

**I** NFORMATYKA  
**K** OMPUTERY  
**S** YSTEMY



Cena 80 zł

DODATEK „ŻOŁNIERZA WOLNOŚCI” NR 11/1987 ISSN 0860—2794



- W szponach Atari
- Oszczędzajmy czas
- Mapa dysku



## W NUMERZE:

**Monitor zamiast bibliotekarki — str. 3**

**Mona Lisa i komputer — str. 4**

**W szponach Atari<sup>(9)</sup> — str. 5—9**

**Mikrokomputery IBM PC<sup>(5)</sup> — str. 13**

**Grafika na ekranie<sup>(2)</sup> — str. 18—19**

**Opowiadanie SF — str. 20**

**Pascal<sup>(7)</sup> — str. 24—25**

**Komputerowy test dla kierowców — str. 28**

**Liga Myślących — str. 31**

**Krzyżówka — str. 32**

UWAGA. Brakujące numery „IKS-a” można jeszcze kupić w księgarniach Wydawnictwa MON, W-wa ul. Grzybowska 77.

W kioskach jest już 3 zeszyt programów. Tym razem przeznaczony jest dla wszystkich posiadaczy mikrokomputerów, które zbudowane zostały na mikroprocesorze Z80. Zeszyt zawiera konstrukcje programowe tego mikroprocesora i z pewnością przyda się tym, którzy chcą wykorzystać wszystkie właściwości swojego mikrokomputera.

### Kolejny błąd ROM-u ZX Spectrum

Napiszemy program:

```
1 CLEAR:REM kasowanie danych
10 LET X$ = „123”
20 PRINT X$
30 SAVE „proba”: DATA x$ ()
50 LOAD „proba”: DATA x$ ()
60 PRINT x$
```

W linii 60 wystąpi błąd. Spowodowany on jest pewnym przeoczeniem ROM-u ZX Spectrum.

Na taśmie została zapisana zmienna tekstowa x\$ jako tablica x\$. Odczyt był identyczny, Spectrum potraktowało x\$ jako tablicę. Ale przecież struktura zmiennej tekstowej różni się od struktury tablicy. Dlatego interpreter, odwołując się w linii 60 do tablicy x\$, napotyka dziwne wymiany tablicy i uruchamia obsługę błędów.

**Krzysztof POŹNIAK**  
Klub „HOBBYTE”

Czy przestajemy się interesować informatyką? Na ile stosunkowo łatwy dostęp do mikrokomputerów pozwolił na upowszechnienie dyscypliny, przed którą bez wątpienia jest jeszcze przyszłość?

Jestem gorącym zwolennikiem informatyki i wierzę, że dobrze oprogramowany komputer może pomóc nie tylko tam, gdzie trzeba przetworzyć ogromne ilości informacji, ale i tam, gdzie szybkość i niezawodność decydują o poprawności podjętej decyzji.

Nie ludźmy się jednak, że komputer pomoże na wszystko. To nieprawda. Komputer to urządzenie, które nie robi niczego, czego nie nauczył go człowiek. A jeśli nawet dobrze zaprojektowany system informacyjny działać będzie w instytucji, w której panuje organizacyjny bałagan, to efektem takiego eksperymentu będzie po prostu jeszcze większy bałagan. Wszystkim menedżerom przypominam, że jeśli trudno im utrzymać równowagę na rowerze, to z pewnością daleko nie pojedą motocyklem.

Trwa kolejna edycja Ligi Myślących. Miłośników zadań logicznych jest wielu — świadczą o tym nadchodzące odpowiedzi. Wiele z nich nosi znamię oryginalności, niektóre rozwiązywane są za pomocą komputerów. Najważniejszy jest jednak efekt.

Tymczasem w redakcji odbyło się wręczenie nagród dla zwycięzców pierwszej edycji Ligi. Dyplomy i nagrody dla najlepszych wręczył redaktor naczelny „Żołnierza Wolności” płk Zdzisław Janoś.

Zwycięzcą został Hubert Nowaczyk, kolejne miejsca zajęli Andrzej Erd, Antoni Goryl, Janusz Morbitzer i Andrzej Wrześniowski.

Niestety tego rodzaju próby już podejmowano, wydano bowiem kilka milionów, zaangażowano sporo pracowników do prac projektowych i ...wkrótce okazało się, że jest to zmarnowany wysiłek. Po nadziejach na postęp pozostał jedynie niesmak i pustka w kasie. Wielu młodych ludzi postąpiło podobnie (tylko Pewex sprzedał ponad sto tysięcy mikrokomputerów ATARI). Wydali spore pieniądze na sprzęt, zainwestowali w programy, tymczasem efekty bywają czasem nikłe. Po okresie zabawy komputer wędruje na szafę, pudełko pokrywa warstwa kurzu i tylko czasami wraca do łask. Wiadomo jednak, że komputer nie tylko zmusza do nauki, ale także wymaga jednoznacznego formułowania problemów i kształci to, czego tak bardzo będzie potrzeba w dojrzałym już twórczym życiu zawodowym. I jeszcze raz powtórzmy: jeśli chcemy działać w przyszłości sprawnie, to nie odkładajmy komputerów na półki, nie rezygnujmy z niczego, co pomoże w przyszłości lepiej organizować nasze twórcze życie.

**Wiesław CETERA**



# Monitor zamiast bibliotekarki

Według wielu specjalistów, społeczeństwo techniczne zostanie zastąpione społeczeństwem informatycznym. Produkcja i przetwarzanie informacji może wysunąć się na pierwsze miejsce przed produkcją towarów. W społeczeństwie amerykańskim blisko połowa czynnej zawodowo ludności pracuje w przemyśle informatycznym — wliczając w to ludzi produkujących komputery, pracujących przy „przenoszeniu” dysków i oczywiście informatyków. Powodem rozrostu przetwarzania danych stał się zalew informacji. Dziś nikt nie jest w stanie przeczytać samych tytułów artykułów ukazujących się w wąskiej dziedzinie. Szacuje się, że w samej medycynie rocznie ukazuje się ponad 20 tysięcy opracowań: pism specjalistycznych, popularnonaukowych, popularnych, biuletynów instytutów naukowych, prac doktorskich, habilitacyjnych, raportów rządowych, organizacji międzynarodowych itp. Nie ma mocnych, którzy byliby w stanie zapoznać się z tymi publikacjami, a przecież drukuje się je po to, by dzielić się doświadczeniami. Aby można było korzystać z już opublikowanego dorobku powstała nowa grupa zawodowa „przeżuwalce wiedzy” — czyli specjaliści zajmujący się lekturą pism, książek itp., to co przeczytają (jeśli napisane jest na odpowiednim poziomie) opracowują pod kątem wprowadzenia do komputerowego banku danych.

O każdej wprowadzonej do banku pozycji przechowywane są kilkanaście bądź kilkadziesiąt informacji. Oprócz podstawowych: tytułu, autora, czasu, języka i miejsca publikacji, szczegółowo opisana jest jej tematyka. Tak, że użytkownik może najczęściej za pomocą odpowiedniego języka wyszukiwania informacji przeszukiwać bazę danych korzystając ze zdań składających się z kombinacji pojedynczych pytań. Daje to kolosalne oszczędności czasu, nie trzeba mozolnie przeszukiwać katalogu biblioteki. Na przykład przed napisaniem wypracowania o Mickiewiczu można sobie zażyczyć spisu opracowań o wieszcu (będzie składał się z kilkudziesięciu stron), z tego spisu (bibliografii) wybrać co lepszych autorów. Można sobie też zażyczyć bibliografię wypowiedzi Słowackiego o Mickiewiczu. Najlepsze efekty uzyskuje się w trakcie przeszukiwania bazy danych w systemie on-line (bezpośrednia konwersacja z komputerem). Użytkownik widzi na ekranie informacje przechowywane w bazie i może zmieniać kombinację pytań, dopóki nie otrzyma bibliografii, która go zadowoli. Problem polega między innymi na tym, że projektanci baz danych nie są w stanie przewidzieć wszystkich potrzeb użytkowników, w związku z czym, przy skomplikowanych i nietypowych kombinacjach pytań mogą powstać rozbieżności między oczekiwaną bibliografią a wydrukowaną przez system.

W pierwszym etapie objęto zautomatyzowanymi bazami danych pojedyncze biblioteki. Początkowo terminale stały gdzieś w holu, potem połączono je siecią transmisji danych z innymi bibliotekami, zaczęły powstawać specjalistyczne bazy danych. Jedną z baz chemicznych obejmuje ponad półtora

miliona związków chemicznych: nazw, wzorów i właściwości, baza uzupełniana jest co dwa tygodnie. Czy wyobrażacie sobie podobne wydawnictwo robione w sposób tradycyjny? Aktualnie istniejące bazy danych dostępne są za pośrednictwem sieci teletransmisji ze wszystkich kontynentów. W Polsce pierwszą biblioteką korzystającą ze zautomatyzowanego systemu informacji bibliotecznej była Główna Biblioteka Lekarska (GBL) w Warszawie. GBL podłączona jest do amerykańskiego systemu **Medline** przez europejski ośrodek systemu w Sztokholmie. Lekarze mogą korzystać z sześciu baz danych między innymi o onkologii, toksykologii (danych toksykologicznych ponad 55 tys. środków chemicznych). Z systemu korzystają pracownicy służby zdrowia — naukowcy i praktycy. Początkowe lęki przed nową techniką szybko mijają, gdy pozna się korzyści. System może wydrukować np. bibliografię wydawnictw w językach: polskim, niemieckim, rosyjskim na temat leczenia odmrożeń oraz oparzeń osób w wieku 75—80 lat, których przyczyną jest benzyna lotnicza (np. ofiar katastrofy lotniczej w górach czy w terenie podbiegunowym). Komputer odszuka wszystkie opracowania, w których wspomniano o podobnych problemach. Z tego spisu wybiera się następnie interesujące pozycje — książki można wypożyczyć, krótsze artykuły skopiować. Lekarz w najmniejszej miejscowości może za pośrednictwem GBL korzystać z opracowań publikowanych na świecie. Teoretycznie nie ma przeszkód w śledzeniu publikacji na całym globie, niestety korzysta się z systemu odpłatnie i to jest jedynym ograniczeniem.

