Atari Writer oraz sposoby ich wykorzystania.

Podstawowym elementem na ekranie jest kursor, migająca pozioma kreska o szerokości znaku, który wskazuje miejsce działania programu: pisanie nowych liter, usuwanie lub zamianę już wprowadzonych, tworzenie akapitu, itp. Położenie kursora określane jest przez jego współrzędne — numer wiersza i numer kolumny w wierszu (L i C w oknie komunikatu — w prawym dolnym rogu ekranu). Kursor może poruszać się po całym tekście. Można przemieszczać go o wiersz (**CTRL↓** i **CTRL↑**) lub całą stronę (**OPTION↓** lub **OPTION↑**), o jeden znak w prawo lub w lewo (**CTRL→** lub **CTRL←**), na początek tekstu (**SELECT T**) lub jego koniec (**SELECT B**).

Tekst pisze się jak na zwykłej maszynie do pisania, linia po linii. Nie używa się klawisza **RETURN**. Podczas wprowadzania słowa, które przekracza prawy margines ekranu, program automatycznie przenosi to słowo na początek następnej linii. Klawisz RETURN ma specjalne znaczenie, używany jest do zakończenia akapitu (którego początek oznaczamy poprzez **CTRL P**). Może być również używany do wstawiania do tekstu pustych linii.

W każdej chwili wprowadzony tekst można zmienić lub poprawić. Dopiero przy tym można docenić zalety redagowania tekstu za pomocą komputera. Podczas pisania na maszynie można poprawić tylko pojedyncze litery w gotowym maszynopisie (za pomocą taśmy korektorskiej). Wstawienie dodatkowych wyrazów lub zdań wymaga przepisania tekstu. Za pomocą edytora tekstu można kasować, wstawiać, wymieniać dowolne litery, wyrazy lub całe zdania. W celu poprawienia błędu lub wymiany słowa lub zdania, należy najpierw wykasować błędne. Używamy do tego klawiszy **CTRL DELETE**. Po wykasowaniu wprowadzamy poprawkę. Program automatycznie robi miejsce na tyle znaków, ile potrzebujemy dodać. Komputer odsuwa tekst napisany uprzednio, nie wymazując go i robiąc miejsce dla wprowadzanych znaków.

Kolejne funkcje odnoszą się do bloków tekstu. Za pomocą **CTRL X** oznaczamy początek i koniec takiego bloku. Tak zaznaczony blok może zostać przeniesiony (**OPTION M**) lub powtórzony (**OPTION D**) w innym miejscu, wskazanym przez kursor, lub wykasowany (**OPTION DEL. BACK SP.**). Interesującą właściwością programu jest przechowywanie przenoszonego, powtarzanego lub kasowanego bloku tekstu w specjalnym buforze. Dzięki temu możliwe jest odtworzenie go (**START INSERT**) w przypadku błędnego wykasowania. Pojemność tego buforu jest jednak ograniczona (maksymalnie 30 linii ekranu) — dlatego wybrane bloki nie powinny przekraczać tej wielkości.

Przydatną funkcją jest wyszukiwanie podanego ciągu (o długości nie większej niż 25 znaków) i zastępowanie go innym. W celu użycia tej funkcji należy ustawić kursor na początku tekstu i nacisnąć **SELECT S**. Komputer pyta, jaki ciąg ma znaleźć (**SEARCH FOR:...**), czy zastąpić go innym (**REPLACE STRING, Y/N?**) i czym go ewentualnie zastąpić (**REPLACE WITH:...**). Można wymieniać podany ciąg krok po kroku lub w całym zbiorze naraz (**REPLACE GLOBAL, Y/N?**). Do pracy krokowej należy nacisnąć **N i RETURN**, wtedy program podmieni tylko pierwsze wystąpienie ciągu i zapyta, czy kontynuować poszukiwania (**CONTINUE SEARCHING, Y/N?**). Po naciśnięciu **Y i RETURN** program odnajduje następne wystąpienie podanego ciągu i daje możliwość wyboru. Możemy przerwać lub kontynuować poszukiwanie i wymianę do końca tekstu. Opisana funkcja może być używana do wyszukiwania i kasowania. W tym celu po pytaniu o nowy ciąg należy nacisnąć **RETURN**. Znaleziony ciąg zostanie zastąpiony odstępem (spacją). Wyszukiwanie i wymiana mogą być przerwane w dowolnej chwili poprzez naciśnięcie klawisza BREAK. Należy pamiętać, że program rozpoczyna poszukiwania od aktualnej pozycji kursora.

Długość tekstu, który możemy wprowadzić do komputera, zależy od ilości wolnej pamięci. Podczas wprowadzania lub poprawiania tekstu dobrze jest sprawdzać od czasu do czasu, ile mamy jeszcze wolnego miejsca. W tym celu należy nacisnąć **OPTION F**. Program wyświetla odpowiedź wyrażoną w bajtach. W celu oszacowania ile to stronic, możemy przyjąć, że każda strona z podwójnymi odstępami wymaga 1500 bajtów.

Kolejną grupą funkcji są operacje na zbiorach. Tekst wprowadzony do komputera może być zapisany w pamięci zewnętrznej (dyskowej lub taśmowej) w celu przechowania dla późniejszego poprawienia, albo jako część większej całości. W tym celu należy przejść do menu (klawisz **ESC**), wybrać opcję przechowania zbioru (Save File). Program poprosi nas o podanie nazwy urządzenia i zbioru (**SAVE DEVICE: FILENAME**). Do stacji dysków trzeba włożyć sformatowaną dyskietkę i napisać **D:** nazwa zbioru. Nazwa może mieć długość do ośmiu znaków. Jeśli dyskietka nie jest sformatowana, program dysponuje opcją Format Disk, którą należy wywołać przed zapisaniem zbioru tekstowego.

W celu przechowania zbioru na taśmie magnetycznej musimy podać C: w odpowiedzi na propozycję podania nazwy urządzenia. Program zarządza włączenia magnetofonu na zapis. Należy włożyć kasetę, przewinąć w odpowiednie miejsce (zanotować stan licznika), wcisnąć klawisze **RECORD i PLAY** oraz **RETURN**.

Wprowadzanie zbioru tekstowego do pamięci komputera odbywa się podobnie. Z menu programu należy wybrać opcję wprowadzania (Load File). Po propozycji podania nazwy urządzenia i zbioru (**LOAD DEVICE: FILENAME**) musimy sprawdzić, czy odpowiednia dyskietka jest włożona do stacji dysków. Wtedy piszemy **D: nazwa zbioru**. Jeśli zbiór tekstowy przechowywany jest na taśmie magnetycznej, niezbędne jest ustawienie taśmy na początek zbioru (według stanu licznika) i zapisanie C:. Po sygnale dźwiękowym należy nacisnąć klawisze **PLAY** oraz **RETURN**.

W przypadku posiadania stacji dysków

przydatne mogą być dwie dodatkowe funkcje programu wybierane z menu: katalog zbiorów i kasowanie zbioru. W celu otrzymania katalogu zbiorów należy włożyć dyskietkę do stacji dysków i wybrać opcję **„Index of Disk Files"**. Program wyświetli katalog zbiorów w kolejności alfabetycznej nazw. Pokaże także, ile sektorów zajmuje każdy zbiór i ilość wolnych sektorów na dyskietce. Jednocześnie może być wyświetlone 21 nazw zbiorów, jeśli jest ich więcej następuje przesuwanie tekstu do góry, można je zatrzymać przez naciśnięcie klawisza spacji. Ponowny przesuw uruchamiany jest również klawiszem spacji. Katalog zbiorów może być wydrukowany po naciśnięciu **Y i RETURN** w odpowiedzi na pytanie **PRINT DIRECTORY, Y/N?**

W celu wykasowania zbioru musimy upewnić się, czy jest on na dyskietce włożonej do stacji dysków, przejść do menu programu i wybrać opcję kasowania zbioru. Następnie trzeba podać nazwę urządzenia i nazwę zbioru (**DELETE DEVICE: FILENAME**). Program upewnia się, czy na pewno chcemy wykasować ten zbiór (**FILE DELETE — ARE YOU SURE, Y/N?**). Po naciśnięciu **Y i RETURN** program wymazuje określony zbiór z dyskietki.

Możliwe jest również łączenie zbiorów tekstowych zapamiętanych na dyskietce lub kasecie z tekstem przechowywanym aktualnie w pamięci komputera. W tym celu należy ustawić kursor w miejscu, gdzie chcemy dołączyć tekst ze zbioru i nacisnąć **OPTION L**. Następnie trzeba podać nazwę urządzenia i zbioru. Dalsze postępowanie jest takie samo jak podczas wprowadzania zbioru.

Trzecią grupę tworzą funkcje przygotowania wydruku. Za pomocą programu Atari Writer możemy wydrukować tekst w dowolnej postaci, dzięki użyciu różnorodnych rozkazów formowania. Pewne z nich, wprowadzane w bloku formowania wydruku u góry ekranu, sterują globalnym kształtowaniem tekstu. Inne, wprowadzane wewnątrz tekstu, podczas tworzenia go lub poprawiania, używane są do określania odstępstw od ogólnego formatu.

Każda litera w negatywie, występująca u góry ekranu, reprezentuje rozkaz, który zmienia formowanie całego tekstu. Liczba z prawej strony litery jest wartością redagowania dla tego rozkazu. W celu ustawienia własnych wartości musimy przesunąć kursor do bloku formowania wydruku, wykasować liczbę, którą chcemy zmienić i wprowadzić nową. W ten sposób możemy ustawić własne wartości dla marginesu dolnego (**B**) i górnego (**T**), odstępu między akapitami (**D**), wcięcie akapitu (**I**), wyrównanie do prawego marginesu (**J**), margines lewy (**L**) i prawy (**R**), odstęp między liniami (**S**), długość stronicy (**Y**) oraz sposób wydruku (**G**).

Powyższe rozkazy mogą być również wprowadzone bezpośrednio do tekstu, można je wprowadzać pojedynczo lub łącznie, podczas tworzenia lub poprawiania tekstu. Program wyświetla odpowiednią literę i liczbę w negatywie (nie występują one na wydruku).

Mozliwe jest umieszczenie, na każdej stronicy wydruku, jedno- lub dwuliniowego nagłówka lub stopki. W szczególności można otrzymać numery drukowanych stronic. Dla określenia nagłówka musimy nacisnąć **CTRL H**, wprowadzić tekst i nacisnąć **RETURN**. Definiowanie stopki odbywa się podobnie, tylko naciskamy **CTRL F**. Domyślnie są one wyrównane do lewego marginesu drukowanej stronicy. Możemy jednak umieścić je na środku (po wprowadzeniu **CTRL C**) lub wyrównać do prawego marginesu (po wprowadzeniu **CTRL C CTRL C**).

W celu numerowania drukowanych stronic musimy nacisnąć @ (Shift 8) w tym miejscu nagłówka lub stopki, w którym ma wystąpić numer stronicy. Jeśli chcemy mieć tylko numery stronic, wprowadzamy @ jako cały tekst nagłówka lub stopki.

Możliwe jest obejrzenie uformowanego tekstu, w takiej postaci, jaka będzie na wydruku, bez drukowania. Służy do tego funkcja przeglądania wydruku, do której przechodzimy po naciśnięciu **OPTION P**. Najpierw musimy określić typ posiadanej drukarki. Następnie program pyta, czy chcemy przeglądać cały dokument, czy tylko wybrane stronice (**PREVIEW WHOLE DOCUMENT, Y/N?**). Naciskamy **Y i RETURN**, gdy chcemy przeglądać cały tekst lub **N i RETURN**, aby obejrzeć wybrane stronice (musimy wprowadzić numery stronic początkowej i końcowej). Po uformowaniu tekstu program wyświetla na ekranie lewy, górny róg pierwszej stronicy. Używając klawiszy sterowania kursorem możemy obejrzeć cały tekst (wskaźniki **L** i **C** pokazują położenie kursora na zredagowanej stronie, nie na ekranie). Należy pamiętać, że pewne tryby wydruku i odstępy między liniami i akapitami nie są dokładnie takie jak na wydruku.

W czasie przeglądania wydruku nie można poprawiać tekstu. Można to zrobić dopiero po przejściu do trybu poprawiania (poprzez naciśnięcie **START**).

Po wszechstronnym sprawdzeniu przygotowanego tekstu możemy przystąpić do drukowania. W tym celu musimy przejść do menu programu i wybrać opcję **Print File**. Podczas pierwszego wybrania tej pozycji musimy określić typ posiadanej drukarki. Następnie udzielamy odpowiedzi, czy chcemy drukować cały tekst (**PRINT WHILE DOCUMENT, Y/N?**). Po naciśnięciu **N i RETURN** możemy drukować wybrane stronice po podaniu numeru początkowego i końcowego. Musimy podać jeszcze liczbę drukowanych egzemplarzy (od 1 do 99).

Z powyższego, dosyć pobieżnego, przeglądu funkcji programu wynika, że posiada on spore możliwości w zakresie redagowania tekstów. Pewną niedogodnością jest konieczność pamiętania wszystkich funkcji i odpowiadających im poleceń. Program w niewielkim tylko stopniu służy nam pomocą. Poważniejszymi wadami, zresztą powszechnymi wśród zachodnich edytorów tekstów są: brak polskich liter na ekranie oraz w repertuarze znaków firmowej drukarki (Atari 1027) i poważne kłopoty w ich uzyskaniu w tym programie. Pomijamy już takie drobiazgi, jak niezbyt ładny krój czcionki (tworzony w polu 7 × 7), podniesienie do góry małych liter z ogonkami, mała liczba znaków w linii ekranu (40), itp. Pomimo tych wad program może pomóc w poznawaniu możliwości edytorów tekstu oraz służyć do modyfikowania programów w języku **BASIC** (przechowanych za pomocą instrukcji **LIST**).

Tomasz MROWIEC,
Ludwik PIELA

GRAFIKA NA SPECTRUM

```
 20 LET z=8
 30 FOR i=0 TO 255 STEP z
 40 PLOT i,0
 50 DRAW 255-2*i,175
 60 NEXT i
 70 FOR i=0 TO 175 STEP z
 80 PLOT 0,i
 90 DRAW 255 ,175-2*i
100 NEXT i
```

MINI-DOS

Każdy programista, wykorzystujący wbudowany w ROM mikrokomputera ATARI 800XL translator języka BASIC z dyskowym systemem operacyjnym (DOS), styka się często z irytującą sytuacją. Dochodzi do niej wtedy, gdy po zakończeniu pisania programu, programista chce zapisać go do zbioru na dyskietce, lecz nie wie, czy jest na niej wystarczająco dużo miejsca, lub musi użyć nowej dyskietki, lecz ta nie jest sformatowana. Mało tego, czasami powstaje problem, jaką nazwę nadać nowo tworzonemu zbiorowi, aby nie powtórzyć nazwy zbioru zapisanego już na dyskietce. W takiej sytuacji nie pozostaje nic innego, jak „wejść pod DOS-a", składując najpierw zawartość pamięci do zbioru MEM.SAV, sformatować nową dyskietkę, bądź „obejrzeć" listę zbiorów na dyskietce, na której chcemy zapisać nowy program, następnie wrócić do BASIC-a, odtwarzając zawartość zbioru MEM.SAV. Wszystkie te operacje pochłaniają wiele cennego czasu. Analogicznie wygląda problem, gdy w opisanej sytuacji chcemy skasować zbiór, zmienić jego nazwę, zabezpieczyć zbiór przed przypadkowym zniszczeniem (LOCK), lub zdjąć zabezpieczenie ze zbioru (UNLOCK).

Bardzo pomocnym w takich sytuacjach jest prezentowany poniżej program o nazwie MINI — DOS. Umożliwia on wykonywanie z poziomu BASICa funkcji normalnie dostępnych w pakiecie DUP (Disc Utility Package) DOS-a. Zasada działania programu jest bardzo prosta. Gdy zostanie on uruchomiony (RUN) zapisuje na dyskietce pięć zbiorów o nazwach: „DIR", „DELETE", „LOCK", „UNLOCK", „RENAME". Do każdego z wymienionych zbiorów program wprowadza po kilka linii programu w języku BASIC, realizujących odpowiednią funkcję. Wprowadzone do zbiorów linie programowe nie są numerowane. Efekt tego rozwiązania jest taki, że gdy zapiszemy komendę np:
ENTER „D:DIR" RETURN w negatywie to komputer zacznie wprowadzać do pamięci RAM linie programowe, zapisane w zbiorze „DIR", a że nie mają one numerów, komputer będzie je natychmiast wykonywał, nie niszcząc aktualnej zawartości RAM-u.

MINI — DOS ma następujące opcje:

E. „D:DIR" — listuje na ekranie wykaz zbiorów na dyskietce.
E. „D:RENAME" — zmienia nazwę zbioru.
E. „D:DELETE" — kasuje zbiór.
E. „D:LOCK" — „nakłada zabezpieczenie" na zbiór.
E. „D:UNLOCK" — „zdejmuje zabezpieczenie" ze zbioru.

Nazwy opcji zostały specjalnie dobrane w języku angielskim, ponieważ najbardziej kojarzą się z odpowiednimi opcjami z MENU DOS-a.

Program bardzo łatwo można rozbudować, wprowadzając do niego dodatkowe funkcje. Np., gdy zastąpimy linię 130 linią poniższą:

```
130 RESTORE 120:GOSUB 140GOTO 200
```

i dodamy dodatkowe linie 200—240

```
200 CLOSE#1:OPEN#1,8,0, „D:FORMAT"
210 DATA ?:? „WŁÓZ DYSKIETKE DO FORMATOWANIA":
  ? „I NACISNIJ START ! CZEKAM..."
220 DATA FOR I=1 TO 3000:IF PEEK (532
  79) <> 6 THEN NEXT I:END
230 DATA ?„FORMATUJE...";XIO 254&#1&0&0&"D: „:?:?"...
  FORMATOWANIE ZAKONCZONE !„ , —1
240 RESTORE 210:GOSUB 140
```

to program zapisze na dyskietce sześć zbiorów (pięć wcześniej wymienionych i szósty o nazwie „FORMAT"), a komenda E. „D:FORMAT RETURN umożliwi formatowanie dyskietki. Ale UWAGA! zalecam szczególną uwagę przy wprowadzaniu linii 200—240 i ostrożność przy ich testowaniu. Jeden drobny błąd może spowodować zniszczenie cennych informacji zapisanych na dyskietce. Postępując podobnie, można dobudować do MINI—DOS-a funkcję kopiowania zbiorów na dyskietce.

