

I NFORMATYKA
K OMPUTERY
S YSTEMY



CENA 120 ZŁ

DODATEK DO „ŻOŁNIERZA WOLNOŚCI” NR 9/1988 ISSN 0860 — 2794

Informatyka w szkole
— str. 14
Co widzi komputer

*Przykładowy tekst
pamiętamy w
pamięci komputera
624 piksele w poziomie
300 w pionie — str. 16*



KOMPUTERY A PSYCHIKA CZŁOWIEKA

Coraz więcej ludzi na całym świecie na co dzień ma bezpośredni kontakt z nowoczesnymi urządzeniami telekomunikacyjnymi i wymiany informacji. W przedsiębiorstwach, biurach, placówkach służby zdrowia i innych instytucjach wykorzystywane są komputery. Automatyzacja i komputeryzacja wkracza także do domów. Pod względem fizycznym z pewnością urządzenia te coraz bardziej ułatwiają pracę i oszczędzają czas. Uczni zaczynają się zastanawiać, jak nowa technika wpływa na sposób myślenia, postrzeganie otaczającej rzeczywistości oraz na zachowanie ludzi.

— *Przede wszystkim następuje ograniczenie wszystkich umiejętności człowieka, które nie łączą się bezpośrednio z tym, co nazywamy racjonalnym działaniem* — podkreślają **Fritz Boehle** oraz **Brigitte Milkau** z Instytutu Nauk Społecznych w Monachium. Ich zdaniem potwierdzają to badania osób zatrudnionych przy obrabiarkach sterowanych numerycznie. Obserwuje się u nich głównie ograniczenie postrzegania.

Tego rodzaju zjawisko występuje także w wypadku komputera, choć człowiek nie odczuwa tu, że ma pod ręką jakieś konkretne urządzenie. Mimo tego zaraz pojawia się grymas na jego twarzy, jak tylko nastąpi coś nieprzewidzianego. Według specjalistów, człowiek znajduje się wtedy jakby w „szizofrenicznej sytuacji”, kiedy powinien zaraz zrobić to, czego się „wyuczył”, ale nie jest w stanie temu sprostać.

Psycholog **Robert Schurz** oraz matematyk **Joerg Pflugler** prowadząc badania osób które na co dzień posługują się komputerem, doszli do wniosku, że reprezentują oni „ma-

szynowy” typ osobowości. Myślenie takiego człowieka charakteryzuje formalizm i operowanie jedynie wyuczonymi algorytmami. Ludzie o takiej osobowości nie znoszą wszelkich niespodziewanych sytuacji, w ogóle tego, co zakłóca ustalony porządek. Mają silną potrzebę kontrolowania wszystkiego, unikają jednocześnie kontaktów z innymi ludźmi, gdyż obawiają się zajścia czegoś, czego się nie spodziewają. W życiu codziennym u takich ludzi wszystko musi przebiegać tak jak w zaprogramowanej maszynie — musi być zaplanowane i systematyczne.

Zdaniem wielu uczonych, tego rodzaju rozdział sfery myślowej w psychice człowieka oraz zmysłowego i emocjonalnego kontaktu ze światem nie jest spowodowany wyłącznie przez automatyzację i komputeryzację. — *To niekorzystne zjawisko pojawiło się znacznie wcześniej* — podkreśla **Michael Wimmer**, politolog z Wiednia — *i związane jest z cywilizacją zachodnioeuropejską, w której liczy się wyłącznie abstrakcyjne myślenie i całkowicie zaprogramowana wraz z ludźmi produkcja.*

Spójrzmy teraz na relację komputer — psychika człowieka nieco z innej strony, a właściwie nawet perspektywy. Na początku lat sześćdziesiątych psycholog **Raymond Fowler** z University of Alabama był jednym z pierwszych, którzy do komputera wprowadzili program Minnesota Multiphasic Personality Inventory — wielofazowy wykaz osobowości. Gdy program dobrze się przyjął, na język cyfr zaczęto przekładać inne testy psychologiczne.

Dzisiaj, gdy ktoś czuje się przygnębiony, czy zestresowany zwraca się o pomoc do psychologa specjalisty. Pacjent sam odtwarza

własną charakterystykę. Dokonuje tego siedząc przed komputerem i odpowiadając na bogatą listę pytań dotyczących własnej osoby. W ciągu kilku minut od naciśnięcia ostatniego klawisza komputer opracowuje i wydrukowuje ścisłą charakterystykę ankietowanego. Jak twierdzi profesor psychiatrii w University of Wisconsin-Madison, **John Greist**, w dziedzinie wywiadów z pacjentem komputery są szczególnie przydatne, gdyż bardzo często przewyższają klinicystów pod względem dokładności i staranności badań. Co więcej, osobisty komputer stojący na biurku terapeuty może błyskawicznie posortować i przeanalizować setki odpowiedzi, a następnie wydrukować opis osobowości pacjenta. Mankamentem jest jednak fakt, że komputery nie potrafią przeanalizować wielu spontanicznych wypowiedzi wyartykułowanych w potocznym języku, chociaż najnowsze eksperymentalne systemy są w stanie zrozumieć zdania w zakresie kilku tysięcy słów.

Najkorzystniejsze dla pacjenta byłoby niewątpliwie sprzęgnięcie komputera z intuicją doskonałego lekarza-naukowca. Komputery bowiem nie potrafią jeszcze samodzielnie przeprowadzać skomplikowanych zadań terapeutycznych. Największe zastosowanie znajduje na razie ich zdolność do przeprowadzania operacji statyczno-obliczeniowych. Prognozą do dalszego rozwoju jest zdolność uczonych do formułowania reguł terapeutycznych, które będzie można wprowadzać do komputera. Wówczas w większym stopniu możliwa będzie penetracja duszy człowieka przez zaprogramowaną przez niego samą maszynę.

(Raj.)

W NUMERZE:

W szponach Atari (18)	— str. 4
Edytor Basic'a	— str. 8
Helios — monitorem	— str. 10
Wyznaczenie rozwiązań układu równań różniczkowych zwyczajnych I-go rzędu	— str. 12
PRIMUM N * N NOCERE, czyli jak chcemy uczyć Podstaw Informatyki	— str. 14
Wprowadzenie do Podstaw Informatyki	— str. 16
Nuty z komputera (4)	— str. 19
Lilavati (8)	— str. 24
Różne sposoby kopiowania ekranu mikrokomputera na drukarce	— str. 26
Dodatkowe zestawy znaków graficznych do tworzenia tablic i ramek	— str. 28
Słownik Informatyczny	— str. 29
Liga Myślących	— str. 31



(Foto: J. Zelman)

Rady i porady programisty, czyli 3 × P (1)

Na początek podprogram, który znaleźć powinien się w każdym programie. Jego głównym zadaniem jest wyświetlenie na ekranie monitora tekstu zawierającego informacje o przeznaczeniu i funkcjach napisanego przez Ciebie programu.

Jednocześnie można go wykorzystać do wywoływania tzw. helpów w trakcie realizacji programu. Eliminuje to konieczność wprowadzania, czasami długich tekstów do ciała programu. Długie teksty w programie powodują bardzo szybkie „pożeranie” pamięci niejednokrotnie niezbędnej do załadowania programu.

Pamiętając o tym zapoznaj się z niżej przedstawionym programem. Zasadniczą jego część stanowi podprogram wyświetlania tekstu, jemu też poświęcę najwięcej uwagi. Aby skorzystać z tego podprogramu należy wcześniej korzystać z edytora tekstu np. TASWORD, AM-SWORD itp. zredagować tekst, który chcemy wyświetlić w naszym programie. Trzeba pamiętać podczas redakcji tekstu o wstawieniu na początku każdej linii dowolnego znaku np. ilustruje to przykład:

```
-----  
: TEST PROGRAMU :  
-----
```

a-p-h-c-z-z-o-s-y-k-z

Jedynym ograniczeniem jest to, że nie możesz użyć w tekście przecinków i średników. Długość tekstu w zależności od potrzeb i możliwości użytego edytora tekstu. W zależności od trybu, w jakim będzie wyświetlany tekst ilość znaków w wierszu ustalony na 40 trybu 1 i 80 dla trybu 2. Jeżeli wykorzystujesz do prezentacji tekstu cały ekran.

Uwaga!
pominięcie znaku na początku linii spowoduje utratę porządku na ekranie — spróbuj, a przekonasz się sam.

Poniżej zamieszczono listing podprogramu wyświetlenia tekstu:

```
70 '####' PODPROGRAM WYSWIETLANIA TEKSTU ###  
80 MODE 2  
90 PEN 1  
100 OPENIN a$  
110 WHILE NOT EOF  
120 LL=LL+1  
130 INPUT#9,zx$  
140 LOCATE 1,LL  
150 PRINT MID$(zx$,2,LEN(zx$))  
160 IF LL<=24 THEN WEND:GOTO 230  
170 WINDOW#1,1,80,25:PEN#1,0:PAPER#1,1:CLSE#1  
180 LOCATE#1,15,1  
190 PRINT#1,"KONTYNUACJA TEKSTU - NACISNIJ KLAWISZ <ENTER>"  
200 IF INKEY(6)=-1 THEN 200  
210 LL=0:CLSE#1:CLS  
220 GOTO 110  
230 CLOSEIN  
240 WINDOW#1,1,80,25:PEN#1,0:PAPER#1,1:CLSE#1  
250 LOCATE#1,15,1  
260 PRINT#1,"POWRÓT DO POCZĄTKU POMOCY - <P> ## POWRÓT DO PROGRAMU - <ENTER>"  
270 CLEAR INPUT  
280 IF INKEY(6)=0 THEN LL=0:CLEAR INPUT: RETURN  
290 IF INKEY(27)=0 THEN LL=0:CLS:GOTO 100  
300 GOTO 280
```

Jak widać jest on przeznaczony do wyświetlania w trybie 2 tekstu zawartego w uprzednio założonym zbiorze o nazwie a\$. Ponieważ ekran w tym trybie pracy ma 25 linii, podprogram jednorazowo powoduje wyświetlenie 24 linii tekstu — w 25 linii zdefiniowano okno 1, przeznaczone na komunikaty. Jeśli tekst jest dłuższy niż 24 linie w oknie 1 pojawi się komunikat:

KONTYNUACJA TEKSTU — NACIŚNIJ KLAWISZ ENTER
W wypadku wyświetlenia na ekranie całej zawartości zbioru o nazwie umieszczonej w a\$ w oknie 1 pojawi się komunikat:

POWRÓT DO POCZĄTKU TEKSTU — P POWRÓT DO PROGRAMU ENTER

Zmiana trybu na 1 (mode 1) pociąga za sobą odpowiednią redakcję tekstu (40 znaków e linii), i zmianę definicji okna 1 na komunikaty. Przykład wywołania tego podprogramu przedstawiono poniżej:

```
10 ' ### SEGMENT STERUJACY ###  
20 INPUT a$  
30 GOSUB 310  
40 GOSUB 70  
50 END  
60 STOP
```

Realizuje on podstawienie pod nazwę a\$ nazwy zbioru, wywołanie podprogramu wyświetlania.

```
310 '###' DEFINICJA POLSKICH LITER ###  
320 SYMBOL AFTER 141  
330 SYMBOL 166,254,134,12,126,48,98,254,0  
340 SYMBOL 224,24,0,126,78,24,48,126,0  
350 SYMBOL 152,240,102,108,120,98,102,126,0  
360 SYMBOL 184,56,26,28,24,56,24,60,0  
370 SYMBOL 173,0,0,120,12,124,204,118,6  
380 SYMBOL 177,0,0,60,102,126,96,60,14  
390 SYMBOL 187,6,12,60,126,102,102,60,0  
400 SYMBOL 175,12,24,60,102,96,102,60,0  
410 SYMBOL 186,6,12,220,102,102,102,102,0  
420 SYMBOL 222,12,24,126,76,24,48,126,0  
430 SYMBOL 191,12,24,60,96,60,6,124,0  
440 RETURN
```

Powyższy podprogram definiuje polskie litery. Kody polskich liter muszą być zgodne z kodami używanymi w edytorze tekstu, za pomocą którego redagowany był tekst. Jest to warunek, aby w wyświetlanym tekście nie zabrakło a, e, ż itd.

Kończąc mam nadzieję, że zaprezentowany wyżej program zainteresuje Czytelników i zastosują go w swoich programach, uatrakcyjniając go i przyczyniając się między innymi do ułatwienia wymiany programów.

Zdzisław ŁUKASZEWICZ



— A ile ty tato pallsz na 100 km.

Stroiciel gitary

Przedstawiony niżej krótki program generuje dźwięki umożliwiające strojenie gitary. Dla zainteresowanych reszta wyjaśnień w programie.

```

10 PRINT 'STROICIEL GITARY':PRINT
20 FOR I=1 TO 6:READ L(I), H(I):NEXT
30 R=54272:FOR I=R TO R+23:POKE I,0:NEXT

40 POKE R+5,102:POKE R+6,102:POKE R+24,1
5
50 PRINT 'NACISNIJ:':PRINT
60 PRINT 'DOWOLNY KLAWISZ ABY WLACZYC LU
B WYLACZYC DZWIEK':PRINT
    
```

```

70 PRINT 'SPACJE ABY UZSKAC DZWIEK
NASTEPNEJ STRUNY':PRINT
80 PRINT 'DOWOLNY KLAWISZ FUNKCYJNY
ABY ZAKONCZYC PROGRAM'
90 GET A$:IF A$=' ' THEN 90
100 POKE R+4,0
110 IF A$=CHR$(32) THEN I=I+1:GOTO 140
120 IF A$>CHR$(132) AND A$<CHR$(141) THEN
POKE R+24,0:END
130 IF X THEN X=0:GOTO 90
140 IF I>6 THEN I=1
150 POKE R,L(I):POKE R+1,H(I)
160 POKE R+4,33:X=I:GOTO 90
170 DATA 71,5,12,7,104,9,142,12,210,15,3
0,21
READY.
    
```

Opracował: Tadeusz CISEK

W szponach ATARI (18)

Okna — ciąg dalszy

Tomasz MROWIEC Ludwik PIELA

W poprzednim odcinku cyklu („IKS” nr 7—8/1988) przedstawiliśmy procedury umożliwiające tworzenie okien, w trybach tekstowych i graficznych, we własnych programach napisanych w języku BASIC. Mieliśmy pewne zastrzeżenia do procedur działających w trybach graficznych. Umieszczały one w oknach teksty kolorowe, ale tylko w trybach wielokolorowych. W trybach dwukolorowych występowały dodatkowe efekty uboczne. Między innymi mało czytelne tło, na jakim umieszczane były napisy oraz zbyt szerokie znaki.

Z tych powodów opracowaliśmy nową procedurę konwersji, przeznaczoną dla trybów dwukolorowych (tryby BASIC-u o numerach 4, 6, 8 i 14). Jej zadaniem jest przygotowanie reprezentacji bitowej znaków tekstu, w której każdy punkt elementarny ekranu odwzorowany jest w jednej pozycji binarnej (a nie dwóch, jak poprzednio). Procedura ta przechowywana jest w zmiennej tekstowej F\$ i wywołujemy ją następująco:

```
X=USR(ADR(F$), ADR(A$), ADR(KOM$), LEN(KOM$))
```

gdzie: F\$ — zmienna tekstowa przechowująca procedurę konwersji,

A\$ — pomocnicza zmienna tekstowa służąca do przechowywania reprezentacji bitowej

znaków tekstu; musi pomieścić 8 razy więcej znaków niż jest w wyświetlanym tekście,

KOM\$ — zmienna tekstowa przechowująca wyświetlany tekst.

Wyświetlenie okna na ekranie realizuje procedura przechowywana w zmiennej W\$. Sposób jej wywołania jest następujący:

```
X=USR(ADR(W$), ADR(A$), O, W, S, L)
```

gdzie: O — wielkość przesunięcia lewego górnego rogu okna względem lewego górnego rogu ekranu (w bajtach),

W — wysokość okna (w liniach obrazu),

S — szerokość linii w bajtach (liczba znaków w linii),

L — przesunięcie między początkiem każdej kolejnej linii, jego wielkość zależy od wybranego trybu graficznego.

W poprzednim odcinku nie napisaliśmy wyraźnie, w jaki sposób usunąć okno z ekranu. Ze względu na zasadę działania procedury tworzenia okna, polegającą na wymianie zawartości ciągu z wybranym obszarem pamięci ekranu, zamknięcie okna uzyskamy poprzez ponowne użycie procedury tworzenia okna. Należy jednak pamiętać, aby parametry używane w wywołaniu były takie same.

Program nr 1 demonstruje wykorzystanie omawianego podprogramu. W zależności od

wybranego trybu graficznego, dwu- lub czterokolorowego, używana jest odpowiednia procedura konwersji. Bądź opisana powyżej, bądź z poprzedniego odcinka. Tak się składa, że tryby dwukolorowe są parzyste (4, 6, 8, 14) a czterokolorowe — nieparzyste (3, 5, 7, 15). Dzięki temu w prosty sposób możemy sprawdzić rodzaj trybu i wywołać odpowiednią procedurę.

Efektywne wykorzystanie opisanych procedur wymaga dokładnego przeanalizowania programu nr 1 oraz opisów zamieszczonych w poprzednim cyklu.

Prezentowane dotychczas procedury tworzenia okien na ekranie wymagały podania szeregu parametrów. Wśród nich był również adres zmiennej tekstowej zawierającej wyświetlaną w oknie informację. Jak pamiętamy, wymagało to używania procedury konwersji znaków tekstu z kodu ATASCII na kod ekranu. Możliwe jest również inne rozwiązanie. Procedura tworzenia okna przygotowuje tylko obszar ekranu, w którym za pomocą instrukcji PRINT możemy umieścić stosowny komunikat. A zatem wykonuje ona te same funkcje co poprzednio, za wyjątkiem wyprowadzania informacji na ekran. Oczywiście może być używana w taki sposób wyłącznie w trybie tekstowym GR.0.

Poprzednio prezentowane procedury włączane były do programu. Początkowo występowały w liniach DATA, a następnie umie-

szczone były w zmiennych tekstowych. Poniżej prezentujemy inne rozwiązanie, zaczerpnięte z czasopisma „Analog Computing” November 1986, w którym podprogramy otwierania i zamykania okien stanowią jakby rozszerzenie systemu operacyjnego. Umieszczone są w pamięci komputera poza obszarem przeznaczonym dla programu w języku BASIC. Odpowiednie dla nich miejsce uzyskujemy poprzez przesunięcie, w górę pamięci, dolnej granicy obszaru BASIC'u.

Rozwiązanie to wydaje się być niezbyt wygodnym, lecz jego zalety ujawniają się podczas pracy ze stacją dysków. Umieszczenie procedur w zbiorze AUTORUN.SYS zapewnia automatyczne wprowadzenie ich do pamięci po włączeniu komputera.

Ze względu na interesujące rozwiązanie oraz fakt, że nie każdy z czytelników dysponuje stacją dysków, opracowaliśmy również wersję taśmową tych procedur. Dlatego procedury otwierania i zamykania prezentujemy w dwóch wersjach: dyskowej i taśmowej.

Program nr 2 automatycznie tworzy na dyskietce zbiór AUTORUN.SYS i wprowadza do niego oba podprogramy. **Program nr 3** tworzy zbiór na taśmie magnetycznej, który może być wprowadzony do pamięci po włączeniu komputera z wciśniętym klawiszem (START).

Aby uzyskać odpowiednie zbiory należy wprowadzić do pamięci komputera, najlepiej za pomocą Edytora BASIC'a, program nr 2, jeśli dysponujemy stacją dysków, lub program nr 3, jeśli możemy korzystać tylko z magnetofonu. Radzimy przechowywać w pamięci zewnętrznej kopie wprowadzonych programów, aby uniknąć ponownego ich wprowadzania w wypadku wystąpienia jakiegokolwiek błędu. Przygotowujemy dyskietkę zawierającą DOS (bez zbioru AUTORUN.SYS) lub czystą taśmę magnetyczną i uruchamiamy program przechowywany w pamięci komputera. Po odliczeniu w dół od 640 do zera program nr 2 utworzy zbiór AUTORUN.SYS i zapisze do niego odpowiednie procedury, a program nr 3 zasygnalizuje chęć zapisania na taśmie magnetycznej. Zapisze je po włączeniu magnetofonu

na zapis i naciśnięciu dowolnego klawisza. Zbiór taśmowy jest typu BOOT, to znaczy samoladujący się.

W obu wypadkach załadowanie procedur wymaga uprzedniego wyłączenia komputera. W wypadku używania stacji dysków wchodzi one do pamięci RAM bezpośrednio po wprowadzeniu DOS-u (bez wyłączenia BASIC-a). W wypadku wprowadzania ich za pomocą magnetofonu należy wcześniej przewinąć taśmę i ustawić na początku zbioru, a następnie włączyć komputer z wciśniętym klawiszem (START).

Obie wersje, taśmowa i dyskowa, różnią się tylko sposobem wprowadzania do pamięci komputera. Ich działanie jest identyczne.

W celu otwarcia okna należy podać sześć parametrów. Pierwszym jest adres procedury otwarcia okna — 7459. Następne dwa określają lewy górny róg okna w postaci współrzędnych X, Y. Wartości X mogą zmieniać się od 0 do 39, a wartości Y od 0 do 23.

Czwarty i piąty parametr określają wielkość okna. Czwarty to szerokość, która może zmieniać się od 1 do 40 znaków. Piąty określa wysokość, od 1 do 24 znaków.

Ostatni parametr jest wskaźnikiem ramki. Jeśli podana jest wartość 0, to nie ma ramki. Wartość między 1 a 127 powoduje wykreślenie normalnej ramki. Wartość powyżej 127 — ramki w negacji.

Poniższe wywołanie otworzy okno o wielkości 10 na 5 znaków, bez ramki, w lewym, górnym rogu ekranu:

X=USR(7459, 0, 0, 10, 5, 0)

Zamykanie okna jest zawsze łatwiejsze. Przekazywany jest tylko jeden parametr, adres procedury zamknięcia okna.

X=USR(7826)

zamknie ostatnio otwarte okno.

Używanie okien jest bardzo proste. Trzeba jednak pamiętać o kilku sprawach. Jeśli otwieramy okno, które jest mniejsze niż 3x3 znaki, nie powinno ono mieć ramki. W przeciwnym wypadku wystąpią dziwne zjawiska. Jeśli

przechodzimy do DOS-u, procedury będą likwidowane, o ile nie użyjemy zbioru MEM.SAV. Podczas otwierania lub zamykania okna zalecane jest ustawianie kursora poza planowanym obszarem okna. Jeśli kursor jest w oknie podczas zamykania, istnieje szansa, że znak nie zostanie podmieniony. Jeśli kursor jest w pobliżu okna, które jest otwierane lub zamykane, możemy nie martwić się o niego. Wstawienie zera do komórki o adresie 8045 przekaże procedurom informację, że aktualnie żadne okno nie jest otwarte.

Podczas uruchamiania programu zalecane jest zerowanie licznika okien, zwłaszcza w wypadku, gdy jakieś okna pozostały otwarte w poprzednio działających programach.

Normalnie do kreślenia okna używane są negatywy spacji. Ten znak wypełniania może być zdefiniowany przez użytkownika. Wstawienie wybranej wartości do komórki 7625 zmieni znak używany do kreślenia tła okna.

Procedura otwierania okna wyznacza adres lewego górnego rogu na podstawie dostarczonych współrzędnych X i Y. Znaki schowane poza oknem przechowywane są na stosie, który rozpoczyna się od góry pamięci i rozrasta w dół, w kierunku programu w BASIC-u. Oprócz znaków na stosie umieszczane są również parametry. Podczas zamykania okna następuje proces odwrotny. Najpierw zdejmowane są ze stosu parametry. Następnie znaki, które zapamiętywane są w obszarze pamięci ekranu, odtwarzają pierwotny stan ekranu.

Podczas otwierania lub zamykania okna zwracaną wartością jest adres dołu stosu. Powinniśmy śledzić tę wartość, aby uniknąć konfliktu z naszym programem, którego górny adres przechowywany jest w komórkach 14 i 15 następująco:

GORPRO=PEEK(14)+256*PEEK(15)

Przed otwarciem okna powinniśmy sprawdzić, czy stos nie zejdzie poniżej górnej granicy programu.

Poprzez użycie prezentowanych procedur, ich wybór pozostawiamy czytelnikom, tworzone amatorsko programy mogą uzyskać profesjonalny wygląd.

```
FK 1 REM *****
FJ 2 REM *
XA 3 REM * Program nr 1 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
SH 100 GRAPHICS 0:POKE 752,1:POKE
82,4
AD 110 ? :? " OKNA TEKSTOW
E "
RL 120 ? :? " W TRYBACH GRAFI
CZNYCH"
KR 130 DIM W$(116),KOM$(11),ROB$(
22*16),E$(263),A$(11*16),F$(20
0)
BN 140 ? :? "WCZYTANIE W$"
ME 150 FOR I=1 TO 116:READ A:W$(I
,I)=CHR$(A):NEXT I
ND 160 ? :? "WCZYTANIE E$"
VL 170 FOR I=1 TO 263:READ A:E$(I
,I)=CHR$(A):NEXT I
OD 171 ? :? "WCZYTANIE F$"
XO 172 FOR I=1 TO 147:READ A:F$(I
,I)=CHR$(A):NEXT I
EB 175 G=15:W=40:GRAPHICS G+16:GO
SUB 260
CY 176 G=14:W=20:GRAPHICS G+16:GO
```

```
SUB 260
WR 180 G=8:W=40:GRAPHICS G+16:GOS
UB 260
WM 190 G=7:W=40:GRAPHICS G+16:GOS
UB 260
US 200 G=6:W=20:GRAPHICS G+16:GOS
UB 260
UN 210 G=5:W=20:GRAPHICS G+16:GOS
UB 260
TX 220 G=4:W=10:GRAPHICS G+16:GOS
UB 260
TS 230 G=3:W=10:GRAPHICS G+16:GOS
UB 260
TW 240 GRAPHICS 0:POKE 82,2: ? :?
"KONIEC PROGRAMU":END
NL 250 REM PODPROGRAM WYSWIETLANI
A KOMUNIKATOW
LI 260 SETCOLOR 0,9,10:SETCOLOR 1
,3,6:SETCOLOR 2,9,0:SETCOLOR 4
,1,2:C=(INT(G/2)=(G/2))
II 270 REM ZEROWANIE CIAGU
UZ 275 COLOR 1:PLOT W*3,0:FOR I=1
TO 40:X=RND(0)*W*4:Y=RND(0)*W
*2:DRAWTO INT(X),INT(Y):NEXT I
VW 280 ROB$(2)=ROB$:KOM$(2)=ROB$
OB 290 FOR L=1 TO 3
```

```
IZ 300 REM TWORZENIE OKNA
LX 305 A$=CHR$(0):A$(11*16)=A$:A$
(2)=A$
MP 310 KOM$="Gr. ":KOM$(6-LEN(STR
$(G)))=STR$(G)
HS 315 IF C=0 THEN ASM=USR(ADR(E$
),ADR(A$),ADR(KOM$),LEN(KOM$),
L)
ML 317 IF C=1 THEN ASM=USR(ADR(F$
),ADR(A$),ADR(KOM$),LEN(KOM$))
KM 320 ROB$=A$
YW 325 IF C=0 THEN WSK=(LEN(KOM$)
-0)*16+1
IM 326 IF C=1 THEN WSK=(LEN(KOM$)
-0)*8+1
HT 330 KOM$="TYP ":KOM$(5)=STR$(L
)
MD 335 A$=CHR$(0):A$(11*16)=A$:A$
(2)=A$
HJ 340 IF C=0 THEN ASM=USR(ADR(E$
),ADR(A$),ADR(KOM$),LEN(KOM$),
L)
MC 342 IF C=1 THEN ASM=USR(ADR(F$
),ADR(A$),ADR(KOM$),LEN(KOM$))
XX 345 ROB$(WSK)=A$
XC 360 REM UMIESZCZANIE OKNA
OD 370 FOR I=1 TO 5
```

```

KQ 380 IF C=0 THEN ASM=USR(ADR(W$
),ADR(ROB$),W*8+W/2-5,16,LEN(K
OM$)*2,W)
AC 382 IF C=1 THEN ASM=USR(ADR(W$
),ADR(ROB$),W*8+W/2-5,16,LEN(K
OM$),W)
WK 385 FOR J=15 TO 0 STEP -15/100
AG 386 SOUND 0,L*75,14,J
HO 387 NEXT J
CE 390 NEXT I:NEXT L:RETURN
BV 400 REM PODPROGRAM UMIESZCZANI
A OKNA
OG 410 DATA 104,104,133,206,104,1
33,205,104,141,1,6,104,141,0,6
NZ 430 DATA 104,133,208,104,133,2
07,104,104,133,209,104,141,3,6
,104,141,2,6
AT 450 DATA 165,88,133,203,165,89
,133,204,24,165,203,109,0,6,13
3
JA 470 DATA 203,165,204,109,1,6,1
33,204,160,0,177,203,170,177,2
05,145
OC 490 DATA 203,138,145,205,200,1
96,209,208,241,169,0,198,207,1
97,207,208
MU 510 DATA 6,197,208,240,30,198,
208,24,152,101,205,133,205,144
,3,230
XF 530 DATA 206,24,165,203,109,2,
6,133,203,165,204,109,3,6,133,
204,24,144,197,96
CR 560 REM PODPROGRAM KONWERSJI Z
NAKOW NA FORMAT CZTEROKOLOROWY
BK 570 DATA 104,104,133,204,104,1
33,203,104,133,212,104,133,211
,104,104,133
MK 590 DATA 216,10,133,217,104,10
4,41,3,24,106,106,106,41,192,1
33,207
XX 610 DATA 133,208,169,0,133,219
,169,4,133,220,169,0,168,145,2
19,200
GL 630 DATA 192,16,208,249,169,25
5,133,215,160,0,177,211,201,12
8,144,4
TV 650 DATA 132,215,73,128,201,96
,144,4,160,3,176,19,201,64,144
,4
RS 670 DATA 160,1,176,11,201,32,1
44,5,176,5,24,144,205,160,2,41
JA 690 DATA 31,10,10,10,133,213,2
4,152,109,244,2,133,214,160,0,
132
RU 710 DATA 209,132,210,162,0,134
,205,164,210,177,213,133,206,1
64,209,165
BJ 730 DATA 206,48,6,177,219,5,20
7,145,219,24,102,207,24,102,20
7,6
IZ 750 DATA 206,232,224,4,208,231
,165,215,240,5,56,241,219,145,
219,165
HE 770 DATA 208,133,207,230,209,2
30,205,165,205,201,2,208,208,2
30,210,164
UG 790 DATA 210,192,8,208,190,240
,3,24,144,160,165,203,72,165,2
04,72
EN 810 DATA 160,0,177,219,145,203
,200,192,2,208,247,177,219,170
,200,177
XA 830 DATA 219,72,200,132,218,16
4,217,138,145,203,104,200,145,
203,24,165
LA 850 DATA 203,101,217,133,203,1
44,2,230,204,164,218,192,16,20
8,220,104
JB 870 DATA 133,204,24,104,105,2,
133,203,144,2,230,204,230,211,
208,2
KP 890 DATA 230,212,198,216,208,1
77,96
WE 999 REM PODPROGRAM KONWERSJI Z
NAKOW NA FORMAT DWUKOLOROWY
ZN 1000 DATA 104,104,133,206,104,
133,205,104,133,204,104,133,20
3,104,104,133,207,133,208,160,
0,177,203,41,127
CK 1010 DATA 168,42,42,42,42,41,3
,170,152,41,159,29,73,251
HN 1020 DATA 133,212,169,0,162,3,
24,10,6,212,144,2,105,0,202,20
8,245,24,109,244,2,133,213
OM 1030 DATA 165,205,72,165,206,7
2,162,0,160,0,169,255,133,209,
177,203,41,128,208,4,169,0,133
,209,177,212,69,209
UR 1035 DATA 145,205,232,224,8,24
0,23
QA 1040 DATA 165,212,24,105,1,133
,212,144,2,230,213,165,208,24,
101,205,133,205,144,2,230,206,
24,144,204
UH 1050 DATA 104,133,206,104,24,1
05,1,133,205,144,2,230,206,24,
230,203,208,2,230,204,198,207,
208,129,96,-1
FK 1 REM *****
FJ 2 REM * *
XU 3 REM * Program nr 2 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
HY 7 REM Wersja dyskowa
EY 8 REM procedur
NO 9 REM
CV 10 DIM A$(64):GRAPHICS 0:POKE
752,1
ZY 15 POSITION 12,10:"Prosze cz
ekac..."
RM 20 FOR I=1 TO 641
EB 25 POSITION 19,14:"641-I,"
"
QF 30 READ A:IF A=-1 THEN 70
YY 40 A$(I,I)=CHR$(A)
IU 60 NEXT I
ZN 70 IF A<>-1 THEN ? "BLAD W DAN
YCH":END
BM 80 OPEN #1,8,0,"D:AUTORUN.SYS"
VV 90 POKE 766,1:"#1:A$":POKE 76
6,0
KX 100 CLOSE #1
NS 110 END
BZ 1000 DATA 255,255,0,29,111,31
NP 1005 DATA 165,12,141,14,29,165
,13,141,15,29
GW 1010 DATA 76,16,29,32,255,255,
169,0,141,231,2,169,32,141,232
,2
SU 1020 DATA 169,13,133,12,169,29
,133,13,96,104,104,104,141,100
,31,104
IU 1030 DATA 104,141,99,31,104,10
4,141,101,31,141,103,31,104,10
4,141,102
VV 1040 DATA 31,141,104,31,104,10
4,141,98,31,41,128,141,106,31,
173,99
DD 1050 DATA 31,133,203,169,0,133
,204,24,6,203,38,204,24,6,203,
38
AG 1060 DATA 204,24,6,203,38,204,
165,203,141,96,31,165,204,141,
97,31
XR 1070 DATA 24,6,203,38,204,24,6
,203,38,204,24,165,203,109,96,
31
OH 1080 DATA 133,203,165,204,109,
97,31,133,204,24,173,100,31,10
1,203,133
GI 1090 DATA 203,165,204,105,0,13
3,204,24,165,203,101,88,133,20
3,141,107
YP 1100 DATA 31,165,204,101,89,13
3,204,141,108,31,56,169,40,237
,101,31
RA 1110 DATA 141,105,31,173,109,3
1,208,13,173,229,2,133,205,173
,230,2
XK 1120 DATA 133,206,32,51,31,238
,109,31,160,0,177,203,145,205,
169,128
OZ 1130 DATA 145,203,32,9,31,32,5
1,31,206,103,31,173,103,31,208
,232
UE 1140 DATA 206,104,31,173,104,3
1,240,23,24,165,203,109,105,31
,133,203
ZK 1150 DATA 165,204,105,0,133,20
4,173,101,31,141,103,31,76,194
,29,32
SW 1160 DATA 37,31,165,203,145,20
5,32,51,31,165,204,145,205,32,
51,31
FB 1170 DATA 173,105,31,145,205,3
2,51,31,173,102,31,145,205,32,
51,31
TL 1180 DATA 173,101,31,145,205,3
2,51,31,173,98,31,208,3,76,0,3
1
VP 1190 DATA 174,101,31,202,202,1
42,110,31,172,102,31,136,136,1
40,111,31
QI 1200 DATA 173,107,31,133,203,1
73,108,31,133,204,169,81,13,10
6,31,160
ZA 1210 DATA 0,145,203,32,65,31,1
72,101,31,136,169,69,13,106,31
,145
XX 1220 DATA 203,32,82,31,169,124
,13,106,31,160,0,145,203,172,1
01,31
YK 1230 DATA 136,145,203,32,82,31
,206,111,31,173,111,31,208,230
,169,90
CA 1240 DATA 13,106,31,160,0,145,
203,32,65,31,169,67,13,106,31,
172
YJ 1250 DATA 101,31,136,145,203,7
6,0,31,104,173,109,31,208,3,76
,0
QU 1260 DATA 31,32,23,31,206,109,
31,160,0,177,205,141,101,31,14
1,103
PY 1270 DATA 31,32,23,31,177,205,
141,102,31,141,104,31,32,23,31
,177
TD 1280 DATA 205,141,105,31,32,23
,31,177,205,133,204,32,23,31,1
77,205
WV 1290 DATA 133,203,32,23,31,177
,205,145,203,32,23,31,32,37,31
,206
HV 1300 DATA 103,31,173,103,31,20
8,238,173,101,31,141,103,31,56
,165,203
ST 1310 DATA 237,105,31,133,203,1
65,204,233,0,133,204,206,104,3
1,173,104
FK 1320 DATA 31,208,210,32,51,31,
165,205,133,212,165,206,133,21
3,96,24
VP 1330 DATA 165,203,105,1,133,20
3,165,204,105,0,133,204,96,24,
165,205
IB 1340 DATA 105,1,133,205,165,20
6,105,0,133,206,96,56,165,203,
233,1
XP 1350 DATA 133,203,165,204,233,
0,133,204,96,56,165,205,233,1,
133,205
LA 1360 DATA 165,206,233,0,133,20
6,96,160,1,174,110,31,169,82,1
3,106
QP 1370 DATA 31,145,203,200,202,2
08,250,96,24,165,203,105,40,13
3,203,165
QE 1380 DATA 204,105,0,133,204,96
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
EI 1390 DATA 0,0,0,0,0,0,255,255,
224,2,227,2,0,29,34,29,-1
FK 1 REM *****
FJ 2 REM * *
YO 3 REM * Program nr 3 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
EA 7 REM Wersja tasmowa
EY 8 REM procedur
NO 9 REM
CK 10 DIM A$(622):GRAPHICS 0:POKE
752,1