Osoby, które miały kontakt z komputerem zdają sobie sprawę, że przy zarządzaniu tak dużą bazą danych musi pracować komputer o dużej szybkości działania, z olbrzymią pamięcią masową, że muszą być zainstalowane sprawne łącza telekomunikacyjne. W naszych warunkach są to rzeczy nieosiągalne, możemy jedynie stosować rozwiązania cząstkowe, wykorzystujące istniejący sprzęt i doświadczenie ludzi. Duże biblioteki wprowadzają zautomatyzowane systemy informacji bibliotecznej. W GBL dzięki automatyzacji szybciej opracowuje się Polską Bibliografię Lekarską, w Bibliotece Politechniki Warszawskiej opracowany został system przechowywania informacji o materiałach z konferencji i sympozjów naukowych (dlatego, że najnowsze odkrycia omawiane są na konferencjach, dopiero potem trafiają do pism specjalistycznych). Biblioteka Uniwersytetu Warszawskiego przygotowuje się do zautomatyzowania informacji o księgozbiorze dydaktycznym, kilka systemów uruchomiono w Bibliotece Politechniki Wrocławskiej, również Centralna Biblioteka Wojskowa zamierza objąć swoje zbiory systemem informatycznym.

Czytelnikom pozostaje nadzieja, że nie trzeba będzie długo czekać na terminale w bibliotekach, że katalogi zostaną zastąpione przez monitory.

**Dariusz OTTO**

# Dysk optyczny do IBM PC

Po latach badań nad stosowaniem nowych technologii w informatyce, na półkach sklepowych pojawiły się dyski optyczne. Jak podaje lipcowy „Byte” za 2795 dolarów użytkownik IBM PC i kompatybilnych z nim mikrokomputerów może kupić stację dysków optycznych WC 525 produkowanych przez Information Storage Inc. wielkości pudełka do butów (męskich sandałów). O dwieście dolarów tańszy jest model montowany wewnątrz komputera o rozmiarach stacji miękkich dysków.

Obecnie produkowane dyski optyczne dzielą się na trzy rodzaje: jednokrotnego zapisu, CD-ROM oraz dyski wielokrotnego zapisu. Wszystkie one mają wspólną cechę — odczyt znaków czy wgłębień dokonywany jest przez laser tak precyzyjnie ustawiony, że każde zgrubienie powierzchni, każdy pyłek utrudnia odczyt.

Jak do tej pory bariera cen powoduje, że przeciętni użytkownicy muszą zadowolić się dyskami jednokrotnego zapisu, takim jest też produkt ISI. Ten typ dysków nazywany jest WORM (write once, read many — raz zapisujesz, odczytujesz wiele razy); niekasowany; OPROM (optical programmable read only memory — pamięć optyczna zapisana, wielokrotnie odczytywana).

Dyski jednokrotnego zapisu produkowane są w rozmiarach od 5,25 cala do 14 cali średnicy, przechowują od 100 megabajtów od 1,8 gigabajtów na stronie. Ceny stacji dyskowych współpracujących z mikrokomputerami wahają się od 3000 do 30 tysięcy dolarów. WC 525 pracuje na dyskach o średnicy 5,25 cala, na jednej stronie mieści 115 megabajtów. Sprzedawane są również dyskietki 115-megabajtowe jednostronne za 100 dolarów oraz dwustronne o dwukrotnie większej pojemności za 125 dolarów. Dostęp do drugiej powierzchni uzyskuje się po odwróceniu dyskietki.

Zapis informacji następuje w wyniku wypalenia laserem otworów w powierzchni dyskietki zawierającej tellur, jest to proces nieodwracalny. Według ISI ten sposób przechowywania informacji zapewnia jej trwałość przez 20 lat! Inne firmy produkujące dyski optyczne przyjęły odmienne rozwiązania, w wyniku tego nie można stosować zamiennie dyskietek ISI, IBM czy Optotech. Dziennikarz „Byte’a”, który przez dwa miesiące testował WC 525, stwierdził, że jest to dobry produkt, nie miał jednak możliwości sprawdzenia, czy dyskietki zapisywane w jednej stacji dysków mogą być odczytywane w innej.

Przepisywanie zbiorów z dysków magnetycznych na optyczne również wymaga wcześniejszej modyfikacji, resztę wykonują instrukcje dostarczone przez ISI. Producent zadbał o to, by użytkownik kupując WC 525 mógł korzystać z najpopularniejszego systemu operacyjnego na IBM PC — DOS. Na podstawie „Byte’a”

opracował D. O.

# Mona Lisa i komputer

Mówiąc najprościej, przetwarzanie obrazów oznacza „manipulowanie” na obrazie, czego przykładem jest np.: zmiana kontrastu na ekranie odbiornika telewizyjnego. Gdy dysponujemy cyfrowym odwzorowaniem postaci obrazu — komputery dostarczają nam niemalże nieograniczonych możliwości zmiany tej postaci informacji. Wykorzystanie do tego celu maszyn uwarunkowane jest dostępem do technologii pozwalającej „zamienić” obraz na wspomniane „cyfrowe odwzorowanie” — szereg liczb reprezentujących stopień szarości, barwę, jej nasycenie w każdym wyróżnianym przez komputer umownym punkcie obrazu (pikselu). Manipulowanie na tych informacjach (w pamięci maszyny) pozwala „udoskonalać” obrazy, odkrywać w nich nowe informacje, a nawet wskazywać (sugerować) poprawną ich interpretację.

Jest to pierwszy krok w rozwoju maszyn, by mogły „widzieć”. Początkowo przetwarzanie obrazów było marginesem praktycznych zastosowań informatyki — dużych wówczas i drogich komputerów. Obecnie — w dobie mikrokomputerów — okazało się to bardzo ciekawym przedmiotem zainteresowań twórców nowoczesnej informatyki. Podobnie jak przetwarzanie liczb i przetwarzanie tekstów — przetwarzanie obrazów, to kolejny etap w upowszechnianiu zastosowań komputerów. Wielu uważa ten etap za najbardziej znaczący w historii elektronicznego przetwarzania informacji.

Bardzo ciekawym kierunkiem tych nowych możliwości komputerów jest wspomaganie pracy konserwatorów dzieł sztuki.

Większość artystycznej spuścizny, którą ulokowano w muzeach, to „sztuka wizualna”

typu: malowidła, rysunki, druki i pismo klinowe oraz trójwymiarowe obiekty. Niestety, bardzo mało z tych dzieł wygląda dzisiaj tak jak wyglądało bezpośrednio po ich stworzeniu. Temperatura i światło spowodowały wyblaknięcie farb i ściemnienie lakierów. Swoją niszczycielski wpływ ma także powietrze, które na skutek reakcji chemicznych powoduje często nieodwracalne zmiany obrazów, rzeźb i innych dzieł sztuki — doświadcza tego na co dzień nasz Kraków. Dzieła zniszczenia dopełnia przypadkowa, a czasem i rozmyślna działalność człowieka. Często na przykład artyści lub konserwatorzy obrazów pokrywali istniejące obrazy „nowym dziełem” lub je „twórczo” poprawiali.

Rozwój techniki ostatnich kilku dekad zrewolucjonizował czynności związane z konserwacją dzieł sztuki, będących przedmiotem zainteresowań archeologów, historyków sztuki i antropologów. Szczególnie przydatne okazały się, nie niszczące badane go obiektu promienie Roentgena (x). Uznanie zdobyło także określanie wieku dzieła na podstawie okresu rozpadu węgla, zawartego w tworzących obiekt składnikach (np. w farbie). Techniki te szybko wzbogacono o wiele innych — między innymi o fotoakustyczną spektroskopię, termoluminescencję, badania laserowe. W efekcie można uzyskać bardzo dużą liczbę obiektywnych danych o konserwowanym bądź badanym obiekcie.

W restauracji dzieł sztuki innowacje te pomogły między innymi odróżnić autentyczne fragmenty przedmiotów artystycznych (by je zachować) od materiałów konserwatorskich (by je usunąć). Metody te są źród-

łem obiektywnych danych, mówiących o postaci dzieła, o materiale, z jakiego jest stworzone, bez względu na jego formę i treść. Najczęściej jednak, właśnie forma i treść okazują się jedynym kluczem do poprawnej analizy całego dzieła. Ocena i zrozumienie tych bardzo „esencjonalnych” charakterystyk decyduje często o poprawności zabiegów konserwatorskich.

Restauracja obrazów opiera się na ludzkich oczach — wykorzystuje mikroskop optyczny, kamerę — częściej niż promienie X. Stąd też wizualna ocena konserwowanego dzieła przesądza o powodzeniu tego typu prac.