Po uważnym wprowadzeniu do pamięci mikrokomputera wszystkich linii programu należy zapisać go na oddzielną dyskietkę. Program najlepiej wykorzystywać według poniższej zasady. Zawsze po sformatowaniu nowej dyskietki, należy załadować MINI—DOS do pamięci i uruchomić go wkładając do stacji dysków nowo sformatowaną dyskietkę. MINI—DOS zapisze na niej pięć lub sześć (gdy dopisane zostaną linie 200—240) zbiorów o wcześniej wymienionych nazwach. Od tego momentu, gdy dyskietka ta będzie znajdowała się w napędzie dyskowym, będzie można używać opcji MINI—DOS-a z poziomu translatora BASIC-a. Wynika stąd wniosek, że chcąc mieć zawsze „pod ręką" powyższe opcje, zbiory tworzone przez MINI—DOS powinny znajdować się na każdej dyskietce, której będziemy używać pracując z translatorem BASIC-a. Jest to wniosek słuszny. Nie należy z tego powodu martwić się o racjonalne gospodarowanie pojemnością dyskietki. Zbiory zapisywane przez MINI—DOS zajmują razem 10—12 sektorów (dla DOS 2.5 każdy zbiór zajmuje dwa sektory), jest to naprawdę niewielka strata, zważywszy na wygodę programowania, jakiej dostarcza ten program.

opracował H. KRASUSKI

```
1 REM ***************************
2 REM MINI - DOS
3 REM ***************************
20 GRAPHICS 0:? "PROSZE CHWILE P
OCZEKAC..."
100 DIM A$(200)
110 OPEN #1,8,0,"D:DIR"
120 DATA CLR:CLOSE#1:DIM A$(20):
OPEN #1&6&0&"D:*.*"
125 DATA ? CHR$(28);"ZBIORY NA D
YSKIETCE":?:FOR X=1 TO 64:INPUT
#1;A$:? A$
126 DATA :IF A$(1&1)=" " OR A$(1
&1)="*" THEN NEXT X,-1
130 RESTORE 120:GOSUB 140:GOTO 2
60
140 READ A$:I=LEN(A$):IF A$="-1"
 THEN RETURN
150 FOR X=1 TO I
160 IF A$(X,X)="&" THEN A$(X,X)=
","
170 NEXT X
180 ? #1;A$:GOTO 140
250 REM
260 CLOSE #1:OPEN #1,8,0,"D:DELE
TE"
270 DATA CLR:CL.#1:CL.#2:DIM A$(
12)&B$(14):O.#1&4&0&"E:":
271 DATA ? CHR$(29);"PODAJ NAZWE
 ZBIORU":? "KTORY MA BYC SKASOWA
NY!"
280 DATA ?CHR$(28);"NAZWA ZBIORU
: ";:I.#1;A$:B$(1&2)="D:":X=LEN(
A$):B$(3&X+2)=A$(1&X)
281 DATA :X.33&#2&0&0&B$:?:?A$;"
 SKASOWANY!"
285 DATA -1
290 RESTORE 270:GOSUB 140
310 CLOSE #1:OPEN #1,8,0,"D:LOCK
"
320 DATA CLR:CL.#1:CL.#2:DIM A$(
12)&B$(14):O.#1&4&0&"E:":?CHR$(2
9);"PODAJ NAZWE ZBIORU"
321 DATA :?"KTORY MA BYC ZABEZPI
ECZONY"
330 DATA ?CHR$(28);"NAZWA ZBIORU
: ";:I.#1;A$:B$(1&2)="D:":X=LEN(
A$):B$(3&X+2)=A$(1&X)
331 DATA :X.35&#2&0&0&B$:?:?A$;"
 ZABEZPIECZONY
340 DATA -1
350 RESTORE 320:GOSUB 140
370 CLOSE #1:OPEN #1,8,0,"D:UNLO
CK"
380 DATA CLR:CL.#1:CL.#2:DIM A$(
12)&B$(14):O.#1&4&0&"E:":? CHR$(
29);"PODAJ NAZWE ZBIORU"
381 DATA :?"KTORY MA BYC ODBEZPI
ECZONY"
390 DATA ? CHR$(28);"NAZWA ZBIOR
U: ";:I.#1;A$:B$(1&2)="D:":X=LEN
(A$):B$(3&X+2)=A$(1&X)
391 DATA :X.36&#2&0&0&B$:?:?A$;"
 ODBEZPIECZONY
400 DATA -1
410 RESTORE 380:GOSUB 140
430 CLOSE #1:OPEN #1,8,0,"D:RENA
ME"
440 DATA CLR:CL.#1:CL.#2:DIM A$(
25)&B$(27):OPEN#1&4&0&"E:":B$(1&
2)="D:"
445 DATA ? CHR$(29);"PODAJ&ODDZI
ELAJAC PRZECINKIEM&":?"STARA I N
OWA NAZWE ZBIORU!"
450 DATA ? CHR$(28);"STARA NAZ.,
NOWA NAZ.: ";:I.#1;A$:X=LEN(A$):
B$(3&X+2)=A$(1&X):X.32&#2&0&0&B$
451 DATA :?:?"NAZWA ZBIORU ZMIEN
NA
455 DATA -1
460 RESTORE 440:GOSUB 140
500 END
```

Program kopiujący

Program umożliwia kopiowanie programów napisanych w języku wewnętrznym komputera. Po uruchomieniu instrukcją RUN, program sygnalizuje pojedynczym sygnałem (beep) gotowość do wczytywania oryginału z kasety. Naciśnięcie, np. klawisza RETURN, powoduje otwarcie kanału 1 w L.175 na czytanie. Gdy czytanie programu zostanie zakończone, komputer powiadomi podwójnym sygnałem dźwiękowym, że może rozpocząć wykonywanie kopii. Nastąpi to w momencie otwarcia kanału 1 jako wyjścia w L.260, przez naciśnięcie np. klawisza RETURN. Po wykonaniu kopii następuje powrót do sytuacji umożliwiającej wprowadzenie następnego programu. **J. J.**

```
65 REM ********************
70 REM *                  *
75 REM *PROGRAM KOPIUJACY*
80 REM *                  *
85 REM *  KASETA-KASETA   *
90 REM *                  *
95 REM ********************
100 DATA 104,174,138,2,134,61,16
0,0,162,0,185,0,4,129,203,200,23
0
105 DATA 203,208,2,230,204,196,6
1,240,3,76,10,6,96
110 DATA 104,169,128,133,61,160,
0,162,0,161,203,153,0,4,200,230
115 DATA 203,208,2,230,204,196,6
1,240,3,76,9,6,198,61,96
120 GRAPHICS 17
125 P=PEEK(742)*256+PEEK(741)-PE
EK(145)*256+PEEK(144)-1500
130 IF P>32767 THEN P=32767
135 DIM P$(P):P$(1)=CHR$(0):P$(P
)=CHR$(0):P$(2)=P$
140 POKE 203,ADR(P$)-(INT(ADR(P$
)/256)*256):POKE 204,INT(ADR(P$)
/256)
145 FOR A=1536 TO 1565:READ D:PO
KE A,D:NEXT A
150 TRAP 200
155 ? #6;" CZYTANIE PROGRAMU"
160 ? #6;" Z KASETY"
165 ? #6:? #6;" wcisnij dowolny
"
170 ? #6;" klawisz"
175 OPEN #1,4,255,"C:":B1=0
180 FOR A=B1 TO P STEP 128
185 GET #1,W:B1=B1+128:X=USR(153
6)
190 NEXT A:? #6;"brak pamieci RA
M do"
195 ? #6;"skopiowania programu":
END
200 IF PEEK(195)=138 THEN CLOSE
#1
205 GRAPHICS 17
210 ? #6;" WGRYWANIE PROGRAMU"
215 ? #6;" NA KASETE"
220 ? #6:? #6;" wcisnij dowolny"
222 ? #6;" klawisz"
225 GOTO 235
230 ? "BLAD - ";PEEK(195):END
235 RESTORE 110
240 FOR A=1536 TO 1566
245 READ D:POKE A,D:NEXT A
250 POKE 203,ADR(P$)-(INT(ADR(P$
)/256)*256)
255 POKE 204,INT(ADR(P$)/256)
260 CLOSE #1:OPEN #1,8,255,"C:"
265 FOR A=0 TO B1 STEP 128
270 X=USR(1536)
275 P=ASC(P$(A+128)):PUT #1,P
280 NEXT A
285 CLOSE #1:GRAPHICS 17
290 ? #6:? #6;" PROGRAM"
295 ? #6;" SKOPIOWANY !"
300 FOR C=0 TO 1000:NEXT C
305 RUN
```

Prawie trzy lata temu, wiosną 1984 roku wdarł się przebojem na europejski rynek mikrokomputerowy AMSTRAD CPC 464. Mało znana firma, która dotąd zajmowała się produkcją domowego sprzętu elektronicznego, zmieniła swoje zainteresowania i natychmiast odniosła ogromny sukces. AMSTRAD stał się popularny w Wielkiej Brytanii, Francji, w Hiszpanii, a nawet w Stanach Zjednoczonych. A niemieckojęzyczną częścią Europy „zajął się" równolegle jego RFN-owski bliźniak SCHNEIDER.

POGWARKI O CPC 464

Znakomicie pamiętam moje pierwsze spotkanie z tymi urządzeniami. Odbyło się ono podczas Dni Kultury Mikroinformatycznej w klubie Abakus. Młodzi ludzie prezentujący owe komputery budzili ogromne zainteresowanie oraz cichą zazdrość zarówno zaawansowanych „mikrofanów", jak i przybyłych incognito informatyków. Te dwadzieścia kilka miesięcy, które minęły od tamtego czasu — prawie epoka w świecie mikrokomputerów — sprawiły, że poczciwa „464" mocno się zestarzała, zdążyła jednak zdobyć dużą popularność również w Polsce. Nic w tym dziwnego, gdyż maszyna to nie byle jaka, bez nadzwyczajnych, nie zawsze sprawdzonych nowinek technicznych, ale i bez większości błędów, jakie dotąd popełniali konkurenci. Słowem — „licząca solidność". W owym czasie parametry techniczne stawiały ją w czołówce mikrokomputerów domowych. Bardzo dobry interpreter BASICa, niewygórowana cena oraz to, że był sprzedawany w zestawie z magnetofonem i monitorem (co ogranicza do niezbędnego minimum nieporozumienia z domownikami i sprzętem) pozwalały sądzić, że długo jeszcze świat (a przynajmniej Europa) będzie stać przed nią otworem. Tym bardziej iż wspomniane zalety AMSTRADa oraz termin promocji sprzętu przyczyniły się w sporym stopniu do chłodnego przyjęcia mikrokomputerów standard MSX. Zagrożenie nadeszło z najmniej oczekiwanej strony. W kilka miesięcy później AMSTRAD wypuścił kolejny model — CPC 664, a po roku CPC 6128. *„Chcąc wygrywać z innymi, musimy być konkurencyjni dla siebie samych"* — usprawiedliwiał się w jednym z wywiadów Alan Sugar — naczelny animator firmy... i w ciągu trzech lat przygotował do produkcji kolejne typy mikrokomputerów: PCW 8256, PCW 8512 oraz „hit" ostatnich miesięcy IBM i podobny PC 1512. We wszystkich modelach, z wyjątkiem ostatniego, widać duży wpływ rozwiązań przyjętych w „czterysta sześćdziesiątce czwórce". Szczególnie zbliżone do niej są modele CPC 664 i CPC 6128, będące dla niej coraz mocniejszą konkurencją.

Nie będę pisał, jaka to porządna komputerowa rodzina — po prostu trochę się zżyliśmy. Sądzę zresztą, że nie tylko my... Pozwolę sobie zatem, zadedykować wszystkim *„wspaniałym mężczyznom (i nie tylko) na tych wspaniałych maszynach"* kilka kolejnych pogawędek poświęconych AMSTRADowi CPC 464. Ta niezobowiązująca forma umożliwia mi dosyć swobodne potraktowanie tematu.

POGAWĘDKA PIERWSZA — COŚ NIECOŚ O „ŻELASTWIE"

Na początku, zamiast cytować starożytnych, przytoczę wypowiedź Bruce'a Goddena z Locomotive Software, dotyczącą założeń, jakie postawiono konstruktorom pierwszego AMSTRADa. *„Naszym celem było stworzenie wyrafinowanego komputera, łączącego dwie rzadko chodzące w parze cechy: jak najmniejszy koszt HARDWAREu i możliwie największą szybkość pracy. Byliśmy także zobowiązani udostępnić wszystkie możliwości maszyny programistom piszącym w BASICu"*.

Przyjęto, że sercem (a może raczej mózgiem) komputera będzie tani, bardzo popularny i co istotne, dobrze znany mikroprocesor Z80A. Wybór tego układu był podyktowany nie tylko jego zaletami technicznymi, chociaż znam kilku fachowców, którzy do dziś uważają go za najlepszy, popularny procesor ośmiobitowy. Liczono przede wszystkim na szybkie pozyskanie oprogramowania. Już na etapie projektowania zakładano, że w zestawach z dołączoną pamięcią dyskową będzie powszechnie wykorzystywana jedna z wersji systemu operacyjnego CP/M. Konsekwentnie — dyskietka z CP/M 2.2 była dodawana przy zakupie stacji dysków. Niestety, brak konsekwencji dał znać o sobie przy doborze standardu nośnika. Zastosowano zupełnie nowy i drogi (co prawda dobrze zabezpieczony) dysk 3-calowy, zamiast powszechnie używanej „piątki". Muszę wyznać, że do dziś nie wybaczyłem tego posunięcia Alanowi Sugarowi.

Wróćmy jednak do środka, do „kości" — ich powiązania funkcjonalne przedstawione zostały na zamieszczonym obok schemacie.

Ponieważ Z80 jest układem powszechnie znanym i doczekał się odpowiedniej „literatury" w języku polskim, przypomnę tylko, że jego praca w AMSTRADzie jest synchronizowana sygnałem zegarowym o częstotliwości 4 MHz (±0,1%).

Najważniejszym, oczywiście po mikroprocesorze, układem komputera jest **Video** GATE ARRAY (często zdrobniale nazywany GATE ARRAY). Można by przyrównać go do ULA w ZX SPECTRUM. Zawarte w nim układy wykonują elementarne działania logiczne i realizują funkcję, od których bezpośrednio zależą niemal wszystkie operacje komputera. GATE ARRAY po prostu koordynuje pracę systemu. W szczególności zaś uaktywnia lub blokuje pamięć ROM, nadzoruje pracę w określonym trybie graficznym, a co się z tym wiąże ponosi odpowiedzialność za „gospodarowanie". W powiązaniu z kontrolerem monitora ekranowego (układ HD 6845S CRTC) generuje sygnały video, synchronizuje i wymusza zmiany obrazu, zgodnie z potrzebami aktualnie realizowanego programu. Istotną funkcją GATE ARRAY jest także zerowanie odpowiednich rejestrów i uaktywnianie procedur obsługi w razie wyłączenia zasilania lub zerowania mikrokomputera przez jednoczesne naciśnięcie klawiszy CTRL, SHIFT i ESC.

Problem generowania sygnałów akustycznych rozwiązano w AMSTRADzie, stosując układ scalony AY—3—8912 firmy General Instruments, oznaczony na schematach literami PSG (Programable Sund Generator). Ma on trzy niezależne kanały dźwięku oznaczone literami A, B i C oraz pseudolosowy generator szumu, dołączany do dowolnego z nich. Kanały A i C to dwa oddzielne tory stereofoniczne, a dźwięk z kanału B jest miksowany z sygnałami w obydwu torach. Efekt stereofonii można uzyskać jedynie przy użyciu zewnętrznego wzmacniacza. Sygnał wprowadzany jest wówczas przez łącze zawsze „małym JACKiem". Wbudowany w komputer głośnik współpracuje jednocześnie ze wszystkimi kanałami, a jego parametry znacznie obniżają jakość generowanego dźwięku. I jeszcze jedno, sygnały akustyczne, jakie powstają podczas wczytywania programów z taśmy kasetowej, nie są niestety wyprowadzane na zewnątrz.

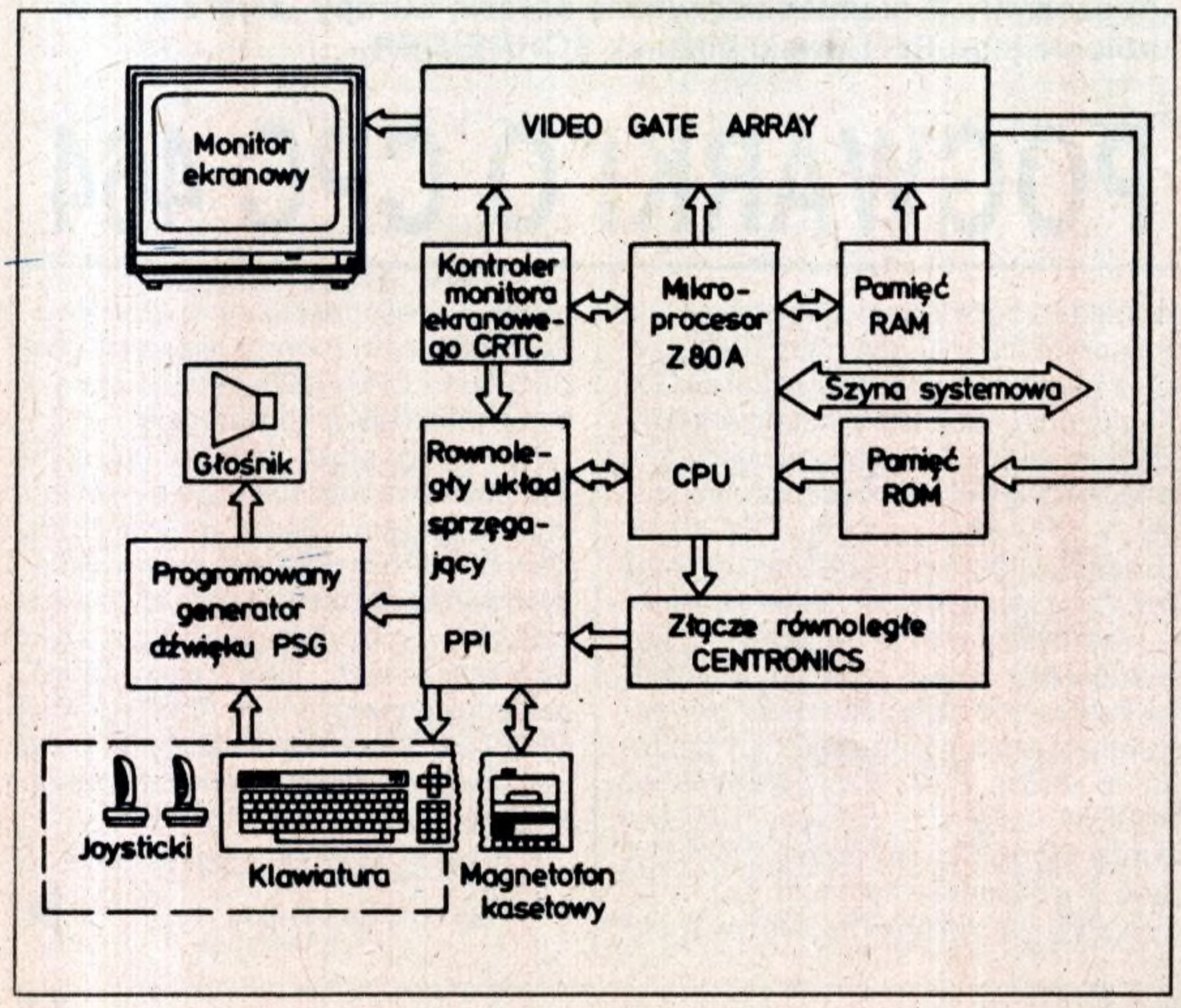
W układzie scalonym programowanego generatora dźwięku znajduje się również I/O PORT — element wejścia-wyjścia, przez który są wprowadzane informacje z klawiatury i JOYSTICKów. Port ten współpracuje z równoległym układem sprzęgającym PPI (Parallel Paripheral Interface — układ scalony 8255). Nazwa równoległy oznacza, że pozwala on przekazywać dane po ośmiu liniach równocześnie. Za pośrednictwem PPI komputer korzysta z pamięci na taśmie kasetowej, steruje pracą silnika magnetofonu, a przy użyciu połączonego z nim i mikroprocesorem złącza równoległego CENTRONICS może współpracować z większością drukarek, przesyłając informacje w przyjętym jako standard międzynarodowy amerykańskim kodzie ASCII.

Konstruktorzy AMSTRADa przewidzieli także potrzebę rozbudowy systemu komputerowego o różnego rodzaju urządzenia dodatkowe, jak np.: modem, łącze szeregowe, moduły pamięci, pióro świetlne, a przede wszystkim pamięć dyskowa. Do tego celu przeznaczona została łącząca się bezpośrednio z układem mikroprocesora szyna systemowa. Podłączyć do niej można równocześnie wiele różnych elementów, nakładając jedna na drugą łączówki kolejnych urządzeń.

Niestety, kiedy wyobrażę sobie, co mógłbym połączyć do zaprzyjaźnionego CPC i jaką to sprawiłoby nam obydwu radość, zaczynają drżeć mi ręce i zupełnie nie mogę się skupić. Nie pomagają proszki od bólu głowy. Dopiero wielokrotne powtarzanie półgłosem, że: nawet najprostszy AMSTRAD może dać mi wiele radości — tylko należy poświęcić mu trochę czasu — przynosi pewną ulgę.

Siadamy zatem do przygotowania **POGAWĘDKI DRUGIEJ — FIRMWARE, ALEŻ TO NIE TAKIE PROSTE.**

ROM-man



Komputer w medycynie

Tomografia komputerowa

W 1895 roku niemiecki fizyk Wilhelm Roentgen wykonał, przy użyciu promieniowania o nieznanej wówczas naturze, zdjęcie układu kostnego dłoni człowieka, otwierając tym samym drogę do bezpośredniej penetracji ludzkiego ciała bez konieczności użycia skalpela. Uzyskana w ten sposób fotografia, oczywiście czarno-biała i nieostra, była efektem utrwalenia na kliszy fotograficznej zjawiska pochłaniania promieniowania X przez duże cząstki, w tym wypadku wapnia — efekt cienia na 1 części rysunku.

Przy takiej technice prześwietleń obraz uzależniony jest od kąta widzenia, czyli kierunku padania wiązki promieniowania. Ponadto zdjęcie rentgenowskie, przy bardziej złożonej strukturze wewnętrznej organizmu może być trudne do interpretacji, gdyż bliższe organy, pochłaniające promieniowanie, mogą zakrywać następne, głębiej położone.

Kolejne dziesięciolecia badań naukowych przyniosły wyjaśnienie natury użytego promieniowania, poprawę jakości uzyskiwanej fotografii i coraz bliższą urzeczywistnienia myśl o nowej metodzie rentgenografii, nie obarczonej brakami metody klasycznej. Myślano zatem coraz śmielej o kolorowym obrazie trójwymiarowym badanego narządu wewnętrznego, przy czym już nie w formie zdjęcia, lecz na ekranie telewizora. Trzeba było jednak na to czekać aż do 1972 roku, gdy geniusz ludzki połączył pozornie odległe techniki: badań rentgenowskich, otrzymywania obrazu video i komputerowego przetwarzania danych. Rezultat, który przyniósł autorowi nagrodę Nobla nazwano tomografem komputerowym. Centralną częścią tego urządzenia jest bęben obrotowy (na rysunku część 2) z trzema źródłami promieniowania rentgenowskiego i setkami czujników tego promieniowania umieszczonymi symetrycznie na jego obwodzie. Pacjent poddawany badaniu umieszczany jest wewnątrz bębna, który powoli obracając się i przesuwając wzdłuż ciała człowieka umożliwia wykonanie serii zdjęć rejestrujących milimetr po milimetrze organy ludzkiego ciała w przekroju poprzecznym. Wyniki kolejnych prześwietleń, zapisane w kodzie cyfrowym wraz z dokładnym określeniem położenia urządzeń naświetlających względem pacjenta, są przechowywane w pamięci komputera. Komputer dysponuje zatem setkami zdjęć badanego narządu wewnętrznego, widzianego z każdej możliwej strony. W tym momencie decydujący staje się kunszt programistów, którzy stworzyli oprogramowanie komputera, będącego częścią składową tomografu. Zebrane dane, stanowiące ułamkowe obrazy anatomiczne w kodzie maszynowym, przetwarzane są bowiem w jeden trójwymiarowy obraz widoczny na monitorze operatora tomografu. Dla podniesienia czytelności obrazu jest on automatycznie nasycany kontrastującymi kolorami. Uwydatnia to subtelne różnice między narządami, doskonale widocznymi na przyciemnionym tle.