```

```

ZY 15 POSITION 12,10:?"Prosze cz
ekac..."
QX 20 FOR I=1 TO 622
DG 25 POSITION 19,14:?"622-I;"
"
QF 30 READ A:IF A=-1 THEN 70
YY 40 A$(I,I)=CHR$(A)
IU 60 NEXT I
ZN 70 IF A<>-1 THEN ? "BLAD W DAN
YCH":END
CI 80 OPEN #1,8,128,"C:"
VV 90 POKE 766,1:?"#1:A$":POKE 76
6,0
KX 100 CLOSE #1
NS 110 END
DL 1000 DATA 0,5,3,29,9,29,169,60
,141,2,211,169,0
MY 1005 DATA 141,231,2,169,32,141
,232,2,169,34
CK 1010 DATA 133,10,169,29,133,11
,24,96,96
EO 1020 DATA 104,104,104,141,100,
31,104
IU 1030 DATA 104,141,99,31,104,10
4,141,101,31,141,103,31,104,10
4,141,102
VV 1040 DATA 31,141,104,31,104,10
4,141,98,31,41,128,141,106,31,
173,99
DD 1050 DATA 31,133,203,169,0,133
,204,24,6,203,38,204,24,6,203,
38
AG 1060 DATA 204,24,6,203,38,204,
165,203,141,96,31,165,204,141,
97,31
XR 1070 DATA 24,6,203,38,204,24,6
,203,38,204,24,165,203,109,96,
31
OH 1080 DATA 133,203,165,204,109,
97,31,133,204,24,173,100,31,10
1,203,133
GI 1090 DATA 203,165,204,105,0,13
3,204,24,165,203,101,88,133,20
3,141,107
YP 1100 DATA 31,165,204,101,89,13
3,204,141,108,31,56,169,40,237
,101,31
RA 1110 DATA 141,105,31,173,109,3
1,208,13,173,229,2,133,205,173
,230,2

XK 1120 DATA 133,206,32,51,31,238
,109,31,160,0,177,203,145,205,
169,128
OZ 1130 DATA 145,203,32,9,31,32,5
1,31,206,103,31,173,103,31,208
,232
UE 1140 DATA 206,104,31,173,104,3
1,240,23,24,165,203,109,105,31
,133,203
ZK 1150 DATA 165,204,105,0,133,20
4,173,101,31,141,103,31,76,194
,29,32
SW 1160 DATA 37,31,165,203,145,20
5,32,51,31,165,204,145,205,32,
51,31
FB 1170 DATA 173,105,31,145,205,3
2,51,31,173,102,31,145,205,32,
51,31
TL 1180 DATA 173,101,31,145,205,3
2,51,31,173,98,31,208,3,76,0,3
1
VP 1190 DATA 174,101,31,202,202,1
42,110,31,172,102,31,136,136,1
40,111,31
QI 1200 DATA 173,107,31,133,203,1
73,108,31,133,204,169,81,13,10
6,31,160
ZA 1210 DATA 0,145,203,32,65,31,1
72,101,31,136,169,69,13,106,31
,145
XX 1220 DATA 203,32,82,31,169,124
,13,106,31,160,0,145,203,172,1
01,31
YK 1230 DATA 136,145,203,32,82,31
,206,111,31,173,111,31,208,230
,169,90
CA 1240 DATA 13,106,31,160,0,145,
203,32,65,31,169,67,13,106,31,
172
YJ 1250 DATA 101,31,136,145,203,7
6,0,31,104,173,109,31,208,3,76
,0
QU 1260 DATA 31,32,23,31,206,109,
31,160,0,177,205,141,101,31,14
1,103
PY 1270 DATA 31,32,23,31,177,205,
141,102,31,141,104,31,32,23,31
,177
TD 1280 DATA 205,141,105,31,32,23
,31,177,205,133,204,32,23,31,1
77,205
WV 1290 DATA 133,203,32,23,31,177
,205,145,203,32,23,31,32,37,31
,206
HV 1300 DATA 103,31,173,103,31,20
8,238,173,101,31,141,103,31,56
,165,203
ST 1310 DATA 237,105,31,133,203,1
65,204,233,0,133,204,206,104,3
1,173,104
FK 1320 DATA 31,208,210,32,51,31,
165,205,133,212,165,206,133,21
3,96,24
VP 1330 DATA 165,203,105,1,133,20
3,165,204,105,0,133,204,96,24,
165,205
IB 1340 DATA 105,1,133,205,165,20
6,105,0,133,206,96,56,165,203,
233,1
XP 1350 DATA 133,203,165,204,233,
0,133,204,96,56,165,205,233,1,
133,205
LA 1360 DATA 165,206,233,0,133,20
6,96,160,1,174,110,31,169,82,1
3,106
QP 1370 DATA 31,145,203,200,202,2
08,250,96,24,165,203,105,40,13
3,203,165
QE 1380 DATA 204,105,0,133,204,96
,0,0,0,0,0,0,0,0,0,0
IR 1390 DATA 0,0,0,0,0,0,-1

FK 1 REM *****
FJ 2 REM * *
ZI 3 REM * Program nr 4 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
EX 100 REM Program demonstracyjny
WN 110 WOTW=7459:WZAM=7826
AG 120 WYP=7625: LICZ=8045
ZW 130 POKE LICZ,0:POKE WYP,128
BL 140 DIM TEKST$(38):POKE 712,PE
EK(710):POKE 752,1:?"CHR$(125)
":LIST 310,510
UU 150 REM TYTUL
NY 160 RESTORE :PX=2:PY=1: SX=36: S
Y=5:RAMKA=1:GOSUB 460:POSITION
PX+8, PY+2:GOSUB 500
WF 170 PX=2:PY=7: SX=36:SY=12:RAMK
A=1:GOSUB 460:POSITION PX+13, P
Y+1:GOSUB 500
GM 180 REM WLASNOSCI
FW 190 FOR X=3 TO 5:POSITION PX+3
, PY+X:GOSUB 500:NEXT X
FR 200 REM ROZNE WIELKOSCI
CY 210 POSITION PX+3, PY+6:GOSUB 5
00
UT 220 FOR X=3 TO 7:PX=X-2:PY=10+
X: SX=X:SY=X:RAMKA=128:GOSUB 46
0:PX=42-X-X:GOSUB 460:NEXT X
MB 230 FOR X=1 TO 400:NEXT X:FOR
X=1 TO 10:GOSUB 480:NEXT X
QC 240 REM WIELE KOLOROW
ZF 250 PX=2:PY=7:POSITION PX+3, PY
+7:GOSUB 500
CU 260 POKE WYP,0:PX=6:PY=16: SX=9
:SY=5:RAMKA=1:GOSUB 460
ZJ 270 POSITION PX+2, PY+2:GOSUB 5
00
RR 280 POKE WYP,128:PX=25:RAMKA=1
28:GOSUB 460
GC 290 POSITION PX+2, PY+2:GOSUB 5
00:FOR X=1 TO 400:NEXT X:GOSUB
480:GOSUB 480
BM 300 REM RAMKI
AE 310 PX=2:PY=7:POSITION PX+3, PY
+8:GOSUB 500
PK 320 PX=2:PY=20: SX=36:SY=3:RAMK
A=0:GOSUB 460:POSITION PX+5, PY
+1:GOSUB 500
QA 330 RAMKA=1:GOSUB 460:POSITION
PX+5, PY+1:GOSUB 500
CM 340 RAMKA=128:GOSUB 460:POSITI
ON PX+5, PY+1:GOSUB 500
ZL 350 FOR X=1 TO 3:GOSUB 480:FOR
Y=1 TO 100:NEXT Y:NEXT X
NG 360 REM ROZWIJANE MENU
BW 370 PX=2:PY=7:POSITION PX+3, PY
+9:GOSUB 500
QJ 380 FOR Z=2 TO 36 STEP 9
SG 390 SX=9:SY=1:RAMKA=0:FOR Y=19
TO 23
IR 400 PX=Z:PY=Y:GOSUB 460:NEXT Y
:POSITION PX,20:GOSUB 500:POSITI
ON PX+4,22:GOSUB 500
IN 410 FOR X=19 TO 23:GOSUB 480:N
EXT X:NEXT Z
OK 420 REM ZAMYKANIE WSZYSTKICH O
KIEN
KC 430 FOR X=1 TO 400:NEXT X
TJ 440 FOR Y=1 TO 2: X=USR(WZAM):F
OR X=1 TO 200:NEXT X:NEXT Y:GO
TO 160
IW 450 REM OTWIERANIE OKNA
FZ 460 WINDOW=USR(WOTW, PX, PY, SX, S
Y, RAMKA):RETURN
VX 470 REM ZAMYKANIE OKNA
FP 480 WINDOW=USR(WZAM):RETURN
WS 490 REM WYSWIETLANIE TEKSTU I
OCZEKIWANIE
HU 500 READ TEKST$:?"TEKST$:POSIT
ION 0,0:?"":FOR W=1 TO 300
:NEXT W:RETURN
MF 510 REM TEKST
KR 520 DATA OKNA Wersj
a i. l. Wlasnoscii, -Caly kod w j
azyku maszynowym
WY 530 DATA -Do 255 okien otwarty
ch naraz, -Odporne na RESET
, -Okna moga miec rozne rozmiar
y
DP 540 DATA -Okna moga byc wielok
olorowe, Czarne, Biale, -Ramki sa
opcjonalne, -Okno bez ramk
i
KZ 550 DATA Okno z normalna ramka
, Okno z ramka w nesacji
, -Rozwijane menu, Menu, 1
MY 560 DATA Menu, 2, Menu
, 3, Menu, 4

FK 1 REM *****
FJ 2 REM * *
AC 3 REM * Program nr 5 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
EX 100 REM Program demonstracyjny
FO 110 WOPEN=7459:WCLOSE=7826
TM 120 POKE 8045,0:POKE 752,1:TRA
P 120
PG 130 ? CHR$(125):?"ILE OKIEN UTW
ORZYC"
HG 140 INPUT N:N=INT(ABS(N))
FE 150 IF N=0 OR N>255 THEN 130
QJ 160 POKE 752,1:?"CHR$(125)":LI
ST
YN 170 FOR C=1 TO N:POSITION 0,0:
?"CHR$(30):CHR$(31):
GZ 180 SX=INT(RND(0)*20)+4
GO 190 SY=INT(RND(0)*10)+4
PY 200 PX=INT(RND(0)*(39-SX))+1
LO 210 PY=INT(RND(0)*(23-SY))+1
BS 220 Z=USR(WOPEN, PX, PY, SX, SY, 12
8)
YM 230 POSITION PX+1, PY+1:?"C
BQ 240 PROG=PEEK(14)+256*PEEK(15)
AG 250 IF PROG+256>Z THEN 270
DW 260 NEXT C
YO 270 FOR C=1 TO N:POSITION 0,0:
?"CHR$(30):CHR$(31):
VP 280 Z=USR(WCLOSE):NEXT C:GOTO
170

```

Tomasz MROWIEC,
Ludwik PIELA

Edytor BASIC'a

Ludwik PIELA

Otrzymujemy sporo listów i telefonów w sprawie programu „Edytor BASIC'a” zamieszczonego w numerze „IKS-a” z 1987 roku. Przyznajemy, jest w nim jeden niewidoczny znak (nie z naszej winy) oraz jeden znak trudny do interpretacji. Poza tym program nie zawiera błędów. Większość problemów z jego uruchomieniem wynika z niedokładnego (lub błędnego) wprowadzenia liczb zawartych w liniach DATA. Ze względu na to, a także na fakt, że w przeciągu minionego półtora roku przybyło posiadaczy komputerów ATARI (i mamy nadzieję naszych czytelników), ośmielamy się zamieścić, głównie dla nich, wydruk programu „Edytor BASIC'a”.

Program zachowuje wszystkie własności edycyjne interpretera, dopuszcza także stosowanie skrótów instrukcji. Umożliwia także sprawdzenie całego programu, jeśli został on wprowadzony w sposób konwencjonalny, z wykorzystaniem wbudowanego edytora.

Wprowadzenie edytora

W celu utworzenia własnej kopii Edytora BASIC'a należy dokładnie wykonać jedną z poniższych instrukcji.

Wersja dyskowa

- 1) Wprowadzić Wydruk nr 1 starannie i bez błędów.
- 2) Przechować program na dysku za pomocą SAVE „D:EDYTORW1.BAS”.
- 3) Oczyszczyć pamięć komputera rozkazem NEW.
- 4) Wprowadzić Wydruk nr 2.
- 5) Dla bezpieczeństwa przechować na dysku za pomocą „D:EDYTORW2.BAS”.
- 6) Uruchomić program instrukcją RUN i wykonać wyświetlone na ekranie polecenia. Zbiór danych o nazwie ML.DAT zapisany zostanie na dysku.
- 7) Załadować do pamięci pierwszy program rozkazem LOAD „D:EDYTORW1.BAS”.
- 8) Dołączyć zbiór utworzony przez program nr 2 instrukcją ENTER „D:ML.DAT”.
- 9) Przechować program wynikowy rozkazem LIST „D:EDYTOR.LST”.

Wersja taśmowa

- 1) Wprowadzić Wydruk nr 1.
- 2) Przechować program na kasecie za pomocą CSAVE (nie przewijać taśmy).

3) Oczyszczyć pamięć komputera rozkazem NEW.

- 4) Wprowadzić Wydruk nr 2.
- 5) Dla bezpieczeństwa przechować program na taśmie.

6) Uruchomić program rozkazem RUN i wykonać wyświetlone na ekranie polecenia. Zbiór zostanie zapisany na kasecie.

7) Przewinąć taśmę i wprowadzić pierwszy program rozkazem CLOAD.

8) Dołączyć zbiór utworzony przez program nr 2 rozkazem ENTER „C:”.

9) Na innej kasecie przechować program wynikowy rozkazem LIST „C:”.

Używanie Edytora

Spójrzmy na wydruki programów zamieszczonych w ramach cyklu „W szponach ATARI”. Zauważmy, że każda linia programu poprzedzona jest przez kod dwuliterowy. Jest to suma kontrolna tej linii, a nie część programu.

W celu wprowadzenia programu załaduj Edytor rozkazem ENTER i uruchom go. Zostaniesz zapytany, czy chcesz używać skrótów instrukcji. Jeśli tak, to wprowadź T i naciśnij RETURN. W przeciwnym razie podaj N. Ekran telewizora (monitora) zostanie podzielony na dwa „okna”. W górnym wyświetlana będzie każda linia po przetworzeniu przez Edytora oraz suma kontrolna tej linii. W dolnym oknie odbywa się wprowadzanie i poprawianie linii.

Gdy program oczekuje na wprowadzenie, kursor pojawia się przy lewym marginesie okna wprowadzania. Wprowadź linię programu i naciśnij RETURN. Nastąpi sprawdzenie jej poprawności oraz obliczenie sumy kontrolnej. Linia programu pojawi się w oknie komunikatu wraz z wyznaczoną sumą kontrolną. Jeśli wyświetlony kod zgadza się z zamieszczonym w wydruku, można przejść do wprowadzania następnej linii programu. W przeciwnym wypadku należy podać rozkaz E (Edit) i naciśnij RETURN. Ostatnio wprowadzona linia pojawi się w dolnym oknie, gdzie może zostać poprawiona. Po wprowadzeniu poprawek naciśnij RETURN.

Można wywołać dowolną linię programu, wcześniej wprowadzoną, którą chcemy poprawić, za pomocą rozkazu E z numerem linii. Na przykład, E230 wyświetli linię 230 w oknie wprowadzania.

Jeśli używamy skrótów instrukcji, obie wersje rozkazu Edit działają różnie. Rozkaz E

bez numeru linii wyświetli ją w takiej postaci, w jakiej została wprowadzona. Po dodaniu numeru będzie ona wyświetlona w postaci rozwiniętej (bez skrótów).

W dowolnej chwili można wyjść z Edytora poprzez naciśnięcie klawisza B (Basic). Można teraz przejrzeć wykonaną pracę. Zauważmy, że linie o numerach 32600 i wyżej, należą do Edytora. Nasz program ma niższe numery. Aby powrócić do Edytora, należy podać GOTO 32600.

W czasie wprowadzania dłuższych programów możemy mieć ochotę na robienie przerw, oczywiście po uprzednim zabezpieczeniu naszej pracy. W tym celu wprowadzamy L i RETURN, a następnie przechowujemy program w wybranej pamięci zewnętrznej w postaci umożliwiającej późniejsze łączenie.

Jeśli ktoś nie chce używać Edytora BASIC'a, preferując wbudowany edytor Atari, także może sprawdzić i ewentualnie poprawić wprowadzony już i przechowany w pamięci zewnętrznej program. W tym celu należy załadować go do pamięci, następnie dołączyć Edytor BASIC'a rozkazem ENTER i uruchomić go (GOTO 32600). Na propozycję używania skrótów musimy odpowiedzieć N. Gdy Edytor pojawi się na ekranie, należy podać rozkaz P i pierwsza linia programu pojawi się w oknie wprowadzania. W celu jej przetworzenia należy nacisnąć RETURN. Linia programu, wraz z sumą kontrolną, pojawi się w górnym oknie. Jeżeli suma kontrolna zgadza się z podaną w wydruku, należy dwa razy nacisnąć RETURN i przetwarzana będzie następna linia programu.

Po stwierdzeniu, że musimy poprawić aktualną linię, podajemy rozkaz E i zostanie ona przesunięta do okna wprowadzania.

Może się zdarzyć, że po wprowadzeniu programu za pomocą Edytora BASIC'a nie działa on właściwie. Możliwe są dwie przyczyny takiej sytuacji. Po pierwsze, mogło się zdarzyć, że nie wykonaliśmy dokładnie instrukcji wprowadzania programu. Dlatego zawsze należy wcześniej przeczytać artykuł dotyczący programu. Pod drugie, chociaż możemy zaufać Edytorowi, że unikniemy błędów wprowadzania linii programu, nie może on nam powiedzieć, czy nie opuściliśmy pewnych linii. Jeśli program nie chce działać, należy upewnić się, czy wszystkie linie zostały wprowadzone.

Edytor BASIC'a zgodny jest z podobnymi programami używanymi aktualnie przez „ANALOG Computing” i „ANTIC”. Opracowany został na podstawie następujących programów: BASIC Editor, BASIC Editor II i Typo II. Dzięki temu można go wykorzystać do wprowadzania programów zamieszczonych w tych czasopismach, a także w „Bajtku”.

```

32595 REM Edytor BASIC'u
32597 REM Wydruk nr 1
32598 REM
32600 IF FL THEN 32616
32602 DIM L$(115),SV$(115),C2$(2),B$(115),M$(119),S$(98),E$(69),A$(1):FL=1:STMTAB=PEEK(136)+PEEK(137)*256
32604 GRAPHICS 0:POKE 710,0:P=0:ABR=0:"CZY CHCESZ UZYWAC S KROTOW INSTRUKCJI";
32605 INPUT A$:IF A$="T" OR A$="t" THEN ABR=1
32606 B$(1)="":B$(115)="":B$(2)=B$
32616 OPEN #17,4,0,"E":L$=""
GOSUB 32662:START=0
32618 POKE 766,1:POKE 83,39:POSITION 1,3:IF LEN(L$)<39 THEN ? L$:GOTO 32624
32620 IF LEN(L$)<77 THEN ? L$(1,38):? L$(39,LEN(L$)):GOTO 32624
32622 ? L$(1,38):? L$(39,76):? L$(77,LEN(L$))
32624 POKE 752,0:POKE 766,0:POKE 559,34:POKE 82,1:POKE 83,38:POSITION 0,10:"":INPUT #17:L$:POKE 766,1
32626 IF (L$="P" OR L$="p") AND START=0 THEN P=1:L$=""
32628 IF L$="E" OR L$="e" THEN E=1:POSITION 1,10:SV$:GOTO 32624
32630 IF L$="L" OR L$="l" THEN GOSUB 32704
32632 IF L$="" AND P=1 THEN 32686
32634 IF L$="" THEN 32624
32636 IF L$="B" OR L$="b" THEN GRAPHICS 0:"WPROWADZ 'GOTO 32600' W CELU KONTYNUACJI":END
32638 IF L$(1,1)="E" OR L$(1,1)="e" THEN E=1:TRAP 32624:EL=VAL(L$(2)):POSITION 1,9:LIST EL:GOTO 32624
32640 SV$=L$:TRAP 32624:X=VAL(L$)
32642 START=1:IF P AND NOT E THEN 32652
32644 GOSUB 32674:IF NOT ABR OR P THEN 32652
32646 POKE 766,0:CHR$(125):POSITION 0,3:L=VAL(L$):LIST L:?"?:?"CONT":L$=B$
32648 POSITION 0,0:POKE 842,13:STOP
32650 POKE 842,12:A=USR(ADR(S$),ADR(L$),4):L$=L$(1,A)
32652 CHKSUM=USR(ADR(M$),ADR(L$),LEN(L$)):CHKSUM=CHKSUM+PEEK(1542)*65536

```

```

32654 CHK=CHKSUM-(INT(CHKSUM/676)*676):HI=INT(CHK/26):LO=CHK-(HI*26):C2$(1)=CHR$(HI+65):C2$(2)=CHR$(LO+65)
32656 IF NOT P OR E THEN E=0:GOSUB 32662:IF NOT P THEN 32660
32658 POKE 83,39:POKE 752,1:FOR X=3 TO 5:POSITION 1,X:?"B$(1,38):POSITION 1,X+7:?"B$(1,38):NEXT X:POKE 83,38
32660 POKE 766,1:POKE 83,38:POSITION 7,7:?"C2$:POKE 752,0:GOTO 32618
32662 GOSUB 32702:POKE 766,0:POKE 752,1:CHR$(125):POKE 82,1:DL=PEEK(560)+256*PEEK(561)+4

```

```

32664 POKE DL-1,70:POKE DL+2,6:POKE DL+3,112:POKE DL+4,112:POKE DL+5,112:POKE DL+13,112:POKE DL+14,112
32666 POKE DL+22,112:POKE DL+23,112:POKE DL+24,65:POKE DL+25,PEEK(560):POKE DL+26,PEEK(561):POKE 83,39
32668 POSITION 20,0:?" edytor basic":POSITION 0,7:?" OKNO WPROWADZANIA
32670 POSITION 0,1:?" OKNO KOMUNIKATU":POSITION 1,7:?"KOD":
32672 POKE 559,34:RETURN
32674 GRAPHICS 0:POKE 559,0:POKE 766,1:POKE 82,0:POKE 83,39:POSITION 0,3:?"L$:?"?:?"CONT":POSITION 0,0
32676 POKE 842,13:STOP
32678 POKE 842,12:TRAP 32682:A=USR(ADR(E$),VAL(L$)):IF A=4 THEN POP:GOTO 32682
32680 RETURN
32682 GOSUB 32662:SOUND 0,75,10,8:FOR X=1 TO 20:NEXT X:SOUND 0,0,0,0:POSITION 1,3:?"BLAD SKLADNI!":POKE 766,1
32684 POKE 83,38:POSITION 1,10:?"SV$:GOTO 32624
32686 LINE=PEEK(STMTAB)+PEEK(STMTAB+1)*256:IF LINE>32599 THEN N 32590
32688 OFS=PEEK(STMTAB+2):STMTAB=STMTAB+OFS:POSITION 1,9:LIST LINE:GOTO 32624
32690 POSITION 1,10:?"Numer 1 inii nie moze byc wiekszy od 32599!":
32692 FOR I=1 TO 1000:NEXT I:GOTO 32624

```

```

32702 POKE 16,112:POKE 53774,12:RETURN
32704 POSITION 5,12:?"KASETA CZY DYSKIETKA":INPUT L$:IF L$<"K" AND L$<"D" THEN 32704
32706 IF L$="D" THEN 32714
32708 POSITION 1,12:?"Przygotuj kasete i naciśnij RETURN":GOSUB 32720
32710 LIST "C:",1,32599:GOTO 32718
32714 POSITION 0,12:?"NAZWA ZBIORU (D:FILENAME.EXT)":CHR$(30);CHR$(30);CHR$(30);INPUT L$
32715 TRAP 32716:OPEN #1,4,0,L$:CLOSE #1:GOTO 32722
32716 LIST L$,1,32599
32718 POSITION 0,12:?" <36 znakow spacji":RETURN
32720 GET #17,A:RETURN
32722 ? CHR$(125):?"ZBIOR JUZ I STNIEJE!":?"CZY WYKASOWAC GO":INPUT A$:TRAP 32722:IF A$="T" THEN 32716
32724 FOR X=12 TO 15:POSITION 0,X:?"B$:NEXT X:GOTO 32714

```

```

2 REM Edytor BASIC'u
3 REM Wydruk nr 2
4 REM
10 DIM L$(120),ML$(119),A$(1)
20 GRAPHICS 0:POKE 710,0:?"Dysk czy Kasete":INPUT A$:IF A$<"K" AND A$<"D" THEN 20
30 IF A$="K" THEN 50
40 ? "Umiesc sformatowana dysk

```

```

ietke w stacji dyskow,":?"n
astepnie naciśnij RETURN":INPUT L$
45 INPUT L$:OPEN #1,8,0,"D:ML.DAT":GOTO 60
50 ?:"Przygotuj kasete, naciśnij RETURN":INPUT L$:OPEN #1,8,0,"C:"
60 L$="32608 M$(1)":L$(13)=CHR$(34)
70 N=119:GOSUB 130:L$(14)=ML$(1,58):L$(LEN(L$)+1)=CHR$(34):?"#1:L$
80 L$(1)="32610 M$(59)":L$(14)=CHR$(34):L$(15)=ML$(59):L$(LEN(L$)+1)=CHR$(34):?"#1:L$
90 L$(1)="32612 S$":L$(10)=CHR$(34)
100 ML$=""N=98:GOSUB 130:L$(1)=ML$:L$(LEN(L$)+1)=CHR$(34):?"#1:L$
110 L$(1)="32614 E$":L$(10)=CHR$(34)
120 ML$=""N=69:GOSUB 130:L$(1)=ML$:L$(LEN(L$)+1)=CHR$(34):?"#1:L$:END
130 FOR X=1 TO N:READ A:ML$(X)=CHR$(A):NEXT X:RETURN
140 DATA 104,104,133,204,104,133,203,104,104,133,205,169,0,141,3,6,141,2,6,141,4,6,141,5,6,150 DATA 141,6,6,238,3,6,32,68,218,172,2,6,177,203,133,212,32,170,217,32,182,221,32,68,218,160 DATA 173,3,6,133,212,32,170,217,32,219,218,32,210,217,165,212,141,0,6,165,213,141,1,6,24,170 DATA 173,0,6,109,4,6,141,4,6,173,1,6,109,5,6,141,5,6,144,3,238,6,6,238,2,180 DATA 6,172,2,6,196,205,208,176,173,4,6,133,212,173,5,6,133,213,96,190 DATA 104,104,133,204,104,133,203,104,104,141,255,6,169,0,133,213,216,165,88,133,205,165,89,133,206,200 DATA 174,255,6,24,165,205,105,40,133,205,144,2,230,206,202,208,242,160,0,177,205,201,64,144,18,210 DATA 201,96,144,19,201,128,144,18,201,192,144,6,201,224,144,7,176,8,24,105,32,144,3,56,233,220 DATA 64,145,203,200,192,114,240,2,208,215,177,203,201,32,208,3,136,208,247,200,132,212,96,230 DATA 104,104,141,254,6,104,141,253,6,169,0,133,213,216,165,136,133,205,165,137,133,206,160,0,177,240 DATA 205,205,253,6,208,8,200,177,205,205,254,6,240,15,160,2,177,205,24,101,205,133,205,144,228,250 DATA 230,206,176,224,160,4,177,205,201,55,240,4,160,0,240,0,132,212,96

```

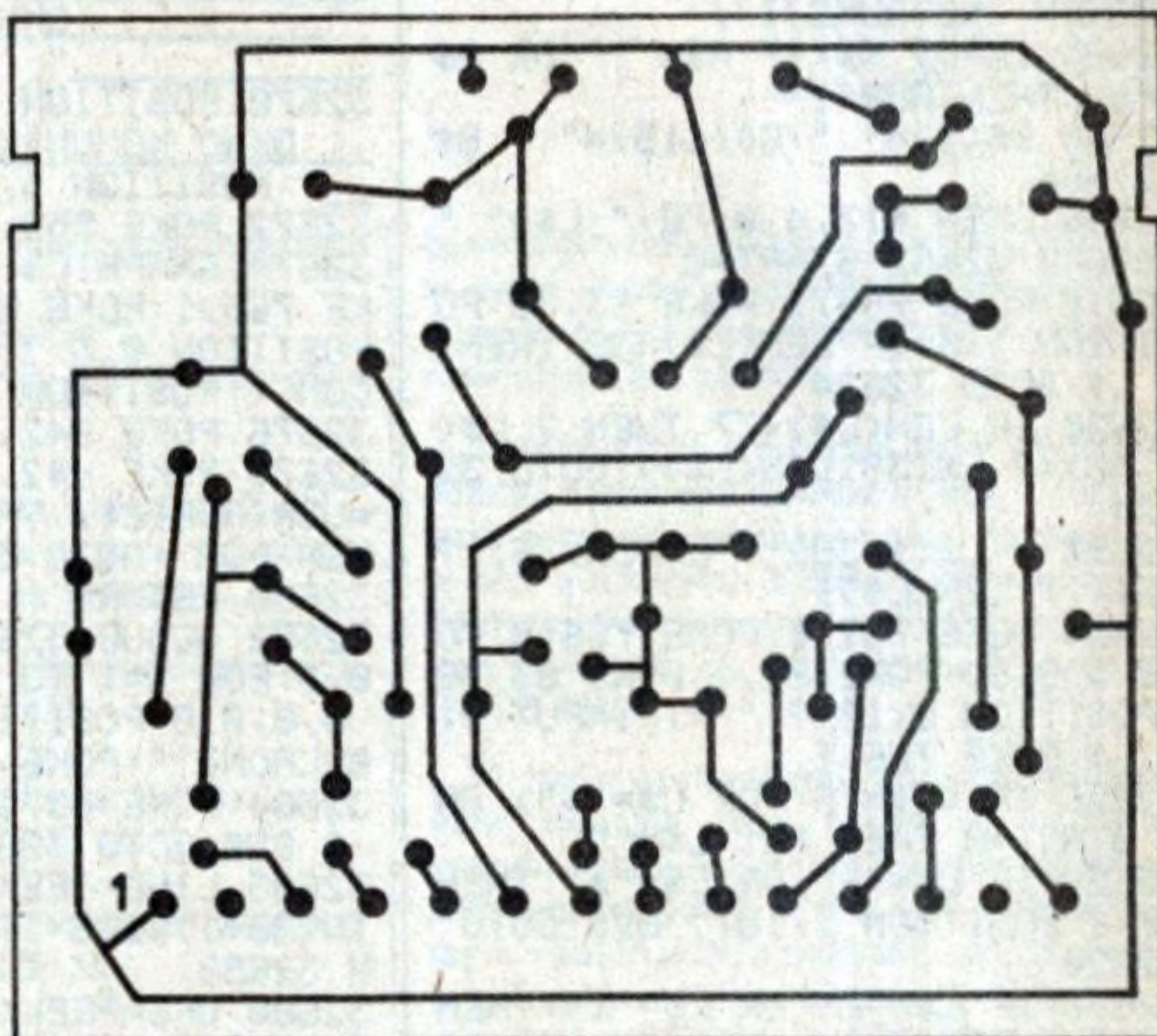
Wobec częstych pytań informujemy, że tylko autorzy dysponują kopiami (na taśmie magnetycznej) wszystkich programów zamieszczonych w ramach cyklu „W szponach Atari”. Zainteresowanych kopiami tych programów (i leniwych) prosimy o bezpośrednie skontaktowanie się z jednym z autorów.