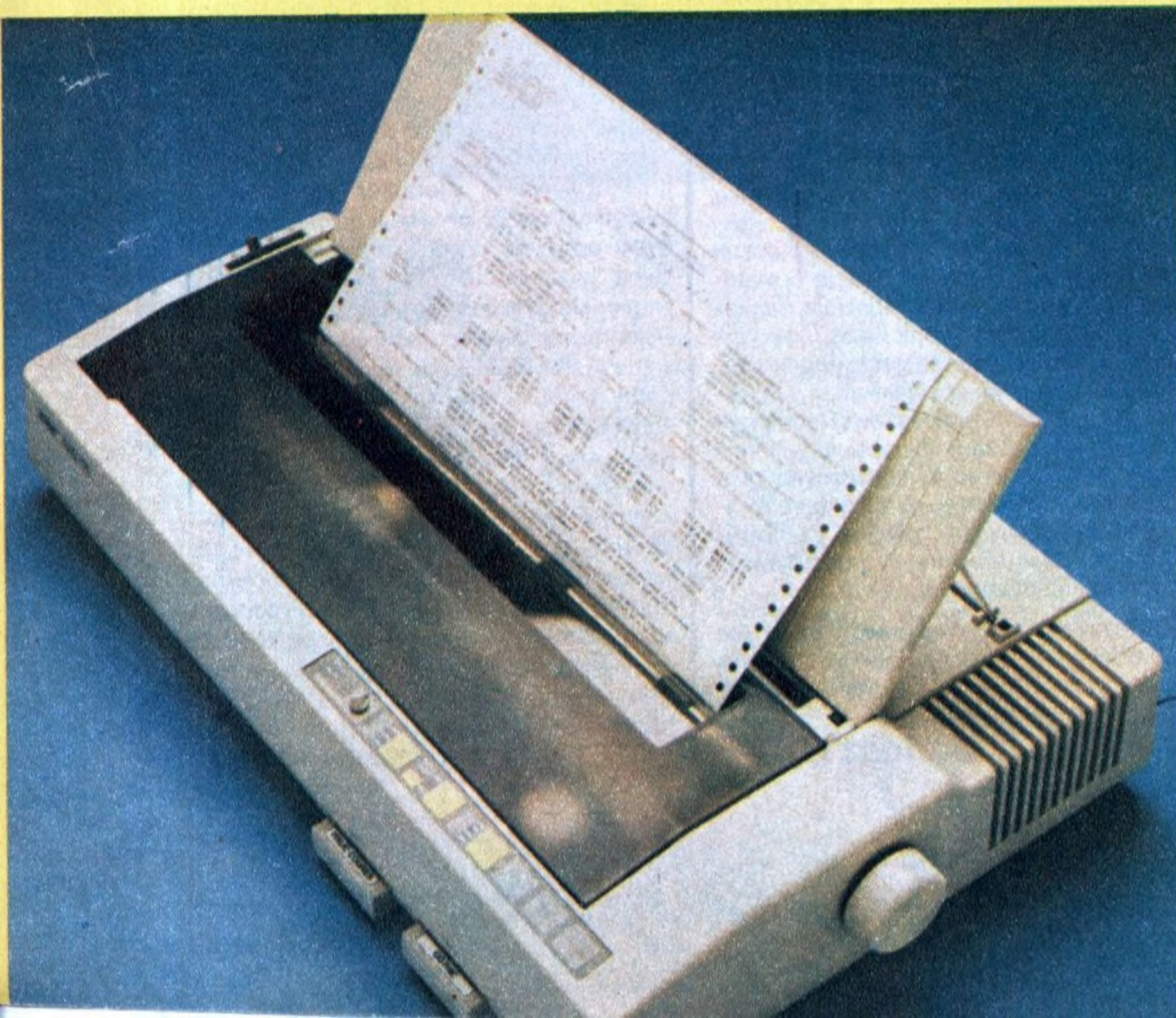
Wspomniany na wstępie dostęp do cyfrowej postaci obrazów znacznie rozszerzył zakres zastosowań sprzętu informatycznego — od astronomii przez diagnostykę medyczną do konserwacji dzieł sztuki.

Niespodziewanym „pośrednikiem” między nowoczesnymi technologiami komputerowymi a sztuką stał się Leonardo da Vinci. Wśród bardzo wielu ciekawych zdarzeń w jego bogatym życiu były także takie, w których nieporozumienia i zatargi ze zleceniodawcą nie pozwalały mu kończyć swych prac. Działo się to w wypadku „Bitwy pod Anghiari”. Giorgio Vasari zamiast tego obrazu (nie dokończonego) stworzył własne dzieło o podobnej treści. Dzisiaj powstał problem, czy można odtworzyć to, czego dokonał Mistrz Leonardo?

W 1975 roku temat ten podjął J. F. Asmus — zdecydowały o tym wiarygodne przesłanki o niezniszczeniu (tylko zamalowaniu) przez Vasariego fragmentów malowidła Leonardo da Vinci. Sprawę komplikował olbrzymi wymiar dzieła, którego zaledwie kilka niewielkich fragmentów wykonał sam Mistrz. W pracach tych badacze wykorzystali termowizję, promienie Roentgena, aktywność neutronową, przewodnictwo elektryczne i mikrofałę. Ostatecznie wybrano ultradźwiękową echolokację (urządzenie podobne do sonaru — hydrolokatora). Technika ta spełniła oczekiwania badaczy i pozwoliła wyróżnić fragmenty wykonane przez Leonarda da Vinci.

Duża część zabiegów konserwatorskich polega na sprawdzaniu i interpretacji fotografii i radiografów. Często promienie X są wykorzystywane do odsłonięcia wcześniejszych kompozycji ukrytych pod aktualną powłoką — innym obrazem. Generalnie obrazy widziane tą techniką bywają mieszaniną powierzchniowego obrazu, „poprzedniego” obrazu oraz struktury nośnika (drewno lub płótno). Po wielu badaniach okazało się, że „czyszczenie” tak otrzymanych zdjęć jest możliwe dzięki pewnym operacjom matematycznym na cyfrowych odwzorowaniach obrazów (stosując „filtrowanie” danych o obrazie szybką transformatą Fouriera). Usunięto w ten sposób interferencję tła, które daje nośnik obrazu — płótno lub drewno. Rezultaty okazały się bardzo dobre — szarość „zamazująca” obraz została wyeliminowana.

Przedstawione uwagi dotyczą rycin „kreskówek” lub obrazów monochromatycznych. W konserwacji obrazów jest to zaledwie szczyt góry lodowej, zdecydowana większość dzieł zgromadzonych w muzeach jest bowiem polichromatyczna. Obraz tworzą nie tylko trzy składowe (czerwień, zieleń i kolor niebieski), ale także ich nasycenie, odcień. Duże znaczenie ma również wizualna percepcja obrazu, której mechanizmy są niezmiernie złożone — decydują o tym, że uważamy piękno lub kiczowatość obrazu.



# Atari i dźwięk

Tomasz MROWIEC  
Ludwik PIELA

**Komputery Atari serii XL i XE, jak wszystkie komputery domowe, mają możliwość generowania dźwięku. Dysponują czterema niezależnie sterowanymi kanałami, które mogą być używane jednocześnie. Instrukcja SOUND języka BASIC, za pomocą której wybieramy numer kanału, wysokość dźwięku, brzmienie i głośność, w niewielkim tylko stopniu umożliwia wykorzystanie wszystkich zalet układu generowania dźwięku. Niezbędne jest dokładne poznanie tego układu i różnych sposobów jegoysterowania.**

Prawdopodobnie jedynym z najstojniejszych obrazów jest Mona Lisa (La Gioconda) Leonarda da Vinci (eksponowana w pańskim Luwrze). Obecnie obraz ten jest ledwo cieniem swej oryginalnej postaci. Przyczyna? Czas i... niestety, zabiegów wielu konserwatorów. Przystępując do konserwacji obrazu jego wysokiej jakości fotografia została „zamieniona” („digityzowana”) na liczby (dla każdego z 6 milionów wyróżnionych punktów i dla każdego z trzech podstawowych kolorów).

Wstępne przetwarzanie danych odwierciedlających postać obrazu zmienilo kolor nieba z brązowego na niebieski, skóry niewiasty z żółtego na alabastrowy, a z sukni bohaterki dzieła wydobyto odcień głębokiej, pięknej zieleni.

Do dalszych, bardziej złożonych badań, wykorzystano zdjęcia (10 × 12,5 cm — piksele o wymiarze 0,02 mm!) fragmentów obrazu. W efekcie otrzymano trzy zbiory danych dla każdego z podstawowych kolorów (RGB). Każdy zbiór zawierał dane o 2600 liniach, po 2200 pikseli w jednej linii. Każdy piksel opisywany był wartością od 0 do 255 — w zależności od jasności każdego z trzech podstawowych kolorów. Zero prezentowało czerń, a 255 maksymalną jasność koloru. Przeprowadzone następnie obliczenia pozwoliły na dokonywanie korekt kolorów. W efekcie każdy niebieski piksel był „dzielony” przez 0,4, czerwony przez 0,2, a czarny przez 1,0.

Większość profesjonalnych artystów, oglądających wynik tej „korekty”, zgodziło się z jej rezultatami. Uznali „poprawione” kolory za bardziej naturalne. Wiek obrazu został „skompensowany”.

Zwrócono także uwagę na odbłaski („świecenie”), które zredukowano (stosując także szybką transformację Fouriera).

Po tych komputerowych zabiegach dostrzeżono wiele nowych detali obrazu Mony Lisy, na przykład fragmenty naszyjnika? Nasunęło to wątpliwości, czy są one dziełem Leonarda, restauratorów, czy po prostu są to zniszczenia?

Dokonano jeszcze wielu innych obliczeń eliminujących lub wskazujących krytyczne miejsca obrazu. Stanowiło to między innymi wskazówkę (naszyjnik) prawdopodobnego pierwowzoru Mony Lisy — La Muta Raphaela.

Coraz częściej w muzeach świata wykorzystywane są komputery do wspomagania zabiegów konserwatorskich, tym bardziej że umożliwia to powszechnie dostępny sprzęt nawet już klasy IBM PC XT (rozdzielczość 1024x1024).

Kończąc warto podkreślić problemy psychologicznych oporów przy stosowaniu maszyn w procesach konserwatorskich. Dotyczy to zarówno profesjonalistów, jak i generalnie publiczności zwiedzającej muzea.