Efekt podróży do wnętrza ciała ludzkiego przy użyciu tomografu komputerowego przypomina do złudzenia badanie przez płetwonurków dna morskiego z jego złożoną florą i fauną. Bogata paleta grafiki komputerowej pozwala przykładowo na eksplorację kręgosłupa przypominającego na ekranie wielką rafę koralową, spękaną i pełną ubytków w przypadku ciężkich urazów.

Lekarz śledząc obraz tomograficzny pacjenta bez trudu natrafi na ślad postępującego zniekształcenia organów, wywołanego nowotworem, co umożliwia szybką diagnozę i bezbłędną akcję chirurgiczną. Niestety tą metodą nie można wykryć wczesnej fazy choroby nowotworowej, fazy, w której nie występują jeszcze zmiany kształtu zaatakowanego narządu — choć i na to jest już metoda, której poświęcimy następną relację ze skomputeryzowanego świata medycyny. Dla podkreślenia wagi tomografii komputerowej nie można nie wspomnieć o innym bardzo ważnym jej zastosowaniu. Dane przechowywane w pamięci komputera zebrane przez tomograf podczas prześwietlenia pacjenta mogą być bardzo przydatne w chirurgii, nie bez przyczyny nazywanej rekonstruującą. Sprzężenie precyzyjnej obrabiarki z komputerem pamiętającym wyniki badania tomograficznego umożliwiają stworzenie plastycznej kopii dowolnej części kości. Chirurg planujący przykładowo operację miednicy dysponuje zatem na podstawie obrazu tomograficznego zdrowej części ciała precyzyjną kopią operowanego elementu, co niezmiernie ułatwia mu pracę.

I pomyśleć, że Roentgen opowiadanie o tomografie komputerowym potraktowałby jako powiastkę science-fiction!

AT

Gra w oczko

Proponujemy Wam grę hazardową. Hasło „hazard" pochodzi z francuskiego, gdzie oznacza: przypadek, traf, zdarzenie losowe. Tak więc formalnie: gra hazardowa = gra przypadkowa.

Rzetelną grą przypadkową powinien rządzić „ślepy los". Na tej zasadzie działają kolektury loteryjne i Toto-Lotek. Ale bywa, że ktoś świadomie *„pomaga losowi"*. W takim wypadku mamy do czynienia z *oszustwem* (vide: Kodeks karny).

Gra w „oczko" wymaga talii z 24 kart (od asów do dziewiątek). Mają one następujące wagi: as — 11 punktów („0 — oczek"), dziesiątka — 10 pkt., król — 4, dama — 3, walet — 2, zaś dziewiątka — zero punktów. Kolor nie gra roli.

Jeden gracz (bankier) wydaje po jednej karcie, zaś drugi gracz starannie kalkuluje:

— jeśli ma 21 punktów (czyli „oczko") to zawsze wygrywa,

— jeśli ma ponad 21 punktów (czyli „furę"), to zawsze przegrywa.

— jeśli ma poniżej 21 punktów (przykładowo 17) to „czeka" i wtedy losuje bankier.

W tym pojedynku zwycięża ten, kto ma więcej punktów.

W przypadku remisu (dwa razy po 17 lub dwie „fury") wygrywa bankier.

Jeśli bankier wylosuje kolejno dwa asy i ma 22 punkty to wygrywa („bankierskie oko").

Program wywołuje etykieta „X" (RUN 10).

Ludowe porzekadło głosi „...nie za to ojciec prał syna, że grał w karty, ale za to, że się ogrywał...".

Zweryfikujcie: u nas nie ma ograniczeń stawek!

Janusz MILLER

SHARP 1500-PC

```
10:"X":CLEAR :
   RANDOM :USING
   :WAIT 0
20:DIM T(6,4),J(2
   4),P(24),H$(6)
   ,K$(4)
30:DATA " 9 "," W
   ALET "," DAMA
   "," KROL "," 1
   0 "," AS "
40:DATA "KIER","K
   ARO","TREFL","
   PIK"
50:RESTORE 30:FOR
   L=1TO 6:READ H
   $(L):NEXT L
60:RESTORE 40:FOR
   L=1TO 4:READ K
   $(L):NEXT L
70:U=0:V=0:F$="FU
   RA ! ":P$=" PK
   T.":O$="*OKO*
   "
80:INPUT "JAKA ST
   AWKA ? ";S
90:IF S<=0THEN
   BEEP 7,100,300
   :GOTO 80
100:WAIT 0:D=0:
   INPUT "DAC KAR
   TE ? T=1 N=0 "
   ;D
110:IF D=0GOTO 150
120:GOSUB "A":X=X+
   1:J(X)=Z:PRINT
   "TY ";H$(H);K$
   (K);
130:GOSUB "C":U=U+
   Q:IF U=21THEN
   CLS :PRINT O$;
   :GOTO 300
140:WAIT 100:PRINT
   " :";U;P$:CLS
   :WAIT 0:GOTO 1
   00
150:GOSUB "A":Y=Y+
   1:P(Y)=Z:PRINT
   "ON ";H$(H);K$
   (K);
160:GOSUB "C":V=V+
   Q:IF V=21THEN
   CLS :PRINT O$:
   GOTO 310
170:WAIT 100:PRINT
   " :";V;P$:CLS
   :WAIT 0:IF (U>
   VAND V<17)GOTO
   150
180:IF Y=0GOTO 200
190:IF (INT P(Y)+
   INT P(Y-1))=12
   THEN PRINT O$;
   :GOTO 310
200:IF U>21THEN
   PRINT F$;:GOTO
   310
210:IF V>21THEN
   PRINT F$;:GOTO
   300
220:IF U>VGOTO 300
230:IF V>=UGOTO 31
   0
300:WAIT 0:PRINT "
   W Y G R A L E
   S ":BEEP 20,25
   ,25:M=M+S:GOTO
   320
310:WAIT 0:PRINT "
   P R Z E G R A
   L E S ":BEEP 3
   ,100,300:M=M-S
320:WAIT 100:PRINT
   "R E Z U L T A
   T"
330:IF M<0THEN
   PRINT "S T R A
   T A :";ABS M:
   GOTO 350
340:PRINT "Z  Y  S
   K :";M
350:IF X>23OR Y>23
   GOTO 500
360:D=0:INPUT "GRA
   MY ? T=1 N=0 "
   ;D
370:IF D=1GOTO 70
380:END
390:"B":H=INT Z:K=
   (Z-H)*10:
   RETURN
400:"A":H=RND 6:K=
   RND 4:IF T(H,K
   )=1AND F<>24
   GOTO 400
410:Z=(K/10)+H:T(H
   ,K)=1:F=F+1:IF
   F<25RETURN
420:WAIT 0:PRINT "
   KONIEC TALII":
   BEEP 10,50,100
   :GOTO "D"
430:"C":IF H=1LET
   Q=0
440:IF (H=2)OR (H=
   3)OR (H=4)LET
   Q=H
450:IF H=5LET Q=10
460:IF H=6LET Q=11
470:RETURN
500:"D":WAIT 50:T$
   =" TY ":O$=" O
   N ":PRINT "K
   A  R  T  Y"
510:FOR L=1TO 24
```

```
520:IF J(L)=0GOTO
    540
530:Z=J(L):GOSUB "
    B":PRINT STR$
    L;T$;H$(H);K$(
    K)
540:IF P(L)=0GOTO
    560
550:Z=P(L):GOSUB "
    B":PRINT STR$
    L;O$;H$(H);K$(
    K)
560:NEXT L:GOTO 10
```

Przykład

```
K  A  R  T  Y
1 TY  WALET TREFL
1 ON  DAMA KIER
2 TY  AS TREFL
2 ON  AS PIK
3 TY  9 KIER
3 ON  DAMA KARO
4 TY  KROL KIER
4 ON  KROL PIK
5 TY  AS KARO
5 ON  9 PIK
6 TY  10 PIK
6 ON  10 KIER
7 TY  WALET KARO
7 ON  WALET PIK
8 TY  10 TREFL
8 ON  KROL TREFL
9 TY  9 TREFL
9 ON  AS KIER
10 TY  WALET KIER
10 ON  KROL KARO
11 TY  10 KARO
11 ON  DAMA TREFL
12 TY  DAMA PIK
13 TY  9 KARO
```

GRA W OCZKO SPECTRUM

```
 5 LET m=0
10 DIM t(6,4): DIM j(24)
15 DIM p(24): DIM h$(6,6)
20 DIM k$(4,5)
25 DATA "9 ","WALET ","DAMA "
30 DATA "KROL ","10 ","AS " .
35 DATA "KIER","KARO","TREFL","PIK"
40 RESTORE
45 FOR L=1 TO 6: READ H$(L): NEXT L
50 FOR L=1 TO 4: READ K$(L): NEXT L
55 LET f=0: LET x=0: LET y=0
60 LET u=0: LET v=0
65 LET f$="FURA! ": LET P$=" PKT."
70 LET o$="*OKO*"
80 INPUT "Jaka stawka ? ";s
90 IF s<=0 THEN  BEEP .2,-15: GO TO 80
100 INPUT "Dac karte? T/N ";a$
110 IF a$<>"t" AND a$<>"T" THEN  GO TO 150
120 GO SUB 400: LET x=x+1: LET j(x)=z
125 PRINT "TY "; INVERSE 1;h$(h);k$(k);
130 GO SUB 430: LET u=u+q
135 IF u=21 THEN  PRINT o$: GO TO 300
140 PRINT " :";u;p$: GO TO 100
150 GO SUB 400: LET y=y+1: LET p(y)=z
155 PRINT  INVERSE 1;"ON" ; INVERSE 0;h$(h);k$(k);
160 GO SUB 430: LET v=v+q
165 IF v=21 THEN  PRINT o$: GO TO 310
170 PRINT  INVERSE 1;" :";v;p$
175 IF u>v AND v<17 THEN  GO TO 150
180 IF y=0 THEN  GO TO 200
190 IF y>1 THEN  GO TO 250
200 IF u>21 THEN  PRINT f$;: GO TO 310
210 IF v>21 THEN  PRINT f$;: GO TO 300
220 IF u>v THEN  GO TO 300
230 GO TO 310
250 IF INT p(y)+INT p(y-1)=12 THEN  PRINT o$;
260 GO TO 310
299 REM
300 PRINT "W Y G R A L E S "
305 BEEP .1,10: BEEP .7,10
307 LET M=M+S: GO TO 320
310 PRINT "P R Z E G R A L E S "
315 BEEP .2,-14: PAUSE 5: BEEP .2,-14
317 LET M=M-S
320 PRINT "R E Z U L T A T"
330 IF m<0 THEN  PRINT "S T R A T A :";-m: GO TO 350
340 PRINT "Z Y S K :";m
350 IF x>23 OR y>23 THEN  GO TO 500
360 INPUT "gramy ? T/N ";a$
370 IF a$="t" OR a$="T" THEN  GO TO 60
380 STOP
390 LET h=INT z: LET k=(z-h)*10: RETURN
400 LET h=INT (RND*6)+1: LET k=INT (RND*4)+1
405 IF t(h,k)=1 AND f<>24 THEN  GO TO 400
410 LET z=k/10+h: LET t(h,k)=1
415 LET f=f+1: IF f<25 THEN  RETURN
420 PRINT " KONIEC TALII"
425 BEEP .2,0: GO TO 500
430 IF h=1 THEN  LET q=0
440 IF h=2 OR h=3 OR h=4 THEN  LET q=h
450 IF h=5 THEN  LET q=10
460 IF h=6 THEN  LET q=11
470 RETURN
500 LET t$=" TY ": LET O$=" ON "
505 PRINT "K A R T Y"
510 FOR 1=1  TO 24
520 IF j(1)=0 THEN  GO TO 540
530 LET z=j(1): GO SUB 390
535 PRINT STR$ 1;t$;h$(h);k$(k)
540 IF p(1)=0 THEN  GO TO 560
550 LET z=p(1): GO SUB 390
555 PRINT STR$ 1;o$;h$(h);k$(k)
560 NEXT 1: PAUSE 0: RUN
```

AMSTRAD

PROGRAM ZNOSI PROTEKCJĘ ODCZYTU

Przedstawiony program znosi „protekcję odczytu". Często chcemy podejrzeć komendą LIST sposób, w jaki dany program został napisany w języku BASIC. Niestety nie zawsze jest to możliwe. Prezentowany poniżej program pozwoli uporać się z tym problemem. Chcąc odczytać program napisany w języku Basic należy najpierw załadować i uruchomić komendą RUN poniższy program, a następnie załadować komendą LOAD interesujący nas program. Możecie być pewni, że większość programów da się teraz przejrzeć komendą LIST.

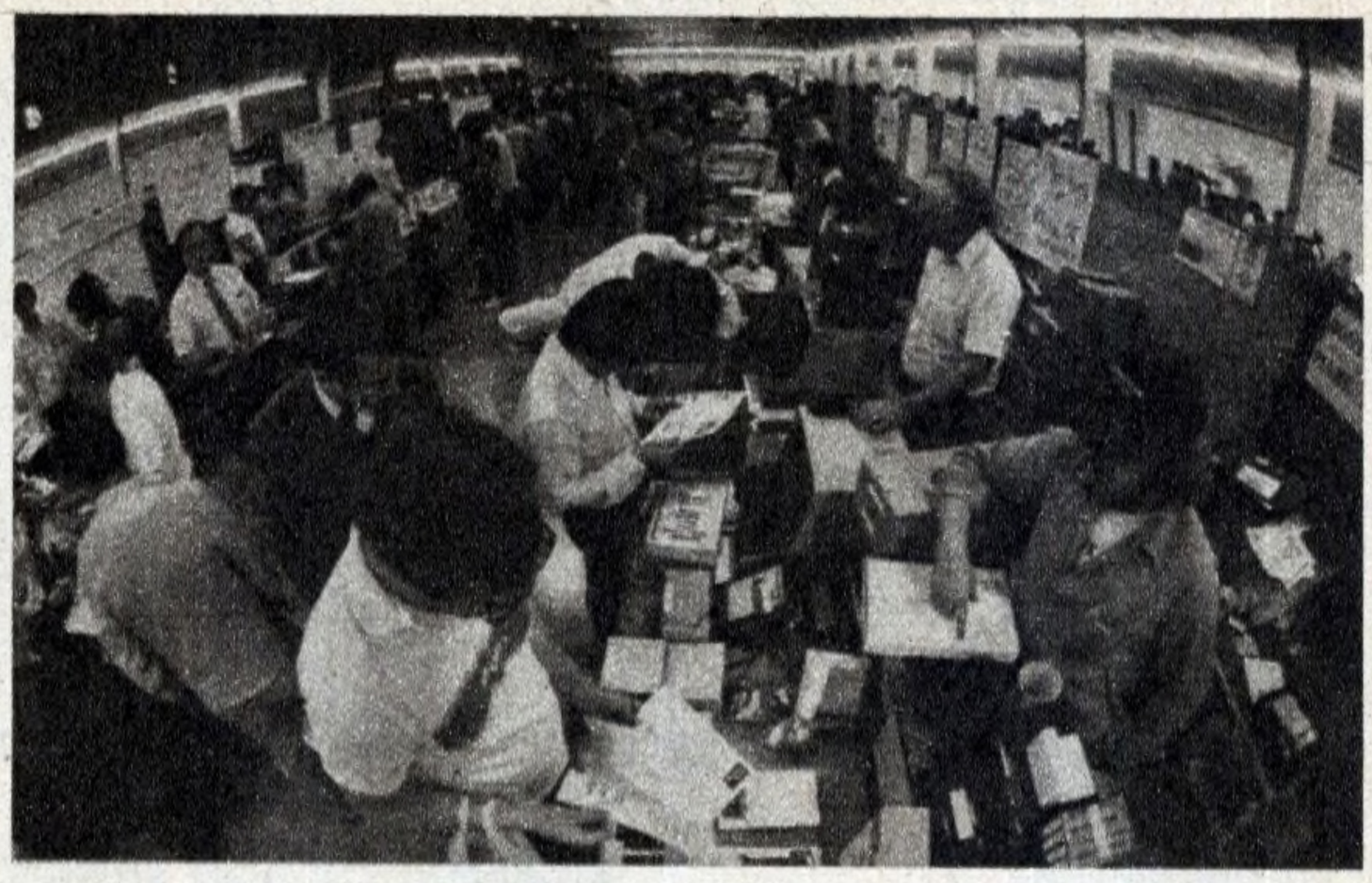
```
10 SYMBOL AFTER 256:MEMORY &A640:SYMBOL AFTER 240
20 FOR x=0 TO 54:READ a:POKE &A641+x,a:NEXT
30 IF PEEK(&AC01)=0 THEN POKE &A669,44
40 CALL &A641
50 CALL &BBBA:CALL &BC02:MODE 1:PEN 1:PAPER 0
60 MODE 2:INK 0,13:INK 1,1:BORDER 13:PRINT"PROGRAM
 ZALADOWANY ":PRINT CHR$(164)" POWODZENIA":PRINT
70 NEW
80 DATA 58,122,188,50,120,166,62,195,50,122,188,42
,123,188,34,121,166,33,89,166,34,123,188,201,245,2
29,58,120,166,50,122,188,42,121,166,34,123,188,175
,50,69,174,225,241,205,122,188,245,229,205,65,166,
225,241,201
```

SPOTKANIA na „Infosystem '8

7"

Foto: Jan Zelman

Kaizen zen-ja — przeddzień wojny. Takie słowa słyszy dziś Japończyk w telewizji, w radio. Czyta je także na łamach gazet. Rozbrzmiewało ono przed laty, niedługo przed atakiem japońskiego lotnictwa na bazę amerykańską w Pearl Harbour, w grudniu 1941 r. Hasło to zwiastowało Japończykom początek „Wielkiej wojny o Pacyfik". A co zwiastuje dziś?



Wielkie centrum sprzedaży komputerów pod San Francisco. Zabrakło tanich japońskich cudów techniki.
(Foto: — „Stern")

Wojna o komputery

USA obłożyły wysokimi cłami wyroby japońskiego przemysłu elektronicznego. Od paru miesięcy wszyscy, i w Japonii i w USA, studiują długą listę produktów, które obciążono cłem, a jak mawiają Japończycy, „karnymi opłatami", w wysokości 100 proc. ich wartości. Spis ten obejmuje m.in. stacje dysków elastycznych do komputerów, mikroprocesory, odbiorniki telewizyjne, radiomagnetofony, lodówki, klimatyzatory, satelity komunikacyjne, silniki elektryczne małej mocy, przyrządy pomiarowe, pompy, filmy światłoczułe, ręczne narzędzia o napędzie elektrycznym. Natomiast cłem nie są obciążone półprzewodniki, których amerykański przemysł komputerowy potrzebuje, a nie może zastąpić własną produkcją. Tak więc zarządzenia ochronne dotyczą przede wszystkim urządzeń elektronicznych. Oblicza się, że Japonia straci na tym ok 300 mln dolarów rocznie.

Czy Japończycy musieli zgodzić się na te restrykcje?

Obserwatorzy zagraniczni w Tokio relacjonują, że postępowanie Amerykanów przypomina mieszkańcom Kraju Wschodzącego Słońca rok 1853, kiedy to amerykański admirał Matthew Perry pojawił się u wybrzeży japońskich ze swą flotą i bez większych ceregieli zażądał przerwania ścisłej izolacji Japonii, którą kraj ten stosował wobec siebie. Władcy jego uznali bowiem, że lepiej nie wiedzieć, co dzieje się u innych, niż narażać się na kontakty z „barbarzyńcami". Tylko że admirał Perry rozporządzał ogniem swych dział i był gotów zrealizować groźby zawarte w ultimatum: otwieracie granice albo...

W Waszyngtonie uznano wówczas, że rynek japoński jest tak nęcący, iż USA nie mogą czekać, aż konkurenci ich ubiegną. Skończyło się na protestach szogunów, ale wymowa straszliwych dział była aż nadto przekonująca.

Dziś jakby sytuacja się „odwróciła". To Amerykanie zamykają drzwi przed japońskim „nosem", przynajmniej, jeśli chodzi o elektronikę. Można powiedzieć: jakże to, USA i Japonia są sojusznikami, współpraca militarna rozwija się doskonale. To wszystko prawda, ale jednocześnie USA mają 170 miliardów dolarów deficytu handlowego, a nadwyżka eksportu japońskiego do USA wynosi 52 mld dolarów. Potwierdza się stare porzekadło: tam, gdzie chodzi o pieniądze, kończą się żarty, a zaczyna... cło!

A czy „żartować" mogą tylko Amerykanie, przecież w końcu, to nie połowa XIX wieku, ale koniec XX!