Podajemy adres:

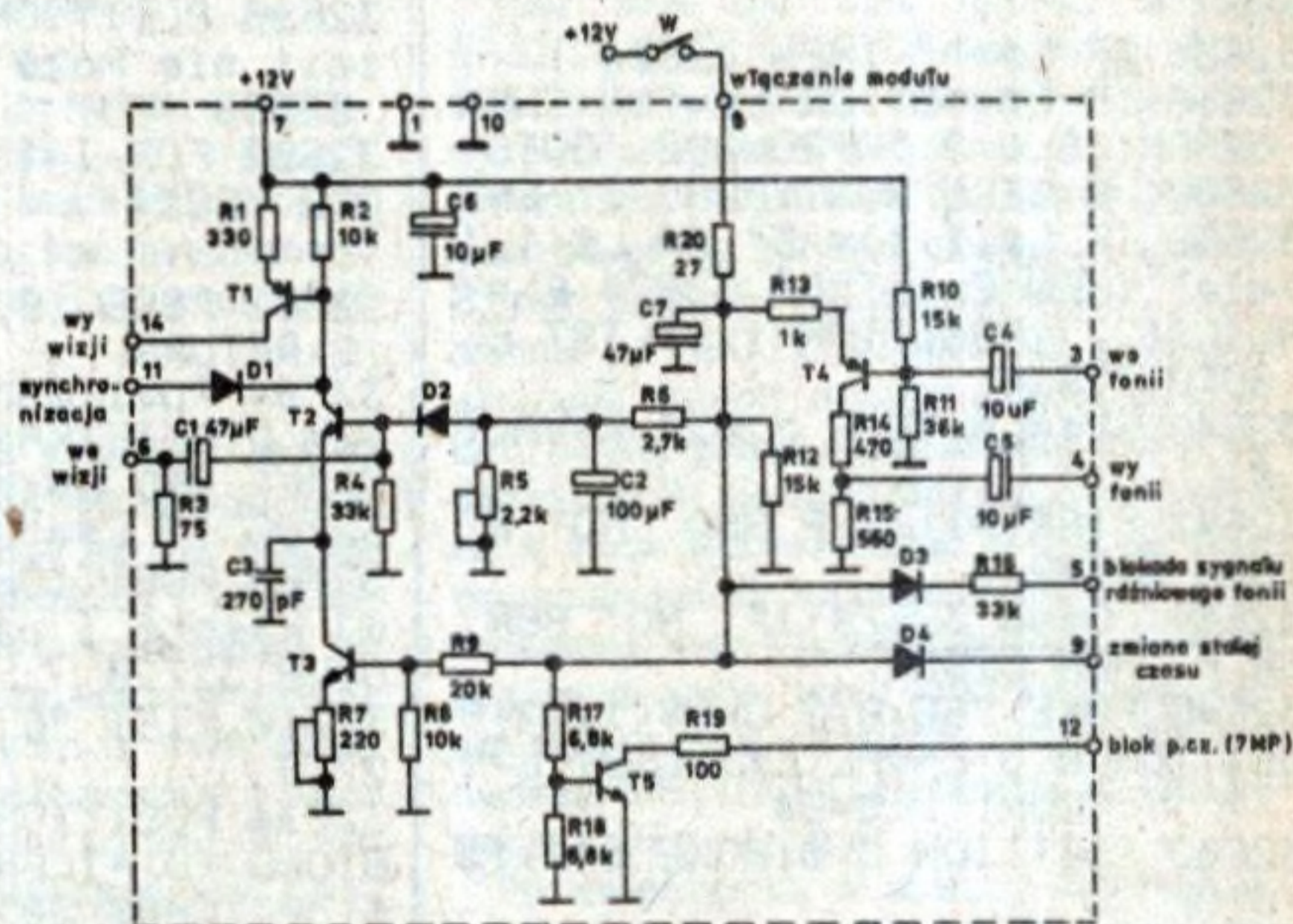
Ludwik PIELA
Ul. Topograficzna 7/2
00-910 WARSZAWA

Helios — monitorem

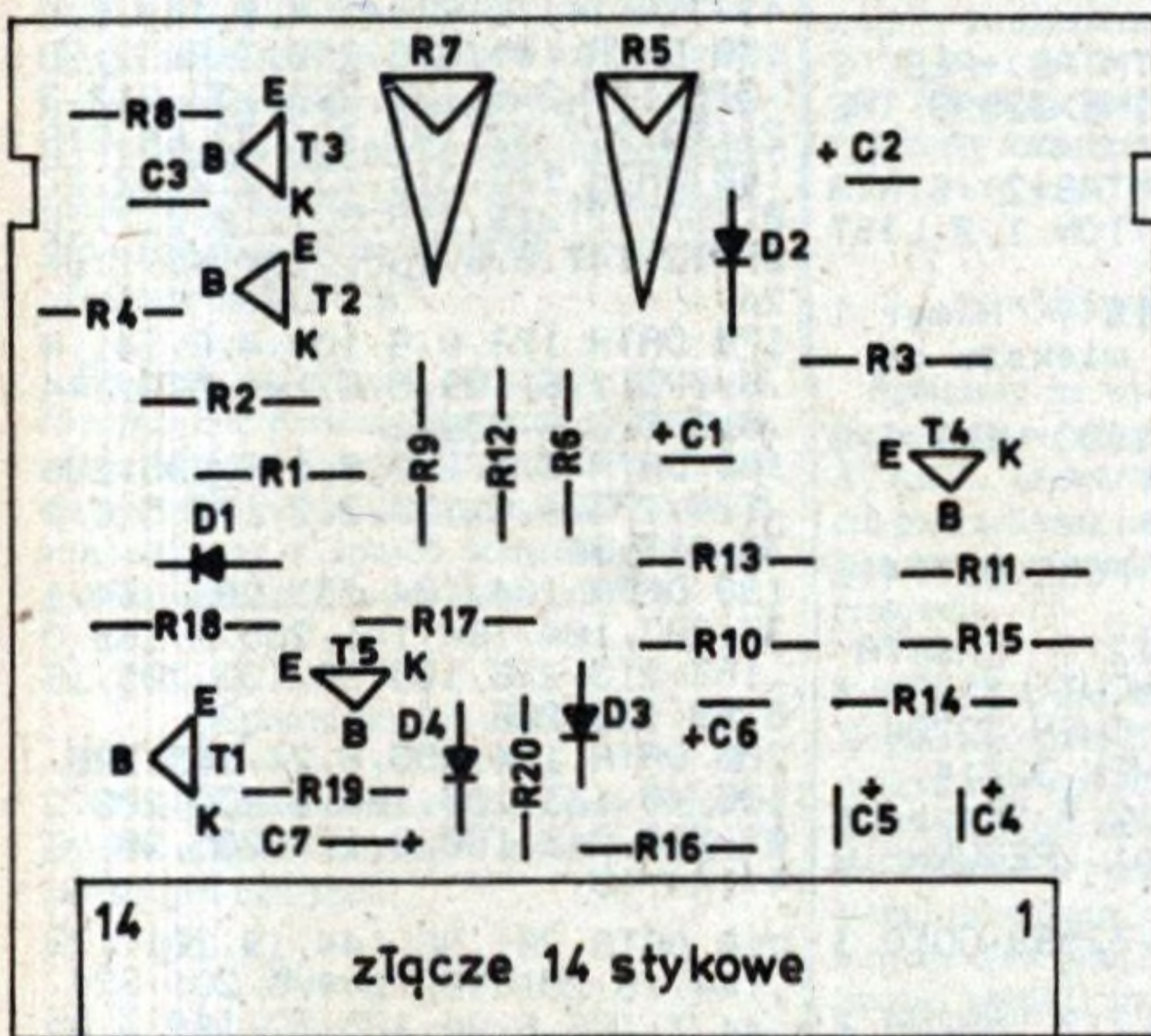
Wielu użytkowników mikrokomputerów korzysta z telewizora połączanego z komputerem poprzez wejście antenowe. Jakość uzyskiwanego tak obrazu nie jest zadowalająca i najczęściej nie ma dźwięku. Rezygnacja z toru wysokiej częstotliwości telewizora (i komputera) daje znaczną poprawę jakości obrazu oraz umożliwia uzyskanie fonii. Opisany w artykule układ — moduł monitora umożliwia wykorzystanie jako monitora telewizora kolorowego HELIOS TC 500. OTV HELIOS jest fabrycznie przygotowany do zainstalowania tego modułu i w związku z tym wymaga niewiele przeróbek. Należy w tym miejscu zaznaczyć, że układ ten nie zapewnia uzyskania kolorowego obrazu z komputera, jeżeli telewizor nie posiada dekodera PAL. Przy wyposażeniu tylko w dekodery SECAM obraz będzie czarno-biały. Moduł monitora umożliwia współpracę telewizora z dowolnym mikrokomputerem mającym wyjścia kompletnego sygnału wizji (ang. composite video) i fonii. W rozwiązaniu modelowym z Heliose współpracuje Atari 130XE. Możliwa jest również współpraca telewizora z magnetowidem.



rys. 2



rys. 3

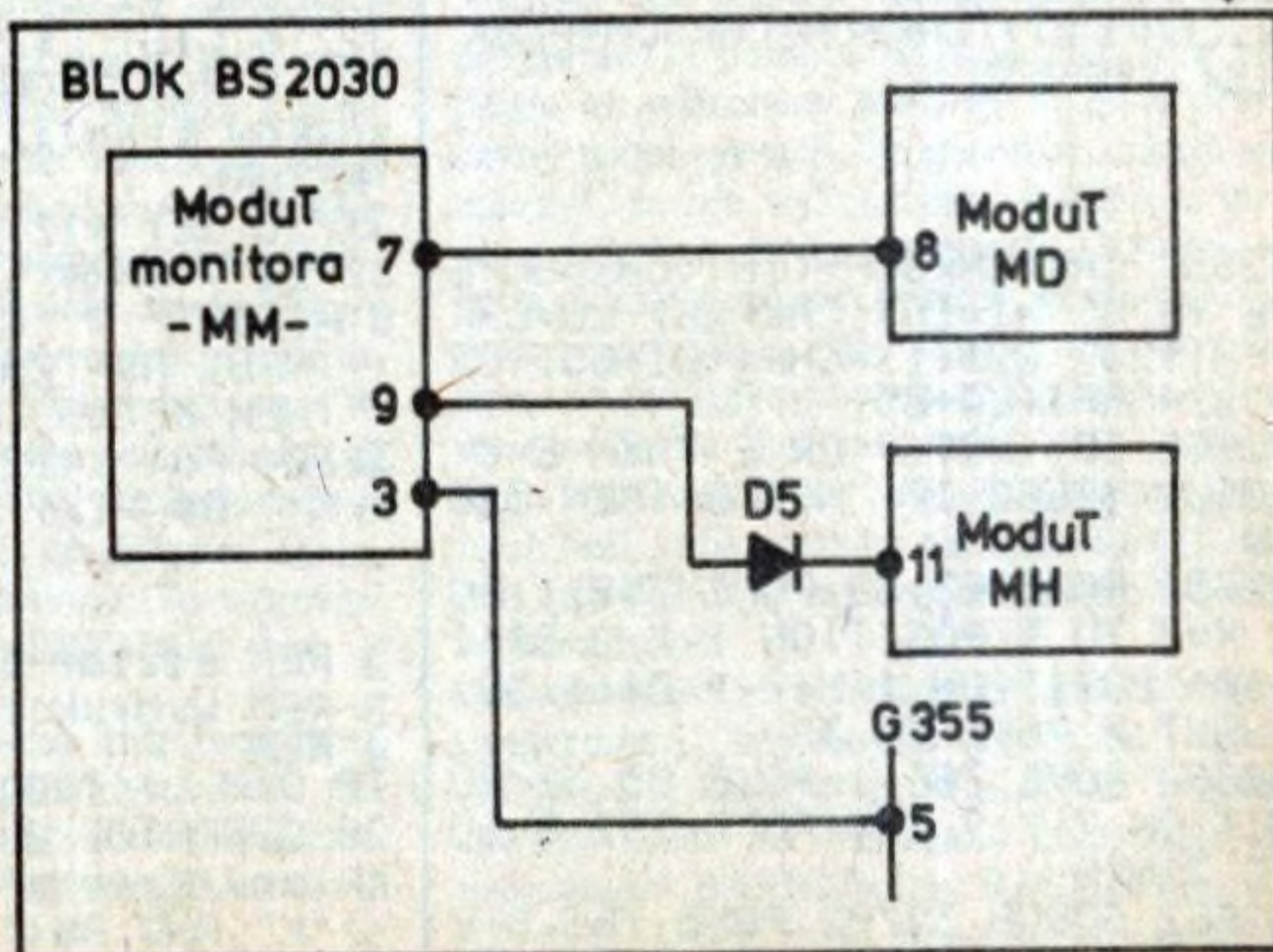


rys. 1

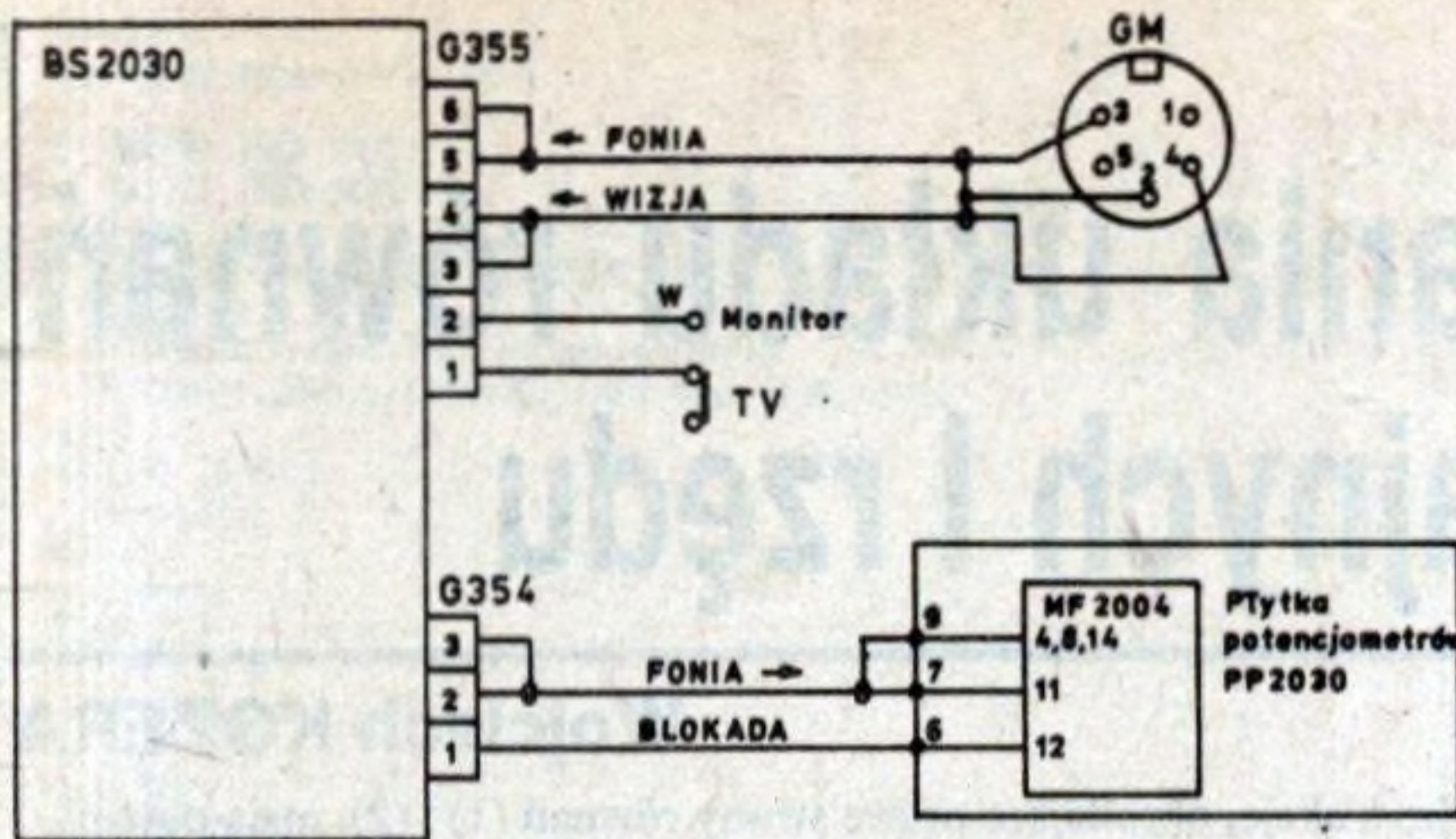
Schemat źródłowy modułu przedstawiono na rysunku nr 1. Zadaniem układu jest dopasowanie sygnałów wizji i fonii z komputera do poziomu wymaganego przez układy telewizora oraz wyłączenie z pracy części w.cz. odbiornika. Przełączenie telewizora w tryb pracy monitora odbywa się przełącznikiem „W”. Przez jego styki podawane jest napięcie +12 V zapewniające pracę wzmacniacza wizji (T1, T2, T3) i fonii (T4) oraz powodujące blokowanie części w.cz. telewizora: modułu pośredniej częstotliwości MP, sygnału różnicowego fonii w module MF oraz zmianę stałej czasu w module synchronizacji MH. Część w.cz. musi być wyłączona, aby umożliwić współpracę z zewnętrznym źródłem sygnału wizji i fonii. Powrót do pracy w trybie odbiornika TV odbywa się poprzez ustawienie przełącznika „W” w przeciwnie położenie.

Układ zamontowano na jednostronnie drukowanej płycie (rys. 2 i 3). Moduł wyposażono w 14-stykowe gniazdo (tego samego typu, jak pozostałe moduły telewizora), a w bloku BS 2030 w miejscu oznaczonym „MM” wlotowano 14 kołeczków stykowych. Ponadto konieczne są jeszcze przeróbki bloku BS 2030 (rys. 4):

- połączenie punktu 3 złącza z punktem 5 gniazda G 355,
- rozcięcie ścieżki dochodzącej do punktu 11 (ST.CZASU) złącza MH i połączenie rozciętej ścieżki diodą D5,
- doprowadzenie napięcia +12 V z punktu 8 złącza MD do punktu 7 złącza MM.



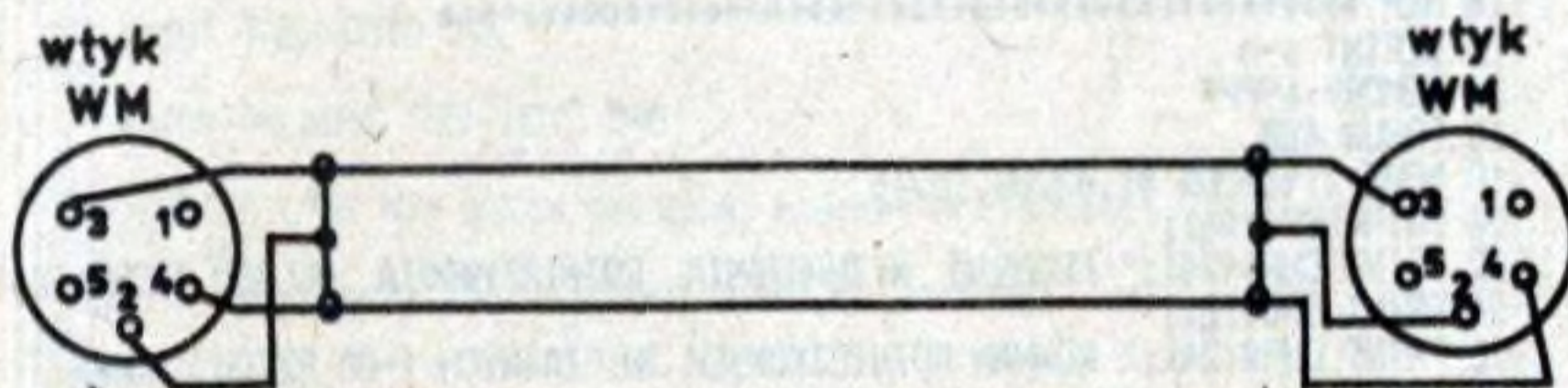
rys. 4



rys. 5

Doprowadzenie zewnętrznych sygnałów wizji i fonii do modułu zrealizowano poprzez 5-stykowe gniazdo magnetofonowe zamontowane w miejscu firmowego gniazda służącego do nagrywania na magnetofon (firmowe połączenia z tym gniazdem należy usunąć). Połączeń między gniazdem, płytką potencjometrów PP 2030 a blokiem BS 2030 należy dokonać ekranowym przewodem (rys. 5). Włącznik „W” umieszczono z tyłu telewizora przy gniazdku antenowym.

Do połączenia komputera z telewizorem wykorzystano ekranowany przewód dwużyłowy zakończony z obu stron 5-stykowymi wtykami magnetofonowymi (rys. 6).



rys. 6

Po wykonaniu powyższych czynności oraz sprawdzeniu ich poprawności umieszczamy moduł w złączu MM w bloku BS 2030 i przystępujemy do regulacji. Potencjometry montażowe R5 i R7 ustawiamy w środkowym położeniu. Włączamy telewizor i przyłączamy w tryb pracy monitora (włącznikiem „W”).

Do gniazda Gm doprowadzamy sygnał z komputera. Regulację

najlepiej jest przeprowadzać gdy wytwarzany przez komputer obraz jest nieruchomy i kolorowy.

Może to być np.: program 256 kolorów z programu demonstracyjnego ATARI. Regulujemy potencjometrem R5 tak, aby uzyskać na ekranie wytwarzany przez komputer obraz. Następnie ustawiamy potencjometr R7 tak, aby obraz był stabilny a odtwarzanie kolorów właściwe. Ewentualnej korekty nasycenia barw możemy dokonać zmieniając ponownie ustawienie R5. Na tym regulację kończymy. W wypadku współpracy z innym komputerem (lub magnetowidem) może zajść konieczność zmiany wartości rezystora R15 w torze fonii modułu. Należy w tym wypadku kierować się zasadą, aby przy takim samym położeniu suwaka regulacji głośności uzyskać podobne natężenie dźwięku w obu trybach pracy odbiornika.

Bieżącej korekty jasności, nasycenia, kontrastu i głośności dokonujemy elementami regulacyjnymi telewizora.

Spis elementów

Tranzystory

T1, T4 — BC 308, BC 178
T2, T3, T5 — BC 238, BC 108

Diody

D1÷D5 — BAVP 17, 18, 19

Rezystory

Typu MŁT 0,125 W

o wartościach jak na schemacie,

R5, R7 — miniaturowe potencjometry montażowe (poziome).

Kondensatory

Elektrolityczne napięcie 16 V

C3 — ceramiczny.

Inne

W — przełącznik suwakowy miniaturowy dowolnego typu.

Gniazdo 14-stykowe.

Kołeczki stykowe — 14 szt.

Gniazdo magnetofonowe 5-stykowe typu GM.

Wtyki magnetofonowe 5-stykowe typu WM.

LITERATURA:

[1] Instrukcja serwisowa HELIOS TC 500 — Wema 1985 Warszawa,

[2] M.DEC. Monitory, monitory... — Komputer 9/1986

Bogusław LIPCZYŃSKI

PROGRAM 57

Makatka

```

1 BORDER 0: CLS : DIM b$(64):
PRINT PAPER 0; b$: LET J=1: LET
d=2: LET t=3: LET cz=4: LET e=11
: LET f=12: LET g=15: LET h=16:
LET i=21: LET k=31: LET l=13: LE
T p=5: LET o=8: DIM a$(60): LET
a$="+++"+"+++"+"+++"+"+++"
:
2 PAPER INT (4+RND*4): INK IN
T (RND*4): LET a=CZ+INT (RND*9):
LET b=INT (RND*p): LET c=INT (R
ND*o): PRINT AT b+d, c; a$(a+j): AT
e-b, c; a$(a+t): AT b+d, g-c; a$(a+d
): AT e-b, g-c; a$(a+cz): AT b+d, c+n
; a$(a+j): AT e-b, c+n; a$(a+t): AT b
+d, k-c; a$(a+d): AT e-b, k-c; a$(a+c
z): AT b+f, c; a$(a+j): AT i-b, c; a$(

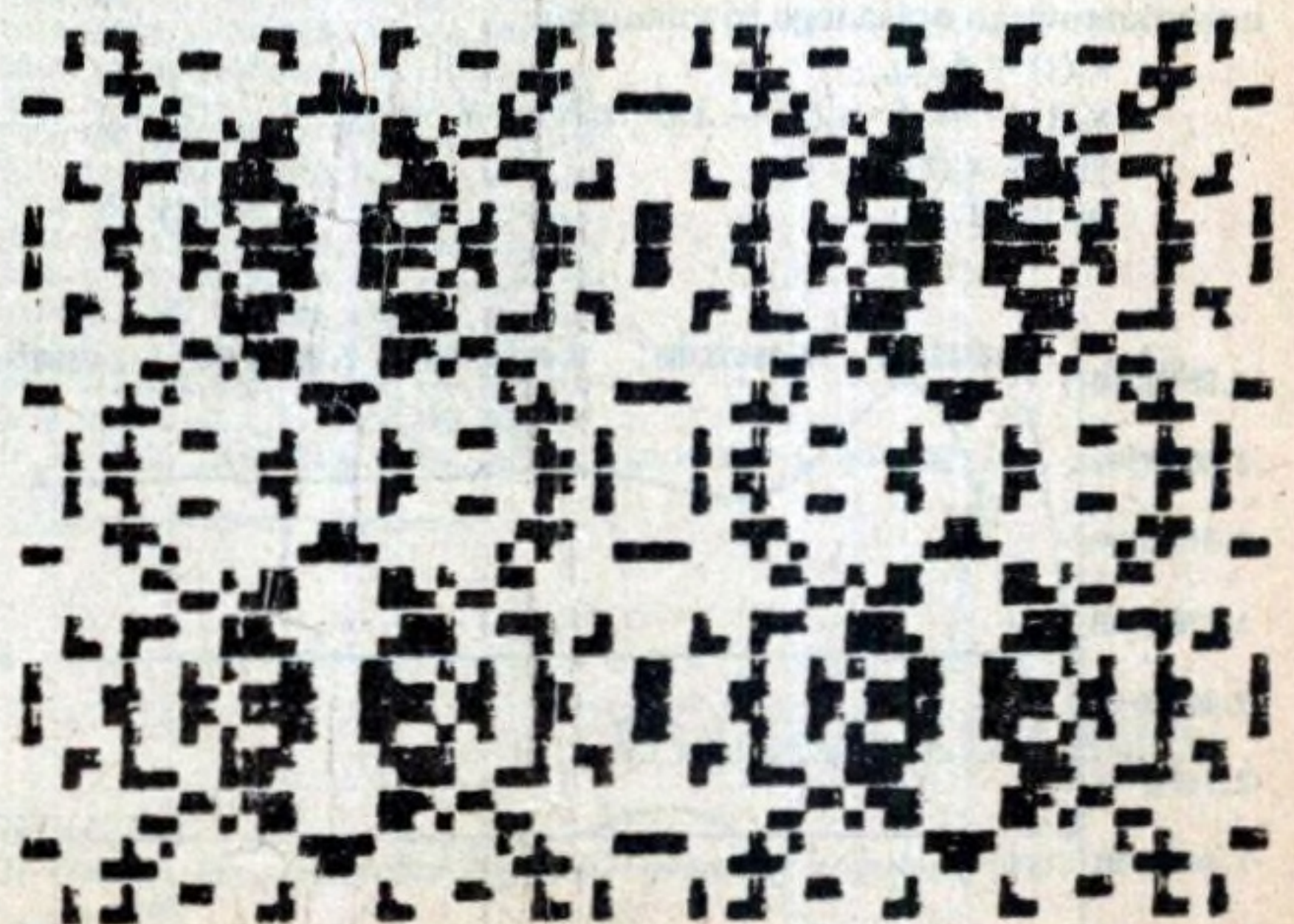
```

ZX SPECTRUM

```

a+t): AT b+f, g-c; a$(a+d): AT i-b, g
-c; a$(a+cz): AT b+f, c+h; a$(a+j): A
T i-b, c+h; a$(a+t): AT b+f, k-c; a$(
a+d): AT i-b, k-c; a$(a+cz): GO TO
d
3 SAVE "Makatka" LINE 1

```



J. Rubinowicz

Wyznaczanie rozwiązania układu równań różniczkowych zwyczajnych I rzędu

Wojciech KOZERA

Program służy do graficznego przedstawienia rozwiązania układu równań różniczkowych, opisujących, w wypadku ogólnym, wielowymiarowe obiekty dynamiczne nieliniowe, w oparciu o metodę Rungego-Kutty czwartego rzędu (i) (str. 325). Treść programu w języku BASIC na komputer Amstrad/Schneider została opracowana na podstawie prac (2), (3).

Dane początkowe

Przed uruchomieniem programu należy wstawić do programu głównej instrukcje określające równania wyjścia (zgodnie z komentarzem w liniach 110—126) oraz instrukcje określające prawe strony równań stanu obiektu dynamicznego (zgodnie z komentarzem w liniach 286—296).

W trybie konwersacyjnym zadawane są wartości następujących danych:

- n — liczba zmiennych stanu obiektu,
- $x^0(m)$ — wartość i — tego elementu wektora stanu w chwili początkowej ($m = 1...n$),
- tp — chwila początkowa,
- tk — chwila końcowa,
- kr — krok wydruku,
- lw — liczba wykresów (patrz komentarz w liniach 110—126 programu głównego,
- mik — najmniejszy dopuszczalny krok,
- eps — względna dokładność rozwiązania,
- eta — oczekiwany rząd odpowiedzi.