Po przedstawieniu omawianych efektów „korekty” Mony Lisy, pewna mistrzyni „sztuki konserwatorskiej”, stwierdziła, że „nigdy mnie nie przekonacie, że Leonardo namalował te drzewa na błękitnym tle”. Malowidła są w rzeczywistości górami, zniekształcenia kolorów myślą bowiem nawet wybitnych profesjonalistów w poprawnej interpretacji obrazów.

W.G.

Opracowano na podstawie:  
J. F. Asmus, Digital Image Processing in Art Conservation, BYTE, march 1987.

W poniższym artykule spróbujemy zapoznać Czytelników z możliwościami układu generowania dźwięku oraz sposobami sterowania nim. Zaprezentujemy programy umożliwiające praktyczne zbadanie większości tych możliwości. Należy jednak pamiętać, że sterowanie dźwiękiem jest skomplikowane, trudne do badania i wymaga konstruowania dosyć rozbudowanych programów.

Do generowania dźwięku przeznaczony jest układ scalony POKEY, który obsługuje także szynę szeregową wejścia/wyjścia oraz klawiaturę. Ma on kilka rejestrów (w pamięci RAM), które są używane podczas generowania dźwięku. Ich nazwy, adresy oraz przeznaczenie są następujące:

- AUDF1 (53760) — rejestr częstotliwości kanału 1
- AUDC1 (53761) — rejestr sterujący kanału 1
- AUDF2 (53762) — rejestr częstotliwości kanału 2
- AUDC2 (53763) — rejestr sterujący kanału 2
- AUDF3 (53764) — rejestr częstotliwości kanału 3
- AUDC3 (53765) — rejestr sterujący kanału 3
- AUDF4 (53766) — rejestr częstotliwości kanału 4
- AUDC4 (53767) — rejestr sterujący kanału 4
- AUDCTL (53768) — rejestr sterujący trybem działania

Aby poprawnie działał układ POKEY musi być zainicjowany. Wymagane to jest po dowolnej operacji na szynie szeregowej (współpraca z magnetofonem, stacją dysków lub drukarką). W celu zainicjowania POKEY w BASIC'u należy wykonać zerową instrukcję SOUND; to jest SOUND 0,0,0,0. W języku maszynowym należy zapamiętać 0 w AUDCTL (\$D208 = 53768) i 3 w SKCTL (\$D207 = 53775).

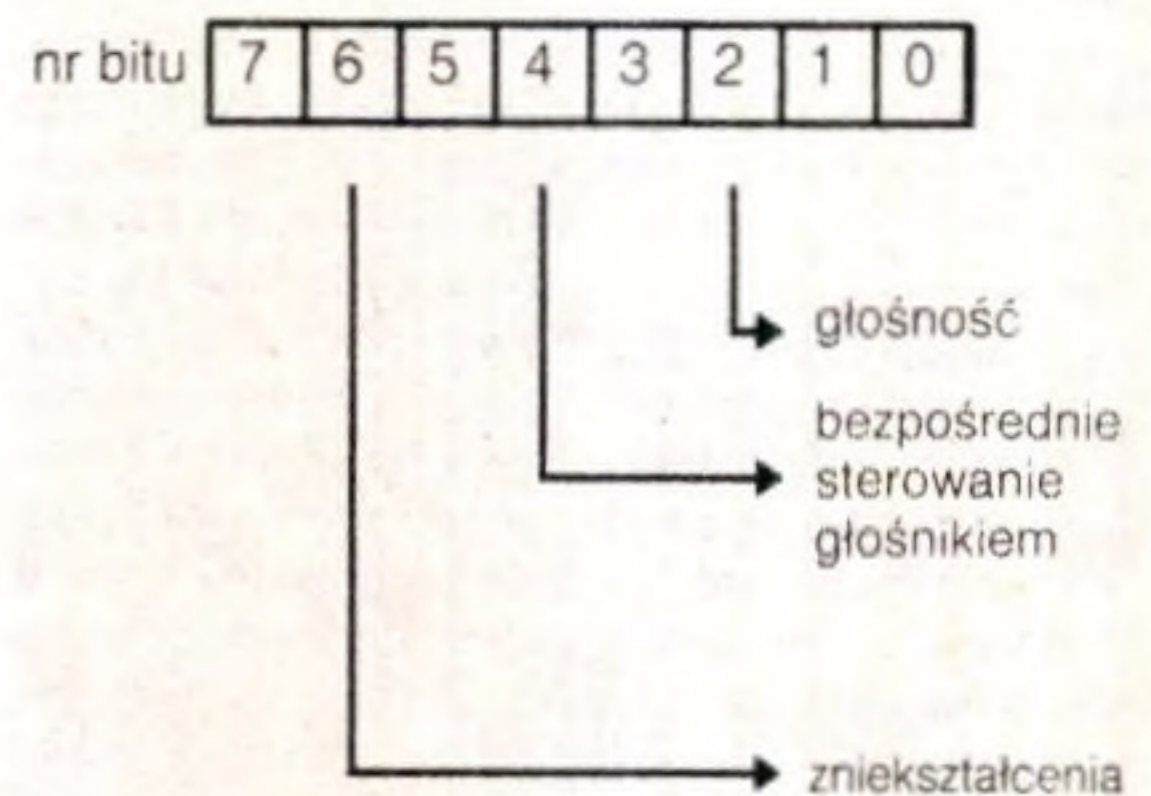
Każdy kanał dźwiękowy ma odpowiadający mu rejestr częstotliwości AUDF1—4, który steruje wysokością generowanego dźwięku. Rejestr częstotliwości, o długości 1 bajta, zawiera liczbę N używaną do sterowania układami dzielników częstotliwości, w celu wytworzenia różnych częstotliwości wyjściowych. Im większa jest wartość N, tym rzadsze są impulsy wyjściowe i tym niższa częstotliwość dźwięku, która może być określona następującym wzorem:

$$f_{wy} = f_z / 2 \times (N + 1)$$

gdzie:  $f_z$  — jest częstotliwością zegara, a N zawartością rejestru częstotliwości. Normalną częstotliwość zegara 64 kHz (a dokładniej 63921 Hz) dysponujemy zakresem częstotliwości od 125 Hz do 32 kHz.

Każdy kanał ma także rejestr sterujący AUDC1—4 umożliwiający ustawienie głośności (cztery młodsze bity) i brzmienia dźwięku (zawartości zniekształceń).

Przydział bitów tego rejestru jest następujący:



Sterowanie głośnością każdego kanału dźwiękowego jest proste. Młodsze bity rejestru sterującego zawierają liczbę czterobitową, która określa głośność dźwięku. Istnieje zatem 16 różnych ustawień głośności. Zera na wszystkich bitach oznaczają głośność zerową, a 15 oznacza maksymalną. Suma głośności czterech kanałów nie powinna przewyższać 32, ponieważ może spowodować przesterowanie wyjścia dźwiękowego.

Każdy kanał dźwiękowy ma w swoim rejestrze sterującym wyróżnione trzy bity. Służą one do sterowania zniekształceniem dźwięku używanym do tworzenia specjalnych efektów akustycznych. Dzięki tej możliwości łatwo jest uzyskać ogromną różnorodność dźwięków, od huków, szczęków i skrzeków do trzasków i szmerów.

Zniekształcenie polega na zmianie kształtu fali (jest ona zawsze prostokątna), lecz na usuwaniu z ciągu wybranych impulsów. Aby w pełni wyjaśnić na czym polega zniekształcenie dźwięku, musimy zrozumieć działanie tak zwanych liczników wielomianowych (polynomial counters). Są one używane jako źródło impulsów losowych stosowanych w generowaniu zakłóceń. Liczniki te wykorzystują rejestry przesuwne pracujące z częstotliwością 1,79 MHz. Zawartość rejestru jest przesuwana w prawo, każdy wychodzący bit poddawany jest pewnej obróbce i wprowadzany ponownie do rejestru. Wytwarza to niby-losowy ciąg bitów na wyjściu rejestru przesuwne.

Liczniki te nie są w pełni losowe, ponieważ po pewnym czasie powtarzają sekwencje bitów. Jak można oczekiwać, okres powtarzania zależy od długości rejestru przesuwne; to znaczy im dłuższy rejestr, tym dłuższy okres powtarzania.

Zakłócenie dźwięku w komputerach Atari uzyskuje się poprzez wprowadzenie impulsów z liczników wielomianowych i impulsów podstawowych do układu wyboru. W rzeczywistości jest to komparator cyfrowy, który przepuszcza na wyjście tylko te sygnały, które są zgodne z impulsami losowymi. Efekt końcowy polega na losowym eliminowaniu pewnych impulsów z układu dzielnika częstotliwości.

Jak wspomnieliśmy, impulsy wyjściowe liczników wielomianowych mają charakter cykliczny. W

Nuta	O k t a w a																	
	1		2		3		4		5		6		7		8		9	
	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.	16-b.	8-b.
C	27357		13675		6834	243	3414	121	1703	60	848	30	421	14	207		100	
C	25821		12907		6450	230	3222	114	1607	57	800	28	397		195		94	
D	24372		12182		6088	217	3040	108	1517	53	755	26	374		183		88	
D	23003		11498		5746	204	2869	102	1431	50	712	25	353		173		83	
E	21712		10852		5423	193	2708	96	1350	47	672	23	332		163		78	
F	20493		10243		5118	182	2555	91	1274	45	634	22	313		153		73	
F	19342		9668		4830	172	2412	85	1202	42	598	21	295		144		69	
G	18256		9125		4559	162	2276	81	1134	40	564	19	278		136		64	
G	17231		8612		4303	153	2148	76	1070	37	532	18	262		128		60	
A	16264		8128		4061	144	2027	72	1010	35	501	17	247		120		57	
A	15351		7672		3832	136	1913	68	953	33	473	16	233		113		53	
H	14489		7241		3617	128	1805	64	899	31	446	15	219		106		50	

Uwaga: 1 — kontra; 2 — wielka; 3 — mała; 4 — rozkreslna; 5 — dwukreslna; 6 — trzykreslna; 7 — czterokreslna; 8 — pięciokreslna; 9 — sześciokreslna

rezultacie zakłócony dźwięk także będzie miał taką samą postać powtarzania. Umożliwia to układowe generowanie takich dźwięków, jak brzęczenie, warkot silnika i podobnych, mających charakter cykliczny.