Ustępliwość Japończyków tłumaczy się zależnością ich kraju od źródeł surowcowych. Wiemy, co potrafią produkować i jak, ale muszą mieć z czego. USA pozostają na razie czołowym producentem w świecie kapitalistycznym wielu rodzajów surowców i są potężnym eksporterem węgla, molibdenu, magnezu, fosfatów, materiałów z zawartością boru. Dla prawie bezsurowcowej Japonii amerykański partner jest tym, na którego przypada jedna trzecia ogólnego importu paliw mineralnych i jedna czwarta surowców nieenergetycznych. USA odgrywają decydującą rolę w zaopatrzeniu gospodarki japońskiej w surowce. I taka jest prawda. I takie są konsekwencje.

Japończycy protestują, ale na tym się skończy. Nie skończyły się natomiast nadzieje Japończyków, że „załatwią" Amerykanów sposobem.

Koniec rozrzutności

Dotychczas USA mogły sobie pozwolić na tolerowanie deficytu budżetowego i handlowego. Olbrzymi potencjał gospodarczy jeszcze znosi brzemię ogromnych wydatków zbrojeniowych. Dolar był i jest międzynarodową walutą rezerwową, w której dokonuje się ponad 60 proc. wszystkich rozliczeń międzynarodowych. Ponadto powiedzmy zwyczajnie i wprost: nie stać już dziś gospodarkę amerykańską na wymienność dolara na złoto, ale stać ją na to, by „przełknęła" wprowadzenie kilku czy kilkunastu miliardów dolarów na rynek, za pomocą maszyn drukujących „zielone", bez komplikacji w postaci inflacji. Przy czym obecnie na skutek nieprawdopodobnego zadłużenia wewnętrznego (deficyt budżetowy) pojawiły się oznaki niepokoju na rynku. Dziś w Białym Domu zdają sobie sprawę, że USA muszą konkurować na rynkach zagranicznych i prezentować atrakcyjną ofertę eksportową. Także i w zakresie elektroniki.

Japończycy obliczają, obserwują i wiedzą. Świadomi są tego, iż roczny wskaźnik wzrostu produkcji w USA w przeliczeniu na godzinę pracy zmniejszył się już od połowy lat sześćdziesiątych. Realne koszty jednostki pracy są wyższe niż w Japonii. Wydajność pracy również wzrasta w tempie wolniejszym. Jakość wyrobów „made in USA" jest gorsza niż japońskich. Występuje zjawisko opóźnienia w inwestycjach technologicznych. W zakresie maszyn numerycznych USA utraciły przodującą pozycję na rzecz Japonii. Amerykańscy producenci skoncentrowali się na osiąganiu szybkich zysków, co wypaczyło ich politykę inwestycyjną w dziedzinie produkcji artykułów elektronicznych codziennego użytku. Ustąpienie miejsca Japończykom, czyli rezygnacja z rynku wewnętrznego, odnośnie tych artykułów jest równoznaczna z osłabieniem eksportu. Teraz próbuje się cłami ochronnymi odwrócić tę sytuację.

Decyduje technologia

Uwagę specjalistów zwraca problem

technologiczny. W ostatnich latach przemysł elektroniczny w USA przeznaczył 67 miliardów dolarów na badania i opracowanie nowych technologii. Połowa tej kwoty pochodzi z kas rządowych. Jest to akurat trzy razy tyle, ile mogą wydać na te cele japońscy konkurenci. A jednak ich wyroby elektroniczne, a szczególnie komputery o pojemności pamięciowej 1 megabita oraz mikroprocesory o pojemności 32 bitów skutecznie rywalizują z analogicznymi urządzeniami „made in USA". Dlaczego?

Japończycy twierdzą, że ani Europejczycy, ani Amerykanie nie potrafią handlować. Towar trzeba umieć nie tylko wyprodukować, ale i sprzedać. Mentalność ludzi Wschodu jest inna niż Zachodu. Na to Amerykanie powiadają: to co, mamy sprzedawać nasze komputery w kimonie? Kłaniać się klientom? Obie strony wiedzą doskonale, że nie o to chodzi, lecz zwyczajnie o „szmal". Godzinę drogi na południe od San Francisco, w pobliżu wzgórz Santa Cruz znajduje się ogromne centrum sprzedaży urządzeń elektronicznych. Można tu nabyć komputery do koloru i wyboru, podzespoły, monitory monochromatyczne, drukarki mozaikowe, laserowe, plottery, digitizery, streamery, zespoły indywidualne, sieci komputerowe, zestawy specjalistyczne z wyposażeniem, moduły do mikrokomputerów, układy scalone itp. Na wieść o podniesionych cłach na wyroby japońskie rozległ się tu (gdzie indziej również) płacz i zgrzytanie zębów. Ceny japońskich produktów elektronicznych były do tej pory tańsze. Niektórzy klienci mówią, że amerykańskie nie są gorsze. Inni, że japońskie lepsze, ale ceny, te ceny!

Skorzysta klient

USA bronią się przed japońską, i nie tylko, konkurencją, machinacjami na giełdach, które powodują kierowany spadek wartości dolara. Dziś za 1 „zielonego" otrzymuje się więc mniej jenów, niż przed rokiem. Czyli że Amerykanin musi wydać więcej dolarów, by kupić japońskie cuda elektroniczne. Zwyżka wartości jena japońskiego już daje o sobie znać w postaci jeszcze nieznacznego, ale wyraźnego spadku eksportu. A przecież Japonia żyje z tego, co sprzeda. W Tokio złośliwie przypominają amerykańskim sprzymierzeńcom: jak w tym świetle wygląda wasze zachwalanie wolnej, nieskrępowanej gospodarki? Najpierw tyle się gada o liberaliźmie, handlu wolnym od barier i restrykcji, a jak są lepsi od was, to sięgacie do metod, które obłożyliście klątwą!

Tzw. wojna handlowa między USA a Japonią nie ogranicza się jedynie do niezadowolenia amerykańskich i japońskich konkurentów „elektronicznych". Trudno sobie wyobrazić, aby jedna ze stron miała skapitulować. W trakcie tej niewątpliwej walki o rynki może dojść do taktycznych kompromisów, ale nic więcej. Z drugiej strony amatorzy domowej elektroniki zyskają na tym. Japończycy wymyślą kolejne nadzwyczajne rzeczy w elektronice. Amerykanie i Europejczycy nie zostaną w tyle.

Henryk KAWKA

W poprzednim odcinku przedstawiony został uproszczony schemat blokowy pakietu systemowego mikrokomputera IBM PC/XT, na którym znajdują się: mikroprocesor INTEL 8088 z koprocesorem INTEL 8087 i ze współpracującymi układami m.in. 8284, 8259A, 8237A, 8253, 8255; pamięć zapisywalna RAM o pojemności 256 KB; pamięć stała ROM.

MIKROKOMPUTERY IBM PC (3)

W mikrokomputerze do pakietu systemowego muszą być dołączone pakiety umożliwiające współpracę z monitorem, drukarką lub plotterem, stacją dysków elastycznych, dyskiem stałym (twardym, sztywnym — ang. hard disk, fixed disk). Nie wyczerpuje to listy możliwych pakietów. Na fot. 1 przedstawiono niektóre z nich.

Karta graficzna służy do organizacji ekranu monitora. Możliwe są dwa mody pracy: **mod alfanumeryczny** 40 lub 80 znaków w 25 liniach oraz **mod graficzny** 160x100, 320x200, 640x200, 640x350 lub 720x348 punktów adresowych bezpośrednio, odpowiednio w standardzie małej, średniej i dużej rozdzielczości. Możliwe mody pracy ekranu monitora zależą od karty graficznej (kart graficznych) przyłączonej do pakietu systemowego.

W mikrokomputerach IBM PC/XT (i kompatybilnych) można spotkać następujące karty graficzne: Monochrome Display Adaptor, Color Graphics Adaptor, Enhanced Graphics Adaptor, Hercules.

Monochrome Display Adaptor (MDA) umożliwia pracę w modzie alfanumerycznym 80 znaków w 25 liniach, w modzie graficznym 720x350 punktów adresowanych bezpośrednio. Podstawowym układem karty jest sterownik MOTOROLA 6845 CRT, karta ma pamięć ekranu 4 KB. MDA może zawierać złącze równoległe, do połączenia drukarki lub plottera.

Color Graphics Adaptor (CGA) umożliwia następującą organizację ekranu: w modzie alfanumerycznym 40 lub 80 znaków w 25 liniach w szesnastu kolorach; w modzie graficznym: małej rozdzielczości 160x100 punktów adresowanych bezpośrednio w szesnastu kolorach, średniej rozdzielczości 320x200 punktów w czterech kolorach, dużej rozdzielczości 640x200 punktów w dwóch kolorach. CGA ma pamięć ekranu 16 KB, podstawowym jej układem jest sterownik 6845 CRT. Istnieje możliwość dołączenia do karty pióra świetlnego.

Enhanced Graphics Adaptor (EGA) w modzie alfanumerycznym organizuje ekran monitora identycznie jak CGA. W trybie graficznym EGA umożliwia: średnią rozdzielczość 320x200 punktów adresowanych bezpośrednio w szesnastu kolorach, dużą rozdzielczość 640x200 w szesnastu kolorach lub 640x350 w dwóch kolorach lub 640x350 w sześćdziesięciu czterech kolorach. EGA ma w wersji firmy IBM pamięć ekranu 64 KB, niektóre wersje kompatybilne mają pamięć ekranu 256 KB oraz umożliwiają jeszcze dodatkowo rozdzielczość 720x348 punktów w dwóch kolorach. Instalacja karty EGA wymaga monitora o dużej rozdzielczości i specjalnej wersji ROM—BIOS-u.

Fot. 1

W mikrokomputerach kompatybilnych z IBM PC/XT można spotkać również Hercules Graphics Card o rozdzielczości w modzie graficznym 720x348 punktów adresowych bezpośrednio w dwóch kolorach.

Kolejnym pakietem, który może być dołączony do pakietu systemowego, jest **karta wielofunkcyjna**. Karta ta umożliwia rozszerzenie pamięci RAM do 640 KB, ma złącze szeregowe RS—232 C i złącze równoległe CENTRONICS oraz zegar czasu rzeczywistego z bateryjką.

Kolejny pakiet, który musi być w mikrokomputerze IBM PC/XT, służy do **sterowania stacjami dysków elastycznych,** takimi jak np. na foto 2. Pakiet może równocześnie sterować dyskiem stałym. W pakiecie służącym jedynie do sterowania stacjami dysków elastycznych głównym elementem jest sterownik NEC μPD 764, mogący obsłużyć dwie stacje dysków.

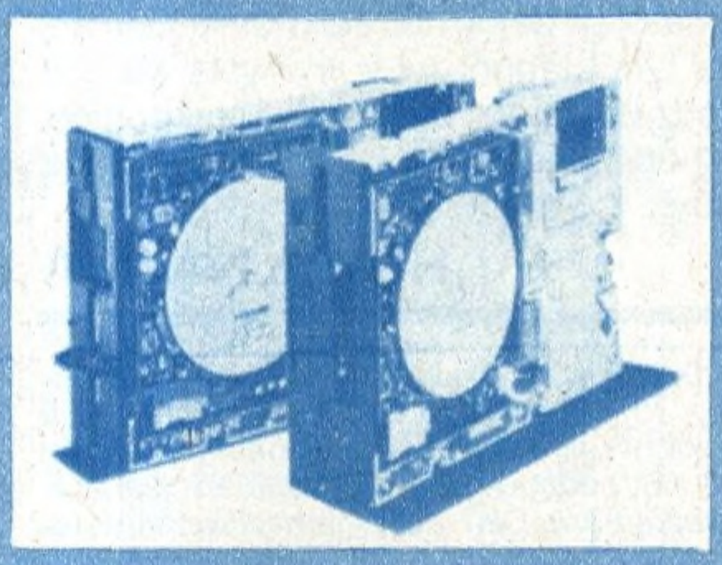


Foto. 2 Stacja dysków elastycznych 5 ½ cala

Stacja dysków (ang. disk drive, diskette drive) może być jedno- lub dwustronna. Stacje dysków jednostronne umożliwiają formatowanie dyskietek (dysków elastycznych, ang. diskette) jednostronnie, w standardzie 8 lub 9 sektorów po 512 bajtów na ścieżce, przy 40 ścieżkach. Umożliwia to osiągnięcie pojemności 160 lub 180 KB. Dla stacji dysków dwustronnych pojemność wynosi odpowiednio 320 lub 360 KB. Ważniejsze parametry stacji: szybkość transmisji danych 250 K bitów/s, czas dostępu między ścieżkami dyskietki 6 ms, czas niezawodnej pracy stacji 20 000 godzin, przy trwałości nośnika 3.10^6 przejść po ścieżce. W zależności od typu stacji wykorzystywane są **dyskietki** jednostronne (single — side, dauble — density) lub dwustronne (double — side, double — density).

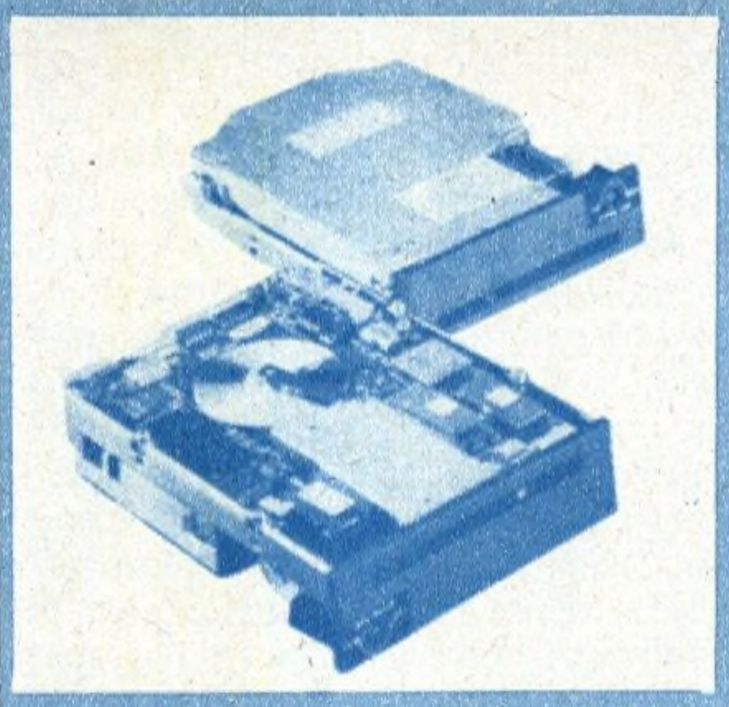


Foto. 3 Dyski stałe typu Winchester o pojemności 20 MB

W mikrokomputerze IBM PC/AT dyskietki formatowane są wg standardu dwustronnie, 9 sektorów na ścieżce, przy 80 ścieżkach na stronie, co umożliwia osiągnięcie pojemności 720 KB lub wg standardu dwustronnie, 15 sektorów na ścieżce, przy 80 ścieżkach na stronie, co pozwala osiągnąć pojemność 1200 KB.

Dysk stały, taki jak np. na fot. 3, z reguły ma pojemność 10 lub 20 MB. Dysk 10 MB formowany jest czterostronnie, po 17 sektorów 512-bajtowych na ścieżce, przy maksymalnie 1226 ścieżkach i 306 cylindrach. Jego najważniejsze parametry to: szybkość transmisji 5 M bitów/s, czas niezawodnej pracy 5 lat, czas dostępu między ścieżkami 3 ms. Dysk o pojemności 20 MB formatowany jest identycznie, ma 615 cylindrów.

W mikrokomputerze można zainstalować pakiet umożliwiający pracę w sieci lokalnej.

Kolejnym elementem mikrokomputera jest **głośnik,** z którym można pracować za pomocą układów 8253 lub 8255.

Mikrokomputer może zawierać złączkę dla **joystick'a.**

Następnym, bardzo ważnym elementem jest **zasilacz** o mocy minimum 130 watów. Obecnie w PC/XT wykorzystywane są zasilacze o mocach 135 watów, 150 watów. W mikrokomputerach PC/AT wykorzystywane są zasilacze o mocach 180 watów lub 200 watów.

Klawiatura umożliwia kontakt z systemem za pomocą 83 klawiszy, które można podzielić na trzy grupy: 10 klawiszy funkcyjnych, klawisze znakowe oraz klawisze ekranowe (numeryczne i specjalne). Obsługą klawiatury zajmuje się mikrokomputer jednoukładowy INTEL 8084, który wykonuje również test po włączeniu zasilania. Obecnie w mikrokomputerach kompatybilnych z IBM PC/XT stosowane są jeszcze dwa inne typy klawiatur: z 84 klawiszami i z 101 klawiszami.

Monitor jest bardzo ważnym elementem systemu, musi on być dopasowany do karty graficznej znajdującej się w komputerze.

Charakterystykę budowy mikrokomputera PC/XT uzupełnia tab. 1, przedstawiająca mapę adresową.

Tablica 1.

adres	przeznaczenie
00000	RAM
10000	RAM
20000	RAM
30000	RAM
40000	RAM
50000	RAM
60000	RAM
70000	RAM
80000	RAM
90000	RAM
A0000	rozszerzenie pamięci ekranu dla EGA
B0000	pamięć ekranu
C0000	rozszerzenie ROM — BIOS, sterownik
D0000	————————
E0000	————————
F0000	ROM — BIOS, interpreter BASIC, diagnostyka systemu.

Następny odcinek dotyczyć będzie dyskowego systemu operacyjnego, który zostanie omówiony na przykładzie MS DOS 3.10.

Jacek WOJTALA

Czym skorupka za młodu nasiąknie ... Fot. Ryszard Rogoń

BASIC 8

Na ekranie możemy rysować w kolorach. Kolory odnoszą się do całej pozycji literowej (kwadracika 8 x 8 pixeli) i nie mogą być określone dla poszczególnych pixeli. Gdy w danym pixelu stawiamy punkt, to jest on w kolorze atramentu, odpowiednim do całego kwadracika 8 na 8 zawierającego ten punkt.

Na ZX Spectrum dostępnych mamy osiem kolorów:

Kolor	Klawisz	
czarny	Ø	(BLACK)
ciemnoniebieski	1	(BLUE)
czerwony	2	(RED)
purpurowy	3	(MGNTA)
zielony	4	(GREEN)
bladoniebieski (lazurowy)	5	(CYAN)
żółty	6	(YELLOW)
biały	7	(WHITE)

Kolejność, w jakiej są ustawione, odpowiada intensywności szarości na ekranie telewizora czarno-białego.

Z pozycją każdego znaku związane są dwa kolory:
— kolor atramentu (INK) odpowiadający czarnym kropkom,
— kolor papieru (PAPER) czyli tło, odpowiadający kropkom białym.

W pisanych i wykonywanych przez nas programach każda pozycja na ekranie miała kolor biały i czarny atrament (pismo czarne na białym papierze). Za pomocą instrukcji postaci:

PAPER *n* n = 0,1,...,7 (numery kolorów)

ustawiamy kolory papieru kwadracików (mogą to być pojedyncze, lub ciąg kwadracików, lub cały ekran), a instrukcją postaci:

INK *n* n = 0,1,...,7 (numery kolorów)

nadajemy kolory pisma (całych kwadracików).
Jasność obrazu zwiększamy za pomocą instrukcji:

BRIGHT Ø lub 1

gdzie: Ø — jasność normalna, 1 — jasność zwiększona.

Przykład 6.9.

Poniższy program pokazuje 8 kolorów (łącznie z białym i czarnym) i dwa poziomy jasności, jakie można otrzymać na kolorowym odbiorniku. Na telewizorze czarno-białym otrzymamy różne odcienie szarości.

Wprowadźmy i wykonajmy ten program.

```
  5 REM Przyklad 6.9
  10 FOR m=0 TO 1: BRIGHT m
  20 FOR n=1 TO 7
  30 FOR c=0 TO 7
  35 REM 4 barwne spacje
  40 PAPER c: PRINT "    ";
  50 NEXT c: NEXT n: NEXT m
  55 FOR i=1 TO 7
  60 FOR m=0 TO 1: BRIGHT m: PAP
ER 7
  70 FOR c=0 TO 3
  80 INK c: PRINT c;" ";
  90 NEXT c: PAPER Ø
 100 FOR c=4 TO 7
 110 INK c: PRINT c;" ";
 120 NEXT c: NEXT m
 125 NEXT i
 130 PAPER 7: INK Ø: BRIGHT Ø
```

Oglądając obraz na ekranie, prześledźmy sami działanie instrukcji **PAPER, INK, BRIGHT.**

Można także wprowadzić błyskanie (miganie) obrazu na ekranie za pomocą instrukcji.

FLASH Ø lub 1

gdzie: Ø — brak błyskania, 1 — błyskanie

Przykład 6.10.

Poniższy program pokazuje, jak działa instrukcja FLASH 1 umieszczona w dwóch różnych miejscach programu.