Ponadto, na pytanie o wydruk wyników na drukarce, należy nacisnąć klawisz (T) dla akceptacji, lub (N) — w przeciwnym wypadku.

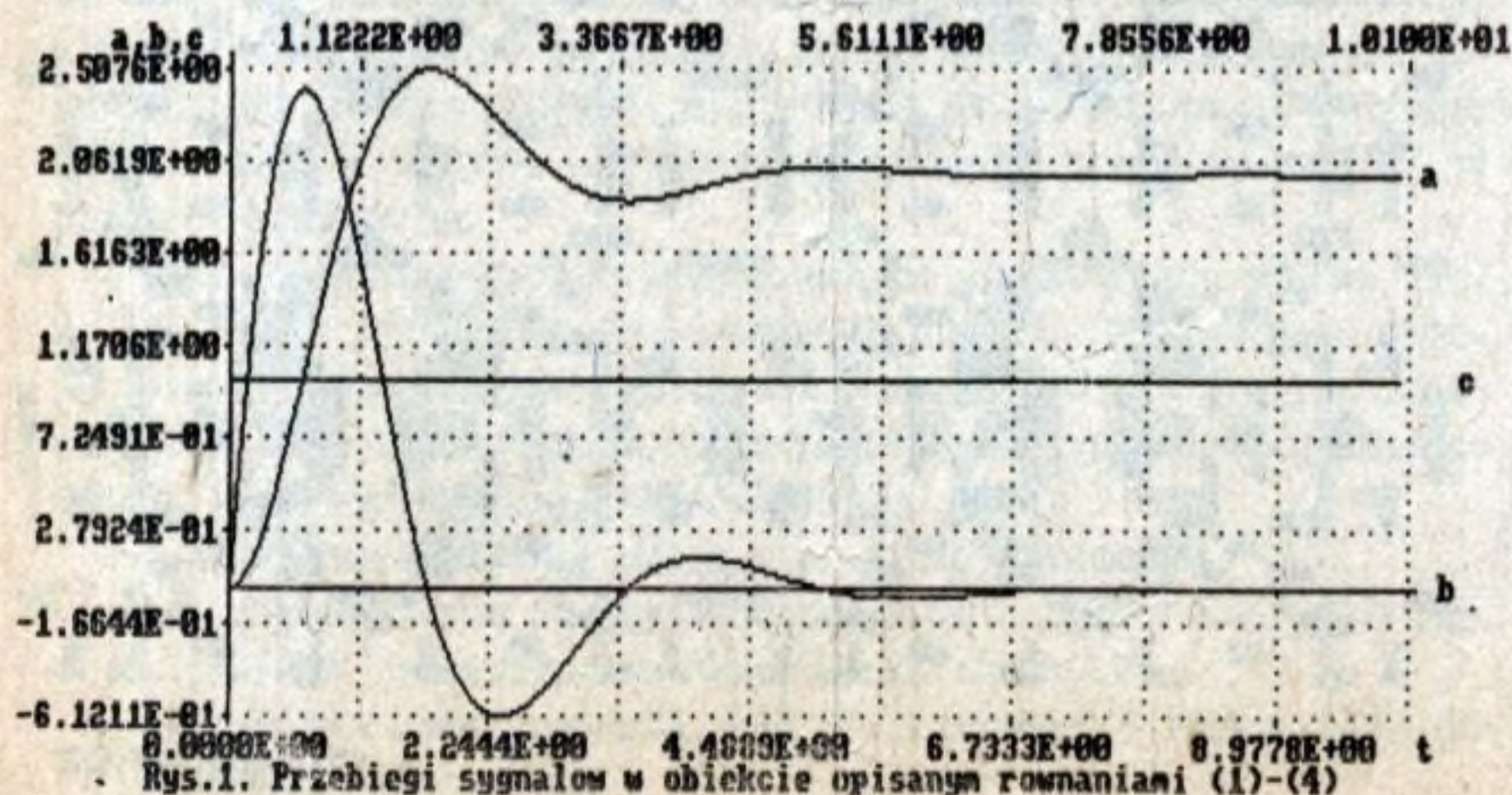
Wyniki końcowe

Efekt działania programu jest wyświetlenie na ekranie monitora, a przy wybranej opcji druku — także wydruk na drukarce, przebiegów w czasie wybranych zmiennych oznaczonych kolejno małymi literami alfabetu (literze a odpowiada przebieg pierwszej zmiennej).

Przykład testowy

Program sprawdzono, między innymi, na przykładzie obiektu jednowymiarowego opisanego równaniami:

$$\begin{aligned} x_1(t) &= x_2(t), & (1) \\ x_2(t) &= -4 \cdot x_1(t) - 1.6 \cdot x_2(t) + 8 \cdot u(t), & (2) \\ y(t) &= x_1(t), & (3) \\ u(t) &= 1. & (4) \end{aligned}$$



Instrukcje, określające prawe strony równań (1) i (2), mają postać:

$$\begin{aligned} d(1) &= yw(2), & (5) \\ d(2) &= 4 \cdot yw(1) - 1.6 \cdot yw(2) + 8 & (6) \end{aligned}$$

a instrukcje, określające równania wyjścia, przyjęto w postaci:

- $w(1e,1) = y0(1)$ — przebieg odpowiedzi $y(t)$ obiektu, oznaczony na wykresie literą a ,
- $w(1e,2) = y0(2)$ — przebieg składowej $x_2(t)$ wektora stanu $x(t)$ obiektu, oznaczony na wykresie literą b ,
- $w(1e,3) = 1$ — przebieg wymuszenia $u(t) = 1(t)$, oznaczony na wykresie literą c .

Dla $n = 2$, $x0(1) = x0(2) = 0$, $tp = 0$, $tk = 10$, $kr = 0,02$, $lw = 3$, $mik = eps = eta = 1E-6$ oraz wybranej opcji druku uzyskano przebiegi przedstawione na rys. 1.

```

2 REM *****
4 REM * WYZNACZANIE ROZWIĄZYWANIA UKŁADU RÓWNAŃ *
6 REM * RÓZNICZKOWYCH ZWYCZAJNYCH I-GO RZĘDU *
8 REM * (C): Wojciech Kozera, Szczecin 1988 *
10 REM *****
12 DEFINT i-n
14 MEMORY &9FFF
16 GOSUB 480
18 MODE 2:WINDOW #1,42,80,15,16
20 PRINT SPC(16);
22 PRINT CHR$(24);" PROGRAM WYZNACZANIA ROZWIĄZYWANIA UKŁADU ";CHR$(24)
24 PRINT SPC(16);
26 PRINT CHR$(24);" RÓWNAŃ RÓZNICZKOWYCH ZWYCZAJNYCH I-GO RZĘDU ";CHR$(24)
28 ir=-1
30 LOCATE 20,5:PRINT CHR$(7);:INPUT "Liczba zmiennych stanu ..... n=",n
32 DIM d(n),y(n),y0(n),y1(n),y2(n),y3(n),yh(n),yw(n)
34 LOCATE 25,6:PRINT "Warunki początkowe : x0(";
36 FOR m=1 TO n
38 LOCATE 50,6:PRINT SPACE$(20):LOCATE 50,6
40 PRINT USING("#");m;:PRINT CHR$(7);";";
42 INPUT "=";y0(m)
44 NEXT m
46 LOCATE 20,6:PRINT SPACE$(59);
48 LOCATE 20,6:PRINT CHR$(7);:INPUT "Chwila początkowa ..... tp=",tp
50 x0=tp
52 LOCATE 20,7:PRINT CHR$(7);:INPUT "Chwila końcowa ..... tk=",tk
54 LOCATE 20,8:PRINT CHR$(7);:INPUT "Krok wydruku ..... kr=",rkr
56 LOCATE 20,9:PRINT CHR$(7);:INPUT "Liczba wykresów ..... lw=",lw
58 LOCATE 20,10:PRINT CHR$(7);"Czy drukować wykresy <T/N>? : ";
60 s$=UPPER$(INKEY$):IF (s$="N") THEN GOTO 60
62 IF (s$="T") THEN PRINT "Tak" ELSE PRINT "Nie"
64 LOCATE 18,12:PRINT CHR$(7);
66 INPUT "Najmniejszy krok dopuszczalny .... mik=",rmak
68 LOCATE 18,13:PRINT CHR$(7);
70 INPUT "Względna dokładność rozwiązania .. eps=",eps
72 LOCATE 18,14:PRINT CHR$(7);
74 INPUT "Oczekiwany rząd odpowiedzi ..... eta=",eta
76 LOCATE 18,15:PRINT CHR$(7);"Podpis pod rysunkiem :";
78 INPUT #1," t$:t$=" "+t$+" "
80 IF (s$<>"T") THEN GOTO 94
82 i=-1
84 WHILE i
86 IF (INP(&F500) AND 64)=0 THEN i=0
88 IF i THEN LOCATE 30,24:PRINT CHR$(7);"Włącz drukarkę !";
90 WEND
92 PRINT #8,CHR$(27);"1";CHR$(10);
94 LOCATE 24,24:PRINT CHR$(24);" Chwila aktualna ..... t=";CHR$(24);
96 mle=4000/lw
98 DIM w(mle,lw)
100 lt=LEN(t$)/2:lt=40-lt:le=0
102 WHILE (x0<tk)
104 LOCATE 51,24:PRINT SPACE$(28):LOCATE 51,24
106 PRINT CHR$(24);INT(x0*1000000+0.5)*0.000001;CHR$(24);
108 le=le+1
110 .....
112 .....
114 .....
116 .....
118 .....
120 .....
122 .....
124 .....
126 .....
128 w(1e,1)=y0(1)
130 w(1e,2)=y0(2)
132 w(1e,3)=1
134 .....
136 .....

```

```

138 x1=x0+rkr
140 eps1=0.025*eps:w1=rkr:sh=0

142 IF (w1<0) THEN sh=-1
144 IF (w1>0) THEN sh=1
146 nt=0
148 IF ir THEN h=w1:prh=h:ir=0:GOTO 156
150 h=prh
152 IF (ABS(h)<rma) THEN h=rma*sh:prh=h
154 IF ((x0+1.25*h-x1)*sh)=0 THEN h=w1 ELSE nt=-1
156 na=-1
158 FOR i=1 TO n
160 y(i)=y0(i)
162 NEXT i
164 x=x0:hp=h
166 GOSUB 274
168 FOR i=1 TO n
170 y1(i)=yh(i)
172 NEXT i
174 hh=0.5*h
176 FOR i=1 TO n
178 y(i)=y0(i)
180 NEXT i
182 x=x0:hp=hh
184 GOSUB 274
186 FOR i=1 TO n
188 y2(i)=yh(i)
190 NEXT i
192 FOR i=1 TO n
194 y(i)=y2(i)
196 NEXT i
198 x=x0+hh:hp=hh
200 GOSUB 274
202 FOR i=1 TO n
204 y3(i)=yh(i)
206 NEXT i
208 w4=0
210 FOR k=1 TO n
212 w1=y1(k):w3=y3(k):w2=ABS(w1-w3):w1=ABS(w1):w3=ABS(w3)
214 IF (w1<w3) THEN w1=w3
216 IF (w1<eta) THEN w1=eta
218 w1=w2/w1
220 IF (w1>w4) THEN w4=w1
222 NEXT k
224 IF (w4<eps) THEN GOTO 246
226 na=0
228 IF (ABS(hh)>rma) THEN GOTO 240
230 CLS
232 PRINT CHR$(7); "** Nie mozna osiagnac zadanej dokladnosci eps =";eps;
234 PRINT " **"
236 PRINT " - nawet przy zmniejszeniu kroku do wartosci kr =";rma;
238 END
240 h=hh:prh=h:nt=-1
242 FOR k=1 TO n:y1(k)=y2(k):NEXT k
244 GOTO 174
246 FOR k=1 TO n
248 w3=y3(k):y0(k)=w3+b.66666667E-02*(w3-y1(k))
250 NEXT k
252 IF (nt=0) THEN GOTO 262
254 x0=x0+h
256 IF ((w4<eps) AND na) THEN h=h+h:prh=h
258 IF ((x0+1.25*h-x1)*sh)=0 THEN h=x1-x0:nt=0
260 GOTO 156
262 x0=x1
264 IF (le>me) THEN GOTO 268
266 WEND
268 GOSUB 320
270 END
272 REM
274 w4=0.5*hp:w1=w4:w2=x
276 FOR k=1 TO n:yw(k)=y(k):yh(k)=0:NEXT k
278 FOR i=1 TO 4
280 ON i GOTO 282,284,284,282
282 w3=w4:GOTO 286
284 w3=hp
286 .....
288 Przepisz zmiennym d(i) wartosci pochodnych zgodnie z ukladem rownan:
290 .....
292 d(i)=fi(yw(1),yw(2),...,yw(n),w2); i=1..n
294 .....
296 .....
298 d(1)=yw(2)
300 d(2)=-4*yw(1)-1.6*yw(2)+8
302 .....
304 .....
306 FOR k=1 TO n
308 w2=d(k):yh(k)=yh(k)+w3*w2:yw(k)=y(k)+w1*w2
310 NEXT k
312 w2=x+w1:w1=w3
314 NEXT i
316 FOR k=1 TO n:yh(k)=y(k)+0.33333333*yh(k):NEXT k
318 RETURN
320 REM =====
322 REM = PROCEDURA SPORZADZANIA WYKRESU =
324 REM =====
326 kk=le MOD 506
328 IF (kk=0) THEN kk=1 ELSE kk=0
330 kk=le\506+kk
332 ymin=1E+10:ymax=-1E+10
334 FOR i=1 TO lw
336 FOR j=1 TO le
338 t=w(j,i)
340 IF (t>ymin) THEN ymax=t
342 IF (t<ymin) THEN ymin=t
344 NEXT j
346 NEXT i
348 ysk=336/(ymax-ymin)
350 FOR k=1 TO kk
352 CLS
354 IF (s="T") THEN GRAPHICS PAPER 0:GRAPHICS PEN 1:GOTO 360

```

```

356 WINDOW 12,75,2,23:PAPER 1:PEN 0:CLS
358 GRAPHICS PAPER 1:GRAPHICS PEN 0
360 WINDOW 1,80,1,25:PAPER 0:PEN 1
362 ia=-8
364 FOR i=1 TO 8
366 ia=ia+48:PLOT 90,ia:DRAWR 2,0:ib=92
368 FOR j=1 TO 63
370 ib=ib+8:PLOT ib,ia
372 NEXT j
374 NEXT i
376 ia=-20
378 FOR i=1 TO 5
380 ia=ia+112:PLOT ia,36:DRAWR 0,2
382 NEXT i
384 ia=36
386 FOR i=1 TO 10
388 ia=ia+56:ib=40
390 FOR j=1 TO 42
392 ib=ib+8:PLOT ia,ib
394 NEXT j
396 NEXT i
398 ia=36
400 FOR i=1 TO 5
402 ia=ia+112:PLOT ia,378:DRAWR 0,2
404 NEXT i
406 tp1=506*(k-1)*rkr+tp:tk1=tp1+505*rkr
408 LOCATE 12-2*lw,1
410 FOR i=1 TO lw:PRINT CHR$(96+i);",":NEXT i
412 ysk1=(ymax-ymin)/7:t=ymin
414 FOR i=1 TO 8
416 LOCATE 1,3*i-1:PRINT USING("###.###^");t:t=ysk1
418 NEXT i
420 LOCATE 76,24:PRINT "t";

422 xsk=(tk1-tp1)/9:t=tp1+xsk:LOCATE 11,1
424 FOR i=1 TO 5
426 PRINT " ";USING("###.###^");t;t=t+2*xsk
428 NEXT i
430 t=tp1:LOCATE 4,24
432 FOR i=1 TO 5
434 PRINT " ";USING("###.###^");t;t=t+2*xsk
436 NEXT i
438 ia=506*(k-1)+1:ib=ia+505:IF (ib>le) THEN ib=le
440 FOR i=1 TO lw
442 l=i:PLOT 92,INT(ysk*(w(ia,i)-ymin)+0.5)+40
444 FOR j=ia+1 TO ib
446 DRAW 1+92,INT(ysk*(w(j,i)-ymin)+0.5)+40
448 l=l+1
450 NEXT j
452 l=YPOS+8:MOVE 593+i*8,l
454 TAG:PRINT CHR$(96+i);TAGOFF
456 NEXT i
458 IF (tp1*tk1>0) THEN GOTO 462
460 PLOT INT(0.5-1/rkr*tp1)+92,40:DRAWR 0,344
462 IF (ymin*ymax>0) THEN GOTO 466
464 PLOT 92,INT(0.5-ysk*ymin)+40:DRAWR 507,0
466 LOCATE 1,25
468 IF (s="T") THEN PRINT t$;CALL &A000 ELSE PRINT CHR$(24);t$;CHR$(24);
470 PRINT CHR$(7);
472 WHILE INKEY$="":WEND
474 NEXT k
476 CLS
478 RETURN
480 REM =====
482 REM = PROCEDURA KOPIOWANIA =
484 REM = EKRANU NA DRUKARKE =
486 REM =====
488 FOR i=&A000 TO &A0BF
490 READ k:POKE i,k:l=l+k:NEXT
492 DATA &cd,&ba,&bb,&cd,&e7,&bb,&32,&bd
494 DATA &a0,&cd,&bc,&a0,&21,&8f,&01,&22
496 DATA &be,&a0,&11,&00,&00,&3e,&07,&32
498 DATA &c0,&a0,&cd,&7c,&a0,&0e,&00,&3a
500 DATA &c0,&a0,&47,&e5,&d5,&c5,&cd,&f0
502 DATA &bb,&c1,&d1,&21,&bd,&a0,&be,&e1
504 DATA &37,&20,&01,&a7,&cb,&11,&2b,&2b
506 DATA &10,&e9,&cd,&af,&a0,&79,&cd,&a6
508 DATA &a0,&13,&e5,&21,&7f,&02,&37,&ed
510 DATA &52,&e1,&38,&05,&2a,&be,&a0,&10
512 DATA &cc,&23,&7c,&b5,&cb,&2b,&11,&00
514 DATA &00,&22,&be,&a0,&3e,&07,&bd,&20
516 DATA &b9,&7c,&b4,&20,&b5,&3e,&04,&32
518 DATA &c0,&a0,&18,&ae,&3e,&1b,&cd,&a6
520 DATA &a0,&3e,&31,&cd,&a6,&a0,&00,&00
522 DATA &00,&00,&00,&c9,&e5,&3e,&42,&cd
524 DATA &1e,&bb,&e1,&28,&02,&e1,&c9,&3e
526 DATA &0d,&cd,&a6,&a0,&3e,&0a,&cd,&a6
528 DATA &a0,&3e,&1b,&cd,&a6,&a0,&3e,&4c
530 DATA &cd,&a6,&a0,&3e,&7f,&cd,&a6,&a0
532 DATA &3e,&02,&cd,&a6,&a0,&c9,&cd,&2e
534 DATA &bd,&38,&fb,&cd,&2b,&bd,&c9,&3a
536 DATA &c0,&a0,&fe,&07,&cb,&af,&cb,&11
538 DATA &cb,&11,&cb,&11,&c9,&00,&00,&00
540 IF 1<>23151 THEN PRINT CHR$(7); "** Bład w DATA ! **":END
542 RETURN
544 REM =====

```

Wojciech KOZERA

LITERATURA:
(1) H. Górecki. Praca zbiorowa: Algorytmy i programy sterowania, WNT, Warszawa 1980.
(2) ELWRO: ODRA—Algol. Procedury. Opis 1204—VIII—11, Wrocław 1970.
(3) Schneider CPC - 6128. Benutzerhandbuch.

INFORMATYKA W SZKOLE

PRIMUM NON NOCERE, czyli jak chcemy uczyć podstaw informatyki

Włodzimierz GOGOLEK

Primum non nocere, zasada ta obowiązuje także nauczycieli. Mają oni bowiem ogromne możliwości kształtowania naszej wiedzy. Pozwala nam ona dalej żyć godnie, dostrzegać dobro i zło, ułatwia poruszanie się w otaczającym świecie. A nie jest to proste u schyłku wieku, który wraz z poprzedzającym go tworzą najburzliwszy okres ze stu wieków, mijających od początku, gdy na Ziemi rozpoczął swoje panowanie homo sapiens.

Po sile pary, potędze elektryczności, poznamy teraz moc informacji. Stało się to za przyczyną stosunkowo młodziutkiej gałęzi wiedzy, jaką jest informatyka. Fundamenty jej stworzono od bardzo dawna, jednak dopiero (już?) 50 lat temu pierwszy komputer wspomógł intelektualną pracę człowieka. Pozwoliło to dostrzec czubek przysłowiowej góry lodowej. Tworzy ją tutaj znaczenie efektywnego (szybkiego!) wykorzystania i przetworzenia informacji. Dzień dzisiejszy pozwala nam zauważyć zarysy, wcześniej niewyobrażalnego ogromu problematyki związanej z informacją i jej przetwarzaniem lub inaczej z INFORMATYKĄ, prościej — z komputerami.

Jednak mimo ogromnej rangi, jaką ostatnio zdobywa sobie informatyka, należy pamiętać, że jest to dziedzina wiedzy dająca nam narzędzia, które mogą wspomagać każde działanie ludzkie. Poza specjalistycznymi kierunkami studiów, informatyka nie powinna występować zatem samodzielnie, jako cel działania — mamy w tym zakresie duże i jednocześnie smutne doświadczenia — piękne ośrodki obliczeniowe — duma dyrekcji..., senne zajęcia o komputerach, programy pisane za resztki stypendium.

Przystępując zatem do nauczania podstaw informatyki (PI), należy pamiętać — „po pierwsze nie szkodzić”, przez zniechęcanie do informatyki. Przyczynia się temu między innymi bardzo ryzykowne traktowanie informatyki jako samodzielnej nauki — przedmiot ten staje się wówczas abstrakcyjny, mało użyteczny. Motywacja do nauki gwałtownie topnieje (widać to po twarzach uczących się, cień zainteresowania pozostaje tylko u nielicznych entuzjastów).

Pierwszym zatem postulatem „by nie szkodzić” jest ściśle powiązanie nauczania podstaw informatyki z jej praktycznymi zastosowaniami. Nie historia komputerów, system dwójkowy, a atrakcyjne (spektakularne) zastosowania komputerów powinny być tematem pierwszych zajęć. Gwarantuje to, iż już podczas pierwszych zajęć nie stracimy „kredytu zainteresowania” naszych uczniów. Kredyt ten, mimo pozorów, jest bardzo niewielki — ukształtowany zazwyczaj „modą” lub trakto-

waniem komputerów jako urządzeń do zabawy. Warto w tym miejscu podkreślić pozytywne efekty nieograniczonego dostępu do komputerów w celu wykorzystania ich do zabawy. Okazuje się, że zabawy te zmniejszają okres poznawania podstawowych czynności manualnych, które należy wykonywać podczas obsługi komputera.

Drugim postulatem w myśl zasady „nie szkodzić” jest wyeliminowanie zajęć z podstaw informatyki w szkołach, gdzie nie można zapewnić swobodnego dostępu wszystkim uczącym się tego przedmiotu (dwie osoby na jedno stanowisko). Przy małej liczbie komputerów jest to oczywiście kwestia odpowiedniego podziału na grupy i liczby nauczycieli...

Trzeci postulat — na każdych zajęciach nauczyciel powinien wykorzystywać komputer. Wiąże się to z koniecznością wcześniejszego przygotowania odpowiednich programów demonstracyjnych, także obrazów komputerowych. Konkretnie propozycje tego typu pomocy dydaktycznych zawarte będą w cyklicznych odcinkach stanowiących rozwinięcie zaproponowanego dalej programu zajęć z PI.

Pominięte zostały założenia zerowe. Jednym z nich jest to, by uczący się nie uczyli nauczycieli. Innym założeniem zerowym jest przygotowanie odpowiedniej literatury. Oczywiście kolejne odcinki zawierały będą propozycje uzupełniającej literatury, jednak konieczne jest bazowanie na jednym, sprawdzonym zwartym wydaniu, jest nim np. „Komputery osobiste” D. Madeja i innych.

Zgodnie z naszą deklaracją, z wakacyjnego numeru „IKS-a”, prezentujemy własną (zweryfikowaną) propozycję programu „Podstaw informatyki”. Zakłada się w nim, że minimalna liczba godzin zajęć powinna wynosić 50, przy proporcji 2 do 3, na korzyść ćwiczeń. Znając naszą „rzeczywistość informatyczną” należy założyć, że owe ćwiczenia odbywać się będą w połowie z wykorzystaniem sprzętu (w sali komputerowej) i w połowie „na sucho” bez komputera. Przeplot ten pozwoli, by na przykład sala z 10 komputerami była wystarczająca dla 40 uczących się (dwie grupy po 20 osób).

Przedstawiony poniżej program będzie cyklicznie uzupełniany wspomnianymi odcinkami, które w miarę dokładnie prezentowały będą treści kolejnych zajęć. Zdając sobie jednak sprawę z tego, że nasza propozycja może być dyskusyjna, przy każdym odcinku PI, będziemy zamieszczali opinie państwa, ale tylko te, które poza krytyką zawierały będą konkretne propozycje. Uzupełnieniem każdego odcinka będą także materiały naszych współpracowników — rozszerzające proponowane treści nauczania, w tym także wspomniane obrazy dydaktyczne i programy. Dopelnienie całości tworzyć będzie lista źródeł materiałów do zajęć — głównie wybrane artykuły z krajowej prasy informatycznej.

PODSTAWY INFORMATYKI (Tematy wykładów wraz z tezami)

1. Wprowadzenie do podstaw informatyki. 2 godz.
Omówienie niektórych (atrakcyjnych) zastosowań komputerów.
Proponowany jest przykład przetwarzania obrazów, a w tym: analiza obrazów uzyskiwanych z kosmosu, komputerowe wspomaganie zabiegów konserwatorskich obrazów na przykładzie Mony Lisy.
Zastosowanie maszyn do identyfikacji autorów tekstów.
Krótka historia rozwoju technologii budowy komputerów (generacje).
2. Sprzęt informatyczny. 2 godz.
Reprezentacja informacji w komputerze — system binarny, a cyfry dziesiętne, litery i obrazy (kod ASCII, atrybuty obrazów).
Budowa mikrokomputera, funkcje, podstawowe parametry jego podzespołów (mikroprocesor, pamięć RAM, ROM).
3. Programowanie. 4 godz.
Algorytm — jego istota i przeznaczenie.
Interpretacja terminu „instrukcja”.
Cechy charakterystyczne niektórych języków programowania: Basic, Pascal, Logo, C i Assembler.
Istota funkcji kompilatora (i interpretera) — „szybkość programu”.
Wybrane instrukcje Basica.

UWAGA Po trzecim temacie, powinien nastąpić początek zajęć praktycznych, zgodnie z odrębnym planem. Celowe jest w tym czasie polecić wykonanie uczącym się (indywidualnie lub w grupach) opracowania nt. „Najciekawsze zastosowania komputerów”.

4. Sprzęt informatyczny — urządzenia zewnętrzne. 2 godz.

Przeznaczenie, zalety i wady urządzeń wejściowych, wyjściowych i pamięci zewnętrznych (magnetycznych). Klawiatura, monitor, drukarka, pióro świetlne, joystick, mysz, OCRy. Przeznaczenie, zalety i wady pamięci typu: dyski elastyczne i sztywne oraz dyski optyczne i streamery.

5. Profesjonalne komputery. 2 godz.

Zastosowania, zalety i wady. IBM PC jako przykład unifikacji podzespołów, urządzeń zewnętrznych i standaryzacji oprogramowania.

Informacja o dużych komputerach serii ODRA, RIAD oraz superkomputerach — CRAY, ETA.

6. Oprogramowanie systemowe. 1 godz.

Podstawowe cechy systemów operacyjnych na przykładzie CPM'a i DOS'a.

7. Oprogramowanie użytkowe. 2 godz. Omówienie oprogramowania („życzeniowości komputerów”) na przykładzie:

— redagowanie tekstów (WordStar, Chiwriter, Fontasy),

— bazy danych (dBase III Plus),

— programy kalkulacyjne (VisiCalc),

— programy graficzne (Paintbrush),

— gry komputerowe.

Proponowane przykłady powinny być zastąpione odpowiednim oprogramowaniem, które jest dostępne w szkolnych komputerach.

8. Oprogramowanie własne, szczególnego przeznaczenia.

Omówienie możliwości praktycznego wykorzystania wcześniej przygotowanych programów. Np. programy: sortujący, zbierający i przetwarzający dane z dziennika lekcyjnego (pensum nauczycieli, obecności, średnie oceny, itp.).

9. Sieci komputerowe, wielodostęp, teletransmisja informacji. 1 godz.

Międzykomputerowa łączność — przesyłanie wszelkiej postaci informacji, także obrazów, przykłady. Szybki dostęp (przekazywanie) do superaktualnych informacji. Zasady dostępu do światowych baz danych na przykładzie International Computing Centre.

10. Zastosowanie komputerów w „szkolnej branży”, np.: w muzyce, sporcie, pedagogice, wojsku. 2 godz.

Wykład gościnny profesjonalisty (z „darem dydaktycznym”) lub wycieczka do firmy świadczącej informatyczne usługi.

11. Problematyka dialogu człowieka z komputerem. 2 godz.

Niepowodzenia sztucznej inteligencji. Ekspert-systemy. Porozumiewanie się z komputerami w języku naturalnym — głosem i za pośrednictwem informacji pisanych. Życzliwość komputerów.

12. Prezentacja najciekawszych opracowań nt.: „Najciekawsze zastosowania komputerów”. 2 godz.

PODSTAWY INFORMATYKI (Tematy ćwiczeń wraz z tezami)

Uwagi wstępne

Założono, że dwugodzinne ćwiczenia odbywać się będą jednocześnie w dwóch grupach — przemianie: jedna godzina zajęć teoretycznych (T) i w sali komputerów godzina zajęć praktycznych (P).

1. (T) Reprezentacja informacji w komputerze. 2×1 godz.

System binarny a cyfry dziesiętne, litery i obrazy, kod ASCII.

(P) Obsługa mikrokomputera na przykładzie dostępnego sprzętu. 2×1 godz.

Włączenie i wyłączenie zestawu mikrokomputera. Rozmieszczenie klawiszy QWERTY. Klawisze funkcyjne (systemowe) itp.

2. (T) Budowa mikrokomputera. 2×1 godz.

Wprowadzanie liczb do pamięci komputera, obliczanie sumy/różnicy/ilorazu/iloczynu, wyświetlanie wyniku.

Mikroprocesor, pamięci półprzewodnikowe, układy współpracy z urządzeniami zewnętrznymi.

(T) Przykłady algorytmizacji zadań. 2×1 godz.

Znaczenie konwersacyjnego wprowadzania danych. Kontrola zakresów danych podczas ich wprowadzania oraz wyświetlanie komentarzy.

(P) Sprawdzenie działania wybranych instrukcji Basica. 4×1 godz. PRINT (PRINT „X”,a), GOTO, FOR, NEXT, DIM, INPUT (INPUT „X”,a) i w dalszej kolejności IF, THEN.

3. (T) Języki programowania. 2×1 godz.

Podstawowe cechy charakterystyczne języków: Basic, Pascal, LOGO, C, Płan, Cobol. Zalety i wady Assemblera (szybkość, unikalność). Kompilatory i interpretry.

(T) Przeznaczenie, zalety i wady urządzeń zewnętrznych komputerów różnej skali miniaturyzacji. Urządzenia wejściowe i wyjściowe, pamięci magnetyczne. Analiza różnic między komputerami: ODRA, RIAD, PC, superkomputery.

(P) Budowanie własnych, krótkich programów w języku Basic. 6 godz.

Zwrócenie uwagi na poprawną konstrukcję komentarzy generowanych przez komputer po uruchomieniu własnego programu — teksty wyświetlane przy wykorzystaniu instrukcji INPUT”” i PRINT””.

4. (T) Analiza dostępnego oprogramowania na komputery szkolne. 4×1 godz.

Nauka korzystania z dokumentacji, określania poprawnych parametrów.

(P) Nauka i ćwiczenie samodzielnego uruchamiania wybranych programów dostępnych na szkolne komputery. 4×1 godz.