Komputery Atari wyposażone są w trzy liczniki wielomianowe, o różnych długościach, które mogą być łączone na wiele sposobów w celu wytworzenia interesujących efektów dźwiękowych. Krótsze liczniki (o długości 4 i 5 bitów) powtarzają się wystarczająco często, aby wytworzyć dźwięki naśladujące warkot, które tak szybko narastają i opadają; podczas gdy dłuższy licznik (o długości 17 bitów) tak rzadko się powtarza, że żaden wzór zakłócenia nie może być wyraźnie rozpoznany. Licznik 17-bitowy może być używany do naśladowania dźwięków eksplozji i krzyku. Jest on na tyle nieregularny, że może służyć do generowania „białego szumu”.

Każdy kanał dźwiękowy oferuje sześć różnych kombinacji trzech liczników wielomianowych. Sterują tym trzy bity rejestrów AUDC 1—4 następująco:

7	6	5	4	3	2	1	0
0	0	0					
0	X	1					
0	1	0					
1	0	0					
1	X	1					
1	1	0					

5-bitowy i 17-bitowy  
5-bitowy  
5-bitowy i 4-bitowy  
17-bitowy  
bez liczników  
4-bitowy

Każda kombinacja liczników oferuje unikalny dźwięk. Ponadto zniekształcone dźwięki mogą brzmieć różnie przy różnych częstotliwościach. Z tego powodu niezbędne jest wykonanie wielu prób w celu znalezienia kombinacji zniekształceń i częstotliwości, które dadzą pożądany efekt dźwiękowy. Poniżej prezentujemy tablicę, która ułatwi nam rozpoczęcie prób.

Bit 4 rejestru AUDC 1—4 określa specjalny tryb pracy. Gdy jest on ustawiony wartość głośności (bity 0—3 AUDC 1—4) wysyłana jest bezpośrednio do głośnika; nie jest on modulowany częstotliwością w rejestrach AUDF 1—4.

Dla pełnego zrozumienia użycia tego trybu działania, musimy przypomnieć sobie jak pracuje głośnik i co się z nim dzieje po odebraniu impulsu. Każdy głośnik ma membranę, która przesuwa się. Pozycja membrany w dowolnej chwili jest wprost proporcjonalna do napięcia, które przychodzi z komputera. Jeśli wysyłane napięcie jest zerowe, wtedy głośnik jest w pozycji spoczynkowej. Kiedy membrana zmienia pozycję, wtedy przesuwa powietrze, co jest odbierane przez nasze ucho jako dźwięk.

Z definicji impulsu wiemy, że składa się on z narastającego napięcia, a następnie opadającego. Jeśli do głośnika wyślemy pojedynczy impuls, powstanie fala powietrza, która może być wykryta przez ucho jako trzask. Przykładem może być realizacja następujących instrukcji:

**POKE 53761.31: POKE 53761.16**

Ciąg impulsów (lub fala) może wprawić membranę w stały ruch i słyszane będzie ciągłe brzęczenie lub dźwięk. Im szybciej wysyłane są impul-

sy, tym wyższy dźwięk. W taki sposób komputer wytwarza dźwięk w głośniku.

Należy zaznaczyć, że w trybie bezpośredniego sterowania głośnikiem, wysłana wartość głośności nie wraca automatycznie do zera, lecz pozostaje stała do czasu, aż zmieni ją program. Dla wytworzenia dźwięku głośność powinna być modulowana wystarczająco często. Spróbujemy teraz wykonać następujące instrukcje, uważnie nasłuchując po każdej:

**POKE 53761.31**  
**POKE 53761.31**

Za pierwszym razem usłyszymy trzask, zgodnie z oczekiwaniem. Głośnik wysunie się i przesunie powietrze. Lecz za drugim razem nic się nie stanie. Membrana głośnika była już w pozycji wysuniętej; kolejny rozkaz wysunięcia niczego nie zmieni, zatem nic nie usłyszymy. Spróbujmy teraz:

**POKE 53761.16**  
**POKE 53761.16**

Tak jak poprzednio, za pierwszym razem usłyszymy trzask, gdy membrana przesuwa się do pozycji spoczynkowej; za drugim razem nic nie usłyszymy, ponieważ głośnik był już w pozycji spoczynkowej.

Jak widać, w tym trybie pracy program ma pełne sterowanie pozycją membrany głośnika w dowolnym czasie. Chociaż powyższe przykłady są tylko binarne (włączone lub wyłączone), nie jesteśmy niczym ograniczeni do tylko tego typu modulacji głośnika. Możemy ustawić membranę w dowolnej z 16 różnych pozycji.

Na przykład, prosta fala trójkątna (podobna do fal wytwarzanych przez instrumenty dęte) może być utworzona poprzez wysłanie wartości głośności 8, a następnie 9, 10, 11, 10, 9, 8, 7, 6, 5, 6, 7 i znowu 8; i powtarzanie tego ciągu bardzo szybko. Poprzez wystarczająco szybką zmianę głośności może być utworzona dowolna postać fali. Przy pomocy tej techniki można łatwo uzyskać syntezę mowy. Wymaga to użycia języka asemblera.

Oprócz rejestrów niezależnego sterowania kanałami (AUDC 1—4), istnieje bajt (AUDCTL) dotyczący wszystkich czterech kanałów. Każdy bit w AUDCTL przydzielony jest do określonej funkcji: AUDCTL (\$D208 = 53768).

Przed przystąpieniem do wyjaśniania opcji AUDCTL musimy wyjaśnić inne pojęcie: taktowanie (ang. clocking). Do synchronizacji milionów operacji wewnętrznych, realizowanych w każdej sekundzie w komputerze, używany jest ciąg impulsów. Jest on generowany przez centralny zegar działający z częstotliwością 1.79 MHz. Istnieje także kilka zegarów pomocniczych o niższych częstotliwościach. Rejestr AUDCTL umożliwia nam wybranie częstotliwości impulsów wejściowych układu dzielenia przez N. Poprzez zmianę zegara drastycznie możemy zmienić częstotliwość wyjściową. Można to łatwo zilustrować prostym wzorem podawanym w niektórych podręcznikach (np.

De Re Atari):

$$f_{wy} = \frac{f_{zeg}}{N}$$

gdzie:  $f_{wy}$  — częstotliwość wyjściowa  
 $f_{zeg}$  — częstotliwość używanego zegara  
N — zawartość rejestru AUDF1—4

W rzeczywistości zależność częstotliwości wyjściowej od częstotliwości zegara i zawartości rejestru AUDF1—4 ma nieco inną postać:

$$f_{wy} = f_{zeg} / 2 \times (N + 1)$$

Ustawienie bitu 1 rejestru AUDCTL przetrąca zegar 64 kHz na 15 kHz. Podobnie, poprzez ustawienie bitów 5 lub 6 możemy taktować, odpowiednio kanały 3 lub 1, częstotliwością 1.79 MHz. Otrzymamy wtedy dźwięk znacznie wyższy, co można zilustrować następującym przykładem:

**SOUND 0.255.10.8** — włączenie kanału, niski ton

**POKE 53768.64** — ustawienie bitu 6 AUDCTL

Osiem bitów w rejestrach sterujących częstotliwością wydaje się oferować więcej niż wystarczającą rozdzielczość dla wybrania dowolnej częstotliwości. Jednak wprawne ucho może zauważyć, że generowane dźwięki, odpowiadające na przykład gamie muzycznej, nie są zbyt dostrojone i nie mogą być. Niech w rejestrze częstotliwości będzie liczba N = 72 (która według instrukcji daje nutę A), wtedy uzyskamy częstotliwość 437.8 Hz. Dla N = 71 otrzymamy 443.9 Hz. A częstotliwość nuty A jest 440 Hz i takiej częstotliwości nie możemy uzyskać.

Istnieją również sytuacje, w których każde ucho może usłyszeć, że ośmiobitowy rejestr częstotliwości jest niewystarczający. Spróbujmy wykonać następujące instrukcje:

**FOR I = 255 TO 0 STEP 1: SOUND 0.1.10.8: NEXT I**

Dźwięk początkowo podnosi się łagodnie, lecz gdy zbliża się do końca swojego zakresu zauważalne są znaczne zmiany częstotliwości. Nie może być inaczej, jeśli dzielimy częstotliwość zegara przez coraz mniejsze liczby. 15 kHz podzielone przez 255 ma zbliżoną wartość, jak 15 kHz podzielone przez 254; lecz 15 kHz podzielone przez 2, znacznie różni się od 15 kHz podzielonych przez 1.

Z tego powodu efektywny zakres dla muzyki ograniczony jest do niepełnych czterech oktaw, ponieważ dokładność dostrojenia odtwarzanych nut staje się postępująco gorsza, gdy częstotliwość staje się wyższa. Przyczyną niedokładności dostrojenia jest ograniczona liczba wartości możliwych do użycia jako dzielniki (256 unikalnych wartości).