Wprowadźmy i wykonajmy ten program. Prześledźmy sami „błyskanie" obrazu na ekranie w zależności od miejsca w programie instrukcji **FLASH 1.** Kolory jasności i błyskanie w danym miejscu nazywamy atrybutami.

```
  5 REM Przyklad 6.10
  10 REM błyskanie całego ekranu
  20 BORDER 2: INK 1: PAPER 6: F
LASH 1: CLS
  30 DRAW 255,175,1
  40 DRAW -255,-175,1
  50 PAUSE 400: FLASH Ø: CLS
  55 REM błyskanie rysunku
  60 BORDER 4: INK 3: PAPER 7: C
LS
  70 DRAW FLASH 1;255,175,1
  80 DRAW FLASH 1;-255,-175,1
  90 PAUSE 400: CLS : FLASH Ø
```

W powyższych instrukcjach można napisać liczbę 8 lub 9. Liczba 8 może być użyta we wszystkich instrukcjach: **PAPER 8, INK 8, BRIGHT 8, FLASH 8** i oznacza „przezroczysty", w znaczeniu takim, że nie zmienia poprzedniego atrybutu (pozostają poprzednie atrybuty).

Liczba 9 może być użyta tylko z PAPER i INK i oznacza „kontrast", tj. aktualny kolor (atramentu lub papieru) jest zmieniany na biały, jeśli drugi jest ciemny (czarny, granatowy, czerwony lub purpurowy) lub na czarny, jeśli drugi jest jasny (zielony, lazurowy, żółty, lub biały). Kolor atramentu jest dobierany tak, aby kontrastował ze starym kolorem papieru.

Przykład 6.11.

Poniższy przykład pokazuje nam, jak działa liczba 9 umieszczona w instrukcji **INK** („kontrast").

Wprowadźmy i wykonajmy ten program.

```
 10 REM Przyklad 6.11
 15 INK 9
 20 FOR c=1 TO 7
 30 PAPER c
 40 FOR i=1 TO 15
 50 PRINT c+i;"  ";
 60 NEXT i
 70 NEXT c
```

Oglądając obraz na ekranie, prześledźmy sami działanie INK 9.

Kropkowy wzór wypisany na ekranie można „kontrolować" za pomocą dwóch instrukcji postaci:

INVERSE Ø lub *1* *oraz* **OVER** Ø lub *1*

Ø — wyłączenie, 1 — włączenie.

Instrukcja **INVERSE 1** powoduje, że rysunek staje się negatywem zwykłej formy, kropki w kolorze papieru zostaną zastąpione przez kropki w kolorze atramentu i vice versa. Np. jeżeli mamy biały papier i czarny atrament, wówczas odpowiedni rysunek pojawi się jako biały na czarnym, ale kolor papieru i atramentu pozostają nie zmienione, zmieniają się tylko kropki.

Instrukcja **OVER 1** powoduje szczególny rodzaj nadrukowywania, tzn. jeżeli coś jest napisane w danym miejscu, to zawartość poprzednia zostaje całkowicie usunięta po wykonaniu **OVER 1.**

Omówiliśmy dotychczas kolory górnej części ekranu. Dolna część ekranu ma jako kolor — kolor papieru, a dla atramentu obowiązuje kod 9, wyłączone jest błyskanie, jasność jest normalna. Kolor ramki, otaczającej ekran, składający się z 24 linii i 32 kolumn, można zmienić instrukcją postaci:

BORDER *n* (*n*=Ø,1,...,7).

Przy wprowadzaniu danych przestrzegana jest reguła o użyciu kontrastującego atramentu dobieranego do papieru.

Poniższy program ustala kontrastowe kolory ekranu:
— część górna : **PAPER** *n*
— część dolna — ramka, **BORDER** 7-n
(*n*=0,1,..., 7)

```
20 FOR n = Ø TO 7
30 BORDER n: BORDER 7 - n : CSL
40 PAUSE 100/ : NEXT n
```

sprawdzenie pozostawiamy czytelnikowi.

Instrukcje **PAPER, INK** itd. można stosować jako elementy instrukcji **PRINT** (stawiając po nich średniki) — ich efekt jest chwilowy (do pierwszego dwukropka lub do następnej instrukcji). Można je także stosować jako elementy instrukcji **PLOT, DRAW, CIRCLE.**

Poniżej podajemy kilka możliwych kombinacji:

PLOT ... — kropka w kolorze atramentu, tzn. ustawia pixel tak, aby pokazywał kolor atramentu,

PLOT INVERSE *1*;... — daje pixel (kropka) w kolorze papieru

PLOT OVER *1*;... — pixel przechodzi z koloru papieru w kolor atramentu lub odwrotnie

PLOT INVERSE *1*; **OVER** *1*;... — zostawia pixel w stanie, w jakim był, ale zmienia pozycję **PLOT**-u (czyni to niezauważalnie)

DRAW OVER *1*;... — pixele przechodzą z koloru papieru w kolor atramentu lub odwrotnie (jeśli coś było napisane, to poprzednia zawartość zostaje usunięta)

DRAW INVERSE *1*;... — pixele (kropki) w kolorze atramentu przechodzą na pixele (kropki) w kolorze papieru i odwrotnie (otrzymujemy negatyw).

Instrukcje **OVER** 1 i **INVERSE** 1 można stosować np. razem z **DRAW** do „wymazywania" rysunków.

Przykład 6.12.

Wprowadźmy i wykonajmy następujący program:

```
 5 REM Przyklad 6.12
10 FOR i=1 TO 22
20 FOR j=1 TO 16
30 PRINT "■■";
40 NEXT j: NEXT i
50 PLOT 0,0
55 DRAW OVER 1,255,175
60 PAUSE 200
70 PLOT 0,0
80 DRAW OVER 1;255,175
```

Na ekranie otrzymamy czarno-białą kratkę, a następnie zostanie wykreślona linia prosta o pixelach koloru białego i czarnego (instr. 55 DRAW OVER 1; 255, 175) po chwilowej przerwie (PAUSE 200) linia zniknie — została zamazana instrukcją 80 DRAW OVER 1; 255, 175. Dlaczego? Odpowiedź zostawiamy czytelnikowi. Rysunek został nienaruszony.

Wprowadźmy instrukcję:

80 **DRAW INVERSE** 1, 255, 175.

i wykonajmy program.

Zauważymy, że instr. INVERSE 1, „wymazuje" linię, ale „narusza" rysunek. Nie zaleca się wymazywać linii w odwrotną stronę, niż była rysowana, np. w naszym programie nie należałoby napisać:

50 **PLOT** 0, 0
55 **DRAW OVER** 1; 255, 175
60 **PAUSE** 200
70 **PLOT** 255, 175
80 **DRAW OVER** 1; — 255,— 175

Przykład 6.13.

Dany jest program:

```
 5 REM Przyklad 6.13
10 FOR i=1 TO 22
20 FOR j=1 TO 16
30 PRINT "■ ";
40 NEXT j: NEXT i
50 FOR i=1 TO 81 STEP 6
55 CIRCLE OVER 1;127,87,i
60 NEXT i
70 PAUSE 200
80 FOR i=1 TO 81 STEP 6
90 CIRCLE OVER 1;127,87,i
100 NEXT i
```

Wprowadźmy i wykonajmy ten program. Ilustruje on rysowanie okręgów instrukcją **CIRCLE** połączoną z instr. **OVER** 1, a następnie w identyczny sposób narysowanych okręgów. Zauważamy, że tło, na którym rysowane były okręgi, po ich wymazaniu jest takie samo jak przed rysowaniem.

Wymieniając instr. 90 na instr. postaci

90 **CIRCLE INVERSE** 1; 127, 87, i

po wykonaniu programu na poprzednim tle otrzymamy białe okręgi. Na zakończenie podajemy cztery programy do wprowadzenia i wykonania.

Przykład 6.14.

Napisać program rysujący na ekranie ciąg prostokątów (od największego do najmniejszego), ale tak aby poprzedni prostokąt „znikał".

```
 10 REM Przyklad 6.14
 15 PAPER 2: INK 4: BORDER 0
 20 FOR n=0 TO 80 STEP 10
 30 LET x=255-2*n
 40 LET y=175-2*n
 50 PLOT n,n
 60 DRAW x,0: DRAW 0,y
 70 DRAW -x,0: DRAW 0,-y
 80 PAUSE 200
100 DRAW INVERSE 1;x,0
110 DRAW INVERSE 1;0,y
120 DRAW INVERSE 1;-x,0
130 DRAW INVERSE 1;0,-y
135 PAUSE 50
140 NEXT n
```

Wprowadźmy i wykonajmy ten program. Jest on dobrą ilustracją działania instrukcji INVERSE 1;

Przykład 6.15.

Dany jest program:

```
 10 REM Przyklad 6.15
 20 PAPER 2: CLS
 30 LET a$="■"
 40 FOR i=1 TO 14
 50 LET a$=a$+"■"
 60 NEXT i
 80 FOR i=1 TO 22
 90 PRINT a$
100 NEXT i
110 REM lewa czesc ekranu zosta
la zamalowana na czarno,prawa je
st w kolorze papieru
120 FOR i=1 TO 500
130 PLOT INK 7;RND*255,RND*175
140 NEXT i
150 REM na ekranie wypisano los
owe punkty w kolorze atramentu
160 REM RND - generator liczb p
seudolosowych z przedzialu <0,1)
```

Wprowadźmy i wykonajmy ten program. Opis programu zamieszczony jest w komentarzach (po słowie REM). Każde uruchomienie programu może dać inny wynik.

Przykład 6.16.

Dany jest program:

```
 10 REM Przyklad 6.16
 20 BORDER 0: PAPER 7: CLS : BR
IGHT 1
 30 FOR i=1 TO 400
 40 PRINT INK 7*RND;AT 21*RND,1
5*RND;"■"
 50 PLOT FLASH 1; INK 7*RND;130
+125*RND,175*RND
 60 NEXT i
```

Wprowadźmy i wykonajmy ten program. Odpowiedź na pytanie dlaczego tak, a nie inaczej wykonuje się ten program, pozostawiamy czytelnikowi.

Przykład 6.17.

Dany jest program

```
 10 REM Przyklad 6.17
 20 BORDER 0: PAPER 5: CLS
 30 LET x=1
 40 FOR l=0 TO 21
 50 FOR c=1 TO 6
 60 PRINT INK c;AT l,c+x;"■"
 70 NEXT c
 80 LET x=x+1
 90 NEXT l
```

Wprowadźmy i wykonajmy ten program. Na telewizorze kolorowym otrzymamy ładny rysunek, a na czarno-białym rysunek w kolorach biało-szaro-czarnych.

ZW

GRAFIKA (3) COMMODORE 64

W poprzednim artykule z tego cyklu przedstawiony został znakowy tryb pracy mikrokomputera C-64. Grafika w tym trybie polega na tworzeniu określonych kompozycji ze znaków standardowych lub samodzielnie zaprojektowanych. Możliwości graficzne są jednak ograniczone. Trudności te znikają, gdy potrafimy korzystać z trybu graficznego o dużej rozdzielczości (nazywanego dalej trybem graficznym). Podstawowe wiadomości z tego zakresu zostaną przedstawione w tym artykule. Tryb graficzny opisany zostanie na przykładzie programu FUNKCJA, służącego do kreślenia funkcji na ekranie.

Mapa bitowa ekranu

Jak pamiętamy z poprzedniej części, obraz w trybie znakowym **VIC** tworzy na podstawie zawartości **PAMIĘCI EKRANU** (i **GENERATORA ZNAKÓW**). W trybie graficznym wykorzystywany jest obszar pamięci zwany **MAPA BITOWA EKRANU** (**MBE**). Każdemu punktowi ekranu odpowiada określony bit pamięci **MBE**. Strukturę ekranu i jej odzwierciedlenie w pamięci przedstawia rys. 1.

Cały ekran podzielony jest na 1000 sekcji rozmieszczonych w 25 wierszach i w 40 kolumnach. Każda sekcja składa się z 64 punktów (8 x 8). W wyniku takiego podziału dysponujemy 320 punktami w poziomie (40 x 8) oraz 200 punktami w pionie (25 x 8). Sekcje w pamięci mikrokomputera zajmuje 8 bajtów (64 bity). Rozmieszczenie sekcji w pamięci przedstawia rysunek. Numery sekcji na ekranie odpowiadają numerom w pamięci. Cała MBE zajmuje 8000 bajtów (1000 x 8).

Istnieją dwie podstawowe metody programowania grafiki w trybie graficznym:

1. Projektujemy obraz na siatce odpowiadającej punktom ekranu. Wyznaczamy wartości liczbowe odpowiednich bajtów MBE. Zapisujemy je w MBE.
2. Dla dowolnego punktu ekranu wyznaczamy odpowiadający mu bit MBE i ustawiamy na określoną wartość (0 lub 1).

Wykorzystanie drugiej metody wiąże się z interpretacją ekranu, jako ćwiartki układu współrzędnych kartezjańskich. W naszych rozważaniach przyjmiemy za środek układu lewy dolny róg, oś X jest pozioma i skierowana w prawo, oś Y jest pionowa i skierowana do góry. Na osi Y wyróżniamy 200 punktów ponumerowanych od 0 do 199. Na osi X wyróżniamy 320 punktów w trybie dwukolorowym lub 160 punktów w trybie wielobarwnym, ponumerowanych od 0 do 319 lub 159. Dla przypomnienia: w trybie wielobarwnym jednemu punktowi na ekranie odpowiada para bitów w pamięci.

Wyznaczając bit MBE, odpowiadający wybranemu punktowi ekranu o współrzędnych *X* i *Y*, stosujemy procedurę (w trybie dwukolorowym):

```
200 REM · PUNKT ·
210 YY = 199 — Y
220 SW = INT (YY/8): REM NR WIERSZA 0 ... 42
230 SK = INT (X/8): REM NR KOLUMNY 0 ... 39
240 S = SW · 40 + SK: REM NR SEKCJI 0 ... 999
250 SL = YYAND7: REM NR LINII NR BAJTU W SEKCJI 0 ... 7
260 BAJT = S · 8 + SL + MAP: REM NR BAJTU PAMIĘCI
270 SP = XAND7: REM NR PUNKTU W LINII 0 ... 7
280 BIT = 7 — SP: REM NR BITU W BAJCIE 7... 0
300 RETURN
```

Dla trybu wielobarwnego należy wprowadzić następujące zmiany:

```
230 SK = INT (X/4): REM NR KOLUMNY 0 ... 39
270 SP = XAND3: REM NR PUNKTU W LINII 0 ... 3
280 BIT = 3 — SP
290 B2 = 2 · BIT: B1 = B2 + 1: REM NR BITÓW W BAJCIE
```

Parametrami procedury **PUNKT** są:
— X, Y — współrzędne punktu
— MAP — adres początkowy MBE w pamięci mikrokomputera.
Wykonując tę procedurę otrzymujemy adres bajtu pamięci BAJT i numer (y) bitu (ów) BIT (B1 i B2) odpowiadające punktowi na ekranie. Na wyznaczonym bicie możemy dokonać operacji:
— ustawić wartość bitu na 1
POKE BAJT, **PEEK** (BAJT) **OR** (2↑BIT)
— ustawić wartość bitu na 0
POKE BAJT, **PEEK** (BAJT) **AND** (255 — 2↑BIT)

Dwukolorowy tryb graficzny

Opis pracy w trybie graficznym przedstawiony jest na przykładzie programu **FUNKCJA** i dlatego w instrukcjach zachowana jest numeracja wierszy programu.

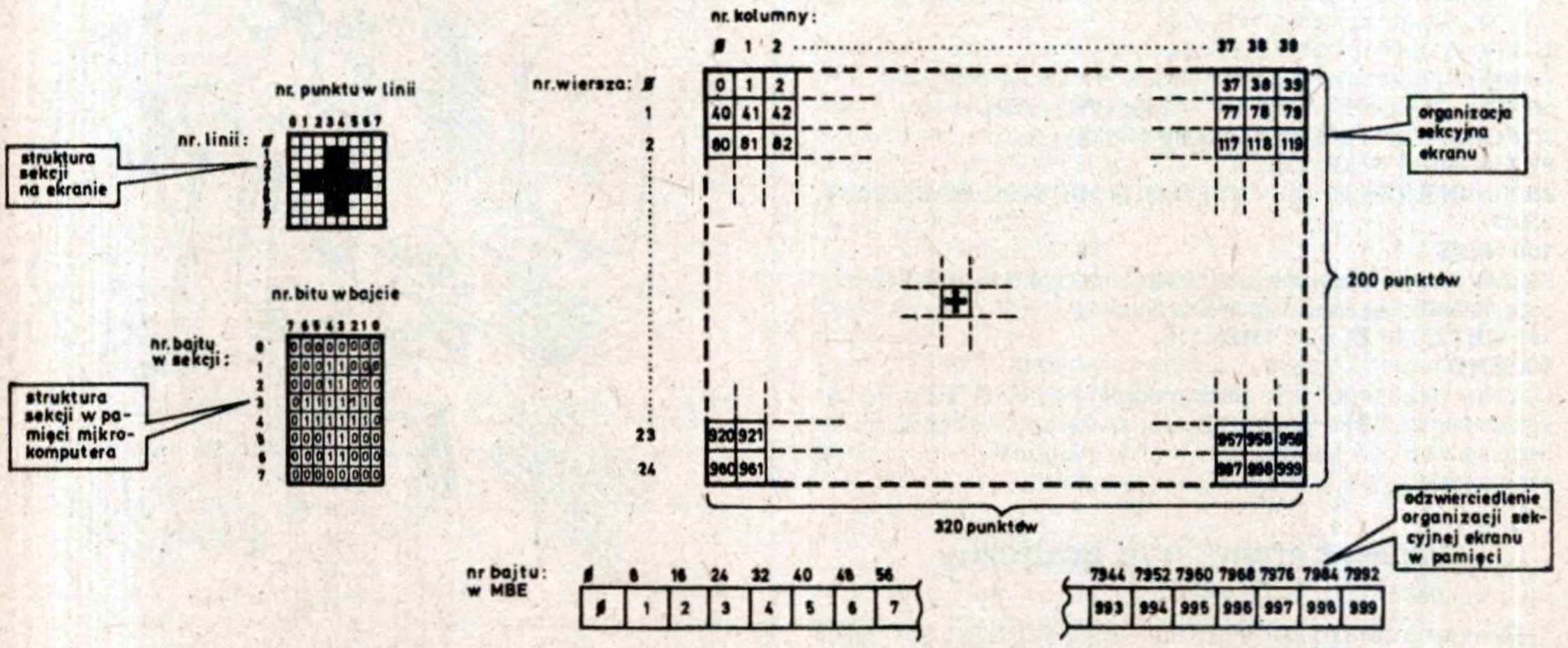
Tryb graficzny dużej rozdzielczości inicjujemy ustawiając piąty bit w 53265 bajcie na 1:

```
20 POKE 53265, PEEK (53265) OR 32
```

Powrót do trybu znakowego otrzymujemy po wykonaniu:

```
120 POKE 53265, PEEK (53265) AND 223
```

W trybie dwukolorowym definiujemy kolory w każdej sekcji osobno. Jako pamięć kolorów VIC wykorzystuje PAMIĘĆ EKRA-



Rys. 1 Struktura ekranu i jej odzwierciedlenie w pamięci mikrokomputera (w trybie graficznym)

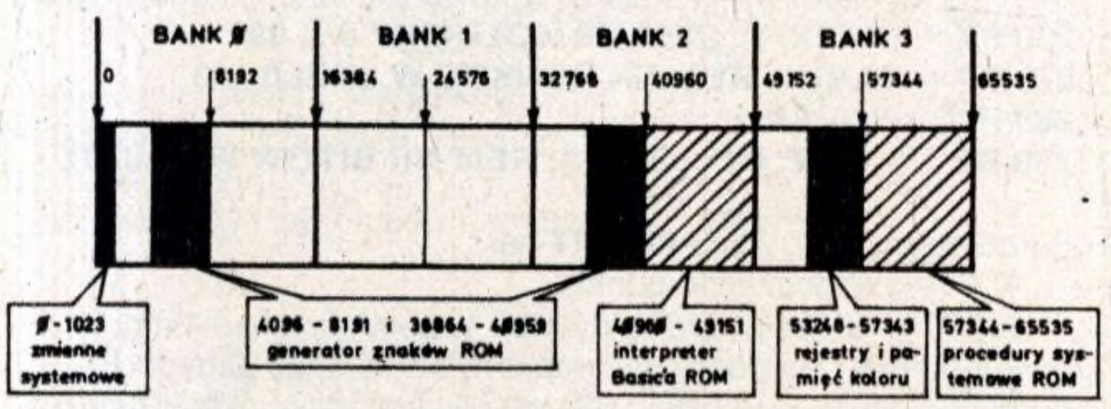
NU (PE). Każdej sekcji przyporządkowany jest bajt PE (kolejność występowania sekcji w MBE odpowiada kolejności bajtów PE). Kolor bitu o wartości 0 określają 4 najmłodsze bity w przyporządkowanym mu bajcie PE. Kolor bitu o wartości 1 określają 4 najstarsze bity.

W programie FUNKCJA tło ma być jednolite na całym ekranie (bity zerowe w MBE) koloru białego — kod koloru 1. Funkcja ma być kreślona (bity o wartości 1) w kolorze czarnym — kod 0. Należy zatem wykonać

40 **FOR** I = MAP **TO** MAP + 7999: **POKE** I,0:**NEXT**
50 **FOR** I = KAP **TO** KAP + 999: **POKE** I,1: **NEXT**

KAP jest adresem początkowym PE.