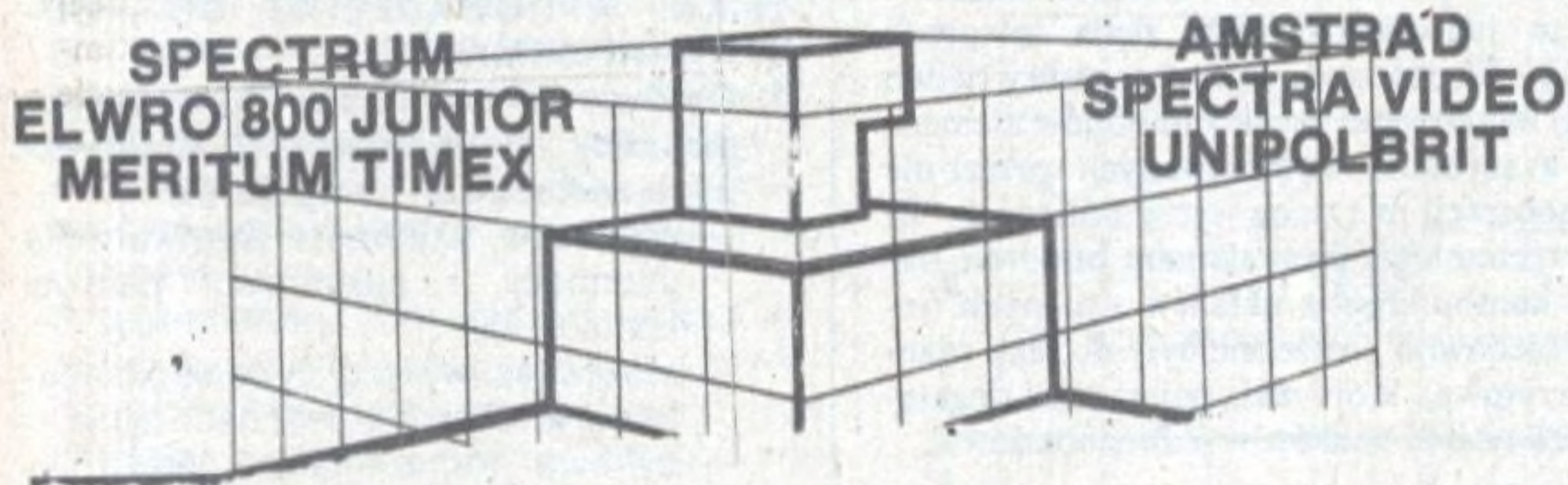
— Pan chce mi wmówić, że myśmy to zaprogramowali?

- a) Fotografia strony tytułowej zeszytu programów komputerowych nr 3
- b) Fotografia wydruku z komputera fragmentu strony



Dodatek „Żołnierza Wolności” Cena 150zł

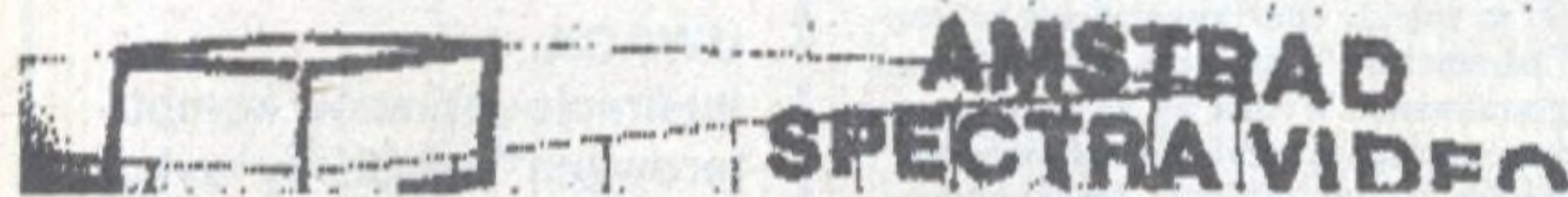
zeszyty programów komputerowych nr 3



Rys. nr 2 Oryginał (a)

I „komputerowa kopia” (b) fragmentu strony jednego z naszych zeszytów.

zeszyty programów komputerowych nr 3



zowego stał się niebieski, skóry dziewczyny z żółtej na alabastrowy, a z sukni bohaterki wydobyto odcień pięknej zieleni.

Oczywiście proces wprowadzania informacji o treści każdego piksela obrazu do pamięci komputera odbywa się automatycznie. Służą temu tak zwane skanery. Urządzenia te wyposażone są w kamery TV. Kamera zamienia światło odbite np. od portretu lub od testowej karty egzaminacyjnej (zob. rys. nr 1) na sygnał

elektryczny (trwa to 1/50 sek.), ten z kolei zostaje zmieniony na liczby, które po chwili „układane” są po kolei w pamięci komputera.

Nie przypadkiem wspominałem tutaj o karcie egzaminacyjnej, mając bowiem jej obraz w pamięci maszyny, nie człowiek, a komputer (jego na „kawę” nie zaprosisz) może sprawdzić jej treść. Gdy treść stanowią zakreślone kratki (egzamin testowy), komputer natychmiast rozpozna czy jedyńki (zakreślone pole

- a) Fotografia „Chopina” zaciemnionego
- b) Fotografia „Chopina” rozjaśnionego

Rys. nr 3 Komputerowe odwzorowanie fragmentu banknotu,



a) po dodaniu wartości 4 do każdego piksela oryginału,



b) po odjęciu wartości 4 od każdego piksela oryginału.

karty testowej) są w odpowiednim miejscu analizowanego dokumentu.

Urządzenie takie zostało zbudowane i z powodzeniem wykorzystane cztery lata temu w WAP. Odczyt i analiza 87 odpowiedzi trwa około 2 sek. Należy w tym miejscu podkreślić, że współczesny sprzęt informatyczny raczej się „nie męczy”, pozwoliło to m.in. wczytać 10 000 kart bez przerwy.

INFORMATYKA W SZKOLE

Zasygnalizowane możliwości komputerów stały się realne dzięki bardzo szybkiemu rozwojowi, najpierw sprzętu, krótko potem oprogramowania.

Początek informatyki (umowny), to maszyny elektromechaniczne. Tworzą one tak zwaną **zerową generację** komputerów. Budowane były głównie z przekaźników elektromagnetycznych. Płynący przez ich uzwojenie prąd powodował, iż „kowadełko” było przyciągane — **jedynka**. Brak prądu — kowadełko odciągała sprężynka — **zero**. (Podobnie jak w dzwonku nad drzwiami — 0,1,0,1,0.... dzwoni). Szybkość, gabaryty, pobór prądu komputerów tej klasy była gorsza niż zła. Niezadowolenie z tego stanu rzeczy spowodowało szybki postęp. Pojawiły się komputery **pierwszej generacji** — lampowe. W większości przekaźniki zastąpiono układami lampowymi. Miniaturyzacji nie osiągnięto. Na przykład reprezentant tej generacji ENIAC ważył 30 ton, ale szybkość działania komputera zwiększono znacznie. W czasie jednej sekundy wykonywał około 5000 operacji dodawania. Jednak moc pobierał ogromną (150 KW), co w połączeniu z ograniczoną żywotnością lamp stanowiło o tym, iż lepiej tych komputerów (gdy pracowały dobrze) nie wyłączać. Ponowne bowiem włączenie niosło z sobą ryzyko (niemal pewność), że parę lamp w tym momencie dokona swego żywota.

Rewolucyjną zmianę w technice cyfrowej spowodowały dopiero półprzewodniki — osławiony (dziś już czterdziestoletni) tranzystor, dioda i inne podobne elementy. Zbudowane z nich komputery — **druga generacja** pracują zapewne do dzisiaj — ODRA 1304. Urządzenia te wykonywały niemal 30 000 prostych operacji arytmetycznych w czasie jednej sekundy. Wielkość ta w połączeniu ze stosunkowo małymi gabarytami oraz dużą niezawodnością pozwoliła by sprzęt tej generacji wysłano w kosmos. To zastosowanie maszyn cyfrowych przyczyniło się do dalszej **miniaturyzacji sprzętu**, tym samym zmniejszenia **poboru mocy, zwiększenia niezawodności przy jednoczesnym zwiększaniu szybkości działania**. Są to (były) w zasadzie cztery podstawowe kryteria pozwalające w miarę jednoznacznie klasyfikować komputer do określonej generacji.

Kolejny etap doskonalenia sprzętu informatycznego — **trzecia generacja** — komputery budowane z układów scalonych. Motorem tego postępu był (i jest) nadal kosmos. Sputniki wymagały bowiem coraz doskonalszych komputerów o małym ciężarze, lepszej niezawodności, a źródłem prądu pozostały nadal mało wydajne baterie słoneczne.

Spowodowało to, iż „upakowano” wiele elementów w jeden układ scalony, tak zwaną „kość”. W zależności od stopnia upakowania („integracji”) mieści w sobie ona różną liczbę elementów (zazwyczaj są to: tranzystory, diody, oporniki, kondensatory). Układ małej skali integracji (SSI) zawiera do 100 elementów, średniej (MSI) do 1000 elementów, dużej skali integracji (LSI) ma aż do 100 tysięcy elementów, a nie jest to ostatnie słowo „upakowania”).

Przykładem komputerów III generacji są nasze ODRY, 1305 i 1325. Te nasze „staruszki” stanowią jeszcze dzisiaj (niestety) podstawowy potencjał obliczeniowy zainstalowany w naszym kraju. Sądzę, że stwierdzenie to wywoła u części czytelników oburzenie — przecież mikrokomputery. Owszem są i to w dużej ilości (biorąc na wagę), ale to już odrębny problem...

Faktem jest natomiast, że mikrokomputery reprezentują już **IV generację**. Wykorzystano w nich dwa charakterystyczne dla tej generacji typy układów, które poprawiają wszystkie parametry sprzętu informatycznego, są to: **mikroprocesor i pamięć półprzewodnikowa**.

Mikroprocesor, to kolejny krok integracji, tym razem nie elementów, a całych układów. Jest to już bardzo duża skala integracji (VLSI). Współczesne mikroprocesory (jeden układ) zbudowane są już z milionów elementów, a szybkość wykonywanych przez nie mikrooperacji mierzona jest w milionach. Ta sama technologia pozwala nam budować pamięci komputerów z układów scalonych (jeszcze niedawno potrzebne były do tego rdzenie ferrytowe), które mają pojemności sięgające setek tysięcy znaków w jednej obudowie.

Czwarta generacja stworzyła warunki do budowy mikrokomputerów przeznaczonych do domu, wysyłanych na Marsa, a także stawianych na biurkach. Budowane są w tym samym czasie superkomputery, nazywane czasami hiperkomputerami (jest ich na świecie mniej niż 300), które dzięki specjalnej metodzie przetwarzania informacji (procesory macierzowe) wykonują np.: dodawanie w czasie, gdy światło przebiega w latarce drogę od żarówki do chroniącego ją szkielek (3 cm!).

Wydawać by się mogło, że osiągnięto szczyt technologiczny doskonalenia komputerów (bariera szybkości światła), prowadzone są jednak nadal badania, słyszymy o komputerach optycznych, pracujących w warunkach nadprzewodnictwa, hybrydach łączących urządzenia z żywymi tkankami.

Sądzę jednak, że najbardziej obiecującym kierunkiem doskonalenia komputerów jest „zrównoleglenie” operacji wykonywanych przez maszynę — na wzór najdoskonalszego modelu, jakim jest mózg ludzki.

Wracając do naszych generacji — kolejną jest **piąta**. W tym względzie zdania uczonych są podzielone. Mało kto kwestionuje fakt, że nie jest to kolejny etap doskonalenia sprzętowych parametrów komputera, a doskonalenie oprogramowania. Wydaje się, że piąta generacja komputerów to „**życzliwość maszyn**”. Zdają sobie sprawę, że jest to dosyć ryzykowne określenie, kryje jednak ono w sobie idee sztucznej inteligencji, ekspert-systemów i analizy języka naturalnego (mówionego i pisanego). Wszystko to zmierza do zaspokojenia naszych oczekiwań już w momencie kupienia komputera — bez względu na planowaną sferę jego zastosowań.

Komputer piątej generacji przede wszystkim z nieograniczoną cierpliwością, także przystępnie, nauczy nas korzystać z jego możliwości, są one ogromne. Dziennikarzowi po-

magają w redagowaniu tekstów, (ortografia, synonimy, tłumaczenia itp.). Maszyna staje się wysokiej klasy partnerem w fachowej dyskusji na wybrane, aktualne tematy. O aktualizację wiedzy naszego komputera dbają inni, nasza maszyna korzysta bowiem (via sieć telefoniczna) z innych baz danych, rozsianych po całym świecie, ale to już odrębny temat naszych podstaw informatyki.

Włodzimierz GOGOLEK

Tezy do pierwszych zajęć z podstaw informatyki

Temat: WPROWADZENIE DO PODSTAW INFORMATYKI

1. Omówienie przykładów ilustrujących potrzeby (i możliwości) praktycznych zastosowań komputerów

- znaczenie szybkości przetwarzania informacji i pojemności pamięci komputerów;
- możliwości wykorzystywania komputerów w różnych dziedzinach życia,
- cyfrowa reprezentacja obrazu — wstęp do systemu dwójkowego, pojęcie „pojemność pamięci”.

2. Podstawowe etapy rozwoju informatyki

Generacje komputerów. System binarny a elementy pamiętające (nie logiczne) np. w zerowej generacji. Pojęcie „szybkość” komputera. Podstawowe parametry charakteryzujące generacje (miniaturyzacja, szybkość, moc pobierana, niezawodność).

Propozycja materiałów pomocniczych:

1. Na początku był ENIAC, „IKS” nr 6/86
2. Krótka historia mikroprocesorów, „IKS” nr 3/86
3. Komputer i stylistyczne linie papilarnie, „IKS”, nr 4/87.
4. Komputer u kompozytora, „Komputer”, nr 2, 3, 4/87
5. Timeworks Publisher, „Komputer”, 5/88.
6. Kosmos, medycyna i komputer, „IKS” nr 12/87
7. Mona Lisa i komputer, „IKS” nr 11/87.

Włodzimierz GOGOLEK

UWAGA.

Ilustracje „obrazów komputerowych” mogliśmy wykonać dzięki uprzejmości Dyrektora Zakładu Projektowo-Wdrożeniowego ATM.N, który udostępnił nam, zbudowany tam skaner sygnału telewizyjnego. Serdecznie dziękujemy.

Nuty z komputera (4)

Dla ciekawskich

Krzysztof POŹNIAK, Jarosław ZIEMBICKI

Zbliżamy się do końca pracy nad edytorem sekwencera. Nie oznacza to niestety zakończenia naszego programu. Pozostaje praktycznie nie omówiony do tej pory problem współpracy z instrumentami muzycznymi. O tym w dalszej części naszego artykułu.

Zamieszczony obecnie program wypełnia ostatnią „dziurę” w oprogramowaniu edytora sekwencera. Jest to edytor utworu. Dzięki niemu z utworzonych do tej pory taktów będziemy mogli budować całe kompozycje, korygować ich struktury itp. Od strony programowej utwór jest ciągiem taktów wykonywanych w ściśle określonej kolejności. Przewidzieliśmy stosowanie pętli (podobnych do pętli FOR—NEXT w Basicu) pozwalających objąć „klamrą” powtarzające się fragmenty utworu.

Przypominamy: zamieszczony wydruk należy dołączyć do opublikowanych poprzednio programów. Istniejące już linie o tych samych numerach należy zamienić na obecnie prezentowane.

EDYTOR GŁÓWNY — nowe funkcje

Klawisz **W** — wywołanie utworu zapisanego w pamięci (zmiennnej tekstowej u\$)

parametry: 1) numer utworu,

Klawisz **U** — utworzenie utworu — wywołanie edytora utworu

parametry: 1) nazwa utworu (zasady takie same, jak przy tworzeniu nazwy taktu)

Klawisz **K** — kasowanie utworu zapisanego w pamięci (zmiennnej tekstowej u\$)

parametry: 1) numer utworu

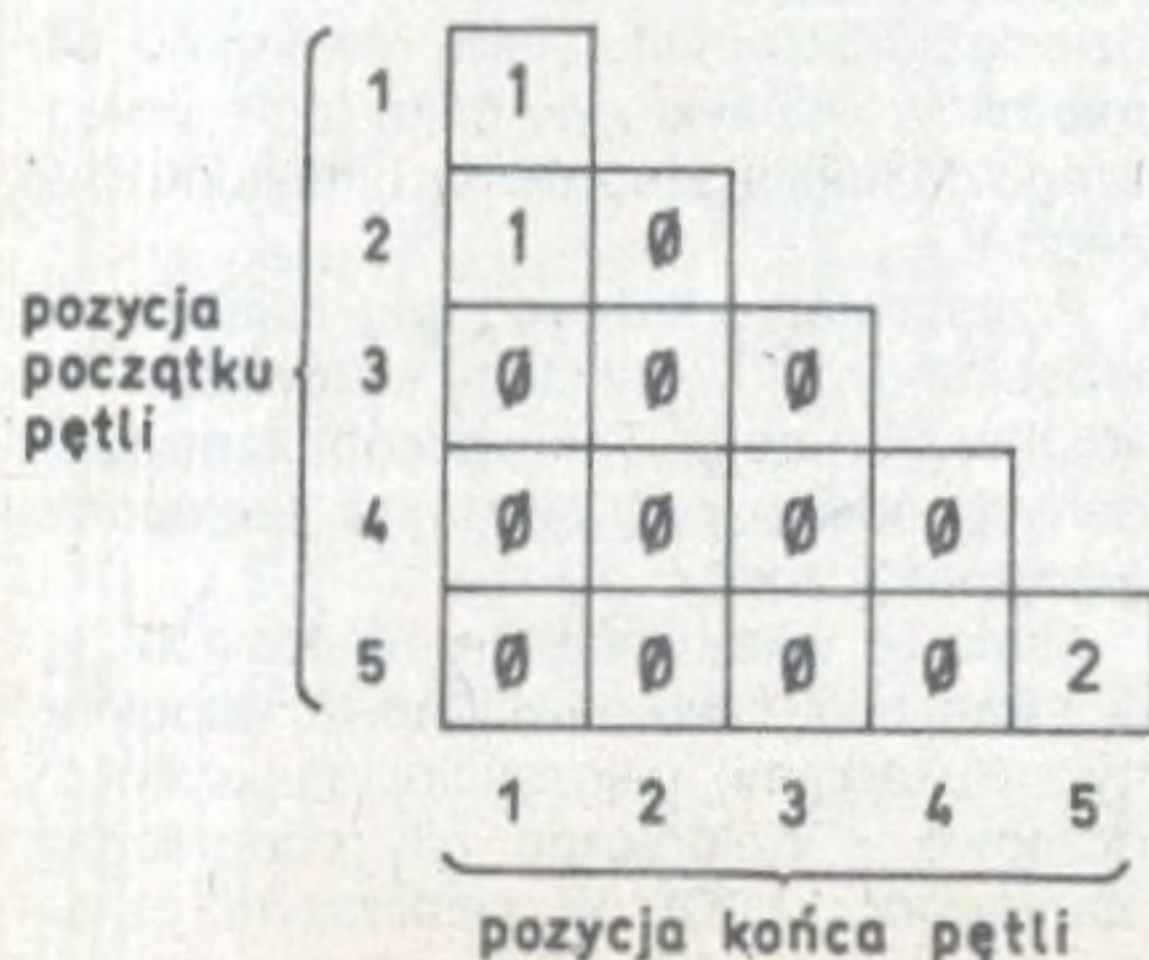
Klawisz **L** — lista utworów

parametry: brak

EDYTOR UTWORU

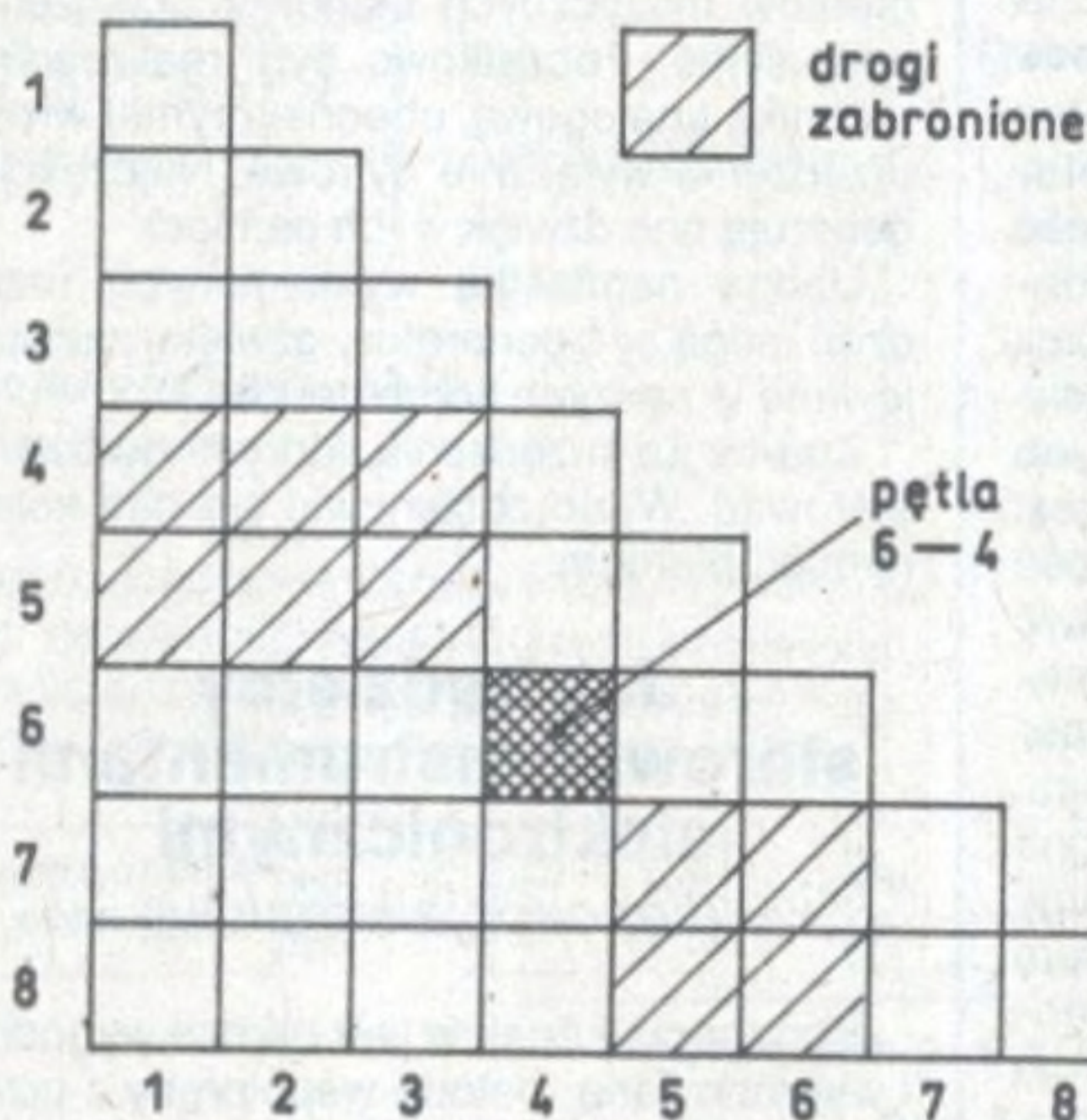
Klawisz **d** — dołączenie taktu do utworu
parametry: 1) numer pozycji, na którą będzie wprowadzony takt, 2) numer taktu,

wartość 0 — brak pętli



Rys. 1

uwaga: wprowadzenie taktu na istniejącą już pozycję powoduje przesunięcie taktów umieszczonych poniżej o jedną pozycję w dół. Podanie pozycji o jedną większą od już istniejących pozwala na umieszczenie nowego taktu na końcu utworu.



Rys. 2

Klawisz **I** — wywołanie listy taktów
parametry: brak

Klawisz **w** — wywołanie taktu z pamięci — wejście do edytora taktu

parametry: 1) numer taktu

Klawisz **k** — kasowanie taktu w utworze
parametry: 1) numer pozycji w utworze, na której znajduje się dany takt

uwaga: następuje kasowanie wszystkich pętli, które bezpośrednio zaczynają się lub kończą na danej pozycji.

Klawisz **p** — utworzenie pętli powtórzeń
parametry: 1) numer pozycji, od której rozpocznie się pętla powtórzenia,

2) numer pozycji, przed którą pętla się zakończy,

3) ilość powtórzeń taktów w pętli (maksym. do 99),

uwaga: pętle nie mogą się wzajemnie przecinać, mogą się natomiast zagłębiać w sobie. Prawidłowo poprowadzone pętle prezentuje rysunek nr 3.

Klawisz **k** — kasowanie pętli powtórzeń
parametry: 1) numer pozycji początku pętli

2) numer pozycji końca pętli

Klawisz **S** — wyświetlanie aktualnej struktury utworu

parametry: brak

Klawisz **z** — zapis utworu do pamięci (zmiennnej tekstowej u\$)

parametry: brak

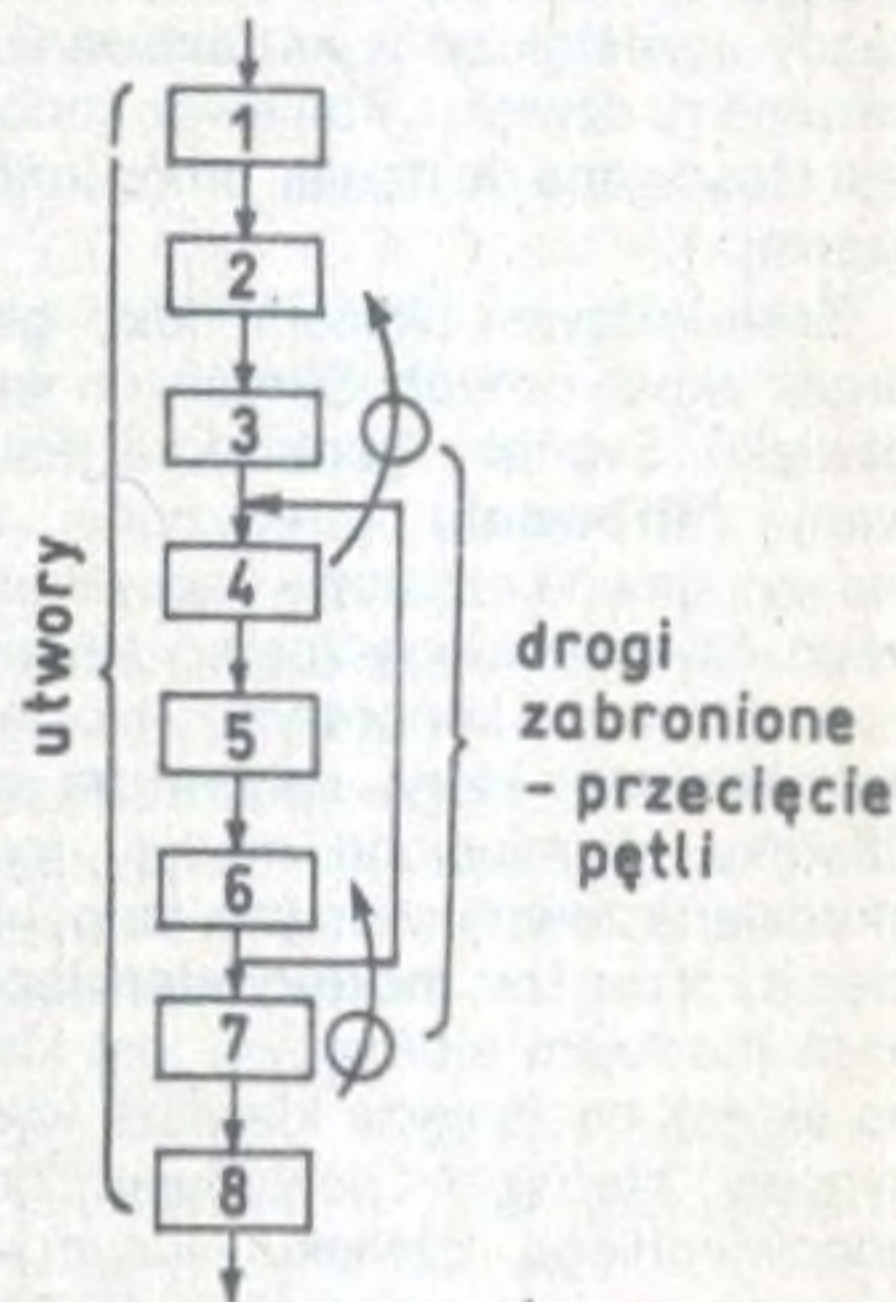
uwaga — stara wersja danego utworu zostaje skasowana.

Klawisz **e** — powrót do edytora głównego.

Na rysunku nr 3 przedstawiliśmy przykładowy utwór utworzony za pomocą edytora utworu.

Utwór aktualnie tworzony za pomocą edytora utworu jest trzymany w zmiennej tekstowej r. Strukturę utworu wygodniej jest przedstawić za pomocą „trójkąta pętli”. Na rysunku nr 1 przedstawiliśmy strukturę utworu z rysunku nr 3. Proponujemy, aby czytelnicy sami znaleźli powiązania obu rysunków.

Dzięki takiej formie zapisu bardzo wygodnie kontroluje się warunki nieprzecinania się pętli. Każda nowo wprowadzana pętla



podlega w programie tego typu kontroli. Rysunek nr 2 przedstawia sytuację wprowadzenia pętli od pozycji nr 6 do pozycji nr 4 i warunki jej umieszczenia. Zakreskowane pola muszą posiadać wartość 0.

tytuł utworu:		Utwór-test
1	Zwrotka	1
2	refren	1
3	solo	
4	Zwrotka	
5	refren	2

EDYTOR UTWORU

Pora zastanowić się, w jaki sposób połączymy komputer z elektronicznym instrumentem muzycznym i co za jego pomocą będziemy sterować. Zanim weźmiemy się do solidnej pracy konstruktorskiej warto poznać trochę historię i budowę syntezatorów.

Krótką historia

Historia elektronicznych urządzeń muzycznych zaczęła się na przełomie XIX i XX wieku. Kiedy to zbudowano pierwsze organy elektromechaniczne. W pewnym sensie były one powtórzeniem koncepcji organów piszczałkowych, ponieważ każdy klawisz miał przydzielony własny generator drgań elek-

trycznych o odpowiedniej częstotliwości. Koncepcja ta z niewielkimi zmianami, dominowała w konstrukcjach instrumentów przez kilkadziesiąt lat. Konsekwencją stosowania dużej liczby generatorów o dosyć prostej konstrukcji było uzyskiwanie mało zróżnicowanych brzmień.

Pierwsze próby rozbudowy syntezatorów umożliwiające kształtowanie dźwięku z dużą dowolnością zaczęły się już w latach pięćdziesiątych, ale dopiero konstrukcje **dr. Roberta Mooga**, bazujące na elementach półprzewodnikowych, spowodowały rewolucję w elektronice muzycznej. Syntezator Mooga składał się z modułów, z których każdy uczestniczył w kształtowaniu innego parametru dźwięku. Ponieważ koncepcja ta jest stosowana do dzisiaj, omówimy ją nieco szerzej.

Zasadniczym blokiem jest **generator** drgań akustycznych. Określa on **wysokość** dźwięku. Sygnał z generatora jest poddawany **filtrowaniu** (precyzyjnej regulacji barwy): pewne częstotliwości składowe danego dźwięku mogą zostać stłumione lub wzmacnione, końcowym blokiem jest **wzmacniacz**, który wpływa na głośność dźwięku. Wymienione moduły mogą być sterowane zewnątrz (np. za pomocą napięcia) przez tzw. **moduły sterujące**. Typowym modułem sterującym jest klawiatura: na skutek naciśnięcia klawisza wysyła ona sygnały sterujące powodując generację odpowiedniego dźwięku. Innym źródłem sterującym jest generator drgań wolnozmennych (LFO), który umożliwia wibrację dźwięku (uzyskujemy efekty typu: vibrato, tremolo). Generator obwiedni kontroluje amplitudę dźwięku w czasie: ustala szybkość narastania, trwania i wybrzmiewania dźwięku. Uniwersalnym modułem jest generator szumu. Może on być źródłem dźwięku, może także tym dźwiękiem sterować wprowadzając przypadkową modulację amplitudy lub częstotliwości. Jest bardzo przydatny przy imitacji naturalnych źródeł dźwięku.

Omówiony układ stanowi tor syntezy pojedynczego dźwięku, stąd pierwsze (najprostsze) syntezatory były jednogłosowe. Okazało się, że podłączenie do każdego klawisza po jednym torze syntezy jest bardzo trudne technicznie i nieekonomiczne. Przeciwnie tak naprawdę podczas gry uruchamiamy nie więcej jak kilka (w wyjątkowych wypadkach kilkanaście) klawiszy

naraz. Rozsądnym wyjściem jest zastosowanie sterownika, który na podstawie stanu klawiatury uruchamiałby odpowiednią ilość torów syntezy. Na skonstruowanie takiego sterownika trzeba było poczekać do lat siedemdziesiątych. Wtedy pojawił się pierwszy wielogłosowy syntezator — **Polymoog** — zawierający wspomniany sterownik klawiatury. Sterownikiem był zazwyczaj mikroprocesor. Przeglądał on stan klawiatury i za pomocą przetworników cyfrowo/analogowych sterował analogowymi modułami dźwiękowymi. Wkrótce pojawiły się syntezatory całkowicie cyfrowe o bardzo różnorodnych koncepcjach syntezy dźwięku.

Odrębną grupę elektronicznych instrumentów muzycznych stanowią urządzenia perkusyjne. Początkowo były realizowane techniką analogową, obecnie prymat wiodą urządzenia wyłącznie cyfrowe. Najczęściej generują one dźwięk w ich pamięci.