Na szczęście Atari dostarcza rozwiązanie tego problemu: dzielniki 16-bitowe, które umożliwiają wybranie 65536 różnych częstotliwości wyjściowych. Lecz dokładniejsze dostrojenie nie przychodzi za darmo. Szesnastobitowe dzielniki uzyskuje się poprzez połączenie rejestrów częstotliwości dwóch kanałów dźwiękowych (AUDF1 z

Głos	Instrukcja SOUND	Równoważne instrukcje POKE
1	SOUND 0, N, Z, G	POKE 53760, N: POKE 53761, 16 * Z + G
2	SOUND 1, N, Z, G	POKE 53762, N: POKE 53763, 16 * Z + G
3	SOUND 2, N, Z, G	POKE 53764, N: POKE 53765, 16 * Z + G
4	SOUND 3, N, Z, G	POKE 53766, N: POKE 53767, 16 * Z + G

```

FK 1 REM *****
WZ 2 REM * Program nr 1 *
FM 3 REM *****
RE 4 GRAPHICS 0:?" :? :? :? "OD JA
KIEJ CZESTOTLIWOSCI CHCESZ"?
"ZACZAC " :INPUT CZEST
AG 5 AUF3=INT(63921/(2*CZEST))-1+
0.5)
ZY 6 AUF2=INT((1789790/(2*CZEST)
-7)/256)
DG 7 AUF1=INT((1789790/(2*CZEST)
)-7-256*AUF2+0.5)
NO 8 IF CZEST>125 AND CZEST<16000
THEN 10
XQ 9 ? "Dopuszczalny zakres 125 d
o 16000." :? "Spróbuj znowu." :
INPUT CZEST:GOTO 5
NJ 10 GRAPHICS 0:?" Manipulator #
1 steruje czestotliwos-
GZ 15 ? "cia o wysokiej rozdzielc
zosci." :? "Nacisnij przycisk,
aby otrzymac" :? "dzwiek."
AZ 20 ? "Przesun w gore lub dol d
la wolnej" :? "zmiany czestotli
wosci lub w lewo" :? "prawo dl
a szybkiej zmiany."
LA 30 ? "Manipulator #2 steruje n
iska roz-" :? "dzielczoscia, pr
zycisk daje dzwiek "
OO 40 ? "lub sora/dol dla zmiany."
FH 55 SOUND 0,0,0,0:SOUND 1,0,0,0
:SOUND 2,0,0,0:SOUND 3,0,0,0
JF 100 POKE 752,1
FZ 110 POKE 53768,80
UQ 130 POKE 53763,10*16+15
VU 140 POKE 53765,10*16+15
BO 160 IF STICK(0)=13 THEN AUF1=
AUF1+1:IF AUF1>255 THEN AUF
2=AUF2+1:AUF1=0
NY 170 IF STICK(0)=14 THEN AUF1=
AUF1-1:IF AUF1<0 THEN AUF2=
AUF2-1:AUF1=255
PK 180 IF STICK(0)=11 THEN AUF2=
AUF2+1
UE 190 IF STICK(0)=7 THEN AUF2=A
UDF2-1
TJ 220 IF STICK(1)=13 THEN AUF3=
AUF3+1
WX 230 IF STICK(1)=14 THEN AUF3=
AUF3-1
XD 240 IF STRIG(0)=0 THEN POKE 53
760,AUF1:POKE 53762,AUF2:GOT
O 260
MY 250 POKE 53760,0:POKE 53762,0
XL 260 IF STRIG(1)=0 THEN POKE 53
764,AUF3:GOTO 280
UA 270 POKE 53764,0
UP 280 CZWY2=1789790/(2*(AUF2*25
6+AUF1+7))
VP 290 CZWY3=63921/(2*(AUF3+1))
ML 300 POSITION 1,10
ZO 310 ? :? "WYSOKA ROZDZIELCZOSC
" :?
OR 320 ? "CZEST = 1789790/(2*(A
UDF2)*256+AUF1+7)) " :?
UG 325 ? " " = " :CZWY2
SE 330 ? :? :? "NISKA ROZDZIELCZO
SC" :?
VD 340 ? "CZEST = 63921/(2*(AUF
3)+1)) " :? :? " " =
":CZWY3
LK 500 GOTO 100
FL 2 REM *****
XW 5 REM * Program nr 2 *
FP 6 REM *****
HR 100 DATA 520,584,655,694,779,8
74,982,1040
IF 110 DIM DZWIEK(8),A$(1)
TL 120 FOR I=1 TO 8:READ X:DZWIEK
(I)=X:NEXT I
RE 150 GRAPHICS 0:?" :? " NORMAL
NA WYSOKA ROZDZIEL." :?
" -----
"
KY 160 ? " P NUTA P1
P2 NUTA" :? " ----
"
JT 170 POKE 752,1
ZJ 200 REM GENEROWANIE 'NORMALNEG
O' DZWIEKU ATARI
PY 210 FOR J=1 TO 8
GV 220 P=INT(63921/(2*DZWIEK(J))-
1)
ZT 230 SOUND 2,P,10,8
DT 240 DZWIEK=63921/(2*(P+1))
LW 250 POSITION 3,J+6:?" P" :DZ
WIEK
ZI 260 FOR W=1 TO 200:NEXT W:NEXT
J
TG 270 SOUND 2,0,0,0:FOR W=1 TO 5
00:NEXT W
YH 300 REM DZWIEK O WYSOKIEJ ROZD
ZIELCZOSCI
GO 305 POKE 53768,80
PW 310 FOR J=1 TO 8
AN 320 P2=INT((1789790/(2*DZWIEK(
J))-7)/256)
FL 330 P1=INT(1789790/(2*DZWIEK(J
))-7-256*P2+0.5)
RE 340 POKE 53760,P1:POKE 53762,P
2:POKE 53763,(16*10)+8
XD 350 DZWIEK=1789790/(2*(256*P2+
P1+7))
ZU 360 POSITION 21,J+6:?" P1:POSIT
ION 25,J+6:?" P2" :DZWIEK
CC 370 FOR W=1 TO 150:NEXT W:NEXT
J
NF 380 POKE 53760,0:POKE 53762,0
WK 400 POSITION 2,19:?" CZY ZNOW
U GRAC (T LUB N)" :INPUT A$:IF
A$="T" THEN 150
DJ 410 ? "WPROWADZ 8 NOWYCH WARTO
SCI DZWIEKU" :? "POJEDYNCZO."
OB 420 FOR I=1 TO 8:INPUT X:DZWIE
K(I)=X:NEXT I:GOTO 150
XP 20000 REM *****
WJ 20010 REM * Program nr 3 *
XX 20020 REM *****
BT 20040 REM
KJ 20050 REM X=USR(ADR(HF1$),N1,G
1,N2,G2,N3,G3)
CB 20060 REM
FU 20100 SOUND 0,0,0,0:X=64+16:PO
KE 53768,X
PY 20110 DIM HF1$(56):RESTORE 201
40
QF 20120 FOR I=1 TO 56:READ X:HF1
$(I,I)=CHR$(X):NEXT I
DN 20130 RETURN
XH 20140 DATA 104,170,104,141,2,2
10,104,141,0,210,104,104,41,15
,9,160,141,3,210
AH 20150 DATA 224,2,240,32,104,10
4,141,4,210,104,104,41,15,9,16
0,141,5,210
DI 20160 DATA 224,4,240,14,104,10
4,141,6,210,104,104,41,15,9,16
0,141,7,210,96
XP 20000 REM *****
XH 20010 REM * Program nr 4 *
XX 20020 REM *****
BT 20040 REM
TP 20050 REM X=USR(ADR(HF2$),N1,G
1,N2,G2)
CB 20060 REM
XD 20100 SOUND 0,0,0,0:X=64+16+32
+8:POKE 53768,X
MO 20110 DIM HF2$(41):RESTORE 201
40
NA 20120 FOR I=1 TO 41:READ X:HF2
$(I,I)=CHR$(X):NEXT I
DN 20130 RETURN
XH 20140 DATA 104,170,104,141,2,2
10,104,141,0,210,104,104,41,15
,9,160,141,3,210
SE 20150 DATA 224,2,240,17
FM 20160 DATA 104,141,6,210,104,1
41,4,210,104,104,41,15,9,160,1
41,7,210,96
FK 1 REM *****
AB 2 REM * Program nr 5 *
FM 3 REM *****
NJ 4 REM
LW 10 SOUND 0,0,0,0
OZ 80 DIM S(5,8),IN$(50)
EH 90 FOR I=0 TO 8:FOR J=0 TO 5:S
(J,I)=0:NEXT J:NEXT I
HE 100 REJ=5000:ZNI=5100:CZE=5200
:WYW=5300:WYL=5400
NI 102 CLD=5900:CLX=6000:GLO=6100
:POKAUD=6200:STA=6300:START=64
00:REJWYS=6500:BUZZ=6600
WN 104 PWYK=6700:SWYK=6800:EWYS=6
900
XA 1000 REM WYSWIETLENIE
OJ 1002 GRAPHICS 0:POKE 752,1
WW 1008 POSITION 2,0:?" (REJ)ESTR
AUDCTL:"
CW 1010 POSITION 2,1:?" " wielom
ian 9 bitowy:(B7):"
PC 1020 POSITION 2,2:?" " 1.79 MH
z w kanale 0:(B6):"
RI 1030 POSITION 2,3:?" " 1.79 MH
z w kanale 2:(B5):"
CO 1040 POSITION 2,4:?" " kanal
1 + kanal 0:(B4):"
HG 1050 POSITION 2,5:?" " kanal
3 + kanal 2:(B3):"
LS 1060 POSITION 2,6:?" "filtr sor
noprzep.k0/2:(B2):"
NW 1070 POSITION 2,7:?" "filtr sor
noprzep.k1/3:(B1):"
CR 1080 POSITION 2,8:?" "
15 kHz:(B0):"
VC 1090 POSITION 2,9:?" (REJ)ESTR
DZWIEKU:"
LS 1100 POSITION 2,10:?" (ZNI)EK
SZTALCENIE:"
DC 1110 POSITION 2,11:?" (CZE)
STOTLIWOSC:"
YM 1120 POSITION 2,12:?" " WYMUSZO
NE WYJSCIE:"
JJ 1126 POSITION 2,13:?" "
(GLO)SNOSC:"
ED 1128 POSITION 2,14:?" "X:"
MA 1130 POSITION 2,15:?" "D:"
BO 1140 POSITION 2,16:?" "REJ ZNI
CZE WYW GLO"
DS 1150 POSITION 2,17:?" "WYL STA
PWYK SWYK"
GX 1500 GOSUB START
NO 2000 REM TABLICA SKOKOW
VR 2008 FOR ZZZ=1 TO 2 STEP 0
HV 2010 POSITION 5,15:POKE 752,0:
INPUT IN$:POKE 752,1
JK 2020 TRAP 2040:A=VAL(IN$):TRAP
40000
RI 2030 POSITION 5,14:?" A:GOSUB C
LD
OO 2040 IF IN$="REJ" THEN GOSUB R
EJ
QM 2042 IF IN$="ZNI" THEN GOSUB Z
NI
RS 2044 IF IN$="CZE" THEN GOSUB C
ZE
FI 2046 IF IN$="WYW" THEN GOSUB W
YW
LI 2048 IF IN$="WYL" THEN GOSUB W
YL
BO 2049 IF IN$="STA" THEN GOSUB S
TA
SX 2058 IF IN$="GLO" THEN GOSUB G
LO
QI 2060 IF IN$="B7" THEN S(4,7)=
NOT (S(4,7)):POSITION 30,1:?" S
(4,7):GOSUB CLD
NA 2061 IF IN$="B6" THEN S(4,6)=
NOT (S(4,6)):POSITION 30,2:?" S
(4,6):GOSUB CLD
JS 2062 IF IN$="B5" THEN S(4,5)=
NOT (S(4,5)):POSITION 30,3:?" S
(4,5):GOSUB CLD
GK 2063 IF IN$="B4" THEN S(4,4)=
NOT (S(4,4)):POSITION 30,4:?" S
(4,4):GOSUB CLD
DC 2064 IF IN$="B3" THEN S(4,3)=
NOT (S(4,3)):POSITION 30,5:?" S
(4,3):GOSUB CLD
ZU 2065 IF IN$="B2" THEN S(4,2)=
NOT (S(4,2)):POSITION 30,6:?" S
(4,2):GOSUB CLD
WM 2066 IF IN$="B1" THEN S(4,1)=
NOT (S(4,1)):POSITION 30,7:?" S
(4,1):GOSUB CLD
TE 2067 IF IN$="B0" THEN S(4,0)=
NOT (S(4,0)):POSITION 30,8:?" S
(4,0):GOSUB CLD
ZL 2070 IF IN$="PWYK" THEN GOSUB
PWYK
EY 2072 IF IN$="SWYK" THEN GOSUB
SWYK