Musimy obecnie wybrać obszary pamięci, w których rezydować będzie MAPA BITOWA EKRANU i PAMIĘĆ EKRANU. Na rys. 2 przedstawiony jest zarys podstawowej konfiguracji pamięci C-64 z podziałem na BANKi i ewentualne obszary, w których można umieścić MBE i PE.



Oznaczenia: ■ - obszary, które nie należy wykorzystywać
▨ - obszary możliwe do wykorzystania, jeśli nie jest konieczne testowanie zawartości pomięci.

Rys. 2 Obszary pamięci C64 do wykorzystania na lokalizację mapy bitowej ekranu i pamięci kolorów (pamięci ekranu)

Jak wynika z rysunku, MBE może zajmować jedną z dwóch połówek BANK-u. PE powinna być umieszczona w innej połowie niż MBE. Określając położenie MBE i PE, możemy skorzystać z tabelek zamieszczonych w I części artykułu („IKS" nr 2/87). Wyboru w BANK-u dokonujemy, ustawiając odpowiedną wartość w 56576 bajcie. Lokalizację ustalamy wprowadzając do rejestru 53272 wartość z kolumny wyznaczającej adres początkowy MBE i wiersza wyznaczającego adres początkowy PE.

30 **POKE** 56572, 150: **POKE** 53272, 9: MAP = 24576: KAP = 16384

Funkcję do wykreślenia definiować będziemy w postaci wyrażenia arytmetycznego:

11 **DEF FN** F1 (XX) = SIN (XX)

Konieczne jest również podanie zakresu zmienności argumentu (XP, XK) i wartości funkcji (YP, YK):

10 XP = 0: XK = 8: YP = —2: YK = 2

Funkcję będziemy kreślić następującą sekwencją instrukcji:

60 DX = 320 {(XK — XP): DY = 200} (YK — YP)
70 FOR I = XP TO XK STEP ABS (1/DX)
80 X = INT (I — XP) DX
90 Y = INT {(FN F1 (I) — YP) DY}: GOSUB200: POKE BAJT, 2↑BIT
100 NEXT

Dla otrzymania gotowego programu FUNKCJA dołączamy procedurę PUNKT i jeszcze dwie linie programu:

100 GETZ$: IF Z$ = "" THEN110
140 END

Możemy teraz uruchomić nasz program komendą RUN. Po zakończeniu kreślenia funkcji program czeka na wciśnięcie dowolnego klawisza, co spowoduje powrót do trybu znakowego i zakończenie pracy.

Wielobarwny tryb graficzny

Wykonanie dwóch lub trzech różnokolorowych wykresów funkcji na tym samym ekranie w trybie dwukolorowym jest niemożliwe. Możemy rozwiązać ten problem wykorzystując tryb wielobarwnej pracy VIC-a, który pozwala nam na zdefiniowanie czterech różnych kolorów w każej sekcji.

Tryb wielobarwny uzyskamy instrukcją:

25 POKE 53270, PEEK (53270) OR 16

Powrót do trybu dwukolorowego umożliwia nam:

125 POKE 53270, PEEK (53270) AND 239

Jak już wspomniałem wcześniej, jednemu punktowi na ekranie odpowiada para bitów wskazująca, skąd pobrać kod barwy tego punktu. Odbywa się to według zależności:

PARA BITÓW	LOKALIZACJA KOLORU
00	rejestr 53281
01	najstarsze 4 bity odpowiedniego bajtu PE
10	najmłodsze 4 bity odpowiedniego bajtu PE
11	najmłodsze 4 bity PAMIĘCI KOLORU

Wprowadźmy do naszego programu definicje nowych funkcji:

12 DEF FN F2 (XX) = COS (XX)
13 DEF FN F3 (XX) = SIN (XX) + COS (XX)

Każda z funkcji ma być w innym kolorze: **F1** — czarna; **F2** — biała; **F3** — zielona. Tło w kolorze niebieskim. W tym celu należy wykonać instrukcję

55 POKE 53281, 6: FOR I = 55296 TO 56295: POKE I,5: NEXT

Ponieważ w trybie wielobarwnym mamy w poziomie tylko 160 punktów, na ekranie należy zmienić wiersz

60 DX = 160/(XK — XP): DY = 200/(YK — YP)

Do sekwencji instrukcji kreślących funkcję wprowadzamy nowe wiersze rysujące poszczególne krzywe:

81 Y = INT {(FN F1 (I) — YP)/DY}: GOSUB200: POKE BAJT, PEEK (BAJT OR (2↑B2) AND (255 — 2↑B1)
82 Y = INT {(FN F2 (I) — YP)/DY}: GOSUB200: POKE BAJT, PEEK (BAJT) OR (2↑B1) AND (255 — 2↑B2)
83 Y = INT {(FN F3 (I) — YP)/DY}: GOSUB200: POKE BAJT, PEEK (BAJT) OR (2↑B1 + 2↑B2)

Musimy jeszcze tylko usunąć wiersz 90 i dokonać modyfikacji procedury PUNKT i już mamy gotowy program FUNKCJA, pozwalający na wykreślenie trzech różnokolorowych krzywych.

O innych ciekawych właściwościach VIC-a będzie można przeczytać w kolejnym artykule.

S. WASIELEWSKI

Zmienne systemowe ROM-Spectrum

Wejście-wyjście

CHANS 23631/2
CURCHL 23633/4
STRMS 123568..605
P-POSN 23679
PRCC 23680/1
ECHO-E 23682/3

Interpretacja komend obsługujących urządzenia wejścia-wyjścia opiera się na dwóch metodach komunikacji: przez porty i kanały. **Porty** to w zasadzie dodatkowa przestrzeń adresowa, z którą wymiana informacji odbywa się rozkazami IN (odpowiednik PEEK) oraz OUT (odpowiednik POKE). Za pomocą tych instrukcji możemy kontrolować stan urządzenia zewnętrznego. I tak, drukarka korzysta z portu 254, joystick Kempstona z portu 31, a Interface I pracuje na portach 254, 239 i 247. Głośnik i magnetofon są również obsługiwane przez 254.

Kanały są formą określenia kierunku i sposobu prowadzenia korespondencji z urządzeniem zewnętrznym. Firmowo określono ich cztery: K-klawiatura (keyboard), S-monitor (screen), P-drukarka (printer) oraz R-bufor edycyjny (editing buffer channel record). Dane kanałowe umieszczone są w ROM-ie, lecz podczas inicjalizacji systemu następuje ich przerzucenie do RAM, a więc można je modyfikować. Obejmują 21 bajtów od adresu 23734, który zmienia CHANS.

Dla przykładu definicja kanału S zawiera:

adres	zawartość	znaczenie
23739 23740	244 9	2548 — adres procedury wyjścia — wydruk znaku na ekranie;
23741 23742	196 21	5572 — adres wejścia — w tym wypadku jest to procedura powodująca raport Invalid I'O device;
23743	83	„s" — identyfikator kanału;

Kanał **K** jest jedynym, który akceptuje zarówno dane wejściowe (dekodowanie przyciśniętego klawisza), jak też dane wyjściowe (wydruk znaku ASCII), co wynika z jego specyfiki. Jeśli kanał nie ma wejścia, to drukowany jest komunikat *Invalid I/O device*.

Tajemniczy kanał **R** jest używany wewnętrznie przez interpreter do przesyłania danych do bufora edycyjnego, co odbywa się po naciśnięciu **EDIT** lub do przestrzeni roboczej (z reguły operacje z łańcuchami tekstowymi). Nie można się nim posługiwać, programując w Basic'u, natomiast można go tak zmodyfikować, aby edycja linii była niemożliwa:

POKE 23744,124
POKE 23745,0

Dane kanałowe kończy bajt 128 będący jednocześnie sygnałem rozpoczęcia obszaru Basic.

Jeśli nie manipulujemy obszarem danych kanałowych (a zwykle nie), można się go pozbyć zwiększając pamięć przeznaczoną dla Basic'a:

```
10 POKE 23635, PEEK 23631
20 POKE 23636, PEEK 23632
30 POKE 23631, 175
40 POKE 23632, 21
50 REM PROG=23734, CHANS=5551
```

Zmienna CHANS wskazuje teraz obszar tych samych danych kanałowych, ale już w ROM-ie.

W danym momencie można przesłać dane tylko jednym kanałem, z którego korzystać można przez wybór skojarzonego z nim strumienia. Dane o strumieniach umieszczone są w zmiennej **STRMS** liczącej sobie 38 bajtów, co pozwala na zdefiniowanie 19 sposobów przesłań. Dwa bajty dla każdego strumienia zawierają liczbę, która dodana do 23733 (CHANS-1) wskazuje początek definicji wybranego kanału. Po włączeniu systemu do STRMS zostaje wysłane 14 bajtów identyfikujących 7 strumieni początkowych:

nr strumienia	kanał skojarzony	przesunięcie
253	K	1
254	S	6
255	R	11
0	K	1
1	K	1
2	S	6
3	P	16

Pierwsze dwa są zarezerwowane dla procedur obsługujących magnetofon i jeszcze do nich wrócimy. Strumień 255 współpracuje z kanałem **R**, strumień **0** z edytorem, 1 z wprowadzaniem danych (INPUT), 2 wysyła dane na ekran, 3 na drukarkę. Reszta 4..15 pozostaje do dyspozycji użytkownika.

Przed przesłaniem danych strumień musi być otwarty, tzn. skojarzony z odpowiednim kanałem. W chwili przesyłania kanał ten staje się dla systemu kanałem aktualnym i początek jego danych zostaje wpisany do zmiennej **CURCHL**. Teraz przykład:

```
10 BORDER 4
20 OPEN #4,"K"
30 LIST
40 LIST #4
50 CLOSE #4
60 GOTO 60
```

Otworzyliśmy (OPEN #) strumień 4 wg parametrów kanału **K**, aby wysłać listing na ekran edycyjny. Fizycznie otwarcie to polega na wpisaniu wspomnianego już przesunięcia do odpowiednich bajtów **STRMS**. Można to uczynić również komendami **POKE**. Zamknięcie strumienia (CLOSE #) polega na wpisaniu przesunięcia równego zeru. Zabronione jest zamykanie strumieni standardowych (0..3).

Po otwarciu przesyłamy dane komendą **PRINT**, a odbieramy rozkazem **INPUT**. Przy pracy w warunkach początkowych nie wyszczególniamy numerów strumieni przy tych instrukcjach, gdyż maszyna robi to automatycznie. **PRINT** oznacza więc naprawdę **PRINT** # 2, a **INPUT** posługuje się strumieniem 1 (INPUT # 1). LLIST znaczy to samo, co **LIST** # 3.

Wychodzić można na wszystkie standardowe kanały, lecz **INPUT** działa tylko w porozumieniu z kanałem K. Próba zaburzenia tego rozwiązania prostymi sposobami będzie nieudana.

```
10 POKE 23741,168
20 POKE 23742,16
30 INPUT #2;AT 1,0; LINE d$
```

Modyfikacja adresu wyjścia się udała, ale tylko na pierwszy rzut oka. Zwykle, jeśli użytkownik ma chęć na swoisty typ przesłania, wtedy definiuje własny kanał i pisze procedury wejścia i wyjścia. Dokładnie pokazuje to wybrany przykład **PRINT 16**. Program ten pozwala otrzymywać na monitorze znaki dwukrotnie szersze od firmowych przy zachowaniu możliwości korzystania ze wszystkich kodów kontrolnych.

Opis programu: (str. 26)
linie 160..240 — dołączenie danych kanału **D** na koniec danych kanałowych przez przesunięcie wskaźników RAM o 5 bajtów w górę (PROG = 23760);
linie 250..260 — otwarcie strumienia 5 przez skojarzenie z kanałem D; linie 280..300 — dane kanału **D**;
linie 310..550 — obsługa kodów kontrolnych w procedurze wyjścia;
linie 560...950 — obsługa kodów drukowanych w ASCII, (grafiki, słowa kluczowe);
linia 970 — obszar tworzenia szerokiego znaku.

Po kompilacji (lub wczytaniu bajtów już skompilowanych) należy zainstalować kanał przez np. **RANDOMIZE USR 6000**, a potem korzystać, stosując komendę wyjścia np.:

PRINT # 5; **AT** 5,5; **INVERSE** 1; *„prusakolep"*

Kod nie jest relokowalny. Zmiana adresu początkowego może odbyć się tylko po wczytaniu tekstu źródłowego do asemblera (linia 10) i ponownej kompilacji. Jeśli uda się Wam „podłączyć" znany program **PRINT 64** do strumienia 4, to będziecie dysponować możliwością jednoczesnego drukowania na ekranie w trzech trybach znakowych.

W zasadzie metoda kanałowo-strumieniowa staje się użyteczna w programowaniu dopiero po dołączeniu INTERFACE 1. Mamy tu rozszerzenie o cztery typy kanałów dających dużo większe możliwości. Pierwszy dotyczy microdrive'u i pozwala zapisywać dane w postaci plików.

```
10 OPEN #4,"m";1;"nazwa pliku"
20 PRINT #4,x
30 CLOSE #4
```

Zapisaliśmy na kasetę (cartridge) zmienną *x*, ale możemy też zapisać listing (LIST # 4), katalog (CAT # 4,1) lub dowolny zbiór liczb, czy znaków nawet bezpośrednio z klawiatury (INKEY$ # 4).

Następne dwa kanały dotyczą złącza szeregowego RS 232. Pozwala ono przesyłać dane w transmisji szeregowej, tzn. bit po bicie. Kanał **B** akceptuje wszystkie kody, lecz kanał **T** tylko podstawowy zbiór drukowanych znaków ASCII ignorując 8 bit. Ich zaletą jest możliwość jednoczesnego otwarcia, co można wykorzystać np. do wysyłania na drukarkę znaków kontrolnych kanałem **B**, a znaków ASCII kanałem **T**.

Czwarty kanał **N** dotyczy pracy w sieci, a więc pozwala na komunikację między poszczególnymi jednostkami Spectrum.

Powróćmy teraz do strumieni wykorzystywanych przez procedury kasetowe. Strumień 253 służy do wydruku napisu *Start tape, then press any key*, a więc używa wyjścia kanału **K**. W wypadku zapisu grafiki ekranowej napis ten niszczy dwie dolne linie obrazka. Jeśli chcemy zapisać cały ekran, postępujemy następująco:

```
10 POKE 23734,124: POKE 23735,0
20 PAUSE 1: PAUSE 0
30 SAVE "nazwaobrazka"SCREEN$
40 POKE 23734,244: POKE 23735,9
```

Zmieniono tu adres procedury wyjścia kanału **K**, który w krytycznym momencie wynosi 124 (pod tym adresem jest tylko instrukcja RET). Strumień 254 jest używany do wydruku typu i nazwy programu po przyjęciu nagłówka. Czasem informacje te nie są istotne, lecz niszczą wczytany wcześniej obrazek pilotujący. Przykładowy loader rozwiązujący tę kwestię:

Tu dla odmiany nastąpiła modyfikacja wyjścia dla kanału s. Jeżeli program wczytywany przez linię 30 ma autostart, to linię 40 trzeba zawrzeć wewnątrz niego, gdyż w przeciwnym razie nie będzie wyjścia na ekran główny.

```
10 LOAD "nazwa"SCREEN$
20 POKE 23739,124: POKE 23740,0
30 LOAD "program"
40 POKE 23739,244: POKE 23740,9
50 RUN
```

Na zakończenie gwoli ścisłości o trzech pozostałych zmiennych wyszczególnionych na początku.

P-POSN i **PR-CC** dotyczą drukarki. Pierwsza trzyma aktualną pozycję znaku w buforze, który rezerwuje obszar od adresu 23296 do 23551 włącznie. Po przesłaniu 32 znaków wysyłany jest kod nowej linii.

Druga natomiast dla tejże pozycji przechowuje adres w buforze. Starszy bajt **PR-CC** (23681) figuruje w instrukcji firmowej jako wolny, co w razie korzystania z drukarki czyni to stwierdzenie nieprawdziwym.

Zmienna **ECHO-E** przechowuje kopię pozycji wydruku w ekranie edycyjnym, czyli zmiennej SPOSNL. W momencie operacji w liniach edycji uaktualniana jest jednak tylko ta ostatnia, więc interpreter otrzymuje informacje skąd i dokąd drukowano. Ma to znaczenie w wypadku dodrukowywania w każdej linii ekranu edycyjnego spacji uzupełniających w kolorach **ATTR-P**.

Kolejnym tematem będą zmienne stanu systemu...

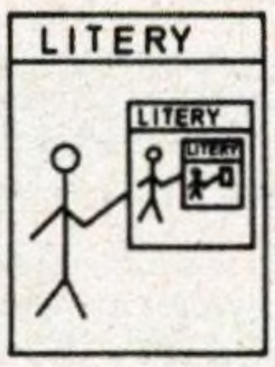
```
10 ; ** PRINT16
20 ; GENS3M2 assembler
30                ORG   60000
40 STR_5          EQU   23584
50 CHARS          EQU   23606
60 UDG            EQU   23675
70 TVDATA         EQU   23566
80 KANALD         EQU   23754
90 GETPOS         EQU   #0B03
100 CTRL          EQU   #0A04
110 PRT?          EQU   #0A69
120 RAPORT        EQU   #15C4
130 EXKAN         EQU   #0A80
140 POCONT        EQU   #0A87
150 MAKERM        EQU   5717
160
170 INSTAL        LD    HL,KANALD
180               PUSH  HL
190               LD    BC,5
200               CALL  MAKERM
210               LD    HL,DEFKAN
220               POP   DE
230               LD    BC,5
240               LDIR
250               LD    HL,21
260               LD    (STR_5),HL
270               RET
280 DEFKAN        DEFW  OUT_5
290               DEFW  RAPORT
300               DEFB  "D"
310 OUT_5         CP    32
320               JR    NC,ZNAKI
330               CALL  GETPOS
340               CP    6
350               JP    C,PRT?
360               CP    #18
370               JP    NC,PRT?
380               CP    16
390               JP    C,CTRL
400               LD    HL,KODPOS
410               CP    22
420               JR    NC,POS
430               LD    HL,KODKOL
440 POS           PUSH  HL
450               JP    GETPOS
460 KODPOS        LD    DE,KANPOS
470               JR    KANKOL
480 KODKOL        LD    DE,WYJ
490 KANKOL        LD    (TVDATA),A
500               JP    EXKAN
510 WYJ           LD    DE,OUT_5
520               JP    POCONT+3
530 KANPOS        LD    DE,WYJ
540               LD    (TVDATA+1),A
550               JP    EXKAN
560 ZNAKI         LD    DE,(CHARS)
570               LD    H,0
580               LD    L,A
590               ADD   HL,HL
600               ADD   HL,HL
610               ADD   HL,HL
620               ADD   HL,DE
630               EX    DE,HL
640               LD    HL,OBSZAR
650               PUSH  HL
660               POP   IX
670               LD    B,8
680 PETLA         LD    A,(DE)
690               PUSH  BC
700               CALL  TWZNAK
710               POP   BC
720               LD    (IX+0),L
730               LD    (IX+8),H
740               INC   IX
750               INC   DE
760               DJNZ  PETLA
770               LD    HL,(UDG)
780               PUSH  HL
790               LD    HL,OBSZAR
800               LD    (UDG),HL
810               LD    A,145
820               CALL  #09F4
830               LD    A,144
840               CALL  #09F4
850               POP   HL
860               LD    (UDG),HL
870               RET
880 TWZNAK        CALL  BITY4
890               LD    L,H
900 BITY4         LD    B,4
910 BIT1          RRCA
920               RR    C
930               SRA   C
940               DJNZ  BIT1
950               LD    H,C
960               RET
970 OBSZAR        DEFS  16
```

Krzysztof MAMCARZ

PASCAL (6)

Ten i kolejny odcinek adresowane są do Czytelników zainteresowanych głębiej możliwościami języka PASCAL: algorytmami rekurencyjnymi, przetwarzaniem tekstów, grafiką. Przedmiotem prezentowanego odcinka są programy rekurencyjne.

Obiekt nazywamy rekurencyjnym, *jeżeli częściowo składa się z siebie samego lub jego definicja odwołuje się do niego samego.* Rekurencję spotykamy nie tylko w matematyce, ale także w życiu codziennym. Wielu Czytelników pamięta jeszcze zapewne pierwszy elementarz, na którego okładce znajdował się następujący obrazek (rekurencyjny):



Rekurencja jest szczególnie przydatnym narzędziem w definicjach matematycznych. Pozwala ona na definiowanie funkcji przez określenie jej wartości w kroku pierwszym i wyznaczenie wartości funkcji dzięki określeniu jej wartości w kroku pierwszym i wyznaczenie wartości funkcji w każdym kolejnym kroku na podstawie wartości otrzymanej w kroku poprzednim. Klasycznym przykładem jest następująca definicja silni: $n! = \begin{cases} 1 & \text{dla } n = 0 \\ n \cdot (n-1)! & \text{dla } n > 0 \end{cases}$

Opis taki można niemal automatycznie przekształcić na deklarację funkcji pascalowej. Program wyznaczania wartości n! dla $n \leq 7$ został zamieszczony w przykładzie 19.