Ubogą namiastką wymienionych urządzeń mogą być generatory dźwięku zainstalowane w naszych komputerach.

Znamy już urządzenia, którymi będziemy sterować. Warto zastanowić się nad kolejnym problemem:

Jak będziemy sterować instrumentami elektronicznymi (koncepcja kanałów)

Koncepcja kanałów jest bardzo wygodną i wypróbowaną metodą współpracy z urządzeniami zewnętrznymi. Zastanówmy się w tym momencie nad prostym przykładem: posiadamy w pamięci zbiór tekstowy (np. program). Możemy wyświetlić go na ekranie, wydrukować na drukarce, zapisać na dysku, nagrać na magnetofonie, wręcz przesłać do innego komputera. Ponieważ każdemu z tych urządzeń przypisany jest określony numer kanału, stąd o tym, gdzie treść zbioru zostanie przekazana decyduje podany przez użytkownika numer kanału. Na tej podstawie zostaje uruchomiona odpowiednia procedura transmitująca. Uniwersalność tej metody jest oczywista:

po pierwsze — możemy dowolnie przypisywać urządzenia do kanałów w zależności od ich ilości

po drugie — możemy współpracować z dowolnymi urządzeniami; wystarczy dopi-

sać procedurę transmisyjną i przypisać ją odpowiedniemu kanałowi.

Dla sekwencera urządzeniami zewnętrznymi będą instrumenty elektroniczne. Każdy z nich wymaga odpowiedniego standardu transmisji i formatu przesyłanych danych. Ten problem rozwiążą procedury transmitujące. Użytkownik będzie je dobierał w zależności od typów wykorzystywanych instrumentów. Dobór odpowiedniego kanału nastąpi automatycznie, ponieważ każda ścieżka w takcie jest przyporządkowana konkretnemu instrumentowi.

Sposoby przesyłania danych

W programie wykorzystamy trzy najbardziej rozpowszechnione sposoby transmisji:

a) *transmisja wewnętrzna* — używana przy przesyłaniu danych w obrębie jednego obszaru pamięci. Wykorzystamy ją sterując generatorami w komputerze.

b) *transmisja równoległa* — pozwala na jednoczesne wysłanie całego bajtu. Często jest wykorzystywana do sterowania instrumentami perkusji elektronicznych.

c) *transmisja szeregową* — bit po bicie. Standard wykorzystywany przez system MIDI.

System MIDI

MIDI to skrót angielskiej nazwy: Musical Instrument Digital Interface (cyfrowy interfejs instrumentów muzycznych). Oznacza ogólnie przyjęty w świecie standard przesyłania informacji między urządzeniami muzycznymi. Pozwala na łączenie instrumentów w większe zestawy, umożliwia współpracę z komputerem. Łączenie instrumentów w jednolitą całość zwiększa możliwości brzmieniowe, pozwala na grę wielogłosową. W epoce syntezatorów analogowych eksperymenty tego typu były bardzo kłopotliwe w realizacji i zmuszały do prowadzenia licznych przewodów sterujących między instrumentami.

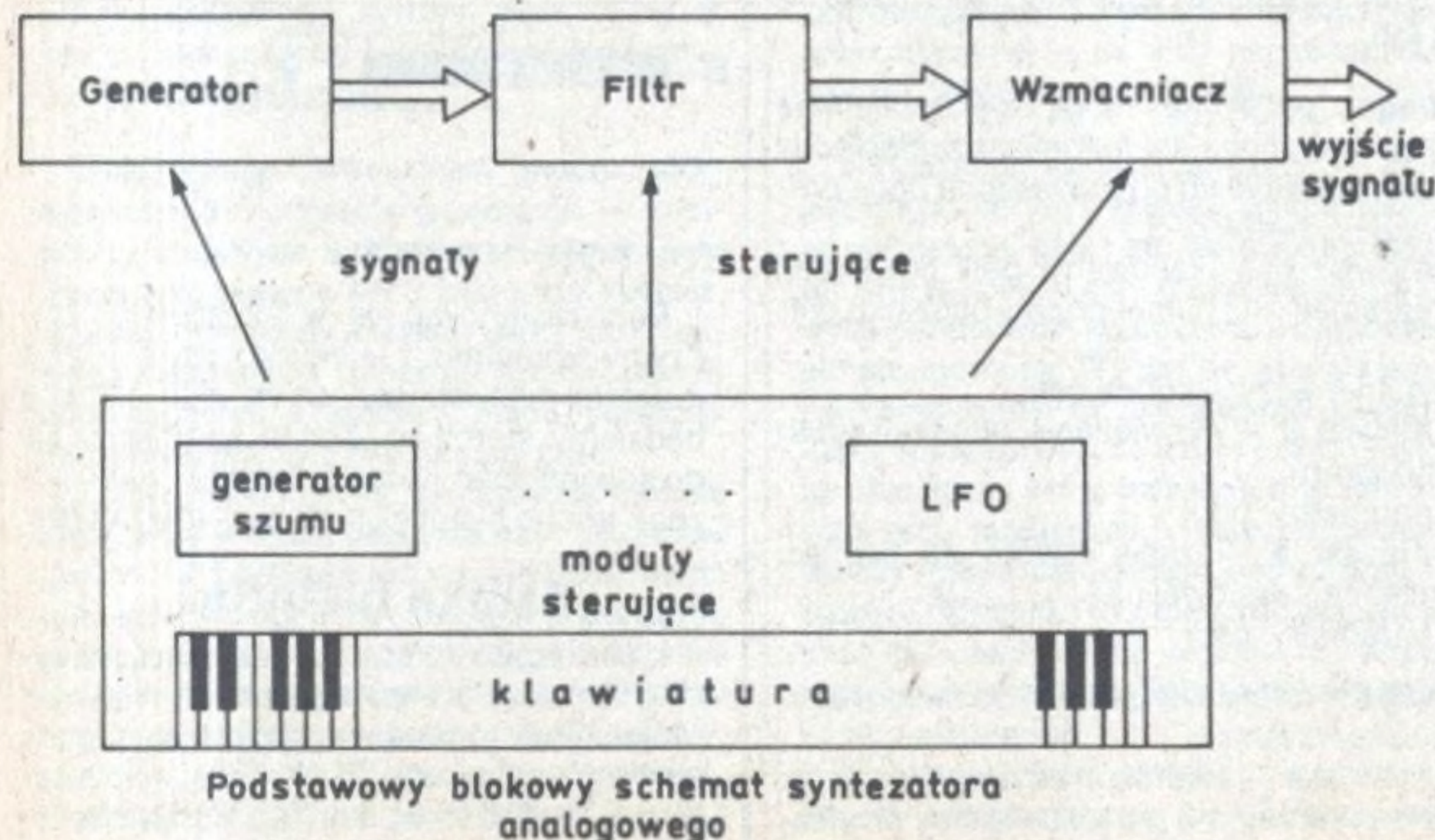
Standard MIDI zaproponowała firma *Sequential Circuits* i jest on dzisiaj jedynym tego typu standardem liczącym się na świecie.

Pojedyncze złącze MIDI składa się z nadajnika, linii łączącej (dwa przewody) i odbiornika. Stanem biernym linii jest brak przepływu prądu — oznacza to logiczną jedynkę. Logiczne zero jest prezentowane przez płynący w linii prąd (standardowo 5 mA).

Najmniejszą jednostką informacji jest bajt. Transmisja bajtu polega na wysłaniu kolejno dziesięciu bitów: bitu startu równego 0, ośmiu bitów przesyłanego bajtu (od najmłodszego do najstarszego) oraz bitu stopu równego 1.

Przestawienie każdego bitu trwa 32 mikrosekundy bitów. Bit stopu wprowadza linię MIDI w stan bajtu. Taki sposób przesyłania danych nosi nazwę *transmisji szeregowej asynchronicznej*.

Mimo że wiele komputerów posiada złącza transmisji szeregową (np. do współpracy z drukarkami), najczęściej ich parametry czasowe i elektryczne nie odpowiadają standardowi MIDI. W najbliższym czasie opublikujemy schemat prostego nadajnika MIDI, który będzie można podłączyć do



Rys. 4

komputera niezależnie od pozostałych jego interfejsów.

Podstawowe konfiguracje urządzeń w systemie MIDI

Do łączenia urządzeń standard przewiduje trzy gniazda:



Forma transmisji MIDI dla pojedynczego bajtu

Rys. 5

- 1) **MIDI IN** — wejście do przyjmowania danych przesyłanych z innych urządzeń, innych urządzeń, wysyłające dane z tego urządzenia,
- 2) **MIDI OUT** — wyjście z urządzenia wysyłające dane z tego urządzenia,
- 3) **MIDI THRU** — („przelotka”), wyjście odwzorowujące dane przychodzące do gniazda MIDI IN pozwala na szeregowe łączenie instrumentów w większe zestawy.

- 2) jedno urządzenie sterujące (np. syntezator, sekwencer, komputer) steruje kilkoma urządzeniami odbiorczymi (mogą być inne syntezatory, ekspandery, itp.).



urządzenia sterowane

Rys. 7

Urządzenia podporządkowane łączymy szeregowo za pomocą gniazd MIDI IN — MIDI THRU.

Komunikaty transmisji MIDI

informacje podstawowe

Wszystkie informacje w systemie MIDI są przesyłane w formie **komunikatów**. Najważniejsze dla naszego programu komunikaty przedstawia tabela 1.

Każde z urządzeń w systemie MIDI posiada przypisany mu **numer kanału**. Stwa-

rza to możliwość wybiórczego sterowania instrumentami. Istnieją także komunikaty nie wymagające numeru kanału — dotyczą wszystkich urządzeń w systemie.

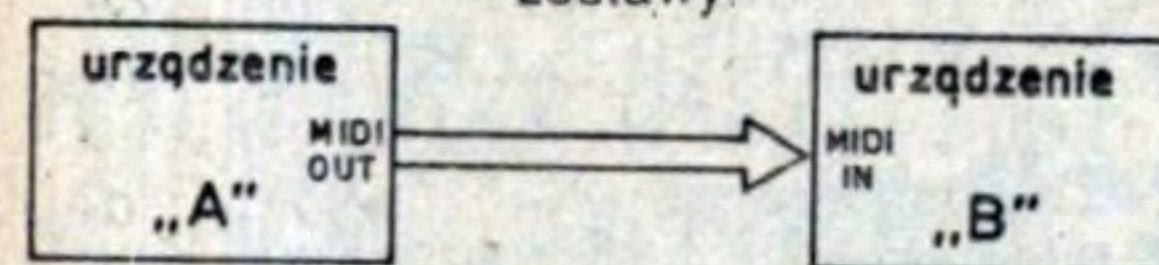
wyjaśnienia:

- 1) **nnnn** — jest numerem kanału MIDI, dla którego komunikat jest przeznaczony: (zakres: 1—16). Np.: nnnn = 0000 — kanał 1, nnnn = 1111 — kanał 16,
- 2) **kkkkkkk** — jest binarnym zapisem numeru klawisza (zakres: 0—127). Klawisze są ponumerowane w kolejności od tonów niskich do wysokich. Np. typowa klawiatura pięciooktawowa (od C do c⁴) obejmuje klawisze o numerach od 36 do 96.

- 3) **ddddddd** — dotyczy siły uderzenia w klawisz (zakres: 0—127). Jest to parametr wykorzystywany przez syntezatory z tzw. klawiaturą dynamiczną. Pozostałe syntezatory ignorują ten parametr ustawiając maksymalną dynamikę (ddddddd = 1111111) lub jej połowę (ddddddd = 1000000),

- 4) **bbbbbbb** — numer brzmienia (zakres: 0—127). Jeżeli instrument posiada mniejszą ilość brzmień, to bierze pod uwagę odpowiednią ilość młodszych bitów wartości bbbbbbb.

- 1) **MIDI IN** — wejście do przyjmowania danych przesyłanych z innych urządzeń, innych urządzeń, wysyłające dane z tego urządzenia,
- 2) **MIDI OUT** — wyjście z urządzenia wysyłające dane z tego urządzenia,
- 3) **MIDI THRU** — („przelotka”), wyjście odwzorowujące dane przychodzące do gniazda MIDI IN pozwala na szeregowe łączenie instrumentów w większe zestawy.



Rys. 6

Przykłady połączeń:

- 1) najczęściej spotykany zestaw składa się z dwóch urządzeń: nadajnika (urządzenia A) i odbiornika (urządzenia B). W tym wypadku wyjście MIDI OUT urządzenia A łączymy z wejściem MIDI IN urządzenia B.

```

9 DEF FN e(y,x)=CODE r$(y-1)
*y/2+x): DEF FN u(y)=256*CODE u$(
(y+1)+CODE u$(y)
55 LET u$=CHR$(0)
1001 LET it=CODE m$(1): LET iu=C
ODE u$(1)
1010 GO SUB 9198: PRINT AT 19,0;
"EDYTOR GLOWNY"
1013 IF x$="U" AND iu>0 THEN GO
SUB 1165: GO TO 1005
1015 IF x$="U" AND it>0 THEN GO
SUB 3000: GO TO 1005
1025 IF x$="K" AND iu>0 THEN GO
SUB 1135: GO TO 1010
1032 IF x$="L" AND iu>0 THEN GO
SUB 1190: GO TO 1010
1122 LET z$="podaj nr utworu:":
LET md=iu: LET q=0: GO TO 9198
1131 LET zapis=0: IF q=1 THEN GO
SUB 8840: RETURN
1435 GO SUB 1122: IF q=0 THEN RE
TURN
1136 LET v=q: GO TO 3820
1165 GO SUB 1122: IF q=0 THEN RE
TURN
1166 LET nu=q: LET l=FN u(2*q):
LET itu=CODE u$(l): LET b$=u$(l+

```

```

1 TO l+10+itu): LET r$=u$(l+11+i
tu TO l+10+itu+itu*(itu+1)/2): G
O TO 3010
1190 CLS: PRINT "nr nazwa
it": PRINT
1191 FOR t=1 TO iu: LET n=FN u(2
*t): PRINT t;TAB 3; INVERSE 1;u$(
(n+1 TO n+10); INVERSE 0;TAB 17;
CODE u$(n)
1193 NEXT t: RETURN
1913 PRINT TAB 5;"U - wywołanie
utworu"
1915 PRINT TAB 5;"U - utworzenie
utworu"
1919 PRINT TAB 5;"K - kasowanie
utworu"
1933 PRINT TAB 5;"L - lista utwo
row"
3000 REM Edytor utworu
3005 LET k=1: GO SUB 3800: IF k<
=iu OR q$="" THEN RETURN
3006 LET b$=q$: LET nu=iu+1
3007 LET itu=0: LET r$=""
3010 GO SUB 3800
3020 GO SUB 9198: PRINT AT 19,0;
"EDYTOR UTWORU"
3025 LET x$=INKEY$: IF x$="" THE
N GO TO 3025

```

```

3030 IF x$="d" AND itu<6 THEN GO
SUB 3100: GO TO 3020
3035 IF x$="t" THEN GO SUB 1180:
GO TO 3020
3040 IF x$="s" THEN GO SUB 3900:
GO TO 3020
3045 IF x$="p" AND itu>0 THEN GO
SUB 3200: GO TO 3020
3050 IF x$="k" AND itu>0 THEN GO
SUB 3300: GO TO 3020
3055 IF x$="K" AND itu>0 THEN GO
SUB 3400: GO TO 3020
3060 IF x$="z" AND itu>0 THEN GO
TO 3500
3065 IF x$="e" THEN GO TO 3600
3070 IF x$="w" THEN GO SUB 1115:
GO SUB 3900: GO TO 3020
3090 GO TO 3025
3100 LET z$="dol. taktu na poz.
nr": LET md=itu+1: LET q=0: GO S
UB 9190: IF q=0 THEN RETURN
3115 LET k=q: GO SUB 1120: IF q=
0 THEN RETURN
3120 LET z$="": FOR l=1 TO k: LE
T z$=z$+CHR$ 0: NEXT l: LET j=k*
(k-1)/2+1: LET r$=r$( TO j-1)+z$
+r$(j TO )
3121 FOR l=k+1 TO itu+1: LET j=l
*(l-1)/2+k: LET r$=r$( TO j-1)+C
HR$ 0+r$(j TO ): NEXT l: LET b$=
b$( TO 9+k)+CHR$ q+b$(10+k TO ):
LET itu=itu+1: GO TO 3900
3200 LET z$="powt. od pozycji nr
": LET md=itu: LET q=0: GO SUB 9
190: IF q=0 THEN RETURN
3210 LET k=q: LET z$="powt. do p
ozycji nr": LET md=k: LET q=0: G
O SUB 9190: IF q=0 THEN RETURN
3220 FOR y=q TO k-1: FOR x=1 TO
q-1: IF FN e(y,x)<>0 THEN GO TO
3250
3225 NEXT x: NEXT y
3230 FOR y=k+1 TO itu: FOR x=q+1
TO k: IF FN e(y,x)<>0 THEN GO T
O 3250
3235 NEXT x: NEXT y
3240 LET l=q: LET z$="ile powtor
zen": LET q=0: LET md=99: GO SUB
9190: IF q=0 THEN RETURN
3245 LET r$((k-1)*k/2+l)=CHR$ q:
GO TO 3900
3250 PRINT AT 19,0; FLASH 1;" BL
AD STRUKTURALNY - Enter ":
PAUSE 1: PAUSE 0: RETURN
3300 LET z$="kas. powt. od poz.
nr": LET md=itu: LET q=0: GO SUB
9190: IF q=0 THEN RETURN
3310 LET k=q: LET z$="kas. powt.
do poz. nr": LET md=k: LET q=0:
GO SUB 9190: IF q=0 THEN RETURN
3320 LET r$((k-1)*k/2+q)=CHR$ 0:
GO TO 3900
3400 LET z$="kasowanie pozycji n
r": LET md=itu: LET q=0: GO SUB
9190: IF q=0 THEN RETURN
3410 LET k=(q-1)*q/2: LET r$=r$(
TO k)+r$(k+q+1 TO ): FOR l=1 TO
itu-q: LET k=q+l: LET k=(k-1)*k
/2-l: LET r$=r$( TO k)+r$(k+2 TO
): NEXT l
3420 LET b$=b$( TO 9+q)+b$(11+q
TO ): LET itu=itu-1: GO TO 3900
3500 LET q=1: LET z$="powrot z z
apisem? t=1:": GO SUB 9190: IF q
=0 THEN GO SUB 9198: GO TO 3010
3502 LET v=nu: IF v<=iu THEN GO
SUB 3820
3505 GO TO 3810
3600 LET z$="powrot bez zapisu 1
-tak": LET md=1: LET q=1: GO SUB
9190: IF q=1 THEN RETURN
3610 GO TO 3020
3800 REM Wskazanie nazwy utworu
3802 LET md=10: LET z$="nazwa ut
woru: ": LET q$="": GO SUB 8980:
IF q$="" THEN RETURN

```

```

3805 FOR k=1 TO iu: LET l=FN U(2
*k)+1: IF U$(l TO l+9)=q$ THEN G
O TO 8979
3807 NEXT k: RETURN
3810 REM Zapis utworu
3812 LET b$=CHR$ itu+b$
3814 IF v>iu THEN FOR l=2 TO 2*i
U STEP 2: LET U$(l TO l+1)=FN z$
(FN U(l)+2): NEXT l: LET U$=U$(
TO 2*iu+1)+FN z$(LEN U$+3)+U$(2*
iu+2 TO )+b$+r$: GO TO 3817
3815 LET n=FN U(2*v): LET k=LEN
b$+LEN r$: FOR l=2 TO 2*iu STEP
2: LET U$(l TO l+1)=FN z$(FN U(l
)+k*(l)=2*v)+2): NEXT l
3816 LET U$=U$( TO 2*v-1)+FN z$(
n+2)+U$(2*v TO n-1)+b$+r$+U$(n T
O )
3817 LET iu=iu+1: LET U$(1)=CHR$
iu: RETURN
3820 REM Kasowanie utworu
3822 LET u=FN U(2*v): LET ur=FN
U(2+2*v)*(v<iu)+(LEN U$+1)*(v=iu
)-FN U(2*v): FOR l=2 TO 2*iu STE
P 2: LET U$(l TO l+1)=FN z$(FN U
(l)-ur*(l)=(2+2*v))-2): NEXT l
3824 LET U$=U$( TO 2*v-1)+U$(2+2
*v TO u-1)+U$(u+ur TO )
3825 LET iu=iu-1: LET U$(1)=CHR$
iu: RETURN
3830 REM Przebieg tekstu utworu
3831 LET v=1
3832 FOR v=v TO iu: LET c=FN U(2
*v): LET ai=CODE U$(c): LET p=1
3833 FOR p=p TO ai: LET ad=CODE
U$(p+10+c): IF ad>t THEN LET U$(
p+10+c)=CHR$(ad-1): GO TO 3835
3834 IF ad=t THEN GO TO 3840
3835 LET p=p+1: GO TO 3833
3837 NEXT p: LET v=v+1: GO TO 38
32
3838 NEXT v: RETURN
3840 IF ai=1 THEN GO SUB 3820: G
O TO 3832
3841 FOR kev+1 TO iu: LET l=2*k:
LET U$(l TO l+1)=FN z$(FN U(2*k
)-ai-1): NEXT k
3843 LET k=c+10+ai+(p-1)*p/2: LE
T U$=U$( TO k)+U$(k+p+1 TO ): FO
R l=1 TO ai-p: LET k=p+l: LET k=
c+10+ai+(k-1)*k/2-l: LET U$=U$(
TO k)+U$(k+2 TO ): NEXT l
3845 LET U$=U$( TO c+9+p)+U$(c+1
1+p TO ): LET ai=ai-1: LET U$(c)
=CHR$ ai: GO TO 3833
3900 REM Prezentacja struktury utworu
3910 CLS: PRINT "tytul utworu:
": INVERSE 1; b$(1 TO 10)
3920 FOR k=1 TO itu: LET l=3+FN
Z(2*CODE b$(10+k)): PRINT TAB 6;
"l"; TAB 0; k; TAB 2; INVERSE 1; m$(
l TO l+9): INVERSE 0; TAB 6; "l":
NEXT k
3930 FOR k=0 TO itu-1: FOR x=1 T
O itu-k: LET y=x+k: LET q=FN e(y
,x): IF q<>0 THEN GO SUB 3940
3935 NEXT x: NEXT k: RETURN
3940 PRINT AT 3*x-2,7;"(-----"
); FOR l=1 TO k: PRINT "----"; N
EXT l
3942 FOR l=3*x-1 TO 3*y-2: PRINT
AT l,14+3*k;"l": NEXT l
3943 PRINT AT l,13+3*k+(q<10);q
3945 PRINT AT 3*y,7;"-----";
FOR l=1 TO k: PRINT "----"; NEX
T l
3950 RETURN
8828 LET it=it-1: LET m$(1)=CHR$
it: LET w$=w$( TO t-1)+w$(t+1 T
O ): IF zapis=0 THEN GO SUB 3830
8829 RETURN
8905 GO SUB 9198: PRINT AT 19,0;
"EDYTOR POLA AKTYWNOSCI"
9025 GO SUB 9198: PRINT AT 19,0;
"EDYTOR TAKTU": GO SUB 150
9225 LET zapis=1: LET q=nt: IF q
<=it THEN LET t=q: GO SUB 8825

```

PROGRAM

58

ZX SPECTRUM

Nowe literary

J. RUBINOWICZ

```

110 REM ...
120 CLEAR 55615: INVERSE 0: BRT
GHT 0: CLS: BORDER 6: LET z$=""
NOWE LITERY": DIM u$(95): GO SU
B 230
130 POKE 23509,50
140 PRINT AT 6,0: PAPER 6;u$
150 FOR d=1 TO 13: PRINT PAPER
2: INK 6:AT 7.2+d+2;z$(d): NEXT
d
160 LET a$="ABCDEFGHIJKLMnopars
t1234567890"
170 GO SUB 220: BEEP .2,25: BEE
P .2,14: BEEP .2,2
200 IF PEEK 55001(<)124 THEN GO
SUB 240
210 PRINT AT 3,0: FOR n=32 TO
143: GO SUB 220: PRINT "": BRIG
HT 1:CHR$ n: BRIGHT 0: GO SUB 2
30: PRINT CHR$ n: BRIGHT 0:"":
NEXT n
214 LET inv=0: LET poz=(32-LEN
a$)/2: PRINT AT 16,0: INVERSE 0:
u$
215 FOR n=1 TO 8: LET inv=ABS i
nv: INVERSE inv: GO SUB 220: PAU
SE 50: BEEP .05,26: PRINT AT 19,
poz: BRIGHT 1;a$: GO SUB 230: PA
USE 50: BEEP .05,23: PRINT AT 19
,poz: BRIGHT 1;a$: LET inv=inv-1
: NEXT n
216 INPUT "tekst (max. 32 znaki
): LINE a$: IF LEN a$>32 THEN
BEEP .2,-10: GO TO 216
217 GO TO 214
220 POKE 23506,64: POKE 23507,2
16: RETURN: REM ...
230 POKE 23506,0: POKE 23507,60
: RETURN: REM ...
240 RESTORE: FOR n=55616 TO 56
383 STEP 8: LET su=0: FOR m=0 TO
7: READ bt: POKE m+n,bt: LET su
=su+bt: NEXT m: READ sm: IF sm(<)
su THEN PRINT "blad w wierszu ";
((n-55360)/8)+1000:
250 PRINT AT 7,14: BRIGHT 1:CHR
$( (n-55360)/8): NEXT n: RETURN
1032 DATA 0,0,0,0,0,0,0,0,0
1033 DATA 0,24,24,24,24,0,24,0,1
20
1034 DATA 0,102,34,0,0,0,0,0,136
1035 DATA 24,86,102,116,110,102,
99,0,640
1036 DATA 0,24,126,89,126,26,126
,24,540
1037 DATA 0,98,100,0,16,38,70,0,
330
1038 DATA 0,0,60,102,124,95,60,6
,440
1039 DATA 0,24,8,0,0,0,0,0,32
1040 DATA 0,12,24,24,24,24,12,0,
120
1041 DATA 0,48,24,24,24,24,48,0,
102
1042 DATA 0,0,102,24,126,24,102,
0,370
1043 DATA 0,0,24,24,126,24,24,0,
202
1044 DATA 0,0,0,0,0,24,24,8,55
1045 DATA 0,0,0,0,62,0,0,0,62
1046 DATA 0,0,0,0,0,24,24,0,48
1047 DATA 0,0,6,12,24,48,96,0,18
6
1048 DATA 0,60,102,110,116,102,6
0,8,552
1049 DATA 0,26,44,12,12,12,30,0,
138
1050 DATA 0,60,102,6,60,96,126,0
,450
1051 DATA 0,60,70,12,6,70,60,0,2
78
1052 DATA 0,12,28,44,75,126,12,0
,298
1053 DATA 0,126,96,124,6,70,60,0
,482
1054 DATA 0,60,96,124,102,102,60
,0,544
1055 DATA 0,126,6,12,24,48,96,0,
312
1056 DATA 0,60,102,60,102,102,60
,0,486
1057 DATA 0,60,102,102,62,6,60,0
,392

```

```

1058 DATA 0,0,24,24,0,24,24,0,96
1059 DATA 0,0,24,24,0,24,24,0,10
4
1060 DATA 0,0,12,24,48,24,12,0,1
20
1061 DATA 0,0,0,62,0,62,0,0,124
1062 DATA 0,0,48,24,12,24,48,0,1
56
1063 DATA 0,60,102,12,24,0,24,0,
222
1064 DATA 0,0,60,6,62,102,62,3,2
95
1065 DATA 0,24,60,102,102,126,10
2,0,516
1066 DATA 0,124,102,124,102,102,
124,0,678
1067 DATA 0,60,102,96,96,102,60,
0,516
1068 DATA 0,120,106,102,102,106,
120,0,660
1069 DATA 0,126,96,124,96,96,126
,0,664
1070 DATA 0,126,96,124,96,96,96,
0,634
1071 DATA 0,60,102,96,110,102,60
,0,530
1072 DATA 0,102,102,126,102,102,
102,0,636
1073 DATA 0,126,24,24,24,24,126,
0,348
1074 DATA 0,6,6,6,102,102,60,0,2
82
1075 DATA 0,102,106,120,106,102,
99,0,532
1076 DATA 0,96,96,96,96,96,126,0
,605
1077 DATA 0,66,102,126,102,102,1
02,0,600
1078 DATA 0,70,102,116,110,102,9
6,0,600
1079 DATA 0,60,102,102,102,102,6
0,8,528
1080 DATA 0,124,102,102,124,96,9
6,0,644
1081 DATA 0,60,102,102,102,102,6
0,6,534
1082 DATA 0,124,102,102,124,106,
102,0,662
1083 DATA 0,60,96,60,6,102,60,0,
324
1084 DATA 0,126,24,24,24,24,24,0
,246
1085 DATA 0,102,102,102,102,102,
60,0,570
1086 DATA 0,102,102,102,102,60,2
4,0,492
1087 DATA 0,102,102,102,102,126,
36,0,570
1088 DATA 0,102,60,24,24,60,102,
0,372
1089 DATA 0,196,102,60,24,24,24,
0,420
1090 DATA 0,126,12,24,24,48,126,
0,360
1091 DATA 0,126,96,124,96,96,126
,3,667
1092 DATA 0,24,26,28,24,56,28,0,
180
1093 DATA 0,24,60,102,102,126,10
2,3,510
1094 DATA 0,24,60,90,24,24,24,0,
246
1095 DATA 0,0,0,0,0,0,0,255,255
1096 DATA 0,96,106,120,112,96,12
6,0,658
1097 DATA 0,0,62,6,62,102,62,0,2
94
1098 DATA 0,96,96,124,102,102,12
4,0,644
1099 DATA 0,0,60,96,96,96,60,0,4
12
1100 DATA 0,6,6,62,102,102,62,0,
340
1101 DATA 0,0,60,102,124,96,60,0
,442
1102 DATA 0,28,48,56,48,48,48,0,
278
1103 DATA 0,0,62,102,102,62,6,60
,394
1104 DATA 0,96,96,124,102,102,10
2,0,522
1105 DATA 0,24,0,56,24,24,60,0,1
36

```

```

1106 DATA 0,12,0,12,12,12,106,56
,212
1107 DATA 0,96,106,120,120,106,1
02,0,554
1108 DATA 0,56,24,24,24,24,28,0,
180
1109 DATA 0,0,106,86,86,86,86,0,
452
1110 DATA 0,0,124,102,102,102,10
2,0,532
1111 DATA 0,0,60,102,102,102,60,
0,426
1112 DATA 0,0,124,102,102,124,96
,96,644
1113 DATA 0,0,62,102,102,62,6,7,
341
1114 DATA 0,0,60,102,96,96,96,0,
460
1115 DATA 0,0,60,96,60,6,124,0,3
46
1116 DATA 0,24,60,24,24,24,26,0,
164
1117 DATA 0,0,102,102,102,102,62
,0,470
1118 DATA 0,0,99,99,54,54,26,0,3
34
1119 DATA 0,0,70,86,86,86,44,0,3
70
1120 DATA 0,0,102,60,24,60,102,0
,346
1121 DATA 0,0,102,102,102,62,6,6
0,434
1122 DATA 0,0,126,12,24,48,126,0
,336
1123 DATA 6,4,60,96,96,96,60,0,4
22
1124 DATA 24,0,126,12,24,46,126,
0,360
1125 DATA 6,4,62,96,60,6,124,0,3
58
1126 DATA 24,126,12,24,24,48,126
,0,384
1127 DATA 12,6,124,102,102,102,1
02,0,552
4000 FOR n=55616 TO 56383 STEP 8
: POKE 23502,255: PRINT CHR$( (n
-55360)/8): "": LET su=0: BEEP
.05,26: FOR m=0 TO 7: PRINT PEEK
(n+m): "": LET su=su+PEEK (n+m)
: NEXT m: PRINT BRIGHT 1;su: PR
INT: NEXT n
4010 GO TO 4000
9998 REM ...
9999 CLS: SAVE "NOWELITERY" LIN
E 1: BEEP .2,2: SAVE "noweliteru
" CODE 55616,768: PRINT INK 2: BR
IGHT 1: FLASH 1:AT 11,12:" VERIF
Y": PRINT AT 0,0: BEEP .5,-10:
VERIFY "": VERIFY "CODE": REST
ORE: GO TO 160

```



J. Janus

Wilk i owce

Danuta KWASIŹUR, Mieczysław SKONIECZNY

W dzisiejszym odcinku mikrokomputerowych anegdot matematycznych chcemy zaproponować grę pod ogólnie znanym tytułem „Wilk i owce”. Jest to gra o nierównych szansach, przeznaczona dla 2 graczy: jeżeli grający owcami nie popełni rażącej pomyłki, powinien wygrać.