```

```

AE 2980 IF BLAD=1 THEN GOSUB BUZZ
US 2989 BLAD=0
WC 2990 NEXT ZZZ
HC 5000 REM REG REGISTER SET
KY 5010 IF A<0 OR A>3 THEN BLAD=1
II 5020 IF A>0 OR A<4 THEN POSITI
ON 24,9: ? A
GV 5030 C=A:REM S(C,B)
MG 5040 GOSUB REJWYS
OF 5088 GOSUB CLD:GOSUB CLX
BE 5090 RETURN
LV 5100 REM ZNI POZIOM ZNIEKSZTAL
CEN
KZ 5110 IF A<0 OR A>14 THEN BLAD=
0:GOTO 5180
IK 5112 IF INT(A/2)-A/2<>0 THEN B
LAD=1:GOTO 5180

```

```

BC 5120 D1=A*16
GS 5130 POSITION 21,10: ? " "
LH 5135 POSITION 21,10: ? A
QC 5140 S(C,1)=D1:S(C,5)=A
JA 5170 S(C,8)=A
NB 5180 GOSUB CLD:GOSUB CLX
BG 5190 RETURN
RQ 5200 IF A<0 OR A>255 THEN BLAD
=1
LN 5218 POSITION 21,11: ? "
"
LF 5220 POSITION 21,11: ? A
GI 5230 S(C,2)=A
ND 5280 GOSUB CLD:GOSUB CLX
BI 5290 RETURN
EE 5300 REM FRC SET FORCE BIT
XU 5310 A=INT(A)
FB 5315 IF A<0 OR A>3 THEN BLAD=1
:GOTO 5380

```

```

LK 5320 S(A,3)= NOT S(A,3)
KT 5350 POSITION 21,12: ? S(C,3)
DB 5380 GOSUB CLD
BK 5390 RETURN
YV 5400 REM WYL WYLACZENIE DZWIEK
U
PS 5410 POKE 53761,0:POKE 53763,0
:POKE 53765,0:POKE 53767,0
DD 5480 GOSUB CLD
BM 5490 RETURN
MC 5900 REM CLD KASOWANIE POZYCJI
D
WD 5910 POSITION 5,15: ? "
"
BW 5990 RETURN
TH 6000 REM CLX KASOWANIE POZYCJI
X
VH 6010 POSITION 5,14: ? "
" : A=0
BF 6090 RETURN
MX 6100 REM GLO USTAWIENIE GLOSNO
SCI
OK 6110 IF A<0 OR A>15 THEN BLAD=
1:GOTO 6180
ZZ 6120 POSITION 21,13: ? "
"
MY 6122 POSITION 21,13: ? A
HB 6130 S(C,4)=A
NC 6180 GOSUB CLD:GOSUB CLX

```

```

BH 6190 RETURN
BW 6200 REM POKAUD POKE WARTOSC A
UDCTL
BO 6208 SUM=0
ZJ 6210 FOR I=0 TO 7
UC 6220 IF S(4,I)=1 THEN SUM=SUM+
2^I
FH 6230 NEXT I
NU 6240 POKE 53768,SUM
BJ 6290 RETURN
ZW 6300 REM STA WLACZENIE KANALU
DZWIEKOWEGO

```

```

RV 6310 GOSUB POKAUD
AU 6320 IF A=0 THEN POKE 53761,S(
0,1)+S(0,4):POKE 53760,S(0,2)
MK 6322 IF A=1 THEN POKE 53763,S(
1,1)+S(1,4):POKE 53762,S(1,2)
YA 6324 IF A=2 THEN POKE 53765,S(
2,1)+S(2,4):POKE 53764,S(2,2)

```

```

JQ 6326 IF A=3 THEN POKE 53767,S(
3,1)+S(3,4):POKE 53766,S(3,2)
MP 6380 GOSUB CLX:GOSUB CLD:GOSUB
REJWYS
BL 6390 RETURN
VD 6400 REM START
AQ 6410 FOR I=1 TO 8:POSITION 30,
I: ? "0":NEXT I
BN 6490 RETURN
ZC 6500 REM REJWYS WYSWIETLENIE R
EJESTRU
IY 6505 POSITION 21,12: ? " "
LH 6506 POSITION 21,12: ? S(C,3) "
KW 6510 POSITION 21,11: ? " "
IW 6511 POSITION 21,11: ? S(C,2) "
KG 6520 POSITION 21,10: ? " "
MY 6521 POSITION 21,10
CQ 6522 ? S(C,1)/16
MO 6530 POSITION 21,13: ? " "
MQ 6531 POSITION 21,13: ? S(C,4) "
BP 6590 RETURN
LM 6600 REM BUZZ
HO 6610 ? CHR$(253)
BR 6690 RETURN
IT 6700 REM PWYS WYSWIETLENIE DAN
YCH POKE
QW 6705 GOSUB EWYS
RN 6710 POSITION 20,18: ? "POKE 53
768, ";SUM
XC 6720 POSITION 2,19: ? "POKE 537
61, ";S(0,1)+S(0,4):POSITION 2
0,19: ? "POKE 53760, ";S(0,2)
MX 6730 POSITION 2,20: ? "POKE 537
63, ";S(1,1)+S(1,4):POSITION 2
0,20: ? "POKE 53762, ";S(1,2)
FE 6740 POSITION 2,21: ? "POKE 537
65, ";S(2,1)+S(2,4):POSITION 2
0,21: ? "POKE 53764, ";S(2,2)
XL 6750 POSITION 2,22: ? "POKE 537
67, ";S(3,1)+S(3,4):POSITION 2
0,22: ? "POKE 53766, ";S(3,2)
DK 6780 GOSUB CLD
BT 6790 RETURN
NK 6800 REM SWYS WYSWIETLENIE DAN
YCH SOUND
QY 6805 GOSUB EWYS
XF 6810 FOR I=0 TO 3
NS 6820 POSITION 2,19+I: ? "SOUND
";I"; ";S(I,2);"; ";S(I,8);";
";S(I,4)
FT 6830 NEXT I
DM 6880 GOSUB CLD
BY 6890 RETURN
WG 6900 REM EWYS KASOWANIE PWYS &
SWYS
XY 6910 FOR I=0 TO 4
MR 6920 POSITION 0,18: ? CHR$(156)
FY 6940 NEXT I
BX 6990 RETURN
ET 7000 END

```

AUDF2 lub ADF3 z ADF4). Umożliwia to wybór jednego głosu 16-bitowego i dwóch ośmiobitowych, lub dwóch głosów szesnastobitowych. Za to możliwe jest uzyskanie częstotliwości wyjściowej mniejszej niż 1 Hz.

W takim przypadku nie możemy już używać instrukcji SOUND, która niszczy ustawienia dla dźwięku 16-bitowego. Musimy zatem poznać inny sposób sterowania dźwiękiem. Trzeba używać instrukcji POKE zamiast SOUND. Nie jest to żadnym problemem, ponieważ możemy zdefiniować dokładną równowagę do SOUND następująco:

Do łączenia rejestrów częstotliwości wykorzystujemy odpowiednie ustawienia bitów rejestru AUDCTL, w szczególności bitów 3 i 4. Ustawienie bitu 4 (POKE 53768,16) łączy rejestry częstotliwości ADF1 i ADF2, a ustawienie bitu 3 (POKE 53768,8) — rejestry ADF3 i ADF4. Dodatkowo, dla uzyskania pełnego zakresu częstotliwości wyjściowych, niezbędne jest włączenie zegara o częstotliwości 1.79 MHz (a dokładniej 1.78979 MHz), a zatem POKE 53768,80 dla połączenia kanałów 1 i 2 oraz POKE 53768,120 dla połączenia kanałów 1 i 2 oraz 3 i 4.