Przykład 19

Występujący w programie zapis {$a –} jest niezbędny dla programów rekurencyjnych. W praktyce wiele problemów można opisać za pomocą mechanizmu rekurencji. Weźmy pod uwagę następujące zadanie: *„należy wyznaczyć wszystkie możliwe rozmieszczenia k obiektów w n pudełkach, tak aby w każdym pudełku znalazł się jeden obiekt (zakładamy, że $k \leq n$)".*

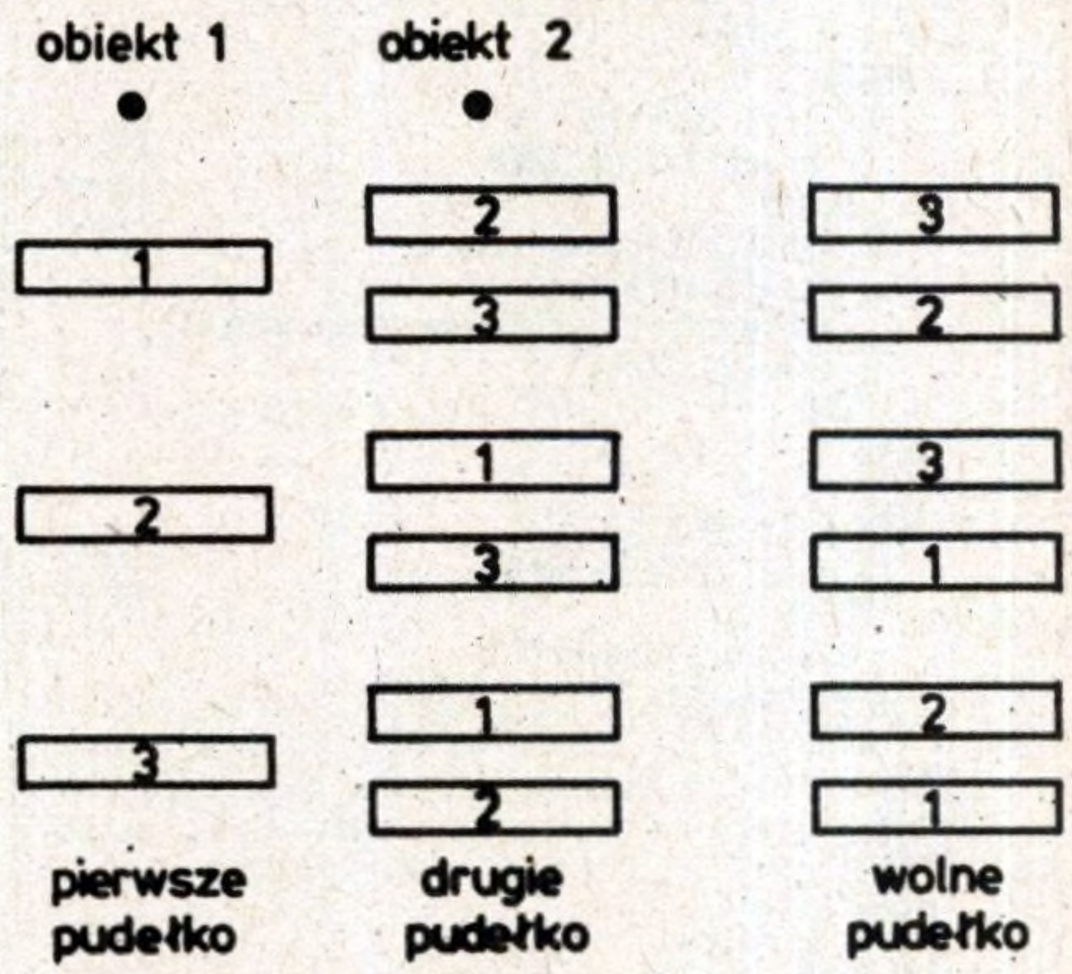
```
PROGRAM prog18;
{$a-}
 VAR
  n:integer;
 FUNCTION silnia(n:integer):integer;
  BEGIN
   IF n=0 THEN
        silnia:=1
   ELSE
        silnia:=n*silnia(n-1)
  END;
 BEGIN
  READ(n);
  WRITELN;
  IF (n>=0) AND (n<=7) THEN
   WRITELN(n:1,'!=',silnia(n))
  ELSE
   WRITELN('za duza wartosc n')
 END.
```

Metodę wyznaczania takich rozmieszczeń prześledzimy dla $k = 2$ i $n = 3$. Można przyjąć następujący sposób postępowania.

a) spośród wszystkich pudełek wybieramy jedno pudełko (zwane dalej pierwszym) i wkładamy do niego pierwszy obiekt.
b) spośród dwóch pozostałych pudełek wybieramy jedno pudełko (zwane drugim) i wkładamy do niego drugi obiekt.
c) nie zmieniając pierwszego pudełka, przekładamy drugi obiekt do następnego pudełka.

Wszystkie możliwe rozmieszczenia uzyskamy, powtarzając czynności a ÷ c dla każdego z możliwych sposobów wyboru pierwszego pudełka.

Jeżeli ponumerujemy pudełka od 1 do 3 i przyjmiemy zasadę, że spośród wolnych pudełek wybieramy pudełko o najmniejszym numerze, to dla tego przypadku uzyskamy następujące kolejne rozmieszczenia obiektów:



Jeżeli przez **P** oznaczymy zbiór numerów pudełek — $FP = \{1, 2,...,n\}$ — to nasze zadanie sprowadza się do znalezienia wszystkich ciągów równowartościowych $\langle P_1, P_2,..., P_i \rangle$ o wyrazach ze zbioru **P** (są to tzw. k-elementowe wariacje bez powtórzeń ze zbioru n-elementowego). Wyraz P_1 takiego ciągu możemy wybrać na n sposobów, wraz P_2, na n-1 sposobów, ogólnie: przy ustalonych wyrazach $P_1, P_2,..., P_{i-1}$ możemy jako pi wybrać dowolny spośród n-i + 1 elementów zbioru $P \setminus \{P_1, P_2,..., P_{i-1}\}$.

Program wyznaczania k-elementowych wariacji bez powtórzeń ze zbioru n-elementowego zamieszczonego w przykładzie 20.

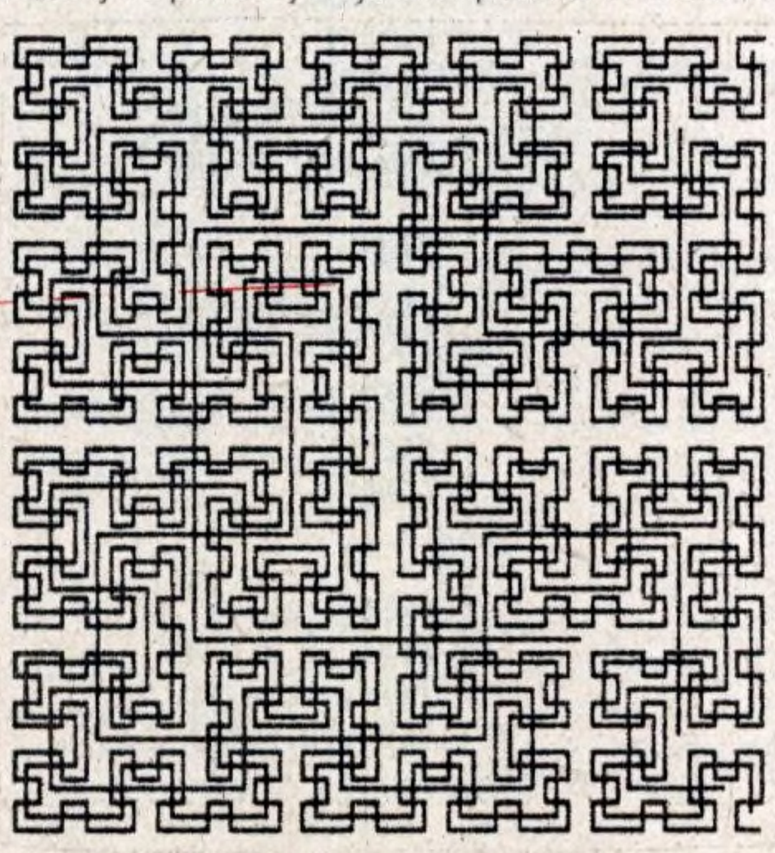
Przykład 20

```
PROGRAM prog19;
{$a-}
 CONST
  nn=9;
  kk=7;
 TYPE
  elem=ARRAY[1..nn] OF char;
  kontr=ARRAY[1..nn] OF boolean;
  sukc=ARRAY[1..kk] OF char;
 VAR
  dowybr:elem;
  praw:kontr;
  wynik:sukc;
  i,n,k,liczw:integer;
 PROCEDURE wyprowadz;
  VAR
   m:integer;
  BEGIN
   IF liczw>20 THEN
    BEGIN
     REPEAT UNTIL keypressed;
     clrscr;
     liczw:=2
    END
   ELSE
    liczw:=liczw+1;
   FOR m:=1 TO k-1 DO
    WRITE(wynik[m]);
   WRITELN(wynik[k])
  END;
```

```
PROCEDURE wybierz(i,n,k:integer; dowybr:ele…,
                  praw:kontr; VAR wynik:sukc);
 VAR
  l:integer;
 BEGIN
  IF i<k THEN
   BEGIN
    FOR l:=1 TO n DO
     IF praw[l]=true THEN
      BEGIN
       wynik[i]:=dowybr[l];
       praw[l]:=false;
       wybierz(i+1,n,k,dowybr,praw,wynik);
       praw[l]:=true
      END
   END
  ELSE
   FOR l:=1 TO n DO
    IF praw[l]=true THEN
     BEGIN
      wynik[k]:=dowybr[l];
      wyprowadz;
     END
 END;
BEGIN
 REPEAT
  READLN(k,n)
 UNTIL (k<=n);
 FOR i:=1 TO n DO
  BEGIN
   praw[i]:=true;
   READ(dowybr[i])
  END;
 clrscr;
 WRITELN;
 WRITE(' ',k,'-elementowe wariacje z ',
       n,'-elementowego zbioru:');
 FOR i:=1 TO n DO
  WRITE(' ',dowybr[i]);
 WRITELN;
 WRITELN;
 liczw:=1;
 wybierz(1,n,k,dowybr,praw,wynik)
END.
```

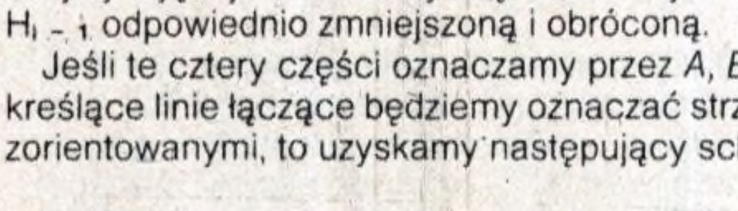
W programie wykorzystano:
— tablice znaków *dowybr* (do przechowania nazw elementów zbioru n-elementowego) i *wynik* (do przechowania kolejnych wariacji); elementami tych tablic są pojedyncze znaki,
— tablice wartości logicznych *praw* (do określania, spośród których elementów zbioru można wybierać kolejny wyraz ciągu),
— funkcję standardową *keypressed* przyjmującą wartość *true*, jeśli zostanie wciśnięty klawisz; w tym wypadku funkcja ta umożliwia stronicowanie wyświetlanych wariacji (po 20 na stronie) — zmiana strony następuje po wciśnięciu klawisza.

Za Wirthem *(Algorytmy + Struktury danych = Programy, s. 142)* przytoczymy jeszcze jeden, bardziej złożony przykład. Atrakcyjny układ graficzny pokazany na poniższym rysunku powstał ze złożenia



Rys. 1

pięciu krzywych. Regularny kształt tych krzywych sugeruje możliwość kreślenia ich za pomocą pisaka sterowanego przez komputer. Naszym celem jest określenie schematu rekurencji, zgodnie z którym mógłby być skonstruowany program kreślący. Trzy spośród pięciu nakładających się krzywych mają kształt przedstawiony na poniższym rysunku; oznaczamy je przez H_1 H_2 H_3.



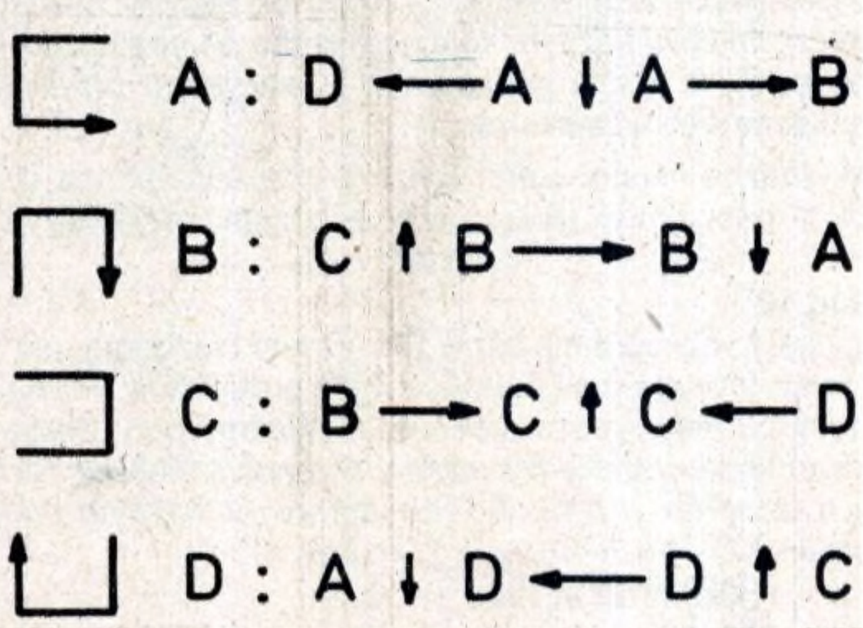
Rys. 2

Po bliższym przyjrzeniu okazuje się, że krzywą H_{i+1} otrzymano ze złożenia 4 krzywych H_i dwukrotnie zmniejszonych, odpowiednio obróconych i powiązanych razem trzema liniami. Zauważmy, że H_1 może być traktowana jako złożenie 4 pustych krzywych H_0, powiązanych że sobą trzema liniami prostymi. H_i jest zwana *krzywą Hilberta rzędu i*, od nazwiska wynalazcy D. Hilberta (1891).

Załóżmy, że naszymi podstawowymi narzędziami do kreślenia są dwie zmienne współrzędne *x* i *y*, procedura *ustaw pióro* (ustawiająca pióro pisaka w punkcie o współrzędnych *x* i *y*) i procedura *kreśl* (przesuwająca pióro z jego aktualnej pozycji do pozycji wskazanej przez x i y).

Ponieważ każda krzywa H_i składa się z 4 dwukrotnie zmniejszonych kopii krzywej H_{i-1}, wydaje się naturalne podzielenie procedury rysującej H_i na cztery części, z których każda rysuje krzywą H_{i-1} odpowiednio zmniejszoną i obróconą.

Jeśli te cztery części oznaczamy przez *A*, *B*, *C* i *D*, a procedury kreślące linie łączące będziemy oznaczać strzałkami odpowiednio zorientowanymi, to uzyskamy następujący schemat rekurencji:



Program kreślący n krzywych Hilberta H_1, H_2, ..., H_n zamieszczono w przykładzie 21.

Przykład 21

```
  PROGRAM prog20;
   {program kresli krzywe Hilberta }
   {zadanego rzedu                 }
{$a-}
 CONST
   h0=256;
 VAR
   i,h,x,y,x0,y0,n:integer;
 PROCEDURE b(i:integer); FORWARD;
 PROCEDURE c(i:integer); FORWARD;
 PROCEDURE d(i:integer); FORWARD;
 PROCEDURE InitGrafik;
  BEGIN
   WRITE(#$1b,'0',#$1b,'y');
  END;
 PROCEDURE leaveGrafik;
  BEGIN
   WRITE(#$1b,'1',#$1b,'x');
  END;
 PROCEDURE ustawpioro;
  BEGIN
   INLINE($2A/x/$EB/$2A/y/$cd/
          $5a/$fc/$EA/$BB);
  END;
```

```
PROCEDURE kresl;
 BEGIN
  INLINE($2A/x/$EB/$2A/y/$cd/$5a/
         $fc/$F6/$BB);
 END;
PROCEDURE a(i:integer);
 BEGIN
  IF i>0 THEN
   BEGIN
    d(i-1); x:=x-h; kresl;
    a(i-1); y:=y-h; kresl;
    a(i-1); x:=x+h; kresl;
    b(i-1)
   END
 END;
PROCEDURE b;
 BEGIN
  IF i>0 THEN
   BEGIN
    c(i-1); y:=y+h; kresl;
   .b(i-1); x:=x+h; kresl;
    b(i-1); y:=y-h; kresl;
    a(i-1)
   END
 END;
PROCEDURE c;
 BEGIN
  IF i>0 THEN
   BEGIN
    b(i-1); x:=x+h; kresl;
    c(i-1); y:=y+h; kresl;
    c(i-1); x:=x-h; kresl;
    d(i-1)
   END
 END;
PROCEDURE d;
 BEGIN
  IF i>0 THEN
   BEGIN
    a(i-1); y:=y-h; kresl;
    d(i-1); x:=x-h; kresl;
    d(i-1); y:=y+h; kresl;
    c(i-1)
   END
 END;
BEGIN
 initgrafik;
 i:=0; h:=h0;
 x0:=(h DIV 2)+75;
 y0:=x0+50;
 READ(n);
 REPEAT
  i:=i+1; h:=h DIV 2;
  x0:=x0+(h DIV 2);
  y0:=y0+(h DIV 2);
  x:=x0; y:=y0;
  ustawpioro;
  a(i);
 UNTIL i=n;
 REPEAT UNTIL keypressed;
 leavegrafik
END.
```

Procedura *A* jest wywoływana w programie głównym po jednym razie dla każdej z nakładanych na siebie krzywych Hilberta. Program główny określa również punkt początkowy krzywej, tj. początkowe wartości zmiennych *x* i *y* oraz przyrost jednostkowy *h*. Przez *ho* oznaczono całą szerokość strony, na której będą kreślone krzywe.

Ponieważ w PASCAL-u obowiązuje zasada, że pierwsze wystąpienie obiektu powinno być jego definicją, zastosowano tzw. deklaracje wyprzedzające procedur *B*, *C*, i *D*.

W deklaracji takiej podaje się pełny nagłówek procedury n blok procedury zastępuje się słowem kluczowym *FORWARD*.

W dalszej części programu deklaruje się blok procedury, poprzedzony jej identyfikatorem (parametrów się nie podaje).

Procedury graficzne omówimy w jednym z kolejnych odcinków.

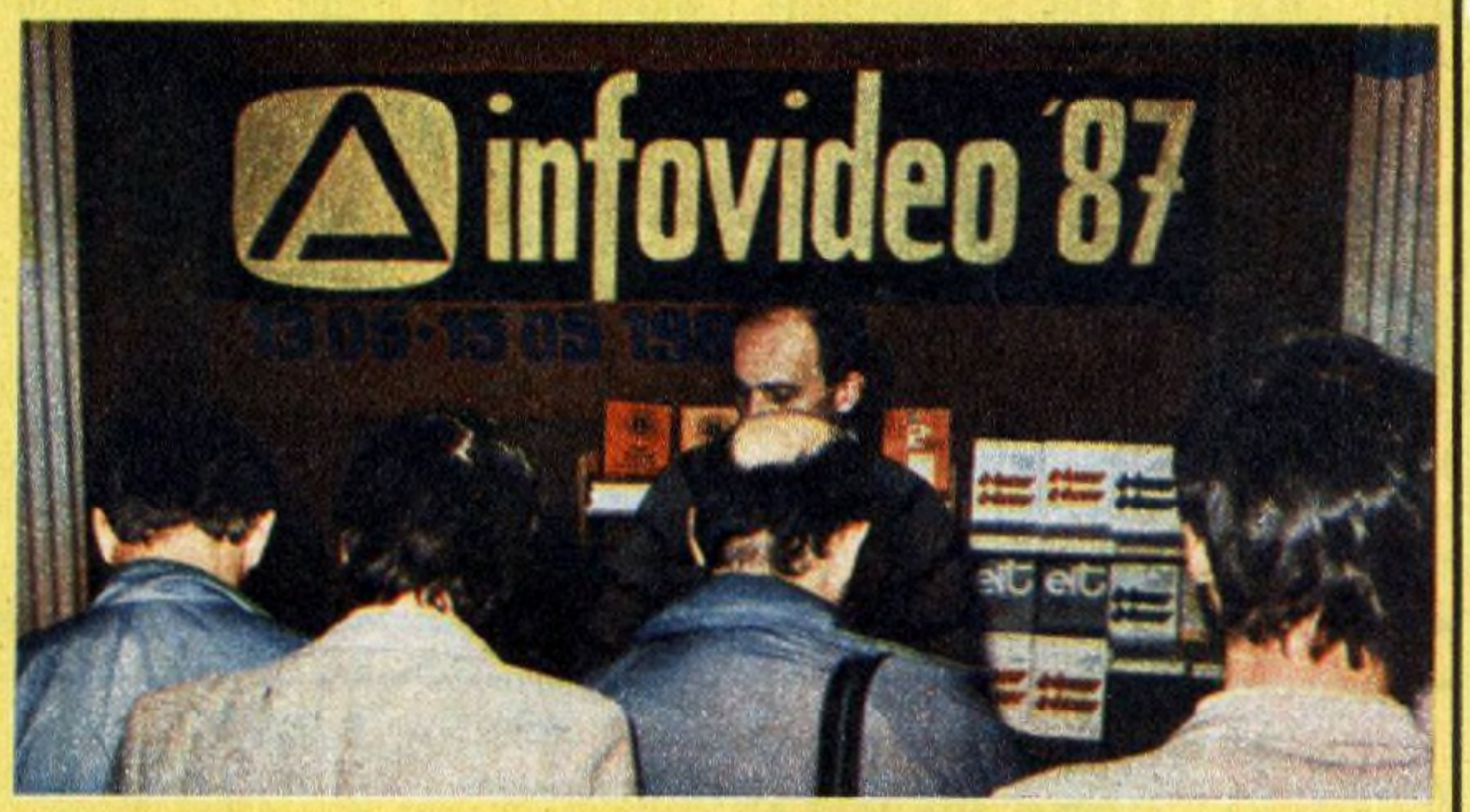
S. ROZMUS

Tym razem nie było sensacji — umiarkowana frekwencja jest rzeczywistym obrazem zainteresowania informatyką w kraju.