Gra jest realizowana na standardowej szachownicy o 64 polach. Jeden z graczy ma 4 owce (ponumerowane 1—4), drugi natomiast jednego wilka. Zarówno wilk, jak i owce mogą się przesuwać na skos o jedno pole tylko tej samej barwy. Owce mogą się przesuwać jedynie do przodu, gdy tymczasem wilk ma również prawo do ruchów do tyłu.

Na początku gry należy odpowiedzieć na 2 pytania wyświetlane w prawym dolnym rogu ekranu:

Na jakich polach gramy:

B — białe

C — czarne

Podaj kod literowy:

oraz

Kto pierwszy robi ruch:

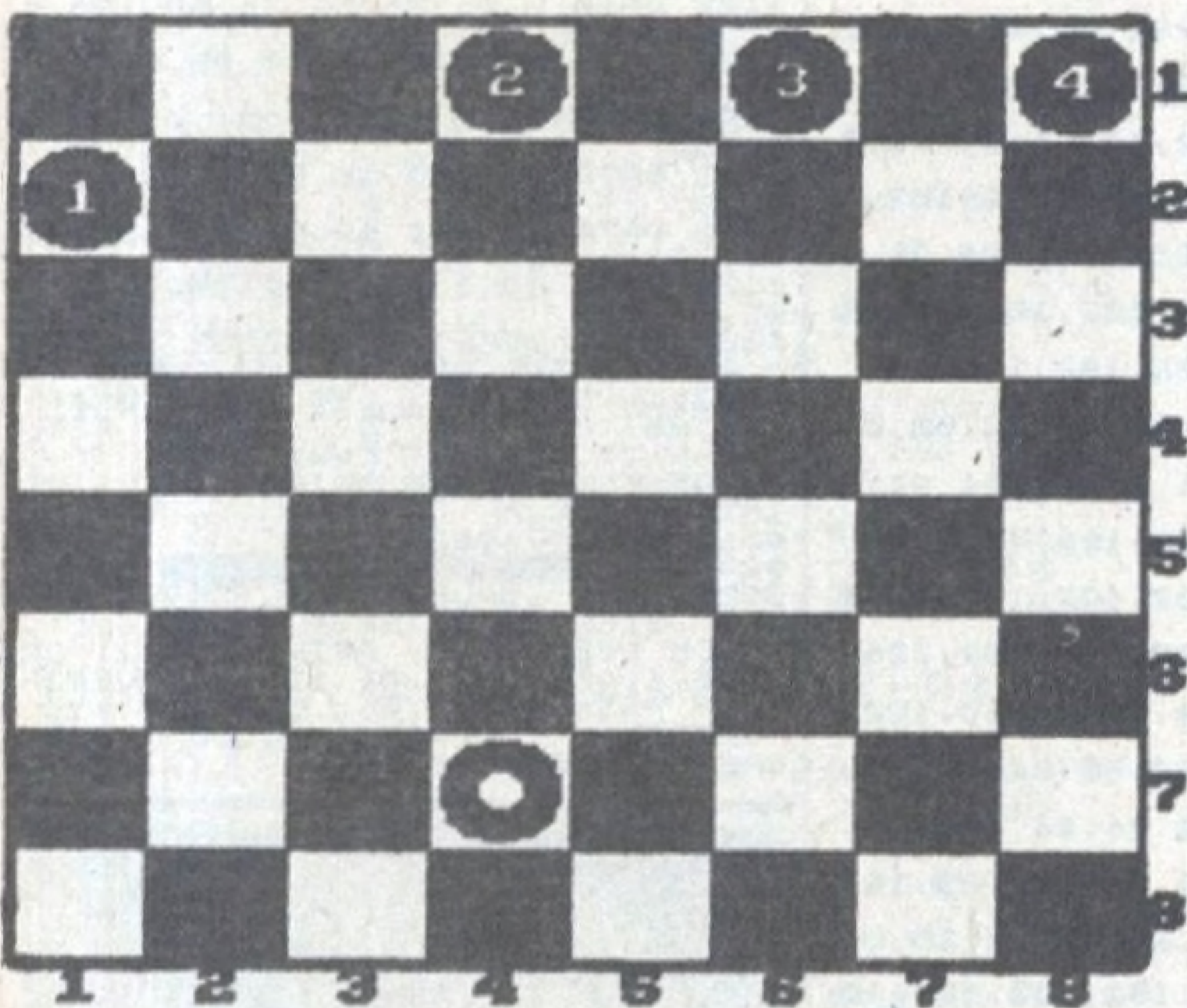
O — owce

W — wilk

Podaj kod literowy:

wciskając odpowiednie klucze literowe.

Następnie rozpoczyna się gra wg parametrów wprowadzonych uprzednio. Początkową sytuację po 2 ruchach prezentuje poniższy rysunek:



RUCH OWCY 1

PODAJ NUMER
KOLUMNY : 1
WIERSZA : 2

Przebiegu gry nie komentujemy, gdyż jest on oczywisty i nie powinien sprawiać Czytelnikom problemów.

Trzecie dolne okienko po prawej stronie ekranu jest przeznaczone na komunikaty o sytuacjach błędnych, które są następujące:

TO POLE JEST ZAJĘTE TAK NIE WOLNO SIĘ PRZESUWAĆ

Gra kończy się wygraną wilka, gdy przedostanie się poza szereg owiec. Owce wygrywają wtedy, gdy osaczą wilka tak że nie może on dokonać żadnego ruchu.

Wersja źródłowa programu jest przedstawiona na załączniku 1.

```

10 SYMBOL AFTER 247
20 SYMBOL 247,0,0,0,3,7,15,31,31
30 SYMBOL 248,0,0,126,255,255,255,255,255
40 SYMBOL 249,0,0,0,192,224,240,248,248
50 SYMBOL 250,31,63,63,63,63,63,63,31
60 SYMBOL 251,195,129,0,0,0,0,129,195
70 SYMBOL 252,248,252,252,252,252,252,248
80 SYMBOL 253,31,31,15,7,3
90 SYMBOL 254,255,255,255,255,255,126
100 SYMBOL 255,248,248,240,224,192
110 'Ekran informacyjny
120 MODE 1:INK 0,26:INK 1,0:INK 2,6:INK 3,19:BORDER 0:PAPER 1:CLS:PEN 0
130 WINDOW 1,1,40,1,3:PAPER 1,0:PEN 1,1:CLSE1
140 WINDOW 2,1,40,23,25:PAPER 2,0:PEN 2,1:CLSE2
150 LOCATE 1,10,2:PRINT1,"PROGRAM - WILK I OWCE"
160 LOCATE 2,8,2:PRINT2,"NACISNIJ DOWOLNY KLAWISZ"
170 CLEAR INPUT:A$=INKEY$
180 IF A$="" GOTO 170
190 'Deklaracja kolorow i okien w trybie 1
200 MODE 1:INK 0,26:INK 1,0:INK 2,6:INK 3,19
210 PAPER 2:BORDER 6:CLS
220 WINDOW 1,28,40,1,3:PEN 1,1:PAPER 1,0:CLSE1
230 WINDOW 2,28,40,5,10:PEN 2,1:PAPER 2,0:CLSE2
240 WINDOW 3,28,40,12,16:PEN 3,1:PAPER 3,0:CLSE3
250 'Rysowanie szachownicy
260 PAPER 0:PEN 1:FOR i=0 TO 6 STEP 2
270 FOR j1=6+i+1 TO 6+i+3
280 FOR j2=1 TO 4
290 PRINT CHR$(143);CHR$(143);CHR$(143);" ";
300 NEXT j2:PRINT: NEXT j1
310 FOR i1=6+i+4 TO 6+i+6
320 FOR j2=1 TO 4
330 PRINT " ";CHR$(143);CHR$(143);CHR$(143);
340 NEXT j2:PRINT: NEXT i1:NEXT i
350 PAPER 2:PRINT" 1 2 3 4 5 6 7 8"
360 FOR i=1 TO 8 :LOCATE 25,i*3-1:PRINT USING"2";i:NEXT
370 'Wybor koloru pol na ktorzych gramy
380 LOCATE 28,19:PRINT"Na jakich":LOCATE 28,20:PRINT"polach gramy:";
390 LOCATE 30,21:PRINT"B-biale":LOCATE 30,22:PRINT"C-czarne
400 LOCATE 28,23:PRINT"Podaj kod":LOCATE 28,24:INPUT"literowy:",kolor$
410 kolor$=UPPER$(kolor$)
420 IF kolor$(">")"B" GOTO 450
430 FOR i=1 TO 4:owce(i,1)=28i:owce(i,2)=1:NEXT
440 wilk(1)=2*(1+INT(RND*4))-1:wilk(2)=8:GOTO 520
450 IF kolor$(">")"C" GOTO 480
460 FOR i=1 TO 4:owce(i,1)=28i-1:owce(i,2)=1:NEXT
470 wilk(1)=2*(1+INT(RND*4)):wilk(2)=8:GOTO 520
480 LOCATE 28,25:PRINT"BLAD !!!"
490 FOR t=1 TO 2000:NEXT
500 LOCATE 28,24:PRINT SPC(12):LOCATE 28,25:PRINT SPC(9)

```

```

510 GOTO 400
520 'Wybor kto zaczyna
530 FOR i=19 TO 24:LOCATE 28,i:PRINT SPC(13);NEXT
540 LOCATE 28,19:PRINT"Kto pierwszy":LOCATE 28,20:PRINT"robi ruch:"
550 LOCATE 30,21:PRINT"O-owce":LOCATE 30,22:PRINT"W-wilk"
560 LOCATE 28,23:PRINT"Podaj kod":LOCATE 28,24:INPUT"literowy:",ruch$
570 ruch$=UPPER$(ruch$)
580 IF ruch$="O" OR ruch$="W" GOTO 630
590 LOCATE 28,25:PRINT"BLAD !!!"
600 FOR t=1 TO 2000:NEXT
610 LOCATE 28,24:PRINT SPC(12):LOCATE 28,25:PRINT SPC(9)
620 GOTO 560
630 FOR i=19 TO 24:LOCATE 28,i:PRINT SPC(13);NEXT
640 'Początkowe ustawienie wilka na szachownicy
650 IF kolor$="B" THEN PAPER 0:PEN 1 ELSE PAPER 1:PEN 0
660 x=3#wilk(1)-2:LOCATE x,22:PRINT CHR$(247);CHR$(248);CHR$(249)
670 LOCATE x,23:PRINT CHR$(250);CHR$(251);CHR$(252)
680 LOCATE x,24:PRINT CHR$(253);CHR$(254);CHR$(255)

```

W trakcie realizacji gry wielokrotnie są wywoływane podprogramy, których rola jest następująca:

1. Podprogram podawania współrzędnych ruchu

```

690 'Początkowe ustawienie owiec na szachownicy
700 FOR i=1 TO 4
710 x=3#owce(i,1)-2:LOCATE x,i:PRINT CHR$(247);CHR$(248);CHR$(249)
720 LOCATE x,2:PRINT CHR$(250);CHR$(251);CHR$(252)
730 LOCATE x,3:PRINT CHR$(253);CHR$(254);CHR$(255)
740 IF kolor$="C" THEN PAPER 0:PEN 1 ELSE PAPER 1:PEN 0
750 LOCATE x+1,2:PRINT USING "£";i
760 IF kolor$="C" THEN PAPER 1:PEN 0 ELSE PAPER 0:PEN 1
770 NEXT
780 'Realizacja kolejnego ruchu
790 GOSUB 860
800 blad=0:GOSUB 1150
810 IF blad=1 GOTO 790
820 GOSUB 1420
830 GOSUB 1670
840 IF ruch$="O" THEN ruch$="W" ELSE ruch$="O"
850 GOTO 780

```

2. Podprogram badania dopuszczalności ruchu

```

860 'Podprogram podawania współrzędnych ruchu
870 IF ruch$="W" GOTO 1040
880 CLS £1:LOCATE £1,2,2
890 PRINT£1,"RUCH OWCY "
900 CLS £2:LOCATE £2,2,2:PRINT£2,"PODAJ NUMER"
910 LOCATE £2,2,3:INPUT £2,"OWCY:",nrowcy
920 IF nrowcy>0 AND nrowcy<5 GOTO 950
930 a=4:GOSUB 1090
940 LOCATE £2,2,3:PRINT£2,SPC(12):GOTO 910
950 LOCATE £1,11,2:PRINT £1,nrowcy
960 LOCATE £2,2,3:INPUT £2,"KOLUMNY:",nrkol
970 IF nrkol>0 AND nrkol<9 GOTO 1000
980 a=8:GOSUB 1090
990 LOCATE £2,2,3:PRINT£2,SPC(12):GOTO 960
1000 LOCATE £2,2,4:INPUT £2,"NIERSZA:",nrwier
1010 IF nrwier>0 AND nrwier<9 GOTO 1080
1020 a=8:GOSUB 1090
1030 LOCATE £2,2,4:PRINT£2,SPC(12):GOTO 1000
1040 CLS £1:LOCATE £1,2,2
1050 PRINT£1,"RUCH WILKA"
1060 CLS £2:LOCATE £2,2,2:PRINT£2,"PODAJ NUMER"
1070 GOTO 960
1080 RETURN
1090 'Komunikat o bledzie
1100 LOCATE £3,3,2:PRINT£3,"TEN NUMER"

```

```

1110 LOCATE £3,3,3:PRINT£3,"MUSI BYC"
1120 LOCATE £3,3,4:PRINT£3,"OD 1 DO ";USING "£";a
1130 FOR t=1 TO 2000:NEXT
1140 CLS £3:RETURN

```

3. Podprogram realizacji ruchu

```

1150 'Podprogram badania dopuszczalności ruchu
1160 'Badanie zajetosci pola
1170 IF nrkol=wilk(1) AND nrwier=wilk(2) GOTO 1220
1180 FOR i=1 TO 4
1190 IF nrkol=owce(i,1) AND nrwier=owce(i,2) GOTO 1220
1200 NEXT:GOTO 1270
1210 'Komunikat o bledzie
1220 LOCATE £3,4,2:PRINT£3,"TO POLE"
1230 LOCATE £3,5,3:PRINT£3,"JEST"
1240 LOCATE £3,4,4:PRINT£3,"ZAJETE"
1250 FOR t=1 TO 2000:NEXT
1260 CLS £3:blad=1:GOTO 1410
1270 IF ruch$="W" GOTO 1330
1280 'Badanie ruchu owcy
1290 IF owce(nrowcy,2)+1<>nrwier GOTO 1360
1300 IF owce(nrowcy,1)-1<>nrkol AND owce(nrowcy,1)+1<>nrkol GOTO 1360
1310 GOTO 1410
1320 'Badanie ruchu wilka
1330 IF wilk(1)<>nrkol-1 AND wilk(1)<>nrkol+1 GOTO 1360
1340 IF wilk(2)<>nrwier-1 AND wilk(2)<>nrwier+1 GOTO 1360
1350 GOTO 1410
1360 'Komunikat o bledzie
1370 LOCATE £3,4,2:PRINT£3,"TAK NIE"
1380 LOCATE £3,3,3:PRINT£3,"MOLNO SIE"
1390 LOCATE £3,3,4:PRINT£3,"PRZESUMAC"
1400 FOR t=1 TO 2000:NEXT:blad=1
1410 CLS £3:RETURN

```

4. Podprogram badania i realizacji końca gry

```

1420 'Podprogram realizacji ruchu
1430 IF ruch$="O" GOTO 1540
1440 'Przesunięcie wilka
1450 x=3#wilk(1)-2:y=3#wilk(2)-2
1460 FOR i=0 TO 2:LOCATE x,y+i
1470 PRINT SPC(3):NEXT
1480 x=3#nrkol-2:y=3#nrwier-2
1490 FOR i=0 TO 2:LOCATE x,y+i
1500 FOR j=0 TO 2:PRINT CHR$(247+3#i+j);NEXT
1510 NEXT
1520 wilk(1)=nrkol:wilk(2)=nrwier
1530 GOTO 1660
1540 'Przesunięcie owcy
1550 x=3#owce(nrowcy,1)-2:y=3#owce(nrowcy,2)-2
1560 FOR i=0 TO 2:LOCATE x,y+i
1570 PRINT SPC(3):NEXT
1580 x=3#nrkol-2:y=3#nrwier-2
1590 FOR i=0 TO 2:LOCATE x,y+i
1600 FOR j=0 TO 2:PRINT CHR$(247+3#i+j);NEXT
1610 NEXT
1620 IF kolor$="C" THEN PAPER 0:PEN 1 ELSE PAPER 1:PEN 0
1630 LOCATE x+1,y+1:PRINT USING "£";nrowcy
1640 IF kolor$="C" THEN PAPER 1:PEN 0 ELSE PAPER 0:PEN 1
1650 owce(nrowcy,1)=nrkol:owce(nrowcy,2)=nrwier
1660 RETURN
1670 'Podprogram badania i realizacji końca gry
1680 'Wygrana wilka
1690 IF wilk(2)=1 THEN a=1:GOTO 1960
1700 FOR i=1 TO 4
1710 IF wilk(2)>owce(i,2) GOTO 1750
1720 NEXT

```

```

1730 a=1:GOTO 1960
1740 'Wygrana owiec
1750 IF wilk(1)<>1 GOTO 1810
1760 FOR b=wilk(2)-1 TO wilk(2)+1 STEP 2
1770 FOR i=1 TO 4
1780 IF owce(i,1)=2 AND owce(i,2)=b GOTO 1800
1790 NEXT i:GOTO 2080
1800 NEXT b:GOTO 1950
1810 IF wilk(1)<>8 GOTO 1870
1820 FOR b=wilk(2)-1 TO wilk(2)+1 STEP 2
1830 FOR i=1 TO 4
1840 IF owce(i,1)=7 AND owce(i,2)=b GOTO 1860
1850 NEXT i:GOTO 2080
1860 NEXT b:GOTO 1950
1870 FOR b=wilk(2)-1 TO wilk(2)+1 STEP 2
1880 FOR c=wilk(1)-1 TO wilk(1)+1 STEP 2
1890 FOR i=1 TO 4
1900 IF owce(i,1)=c AND owce(i,2)=b GOTO 1930
1910 NEXT
1920 GOTO 2080

```

```

1930 NEXT:NEXT
1940 'Czy koniec gry
1950 a=2
1960 CLS £1:LOCATE £1,2,2:PRINT£1,"KONIEC GRY"
1970 CLS £2:LOCATE £2,4,2
1980 IF a=1 THEN PRINT£2,"WYGRAL" ELSE PRINT£2,"WYGRALY"
1990 LOCATE £2,5,4
2000 IF a=1 THEN PRINT£2,"WILK" ELSE PRINT£2,"OWCE"
2010 LOCATE £3,4,2:PRINT£3,"CZY GRE":LOCATE £3,3,3:PRINT£3,"POMIOTRZYC"
2020 LOCATE £3,5,4:INPUT £3,"(T/N):",dalej$
2030 dalej$=UPPER$(dalej$)
2040 IF dalej$="T" GOTO 190
2050 PAPER 2:CLS
2060 LOCATE 16,12:PRINT"KONIEC GRY"
2070 GOTO 2070
2080 RETURN

```

Danuta KWASIŹUR
Mieczysław SKONIECZNY

Różne sposoby kopiowania ekranu mikrokomputera na drukarce

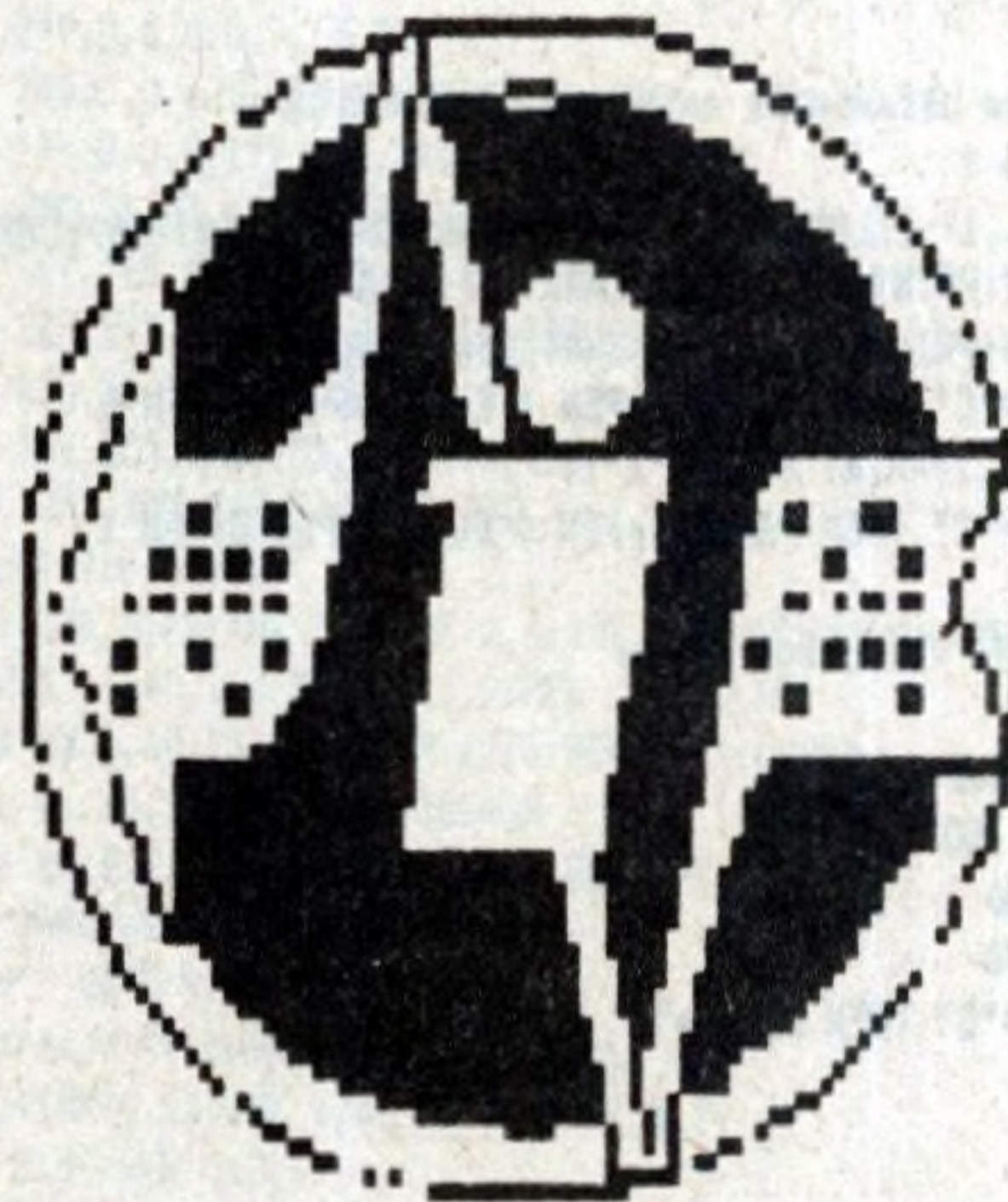
Wśród użytkowników mikrokomputerów znanych jest sporo programów kopiujących zawartość ekranu na drukarkę. Stanowią one wygodną formę uzyskiwania ciekawych rysunków. Technika współpracy z takim programem jest najczęściej następująca:

1. Przygotowanie rysunku jako zbioru na dysku. Realizuje się to instrukcją: SAVE „rysunek”, b, &c000, &4000, wydrukowany zbiór zajmuje 17 kB pamięci i ma dopełnienie. BIN.
2. Wprowadzenie do pamięci mikrokomputera programu kopiującego, zainicjowanie jego pracy i wprowadzenie niezbędnych parametrów (jest to najczęściej nazwa zapisana na dysku zbioru).
3. Wydruk rysunku.

Chcemy zaproponować Czytelnikom wygodny w użyciu program, który wykonuje kopie ekranu dwoma różnymi sposobami:

1. **Normalnym** — ściśle odwzorowanie ekranu na papierze. Wydruk powstaje przez kolejne kopiowanie wierszy od góry do dołu.
2. **Powiększonym** — odwzorowanie ekranu na papierze w skali 1:2. Wydruk powstaje przez kolejne kopiowanie kolumn od strony prawej do lewej.

Poniżej 2 rysunki pokazują kopie tego samego rysunku wykonane ww. sposobami.



rys. 2

Jak widać, pierwszy sposób nieznacznie spłaszcza obiekt, natomiast drugi go wydłuża. Mamy jednak nadzieję, że te drobne mankamenty nie będą przeszkadzały użytkownikom tego programu. Program pracuje w trybie konwersacyjnym, a więc na kolejne pytania należy udzielać następujących odpowiedzi:

1. **PODAJ WARIANT KOPIOWANIA:**
Wprowadzamy liczbę 1 lub 2 w zależności od sposobu, którym chcemy kopiować.
2. **PODAJ NAZWĘ ZBIORU:**
Wprowadzamy nazwę zeskładowanego na dysku zbioru bez dopełnienia (program sam uzupełnia ją o dopełnienie. BIN).
3. **PODAJ TRYB, W KTÓRYM UTWORZONO OBRAZ:**
Wprowadzamy liczbę całkowitą z przedziału (0,2).

Na bieżąco jest przeprowadzana kontrola formalna wprowadzanych danych. W wypadku próby podania błędnego parametru program sygnalizuje to jednym z komunikatów:

BŁĘDNY WARIANT
ZA DŁUGA NAZWA ZBIORU
BŁĘDNY TRYB

i żąda ponownego wprowadzenia tej danej. Następnie po włożeniu w stację odpowiedniej dyskietki i przygotowaniu drukarki wykonuje się kopie ekranu. Po wykonaniu rysunku program wraca do stanu początkowego, co pozwala wykonać jednorazowo więcej niż 1 rysunek.

Segment sterujący programem opracowano w języku BASIC mikrokomputera AMSTRAD CPC 6128 i ma on następującą postać:

```

10 REM 'Program kopiowania ekranu mikrokomputera AMSTRAD CPC 6128'
20 MODE 1:INK 0,0:INK 1,14:INK 2,26:INK 3,6:PAPER 0:CLS: BORDER 0
30 WINDOW 1,7,33,1,3:PAPER 1,1:CLS 1:PEN 1,0
40 REM 'Mozliwosci programu'
50 LOCATE 1,2,2:PRINT 1,"PROGRAM KOPIOWANIA EKRANU"
60 PEN 1:LOCATE 2,7:PRINT "1. KOPIOWANIE NORMALNE":PRINT:PRINT " 2. KOPIOWANIE POWIEKSZONE (Skala 2:1)"
70 WINDOW 1,40,13,20:PAPER 1,2,2
80 REM 'Mprowadzanie wariantu kopiowania'
90 CLS 1:PEN 1,0:LOCATE 1,2,2:INPUT 1,"PODAJ WARIANT KOPIOWANIA: ",WK
100 IF WK<1 OR WK>2 THEN PEN 3:LOCATE 1,2,4:PRINT 1,"BLEDNY WARIANT":FOR T=1 TO 1000:NEXT:GOTO 90
110 REM 'Mprowadzanie nazwy zbioru'
120 LOCATE 1,2,4:INPUT 1,"PODAJ NAZWE ZBIORU: ",A$
130 IF LEN(A$)>8 THEN PEN 3:LOCATE 1,2,6:PRINT 1,"ZA DLUGA NAZWA ZBIORU":FOR T=1 TO 1000:NEXT:LOCATE 1,2,6:PRINT 1,SPC(30):PEN 0:LOCATE 1,2,1,4:PRINT 1,SPC(19):GOTO 120
140 A$=A$+".BIN"
150 REM 'Mprowadzanie trybu obrazu'
160 LOCATE 1,2,6:INPUT 1,"PODAJ TRYB W KTORYM Utworzono obraz: ",TRYB
170 IF TRYB<0 OR TRYB>2 THEN PEN 3:LOCATE 1,2,8:PRINT 1,"BLEDNY TRYB":FOR T=1 TO 1000:NEXT:LOCATE 1,2,8:PRINT 1,SPC(11):PEN 0:LOCATE 1,2,3,6:PRINT 1,"":GOTO 160
180 PAPER 0:PEN 1:LOCATE 1,22:PRINT "Jeżeli masz w stacji dysku dyskietkę z obrazem i drukarka jest gotowa do pracy to naciśnij dowolny klawisz."
190 CLEAR INPUT
200 b$=INKEY$
210 IF b$="" THEN GOTO 200
220 REM 'wydruk obrazu na drukarce'
230 MODE tryb
240 LOAD a$
250 MEMORY 39999
260 ON WK GOSUB 470,280
270 GOTO 10

```

Poszczególne sposoby kopiowania są realizowane przez następujące podprogramy, opracowane w języku wewnętrznym:

1. Kopiowanie powiększone

Podprogram ten zajmuje obszar 415 bajtów o adresach od 40000 do 40414. Jego postać poszerzona o instrukcje ładowania go do pamięci i wywoływania jest następująca:

```

280 REM 'Podprogram kopiowania powiekszonego'
290 RESTORE 370
300 FOR a=40000 TO 40414
310 READ x
320 POKE a,x
330 NEXT a
340 CALL 40000
350 COPY
360 RETURN
370 DATA 1,96,156,33,92,156,205,209,188,201,164,74,46,87,46,47,49,57,56,55,47,87,73,73,67,79,80,217,0,0,0,0,88,156,237,115,223,157,205,190,157,62,27,205,184,157,62,49,205,184,157
380 DATA 62,255,50,113,157,33,79,192,62,81,50,225,157,62,8,50,226,157,205,203,156,58,225,157,61,40,35,50,225,157,205,189,156,58,226,157,71,61,50,226,157,62,8,144,71,205,43,157,58,226
390 DATA 157,61,183,32,219,6,1,205,193,156,24,207,62,240,50,113,157,205,189,156,205,158,157,201,58,226,157,71,205,43,157,205,93,157,205,131,157,201,229,253,33,174,158,6,25,17,0,8,229,14
400 DATA 8,126,205,239,156,253,119,0,253,35,25,13,32,243,225,17,80,0,25,16,230,225,43,201,197,6,0,79,230,85,183,40,2,6,15,121,230,170,183,40,2,62,240,176,193,201,213,197,79,6
410 DATA 4,175,203,1,203,23,203,1,203,1,203,1,203,1,203,23,87,203,15,203,19,203,15,203,19,122,179,95,16,227,193,209,201,201,4,5,200,221,33,230,157,253,33,174,158,175,14,200,197,253
420 DATA 203,0,14,221,203,0,22,221,182,0,16,243,221,35,253,35,193,13,32,234,71,58,227,157,79,120,50,227,157,177,50,228,157,201,58,228,157,183,200,6,200,221,33,230,157,205,167,157,221,126

```

```

430 DATA 0,203,47,230,255,205,184,157,205,184,157,205,184,157,205,184,157,221,35,16,233,201,62,13,205,184,157,62,10,205,184,157,221,229,253,229,229,205,9,187,218,215,157,225,253,225,221,225,201,62
440 DATA 27,205,184,157,62,64,24,17,62,27,205,184,157,62,76,205,184,157,62,32,205,184,157,62,3,205,43,189,48,251,201,33,209,157,205,17,188,135,79,6,0,9,94,35,86,237,83,218,156,201
450 DATA 239,156,5,157,42,157,205,158,157,237,123,223,157,201

```

2. Kopiowanie normalne

Podprogram ten zajmuje obszar 177 bajtów o adresach od 40960 do 41136. Jego postać poszerzona o instrukcje ładowania go do pamięci i wywoływania jest następująca:

```

460 REM 'Podprogram kopiowania normalnego'
470 RESTORE 530
480 FOR i=&A000 TO &A0B0
490 READ v:POKE i,v
500 NEXT i
510 CALL &A000
520 RETURN
530 DATA 205,166,160,62,27,205,157,160
540 DATA 62,49,205,157,160,205,186,187
550 DATA 205,231,187,50,180,160,17,0
560 DATA 0,33,143,1,34,178,160,62
570 DATA 7,50,177,160,62,10,205,157
580 DATA 160,62,13,205,157,160,62,27
590 DATA 205,157,160,62,76,205,157,160
600 DATA 62,127,205,157,160,62,2,205
610 DATA 157,160,14,0,58,177,160,71
620 DATA 229,197,213,205,240,187,209,193
630 DATA 33,180,160,190,225,55,32,1
640 DATA 167,203,17,43,43,16,233,58
650 DATA 177,160,254,7,40,7,175,203
660 DATA 17,203,17,203,17,121,205,157
670 DATA 160,19,229,33,127,2,55,237
680 DATA 82,225,56,5,42,178,160,24
690 DATA 193,35,124,181,40,32,43,17
700 DATA 0,0,34,178,160,62,7,189
710 DATA 32,146,124,180,32,142,62,4
720 DATA 50,177,160,24,135,205,46,189
730 DATA 56,251,205,43,189,201,62,27
740 DATA 205,157,160,62,64,205,157,160,201

```

Jeżeli Czytelnicy dysponują innymi programami kopiowania ekranu, to w prosty sposób mogą je dołączyć do tego programu wykonując następujące modyfikacje:

1. Poszerzyć liczbę wariantów i zmienić sposób kontroli (instrukcje 60—100).
2. Zmienić instrukcję 260 dopisując do skoku z przełącznikiem etykiety dołączonych podprogramów.
3. Dopisać nowy podprogram kopiujący wraz z jego wywołaniem.