Infowideo'87

Największą frajdę na tegorocznym Infowideo miały maluchy. Pierwszy kontakt z komputerem z pewnością na długo pozostanie w pamięci przedszkolakom.

Foto: Jan Zelman



ZŁOŚLIWE ŁADUNKI

Obwody scalone („mądre" kostki) zawierają coraz więcej coraz mniejszych funktorów zerojedynkowych. Ta miniaturyzacja powoduje, że te najnowocześniejsze, a więc MOS (Metal Oxide Semiconductor) wymagają szczególnej ostrożności: są uczulone na ŁADUNKI ELEKTROSTATYCZNE.

Przypominamy: ładunki elektrostatyczne powstają nie tylko przy pocieraniu bursztynu o wełnę, ale również, gdy trzemy SYNTETYK o... WSZYSTKO!

Kto nie wierzy, niechaj gwałtownie ściągnie sweter z anilany: może usłyszeć trzeszczenie iskierek, a jeśli będzie ciemno, to je zobaczy.

Ładunki elektrostatyczne skutecznie BLOKUJĄ obwody scalone MOS. Efekt jest taki, że komputer (kalkulator) nagle... głupieje. Coś pokazuje, ale źle!

Niektórzy „dowcipni" radzą w takiej sytuacji... wyjazd do Japonii.

Nie jest to konieczne, gdyż często możemy zreperować samodzielnie. Wymagana tu jest zręczność radioamatora i jego narzędzia (wkrętaki, pinceta, lutownica, lutowie...). Jeśli tego nie mamy, to radzimy poszukać zaprzyjaźnionego radiomajsterkowicza.

Dodatkowe wyposażenie to arkusik blachy (duży talerz), 30 centymetrów przewodu izolowanego (tzw. linka LY), lampa „kreślarska", spodeczek i ewentualnie lupka.

Z przewodu usuwamy po 10 milimetrów izolacji. Z jednego końca robimy „pędzelek", a drugi bielimy cyną.

Teraz strój: zdejmujemy SYNTETYCZNĄ odzież (sweter z anilany, nylonową koszulę...) i zakładamy bawełniane. Bardzo starannie myjemy ręce i dokładnie suszymy.

Uwaga: suche, czyste ręce bardzo często kładziemy na arkuszu blachy, aby usunąć ładunki. **Lutownice małej mocy starannie uziemiamy (korpus). Wyjmujemy komputerowe bateryjki.**

Na blachę kładziemy „zwariowany" komputer. Uważnie lokalizujemy wkręty (warto zrobić szkic rozmieszczenia). Ostrożnie wykręcamy — niektóre są przyklejone lakierem. Wykręcone kładziemy na spodeczek.

Gdy wszystkie wykręciliśmy, bardzo ostrożnie „otwieramy" komputer: *bez użycia noża, szczypiec — samymi palcami.* **Jeśli napotkamy opór, to szukajmy zapomnianego wkrętu.**

Po „otwarciu" warto zrobić szkic wnętrza: wynotować symbole obwodów scalonych.

Teraz często kładźmy ręce na blachę. Często róbmy przerwy i dużo, dużo oglądajmy: „połówki" komputera są z reguły połączone cienkimi linkami lub kruchymi taśmami.

Wyszukujemy „masę": ma ona liczne punkty lutownicze połączone zazwyczaj CZARNYM przewodem.

Do masy lutujemy jeden koniec naszej linki. Rozładowujemy ręce i drugim końcem linki „pędzlujemy" po WSZYSTKICH METALOWYCH elementach: końcówkach obwodów scalonych, metalowych ścieżkach druku, wszystkich metalowych punktach (jeśli jest wygodny dostęp, to również po drugiej stronie płytki drukowanej).

Ten zabieg LIKWIDUJE ŁADUNKI ELEKTROSTATYCZNE. Radzimy zrobić krótką przerwę, obejrzeć notatki i znów... rozładować ręce, kładąc je na blachę.

Zabieg „pędzlowania" powtarzamy (może coś przeoczyliśmy).

Znów przerwa i rozładowanie rąk. Przylutowany koniec odlutowujemy i bardzo uważnie oglądamy, czy „coś" nie odłączyliśmy.

Powoli i co najmniej uważnie składamy „połówki". Ze spodeczka pincetką bierzemy po jednym wkręcie i „dociągamy częściowo". Gdy spodeczek pusty, to „z wyczuciem" dokręcamy wkręty.

Bardzo uważnie oglądamy złożony komputer: NIE POWINNO BYĆ ŚLADÓW!

Zakładamy bateryjki (jeśli je ma). Jeszcze raz oglądamy i START. Życzymy „ożywienia komputera".

Janusz MILLER

AMSTRAD

GRA „VIA MALA"

Jest to prosta gra zręcznościowa, polegająca na przejechaniu jak najdłuższego odcinka krętej drogi. Wygrywa ten, kto zdobędzie większą liczbę punktów. Zetknięcie się w czasie jazdy z krawędzią drogi grozi katastrofą. Życzymy udanej zabawy.

```
10 CLS
20 REM 'VIA MALA'
30 PRINT:PRINT:PRINT SPC(12),"OPIS GRY"
40 PRINT:PRINT:PRINT:PRINT
50 PRINT SPC(8);CHR$(242);" - RUCH W LEWO"
60 PRINT:PRINT
70 PRINT SPC(8);CHR$(243);" - RUCH W PRAWO"
80 FOR I=1 TO 4000:NEXT
90 ENV 1,1,15,30,15,-1,12:MODE 1:INK 0,0:PAPER 0:B
ORDER 0:INK 1,6:INK 2,26:INK 3,21
100 CLS:P=100:Z=50:X=15:PEN 2:          FOR I=1 T
O 25:PRINT TAB(X);CHR$(143);SPACE$(10);CHR$(143):N
EXT I:WX=20:LOCATE WX,1:PEN 1:PRINT CHR$(239)
110 Z=Z-1:IF Z<1 THEN Z=1
120 FOR I=1 TO Z:SOUND 2,500+Z*10,1,15-(Z\8),0,0,4
:NEXT I:SOUND 2,200+Z*20,17,15-(Z/8),0,0,4
130 LOCATE WX,1:PRINT" ":T$=INKEY$:IF T$="" THEN 1
50
140 IF T$=CHR$(243) THEN WX=WX+1 ELSE IF T$=CHR$(2
42) THEN WX=WX-1
150 P=P+(50-Z):X=X+SGN(RND-RND):IF X<1 THEN X=1 EL
SE IF X>29 THEN X=29
160 IF WX>X AND WX<X+11 THEN PEN 2:LOCATE 1,1:PRIN
T CHR$(11):LOCATE 1,1:PRINT CHR$(30);TAB(X);CHR$(1
43);SPACE$(10);CHR$(143):PEN 1:LOCATE WX,1:PRINT C
HR$(239):GOTO 110
170 CLS:FOR I=26 TO 0 STEP -1:INK 0,I:BORDER I:SOU
ND 1,0,4,15,0,0,I:NEXT          I:SOUND 1,0,100,0,1
,0,26:PEN 3:PRINT"PUNKTY:";P:PRINT:INPUT "GRASZ DA
LEJ? T/N",NO$:IF UPPER$(LEFT$(NO$,1))="T" THEN 100
```

LIGA MYŚLĄCYCH

Zadanie 1

Z pełnego naczynia zawierającego kwas o stężeniu 96 proc. odlano 2,5 litra i dopełniono naczynie kwasem o stężeniu 80 proc., a następnie odlano 2,5 litra i znowu dopełniono naczynie kwasem o stężeniu 80 proc. W wyniku tych czynności otrzymano kwas 89-procentowy.

Należy znaleźć pojemność naczynia.

Zadanie 2

Cyfry liczby trzycyfrowej tworzą postęp geometryczny. Suma cyfr jedności i setek stanowi $\frac{5}{2}$ cyfry dziesiątek. Jeżeli od szukanej liczby odejmiemy liczbę ułożoną z tych samych cyfr, lecz napisanych w odwrotnym porządku, to otrzymamy 297. Należy znaleźć tę liczbę.

Zadanie 3

Dziadek i babcia mają w sumie 154 lata. Dziadek ma 2 razy tyle lat, ile babcia miała wtedy, kiedy dziadek miał tyle, ile babcia ma obecnie.

Proszę podać, ile lat ma dziadek, a ile babcia?

Zadanie 4

Na szalki wagi postawiono dwa wiadra napełnione po brzegi wodą. W jednym z wiader pływa kawałek drewna. Które z wiader jest cięższe? Odpowiedź należy uzasadnić.

Zadanie 5

Marek jadąc autobusem zauważył Pawła, który szedł wzdłuż ulicy po chodniku, ale w kierunku przeciwnym do ruchu autobusu. Autobus niebawem zatrzymał się na przystanku, a Marek po wyjściu z niego pobiegł w kierunku oddalającego się Pawła. Od momentu zauważenia Pawła przez Marka do chwili wyjścia Marka z autobusu minęło 10 sekund. Przyjmując, że prędkość biegu Marka jest dwa razy większa od prędkości chodu Pawła i pięć razy mniejsza od prędkości autobusu, obliczyć po upływie ilu sekund Marek dogoni Pawła?

Zadanie 6

Pewien robotnik kopie dół. Na zapytanie przechodnia, jak głęboki będzie dół, odpowiedział: „mój wzrost wynosi 180 cm, gdy wykopię dół do końca, moja głowa będzie o tyle poniżej powierzchni ziemi, o ile teraz, gdy już wykopałem połowę głębokości dołu, jest powyżej niej". Jaka będzie głębokość wykopanego dołu?

Rozwiązania zadań prosimy przesyłać do redakcji do końca września br., z dopiskiem „Liga Myślących". Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi" czekają dodatkowe cenne nagrody niespodzianki.

Uwaga! **Do programu „Sumy kontrolne" (IKS nr 4) wkradł się błąd. Poniżej zamieszczamy prawidłowy wydruk:**

```
310 GOSUB 600
345 NM$=STR$(HOLD):POSITION
    28-LEN(NM$),1:PRINT
    NM$:NM$="":SYN$=H1$:DOT=0
    :NUMALL=0:GOTO 130
```

Pocztowa giełda

Rozwiązanie krzyżówki z "IKS-a" nr 3/87
Hasło brzmi: „IKS" — PISMO DLA CIEBIE
Bony pieniężne (1000 zł) wylosowali: Maria Wawrzyniak — Konin, Dariusz Jankowski — Siedlce, Zdzisław Ziółkowski — Łódź, Piotr Nowak — Sosnowiec, Marek Zemfler — Gdańsk.
Nagrody książkowe otrzymują: Ryszard Galiński — Warszawa, Sławomir Kossakowski — Wałbrzych, Mieczysław Cetnar — Czchów, Marek Oleszczuk — Lublin, Zbigniew Kaliszyk — Parczew, Joanna Pasternak — Brzeg, Eugeniusz Wojciechowski — Poznań, Witold Chwiałkowski — Sandomierz, Jacek Piska — Płock, Tadeusz Hermanowski — Białogard.

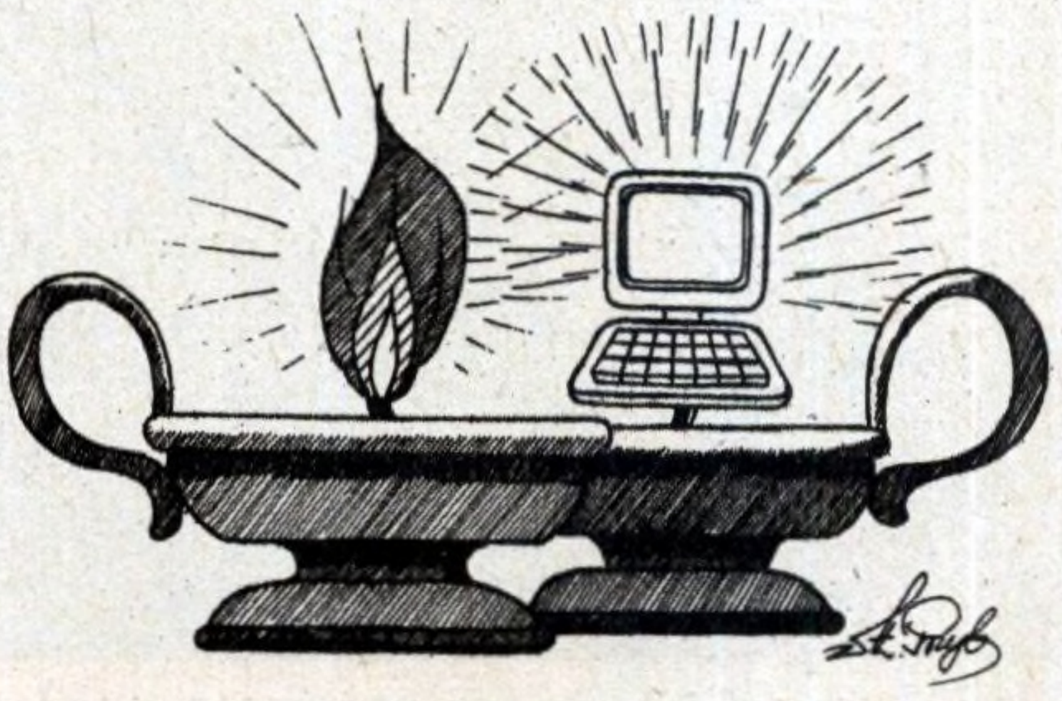
● Kupię nową klawiaturę do ZX-81. Tomasz Sadzikowski, ul. Grzybowa 1/30 32-503 Chrzanów 5.

● Wymienię programy na Commodore 64 oraz zakupię. Wioletta Kryza, ul. Żeromskiego 36/11, 97-425 Zelów.

● Kupię komputer Commodore 16 + magnetofon 1531. Adam Wojcieszyk, ul. Świerczewskiego 5 m 43, 86-200 Chełmno.

● Literatura na Atari, informacja — koperta zwrotna + znaczek. Adam Gawlikowski, ul. Z. Wolskiego 5/1 m 6, 09-400 Płock.

● Pilnie poszukuję literatury, programów do mikrokomputera ZX-81 (1K). Piotr Kempys, ul. Bociania 9, 43-316 Bielsko-Biała.



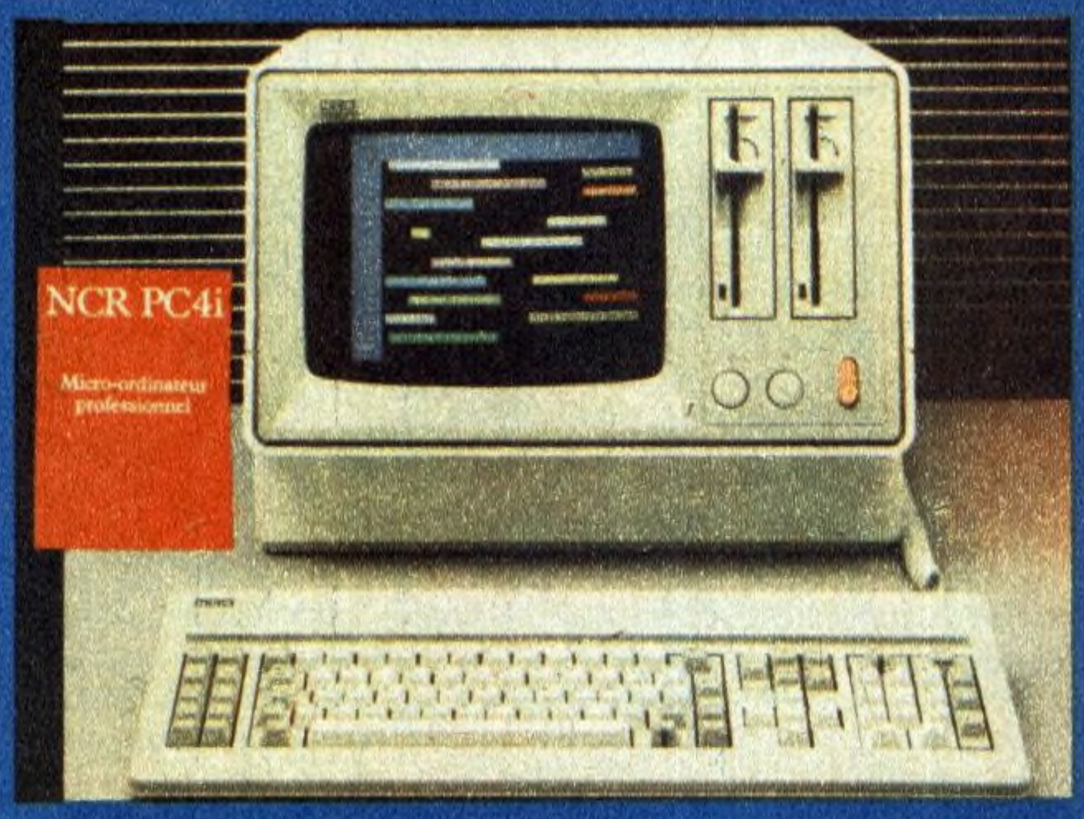
Rys. Michał Przybyłowski

KRZYŻÓWKA NR 6

POZIOMO: 5) twarda, cienka tektura, stosowana w elektrotechnice jako materiał izolacyjny, 6) zaprawa do zupy z tłuszczu roztartego z mąką i zasmażonego, 7) niemożność odbierania wrażeń słuchowych wskutek wrodzonych lub nabytych zmian w narządzie słuchu, 8) oddzielanie się czerwonych krwinek od osocza, 13) język literacki starożytnych i średniowiecznych Indii, 14) Homo sapiens, 15) tech. okrągła tarcza, której środek nie leży na osi jej obrotu (ekscenter),

PIONOWO: 1) urządzenie do łączenia dwóch elementów w celu przeniesienia ruchu z jednego na drugi, 2) tech. układ ziarn, warstw lub włókien w wewnętrznej budowie materiału, 3) stan społeczny wywodzący się z rycerstwa, 4) dotykanie ciała w miejscach wrażliwych i pobudzanie tym do śmiechu, 9) hodowca, sprzedawca ptaków lub gat. pająka, 10) obelga, zniewaga, 11) amator „białego szaleństwa", 12) pomost na słupach do przeprowadzenia trasy komunikacyjnej ponad poziomem terenu.

Litery z kratek ponumerowanych dodatkowo w prawym dolnym rogu od 1 do 14 utworzą hasło, które wystarczy nadesłać jako rozwiązanie zadania, pod adresem redakcji do końca września 1987 r. Wśród czytelników rozlosujemy bony pieniężne i nagrody książkowe.



NCR PC4i, NCR PC6 i NCR PC8 są szesnastobitowymi komputerami profesjonalnymi. Pierwszy z nich oparty jest na mikroprocesorze 8088 (4.77MHz). Może być wyposażony opcjonalnie w koprocesor arytmetyczny 8087. Pamięć operacyjna ma pojemność 256 Kb (z możliwością rozszerzenia do 640 Kb). Na kolorowym bądź czarno-białym monitorze o rozdzielczości 640 na 400 punktów standardowo mieści się 25 80-znakowych wierszy. W obudowie monitora znajdują się dwie dyskietki 5"1/4 (po 360 Kb).

NCR PC6 (mikroprocesor 8088-2; opcjonalnie 8087) wyposażony jest w dwa poziomy grafiki. Standard dla monitora monochromatycznego 12" wynosi 720x350 punktów, a dla kolorowego 14" 640 na 200.

Podobne parametry ma NCR PC8. Systemem operacyjnym, pod którym pracują wszystkie te modele, jest DOS (NCR—DOS 2.11 lub NCR—DOS 3.1). NCR PC8 jest również wyposażany w XENIXa V. Oczywiście, istnieją możliwości rozbudowy konfiguracji podstawowej tych mikrokomputerów: zwiększenia ich pamięci zewnętrznych o floppy dyski i dyski twarde.

Francuskie mikrokomputery (oparte na sprawdzonych już mikroprocesorach) są całkowicie zgodne ze standardem IBM, a produkowane są głównie na francuski rynek. C.

„IKS" — dodatek „Żołnierza Wolności". Redagują Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogołek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe", Warszawa ul. Grzybowska 77, Fotoskład i druk rotograwiurowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 8896 Nr. ind. 361682 K-62