K. WALCZAK

FIRST-COME FIRST-SERVED - pierwszy zgłoszony pierwszy obsłużony (w teorii kolejek),
 FIRST-COME LAST-SERVED - pierwszy zgłoszony ostatni obsłużony (w teorii kolejek),
 FIRST FIT - metoda pierwszego odpowiadającego (metoda podziału pamięci),
 FIRST FIT ALGORITHM - algorytm dopasowywania,
 FIRST FIT DECREASING ALGORITHM - algorytm dopasowania uporządkowanych zadań,
 FIRST-IN FIRST-OUT - pierwszy na wejściu pierwszy na wyjściu (w teorii kolejek),
 FIRST-IN LAST-OUT - pierwszy na wejściu, ostatni na wyjściu (w teorii kolejek),
 FIRST-IN-NOT-USED-FIRST-OUT - metoda wymiany stronic (segmentów) pamięci w systemach z pamięcią wirtualną, polegająca na usuwaniu stronic do których przez długi czas nie było odwołań,
 FIRST-ITEM - pierwsza pozycja,
 FIRST-LEVEL ADDRESS - adres pierwszego poziomu,
 FIRST LEVEL INDEX TABLE - tablica indeksów pierwszego stopnia,
 FIRST LEVEL OVERFLOW BUCKET - porcja (blok programowy) w obszarze przepełnień pierwszego stopnia,
 FIRST LEVEL OVERFLOW ROUTINES - procedury przepełnień pierwszego poziomu,
 FIRST-ORDER LOGIC - logika pierwszego rzędu,
 FIRST-ORDER SUBROUTINE - podprogram pierwszego rzędu,
 FIRST RULES OF ARITHMETIC - działania arytmetyczne podstawowe,
 FITNESS FOR USE - zdatność użytkowa,
 FITTING - dopasowanie, montaż,
 FIT TOGETHER - pasować do siebie,
 FIVE-LEVEL CODE - kod pięciostopniowy,
 FIX - poprawianie, sprostowanie, ustalanie, także: poprawiać, sprostować, ustalać, umacniać,
 FIXED - stały, ustalony, nieruchomy,
 FIXED ADDRESS - adres stały,
 FIXED AREA - obszar ustalony,
 FIXED DISK - dysk stały, dysk wymienny,
 FIXED HEAD DISC - dysk z nieruchomą głowicą,
 FIXED INFORMATION LENGTH - stała długość informacji,
 FIXED INSTRUCTION LENGTH - stała długość rozkazu,
 FIXED-LENGTH ADDRESSING - adresacja ze stałą długością adresu (dot. sposobu adresacji w sieci komputerowej),
 FIXED-LENGTH COMPUTER - komputer ze słowem o stałej długości,
 FIXED-LENGTH FIELD - pole stałej długości,
 FIXED-LENGTH RECORDS - zapisy (rekordy) o stałej długości,
 FIXED LINE - łącze stałe,
 FIXED POINT - przecinek stały,
 FIXED-POINT ARITHMETIC - arytmetyka stałopozycyjna, arytmetyka stałoprzecinkowa,
 FIXED-POINT CALCULATION - rachunek stałoprzecinkowy,
 FIXED-POINT INSTRUCTION - rozkaz stałoprzecinkowy,
 FIXED-POINT NOTATION - zapis stałoprzecinkowy,
 FIXED-POINT NUMBER - liczba stałoprzecinkowa,
 FIXED-POINT OPERATION - operacja stałoprzecinkowa,
 FIXED-POINT REPRESENTATION - przedstawienie liczb stałoprzecinkowych,
 FIXED-PROGRAM COMPUTER - komputer o stałym programie (mający na stałe, ustaloną kolejność wykonywanych rozkazów),
 FIXED RADIX NUMERATION SYSTEM - system liczbowy o stałej podstawie,
 FIXED RETENTION - stałe członkostwo, ustalone członkostwo,
 FIXED ROUTING - ustalona trasa (droga),
 FIXED STORAGE - pamięć stała, pamięć wielkości stałych, pamięć martwa,
 FIXED WIRE - sznur do łączenia na stałe, tablicy połączeń,
 FIXED WORD LENGTH - ustalona długość słowa,
 FIXING - ustalanie, utrwalanie,
 FLAT BED PLOTTER - ploter stołowy (płaski),
 FLAG - znacznik, flaga, sygnalizator, znak sygnalizujący, także: bit oznaczający granice pola,
 FLAG BIT - znacznik,
 FLAG FIELD - znacznik pola,
 FLAG PAGE - strona tytułowa,
 FLAG REGISTER - rejestr flagowy,
 FLAG SEQUENCE - znacznik transmisji oddzielający obrazy przesyłane na monitor w sieci komputerowej, także: znacznik wielobitowy, ramka,
 FLASHING - podświetlanie szablonu (jako tła obrazu),
 FLASH PHOTOGRAPHIC PRINTER - drukarka fotograficzna błyskowa,

FLAT - płaski, także: spłaszczać, równać,
 FLAT ADDRESSING - adresacja prosta,
 FLATBED PLOTTER - pisak planszety, ploter planszety,
 FLAT FILE - plik (zbiór) składający się z zapisów (rekordów) tego samego typu,
 FLATPACK - płaska obudowa typu krab (mikroukład),
 FLAT SUBASSEMBLY - pakiet,
 FLATTEN - równać, spłaszczać,
 FLEXIBILITY - giętkość, podatność, elastyczność,
 FLEXIBLE - giętki, podatny, elastyczny,
 FLEXIBLE ARRAY - tablica o zmiennym rozmiarze,
 FLEXION - druga pochodna,
 FLEXOWRITER - kodopis,
 FLEX POINT - punkt przegięcia,
 FLICKER - niepożądane migotanie, miganie, mruganie,
 FLICKER NOISE - szum migotania,
 FLIGHT PATH COMPUTER - przelicznik toru lotu,
 FLIGHT COMPUTER - komputer do celów nawigacji (lotniczej lub kosmicznej),
 FLIMSY - rozdzielaająca karta dziurkowana,
 FLIP - przetrzącać (z jednego położenia w drugie),
 FLIP-FLOP - przerzutnik, multiwibrator bistabilny,
 FLIP OFF - wyłączyć, zgasić (np. światło),
 FLIP ON - włączyć, zapalić (np. światło),
 FLIPPY DISK - dyskietka dwustronna z niezależnie używanymi stronami,
 FLOAT - płynny,
 FLOATING - wahający się, zmienny, pływający,
 FLOATING ADDRESS - adres symboliczny, adres pływający, adres zmienny,
 FLOATING GATE - pływająca bramka,
 FLOATING POINT - zmienny przecinek,
 FLOATING-POINT ARITHMETIC - arytmetyka zmiennopozycyjna, arytmetyka zmiennoprzecinkowa,
 FLOATING-POINT BAR - podstawa liczenia ze zmiennym przecinkiem,
 FLOATING-POINT CALCULATION - rachunek zmiennoprzecinkowy,
 FLOATING POINT CHARACTERISTIC - cecha liczby zmiennoprzecinkowej,
 FLOATING POINT COMPUTATION - rachunek zmiennoprzecinkowy,
 FLOATING-POINT INSTRUCTION - rozkaz zmiennoprzecinkowy,
 FLOATING-POINT NOTATION - zapis zmiennoprzecinkowy,
 FLOATING-POINT NUMBER - liczba zmiennoprzecinkowa,
 FLOATING-POINT OPERATION - operacja zmiennoprzecinkowa,
 FLOATING-POINT OPERATIONS PER SECOND - liczba operacji zmiennoprzecinkowych na sekundę (określenie szybkości działania procesora),
 FLOATING-POINT PACKAGE - program zmiennoprzecinkowy, program zmiennopozycyjny,
 FLOATING POINT REPRESENTATION - przedstawienie liczb zmiennoprzecinkowych,
 FLOATING-POINT WORD - słowo zmiennoprzecinkowe,
 FLOODING - kaskadowy wybór trasy, także: rozpiwowanie,
 FLOPOVER - przesuwanie się obrazu w pionie (na ekranie monitora),
 FLOPPY-BASED - urządzenie pamięciowe z dyskami elastycznymi,
 FLOPPY DISK - dysk elastyczny, minidysk, dyskietka,
 FLOPPY DISK DRIVE - napęd dyskietkowy,
 FLOPS - patrz: FLOATING-POINT OPERATIONS PER SECOND,
 FLOW - przepływ, także: sterowanie,
 FLOW AUGMENTING PATH - droga powiększająca przepływ,
 FLOW CHART - schemat czynności, diagram przebiegu, sieć działań, schemat działania programu, schemat blokowy programu,
 FLOWCHARTER - program sporządzający schemat blokowy,
 FLOWCHARTING - sporządzanie schematu blokowego,
 FLOW CHART SYMBOL - element schematu blokowego,
 FLOWCHART TEMPLATE - szablon do projektowania,
 FLOW CONTROL - sterowanie strumieniem danych, sterowanie przepływem (w sieciach komputerowych),
 FLOW DIAGRAM - schemat działania programu, schemat blokowy programu, sieć działań,
 FLOW DIRECTION - kierunek przebiegu (przepływu) na schemacie blokowym,
 FLOWLINE - linia obiegu, linia łącząca dwa symbole na schemacie blokowym,
 FLOW OF CURRENT - przepływ prądu elektrycznego,
 FLOW OF INFORMATION - strumień informacji,
 FLOW SHOP PROBLEM - przepływowy problem obsługi,
 FLOW TIME - czas przepływu (dot. przebywania zadania w systemie obsługi),
 FLUCTUATION - wariacja,
 FLUID - płyn, także: płynny,
 FLUID COMPUTER - komputer na elementach strumieniowych,
 FLUTTER - drżenie dźwięku,
 FLUX - strumień,

FORMAL LOGIC - logika matematyczna, logika symboli-
czna, logistyka,
FORMAL PARAMETER - parametr formalny,
FORMAL SPECIFICATION - opis formalny,
FORMAT - format (sposób przedstawienia danych na no-
śniku),
FORMAT BUFFER - pamięć formatu formularza,
FORMAT CHARACTER - znak sterujący formatem,
FORMAT EFFECTOR - formatyzator,
FORMAT ERROR - błąd formalny, błąd formatu,
FORMATION - tworzenie się, powstawanie,
FORMAT LABEL - etykieta formatu,
FORMATLESS - bezformatowy,
FORMATTED - formatowany, sformatowany,
FORMATTED CAPACITY - pojemność nośnika danych, poje-
mność użytkowa,
FORMATTER - program edycji tekstów tworzący formaty,
FORMATTING - formatowanie (rozplanowywanie tekstów
na ekranie monitora),
FORM FEED - przesunięcie strony, przesuw formularza,
nowa strona,
FORMFEED - patrz: FORM FEED,
FORM FEED CHARACTER - znak zasilenia (drukarki) for-
mularzem, także: znak przesuwu strony,
FORM FEEDING - przesuw formularza,
FORMING - kształtowanie, formowanie, tworzenie,
FORM OVERLAY - szablon (tabela, siatka lub mapa sta-
nowiąca element tła obrazu),
FORM SET - zestaw formularzy,
FORMS FLASH - podświetlanie szablonu, (jako tła ob-
razu),
FORMULA - wzór, wyrażenie,
FORMULA LANGUAGE - język formuł,
FORMULATION - sformułowanie, wyrażenie za pomocą
wzoru,
FORMULA TRANSLATION - patrz: FORTRAN,
FOR TECHNICAL PURPOSES - do celów technicznych,
FORTH - wysokopoziomowy język programowania,
FOR THE TIME BEING - na pewien czas, tymczasowo,
FOR THIS PURPOSE - w tym celu,
FORTRAN - język proceduralny do zastosowań naukowych
i technicznych,
FORWARD - nadawać, transmitować, komunikować,
FORWARD CHAINING - także: wnioskowanie uprzedzające
(w systemach ekspertowych),
FORWARD CHANNEL - kanał docelowy,
FORWARD-COMPATIBLE - zgodny, kompatybilny (o progra-
mie, który może pracować na innym typie komputera),
FORWARD DECLARATION - wstępny (częściowy) opis obie-
ktu programu,
FORWARD DIRECTION - kierunek przewodzenia,
FORWARDER - urządzenie przesyłania danych (w sieci),
FORWARDING - przesyłanie danych (do miejsca wykorzy-
stania),
FORWARD REFERENCE - odsyłacz, odnośnik użytkowany
wstępnie,
FORWARDS - do przodu,
FORWARD SPACING - przesunięcie do przodu,
FORWARD SUPERVISION - sterowanie wprost,
FOUNT - patrz: FONT,
FOUR-ADDRESS - czteroadresowy,
FOUR-ADDRESS INSTRUCTION - rozkaz czteroadresowy,
FOURTH-GENERATION COMPUTER - komputer czwartej gene-
racji,
FOUR-WIRE COMMUNICATION - przesyłanie czteroprzewo-
dowe,
FOUR-WIRE LINE - łącze czteroprzewodowe,
FOX MESSAGE - standardowy komunikat kontroli popraw-
ności transmisji, zawierający wszystkie litery alfa-
betu łacińskiego (postać komunikatu stanowi zdanie:
"A quick brown fox jumps over the lazy dog 123456789
0"),
FPLA - patrz: FIELD-PROGRAMMABLE LOGIC ARRAY,
FPRM - patrz: FIELD-PROGRAMMABLE ROM,
FRACTION - ułamek, także: frakcja, część,
FRACTIONAL - ułamkowy, także: frakcyjny, częściowy,
FRACTIONAL DIGIT - cyfra po przecinku,
FRACTIONAL EXPONENT - wykładnik ułamkowy,
FRACTIONAL NUMBER - liczba ułamkowa,
FRACTIONAL PART - mantysa,
FRACTION BAR - kreska ułamkowa,
FRAGMENT - fragment, odłamek, także: fragmentować,
FRAGMENTARY - częściowy, fragmentaryczny,
FRAGMENTATION - fragmentacja, rozdrabnianie,
FRAME - 1. grupa (blok) danych, 2. rząd bitów po-
łożonych w poprzek taśmy, 3. system odczytu, 4. kadr
(w sieciach - przesyłana porcja danych), 5. karkas,
korpus, ramka,
FRAME BUFFER - bufor obrazu, bufor ramki,
FRAME CHECK SEQUENCE - ciąg kontrolny ramki,
FRAME DELETION - skasowanie obrazu, skasowanie ramki
(fragmentu lub całego ekranu monitora),
FRAME GRABBER - urządzenie wprowadzania i zapisywa-
nia obrazów (fragmentu lub całego ekranu monitora),

FLUXING - upłynianie (nadawanie płynności), także:
uplastycznianie, zmiekczenie,
FLX - patrz: FILE EXCHANGE UTILITY,
FLYBACK - szybki ruch powrotny, powrót,
FLYING HEAD - głowica magnetyczna zapisu/odczytu nie
dotykająca nośnika danych lecz unosząca się nad nim
na cienkiej warstwie powietrza,
FLYING LEAD - cienki drut, łączący pole kontaktowe
z nóżką obudowy np. tranzystora, układu scalonego,
FLYING SPOT - ruchoma plamka świetlna,
FLYING SPOT SCAN - skaning (system kodowania obrazu
za pomocą techniki oświetlenia rastrowego, uwzględnia-
jący jasność obrazu),
FLYING-SPOT SCANNING - wybieranie ruchomą plamką
świetlną,
FM - patrz: FREQUENCY MODULATION,
FM-RECORDING - patrz: FREQUENCY MODULATION RECOR-
DING,
FMT - patrz: FORMAT,
FOCAL POINTS - punkty skupiające,
FOLD - składać (np. dane w celu zwartej przedsta-
wienia),
FOLDABILITY - podatność na zginanie,
FOLD DISTANCE - odstęp zagięcia,
FOLDER - określenie kartoteki plików (zbiorów) teks-
towych w systemach,
FOLDING - składany,
FOLIATE - rozwarstwiać się,
FOLLOWED BY ... - także: zakończone ...
FOLLOWER - powtarzacz, także: człon bierny, człon
napędzany,
FOLLOWING - kolejny, następny,
FOLLOW INSTRUCTIONS - postępować według wskazówek,
FOLLOW UP - nadać,
FOLLOW-UP - dalsza obróbka, dalsze opracowanie,
FOLLOW-UP FILE - plik uporządkowany według klucza,
inicjowanego określonym zdarzeniem w systemie,
FONT - zbiór znaków graficznych o określonym kroju,
krój pisma,
FONT-CHANGE CHARACTER - znak zmiany kroju czcionki,
FONT RECTICLE - miejsce na papierze lub ekranie,
na jeden znak,
FOOLPROOF - bezpieczny przy nieostrożnym lub nie-
umiejętnym obchodzeniu się, także: zabezpieczać
urządzenie przed skutkami nieostrożnego lub nieumie-
jętnego obchodzenia się,
FOOL-TOLERANCE - odporność (wytrzymałość) na nie-
umiejętne obchodzenie się,
FOOTAGE COUNTER - licznik długości taśmy,
FOOTER - stopka (opis powtarzający się na dole każ-
dej strony tekstu),
FOOTING - zsumowanie,
FORBIDDEN - zakazany, niedopuszczalny, wzbroniony,
FORBIDDEN TERM - askryptor, nondeskryptor, termin
zakazany,
FORCE - siła, także: wciskać, wtykać, wymuszać,
FORCED - wymuszony,
FORCE OUT - wypychać,
FORCING INPUT - wejście przełączające (przerzutnika)
FORECAST - prognoza,
FORECAST DEMAND - zapotrzebowanie przewidywane,
FORECASTING - prognozowanie, przewidywanie, spekulac-
ja,
FOREDESIGN - projekt wstępny,

FOREGROUND-BACKGROUND SCHEDULING - szeregowanie wie-
lopoziomowe,
FOREGROUND COLOR - kolor przedniego planu obrazu,
FOREGROUND IMAGE - przedni plan obrazu,
FOREGROUNDING - wykonywanie zadań o wyższym priory-
tecie, obsługa priorytetowa,
FOREGROUND MUSIC - muzyka pierwszoplanowa,
FOREGROUND (PRIORITY) - o wyższym priorytecie, pierw-
szoplanowy,
FOREGROUND PROCESSING - przetwarzanie pierwszopla-
nowe,
FOREGROUND PROGRAM - program pierwszoplanowy,
FOREIGN KEY - klucz zewnętrzny,
FOREIGN MATTER - obce ciało,
FORESEE - przewidywać,
FORESEEABLE - przewidywalny, możliwy do przewidzenia,
dający się przewidzieć,
FOREST - las, wielka liczba drzew (w teorii grafów),
FORGETFULNESS - brak działania następczego, wtórne-
go (w teorii masowej obsługi),
FORK - rozgałęzienie, rozwidlenie,
FORM - formularz, druk, sformatować, także: druk,
formularz,
FORMAL - formalny,
FORMALIZATION - formalizacja (opisu, problemu),
FORMALIZE - formalizować,
FORMAL LANGUAGE - język formalny,

LIGA MYŚLĄCYCH

ZADANIE 1

Od dwóch kawałków stopu, o różnej zawartości procentowej miedzi, ważących m kilogramów i n kilogramów, odcięto jednakowe wagowo kawałki. Każdy z odciętych kawałków stopiono z pozostałą resztą drugiego kawałka stopu i okazało się, że procentowa zawartość miedzi w otrzymanych stopach jest jednakowa. Ile ważył każdy z odciętych kawałków?

ZADANIE 2

Łódź motorowa płynąc z prądem rzeki przebywa odległość między dwoma miastami w czasie $a = 4$ h, a płynąc pod prąd odległość tę przebywa w czasie $b = 12$ h. W jakim czasie belka niesiona prądem rzeki przebędzie tę odległość?

ZADANIE 3

Zboże zawiera k % zanieczyszczeń. Po każdorazowym oczyszczeniu ilość zanieczyszczeń zmniejsza się o q %, przy

czym ilość czystego ziarna nie ulega zmianie. Ile razy trzeba powtórzyć oczyszczenie zboża, aby jego zanieczyszczenie zmniejszyło się do f % ($f < L < q$)?

ZADANIE 4

Dziesięć litrów wody o temperaturze 13°C zmieszano z 12,5 litrami wody o takiej temperaturze, że po ustaleniu się temperatury mieszanina miała temperaturę wyższą od 25°C . Lecz mniejszą od 30°C . Jaka temperatura miała woda dolana?

ZADANIE 5

Mamy 24 gramy radioaktywnego pierwiastka o czasie połowicznego rozpadu $T_1 = 7$ lat i 3 gramy innego radioaktywnego pierwiastka. Jaki jest czas połowicznego rozpadu T_2 drugiego pierwiastka, jeżeli po 21 latach ilości obu pierwiastków są równe?

Rozwiązania zadań prosimy przysyłać do redakcji do końca listopada br. z dopiskiem „Liga Myślących”. Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi” czekają dodatkowe nagrody.

Tylko dla informatyków

JĘZYK D

Zdecydowana większość oprogramowania napisana została indywidualnie na określone typy komputerów. Natomiast niewiele jest programów, które pracują na wszystkich komputerach.

Obecnie oprogramowanie tworzone jest głównie na komputery „marki” VAX. Jest to zapewne przyszłościowy komputer, w każdym razie w naszych warunkach. Stanowi on ogniwo pośrednie między sprzętem PC, a superkomputerami. Stąd też często nazywa się ten sprzęt — minikomputery, cena wywoławcza (umowna?) około 100 (stu) milionów zł.

Wśród 25 pakietów przetwarzania tekstów tylko LEX (stworzony przez Ace Microsystem) pracuje na wszystkich VAXach, PDP-11 i systemach Rainbow'a. Stało się to możliwe dzięki temu, iż LEX napisany jest w języku D.

LEX był pierwszym zintegrowanym systemem przetwarzania tekstów z jednoczesną obsługą bazy danych. Oryginalnie (w 1979 roku) napisano go na PDP-11. Komputer ten święcił wówczas swe najlepsze dni.

LEX napisany został w języku RPL. Język ten miał pewne ograniczenia, które dosyć szybko dały o sobie znać. Wynikało to z tego, iż kompilator RPLa został napisany w MACRO assemblerze. Decydowało to o tym, że tworzone oprogramowanie mogło wyłącznie pracować na PDP-11. By pokonać ten problem stworzono właśnie język D.

D jest językiem unikalnym, opartym o elementy, które jego twórcy (bardzo młodzi — D. Medford i T. Barnard) nazywają — „tabele decyzyjne”. Język ten rozpowszechniany jest przez National Computing Centre jako Filetab-B.

Języki: RPL i D mają podobną syntaktykę, która pozwala przekładać programy z RPLa na D.

Wydaje się, że najważniejsza jest ta możliwość kompilatora języka D, która polega na tym, iż tworzy on kod maszynowy dla różnych (wybranych) maszyn.

Należy jednak pamiętać, że D nie jest narzędziem na zmianę istniejącego oprogramowania napisanego na PDP-11 na

programy pracujące na VAXie (za wyjątkiem gdy użytkownik napisał swoje oprogramowanie na PDP-11 w jednym z języków: D lub RPL).

Omawiana własność języka D stwarza całkiem nowe warunki dla nowego oprogramowania, które bez żadnych zmian może pracować nie tylko na PDP, VAXie, ale na wielu innych komputerach. Podstawową zaletą języka D okazuje się właśnie to, że użytkownik (posługujący się nim) nie jest skazany na PDP lub VAXa. Ilustracją tego może być przykład przeniesienia programu źródłowego z RIADA na ODRĘ...

Program w języku D składa się ze zbioru tabel decyzyjnych. Zawierają one szereg warunków tworzących kolejne „strony” programu.

Jak już zostało wspomniane, język D był projektowany z zamiarem stworzenia możliwości jego implementacji na różne komputery.

Zbiór dostępnych komend i operatorów oraz odpowiadające im kody maszynowe, które muszą być generowane, są przechowywane jako modyfikowalny zbiór danych — TMC (wybrany kod maszynowy) — zbiór ten wykorzystywany jest przez kompilator.

TMC tworzony jest ze zbioru danych ASCII przez program nazywany D Compiler Compiler. Zbiór ten nazywany jest CGT. Liczba zbiorów tego typu jest równa liczbie procesorów, na które język D może być implementowany.

Specyfikacja zbiorów CGT przewiduje wymagania odpowiedniego procesora — jego architekturę, specjalistyczne zadania itp. Zatem, gdy na rynku pojawia się nowy procesor, tworzony jest odpowiedni zbiór CGT. Następnie używając Compiler Compiler powstaje zbiór TMC — kompilator języka D na nowy procesor jest gotów. Możliwe staje się tworzenie kodu maszynowego dla komputera z nowym typem procesora, przy użyciu do prac programowych oczywiście języka D.

Należy równocześnie dokonać pewnych zabiegów przystosowujących eksploatowany system operacyjny do programów napisanych w języku D. Obecnie dostępne są wersje LEXa na 28 różnych systemów operacyjnych.

Język D okazał się bardzo użyteczny, także na popularny u nas sprzęt mikrokomputerowy oraz na minikomputery (od PDP, VAXa do sprzętu budowanego na Motoroll 68000 i Intelu 8086). Wydaje się, że język D staje się bardziej użyteczny, niż jakikolwiek inny, taki jak na przykład język C.

Opracowane na podstawie:

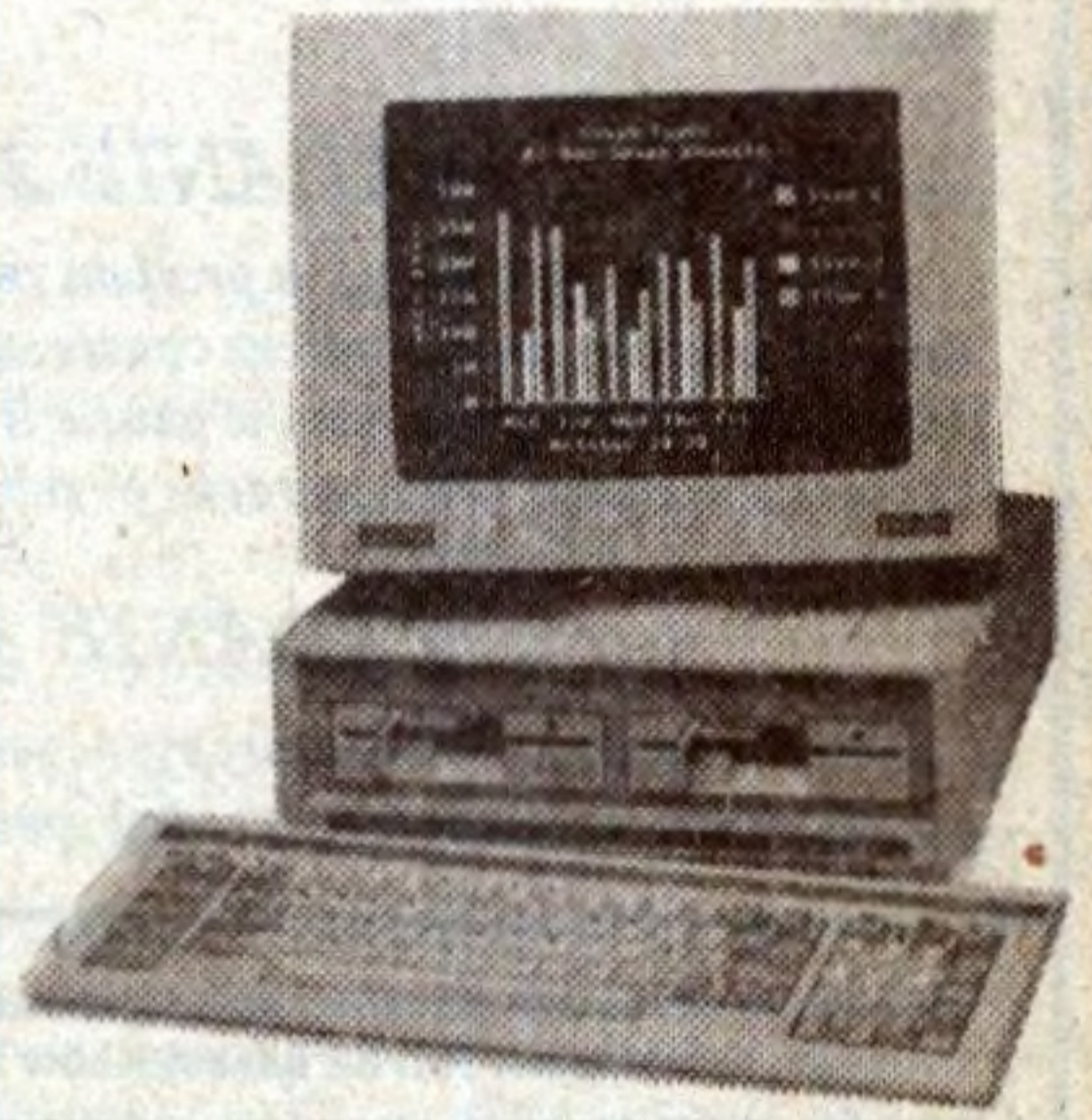
D. Medford, A Language Created for Riding the Range, dec/computing, 20 stycznia 1988.

KRZYŻÓWKA NR 7

		C ₁	Y	L	U	O ₂	T	A	K	
GÓRNA CZĘŚĆ ŚRODKI RODZINY	S	K	ŁÓTWA	S	PRZEPOC	K	MIELONY	L ₅	ODPOCZY- NEK	S
B ₆	A	A	J	U	S	T	E	E	R	
N	WŁAŚCI- CIEL BARU	R	SARWNA WOCZKA TRAWA Z GRUB. WŁOS.	K	PODWIETA	A	NA URZE- JOWE PAPIERY	C ₁₀	NA KUCHEN- NYM STOLE	A
A	M	K	A ₇	O	P ₂₄	K	Z	A	T	
D	SPOSOB	E ₁₇	CZŁONIEK UŁOMNY	K	POTRZEBNY PRZY DOJENIU	E	NADMIAR PRZESADA	B ₁₅	SPRYCIARZ ZIÓMKO	
O	T	L ₂₅	A	S	K	T	Y	R	A	
Ł	UŻYWANY PODCZAS LICYTACJI	E	GLINANY GARNUSZ Z DAWNEGO LAB. CHEM.	L ₁₉	NAD WISBĄ	A	DŁUGA WYPowiedź	R ₁₈	ZDNIERZ RASTARII	J
M	K			O	P	D	A	A	T	
O	KAWATEK CHLEBA	A	MYSLENIE O CZYMŚ	M ₁₄	FORMACJA ROSLINNA Z DN. AMBÓW	A	LEKARSKI LUB PRAWNA	P ₂₂	ŚNIEDZ	Y
R	K	Z	A			R	O	A	N	
O ₄	SAMIEC KACZKI	A	GRA ROZRYWKA	B ₃	DUZA WIOSNOWA ŁÓDZ	K	WYKUIT NA SKÓRZE	S ₁₂	GAGATEK	A
Z	C	W	A	S ₈	A	A	T	A	N	
S ₂₁	PLEMIE	Z	WISPA	P	RODZAJ GRUSZKI	K	KŁOPOT	R	PRZYRODA	A
P	E			A	L ₁₆	S	O	U	T	
A	SZTUCZNE WŁOSY	R	MASZYNA DO GORTO- WANIA		STAN W INDIACH		DRZEWO Z RODZI- NY WIE- RZBOWAT.	K	KARA ZA JAK- IES WYK- ROZCZENIE	A ₂₃
K	U	E ₁₃	T	E	K	R	O	O	P	
K	DOMEK DLA DROBKU	R	TAJEMNIC	S	JEDN. MIA- RY DŁUG. W ASTRONOM.	P	GAZ PALNY	P	"....." DIAMENT	I ₉
I ₂₀	N	K	E	R	A ₁₁	N	A	Ł	Ó	

AMSTRAD

PC 1640 CD



CPC - grafika

```

10 REM ***** GRAFIKA *****
20 DEFINT b,c,x
30 MODE 2
40 b=RND*5 + 1:c=RND*5 + 1
50 ORIGIN 320,200
60 FOR a=0 TO 1000 STEP PI/30
70 x=100*COS(a):MOVE x,x
80 DRAW 200*COS(a/b),200*SIN(a/c)
90 IF INKEY$<" " THEN 30
100 NEXT: GOTO 30
    
```

Litery z ponumerowanych kratek od 1 do 25 utworzą 5 terminów związanych z komputerami, które wystarczy podać jako rozwiązanie zadania przesyłając pod adresem redakcji na kartach pocztowych w terminie do końca listopada naklejając kupon „IKS-a”. Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.



„IKS” — dodatek „Żołnierza Wolności”. Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarna, Romuald Głęb, Włodzimierz Gogołek, Janusz Janiec, Henryk Krasuski, Ireneusz Miernik, Ludwik Piela, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-81 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77. Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 1775. Nr ind. 361682.