Zmienia się również nieco wzór dla określenia częstotliwości wyjściowej:

$$f_{wy} = f_{zeg} / (2 \times (256 \times N_2 + N_1 + 7))$$

gdzie: N2 i N1 są zawartościami rejestrów częstotliwości, drugi rejestr pary jest młodszy bajtem, ADF2 lub ADF4.

Nowymi głosami możemy sterować następująco:

Głos 1 + 2:  
**POKE 53760,N1: POKE 53762,N2: POKE 53763,16 \* Z + G**

Głos 3 + 4:  
**POKE 53764,N1: POKE 53766,N2: POKE 53767,16 \* Z + G**

Odwrotnie, jeżeli chcemy określić, jakie wartości należy wstawić do rejestrów częstotliwości w celu wygenerowania dźwięku o wybranej częstotliwości, możemy rozwiązać (lub zlecić komputerowi) następujące równania:

$$N_2 = \text{INT}((f_{zeg} / (2 \times f_{wy}) - 7) / 256)$$

$$N_2 = \text{INT}((f_{zeg} / (2 \times f_{wy}) - 7) - 256 \times N_2 + 0.5)$$

W celu wygenerowania nuty o częstotliwości 440 Hz otrzymujemy N2 = 7 i N1 = 235 i wstawiamy je do rejestrów. W rezultacie uzyskujemy częstotliwość 439,97 Hz — bardzo dobre dostrojenie.

Rejestr sterujący niższego kanału nie jest używany i powinien być ustawiony na zero. Głośność sterowana jest zawartością drugiego rejestru sterującego (AUDC2 lub AUDC4).

Program nr 1 demonstruje użycie dwóch manipulatorów do sterowania częstotliwością jednego głosu o wysokiej rozdzielczości i jednego „normalnego”. Wyświetla on na ekranie odpowiednie równania i aktualizuje na bieżąco zmieniane w nich parametry. Można zaobserwować, jakim zakresem sterowania dźwiękiem dysponujemy w każdym przypadku.

Program nr 2 umożliwia nam eksperymentowanie z różnymi skalami nut — używając najlepszej normalnej aproksymacji, a następnie wykorzystując sterowanie dźwiękiem o wysokiej rozdzielczości.

Instrukcje obsługi programów wyświetlane są na ekranie.

Programy nr 3 i 4 są procedurami w języku maszynowym, które ułatwiają korzystanie z dźwięku o wysokiej rozdzielczości z BASIC'u. Pierwszy z nich wykorzystuje jeden głos 16-bitowy i dwa ośmiobitowe, a drugi dwa głosy szesnastobitowe. **SOUND 0,0,0,0** na początku każdej procedury inicjuje układ **POKEY** dla dźwięku. **POKE 53768,X** ustawia zawartość rejestru AUDCTL dla dźwięku 16-bitowego, jednego lub dwóch głosów. Pamiętajmy, że dowolna instrukcja SOUND, wykonana później, wyzeruje ten rejestr.

W celu użycia tych procedur, należy skopiować jedną z nich dla własnego programu i wykonać GOSUB 20100 (raz, na początku programu). Później, dla uzyskania dźwięku, należy wywołać procedurę w języku maszynowym:

X=**USR**(ADR(HF1\$),N1,G1,N2,G2,N3,G3) lub  
X=**USR**(ADR(HF2\$),N1,G1,N2,G2)

Nx jest określeniem częstotliwości dźwięku, a Gx — głośnością. N1 jest głosem 16-bitowym w procedurze trzygłosowej. Nie musimy przekazywać parametrów dla nieużywanych głosów. Na przykład, jeśli chcemy usłyszeć tylko dźwięk 16-bitowy w procedurze trzygłosowej, możemy użyć X = **USR**(ADR(HF1\$),N1,G1), lecz dla użycia tylko głosu ośmiobitowego musimy dać: X = **USR**(ADR(HF1\$),0,0,N2,G2).

Zamieszczona poniżej tablica prezentuje najbardziej dokładne wartości nut dla czterech oktaw ośmiobitowych i dziewięciu oktaw 16-bitowych.

Praktycznie pierwsza oktawa nut 16-bitowych nie nadaje się do wykorzystania, ponieważ powstają pewne dźwięki harmoniczne, które maskują odtwarzaną aktualnie nutę.

Bit 1 i 2 AUDCTL sterują filtrami górnoprzepustowymi w kanałach 2 i 1 odpowiednio. Filtr górnoprzepustowy umożliwia przejście tylko wyższym częstotliwościom. W przypadku tych filtrów wyższe częstotliwości zdefiniowane są jako będące nieco wyższe niż wyjście innego kanału wybranego przez kombinację bitów AUDCTL. Na przykład, jeśli dźwięk w kanale 3 naśladuje ryk krowy i ustawiony jest drugi bit AUDCTL, wtedy tylko dźwięki o częstotliwościach wyższych od tego ryku będą

styszane w kanale 1 (niższe częstotliwości będą odfiltrowane).

Filtr może być programowany na bieżąco, ponieważ kanał filtrujący może być zmieniany „w locie”. Daje to programiście duże możliwości. Filtry używane są głównie do tworzenia efektów specjalnych. Dla przykładu sprawdźmy następujące instrukcje:

```
SOUND 0,0,0,0
POKE 53768,4
POKE 53761,168: POKE 53765,168
POKE 53760,254: POKE 53764,127
```

Bit 7 AUDCTL przełącza licznik 17-bitowy na 9-bitowy. Im krótszy licznik, tym częściej powtarza się jego wzór zakłóceń lub jest bardziej dostrzegalny. Zatem zmieniając licznik 17-bitowy uczynimy wzór zniekształceń bardziej powtarzalnym lub bardziej dostrzegalnym. Spróbujmy wykonać następujące instrukcje, słuchając uważnie, kiedy wykonywana jest instrukcja **POKE**:

```
SOUND 0,80,8,8 użycie licznika 17-bitowego
POKE 53768,128 zmiana na licznik 9-bitowy
```

#### Podsumowanie

Dźwięk jest jedną z najciekawszych i jednocześnie najtrudniejszych do poznania możliwości komputera Atari. Istnieją cztery niezależne kanały dźwiękowe, których częstotliwość, zniekształcenie i głośność mogą być sterowane niezależnie. Są one adresowane przez instrukcję SOUND za pomocą liczb 0 do 3. Dane o dźwięku mogą być niezależnie wstawiane (POKE) do rejestrów o adresach 53760 do 53767. Komórki nieparzyste steru-

ją głośnością i zniekształceniem, a parzyste sterują częstotliwością. Rejestr 53768 (AUDCTL) steruje wszystkimi kanałami dźwiękowymi w określony sposób. Jeśli użyjemy instrukcji SOUND, dowolne zmiany, które zostały zrobione w AUDCTL, są zerowane — AUDCTL ustawiany jest na zero. Zatem instrukcja SOUND nie zapewnia pełnego wykorzystania możliwości dźwiękowych komputera.

Program nr 5 przeznaczony jest do demonstracji przedstawionych powyżej możliwości. Próbuje dać łatwe sterowanie wszystkimi parametrami. Wymaga nieco wprawy w obsłudze, lecz pozwala na totalne sterowanie wszystkimi rejestrami plus AUDCTL, włącza kanały indywidualnie i wyłącza je wszystkie naraz. Kiedy otrzymane dźwięki odpowiadają nam, możemy wyświetlić odpowiednie instrukcje BASIC'u w postaci POKE albo SOUND.

Po uruchomieniu programu na ekranie pojawia się szereg linii. Pierwsze osiem z nich, oznaczone B7 do B0, reprezentują zawartości bitów rejestru AUDCTL. W celu zmiany siódmego bitu na 1, należy wprowadzić B7 i RETURN. Dla zmiany z powrotem na zero, wprowadzamy B7 i RETURN. Są to zmiany techniczne, które nie dają wskazań, jak będzie brzmiał nowy dźwięk. Najlepiej eksperymentować.

Dla omówienia następujących pięciu linii musimy przejść w dół, do linii oznaczonych D i X. Program przyjmuje dwa typy danych: komendy i liczby. Jeśli chcemy mieć liczbę, wprowadzamy ją i naciskamy RETURN. Jeśli komendę, to postępujemy podobnie. W celu wybrania zerowego kanału dźwiękowego wprowadzamy następujący ciąg: 0, RE-

TURN, REJ, RETURN. Zero pojawi się na ekranie po REJ ESTR DŹWIĘKU. Dla uzyskania czystego tonu wprowadźmy: 10,RETURN, ZNI, RETURN. 10 pojawi się po ZNI EKSZTALCENIE. Podobnie: 100, RETURN, CZE, RETURN, 8, RETURN, GLO, RETURN, wypełni pozostałe linie.

Dla usłyszenia dźwięku musimy wprowadzić: 0,RETURN, STA, RETURN, a dla wyłączenia: WYL, RETURN. Aby zobaczyć wartości POKE dla tego dźwięku, należy wprowadzić PWYK, RETURN i na ekranie pojawi się wykaz dziewięciu instrukcji POKE. Możemy skopiować je do naszego programu, aby odtworzyć dźwięk, który usłyszeliśmy. Prawy górny POKE dotyczy rejestru AUDCTL. Następne cztery wiersze są kanałami 0 do 3 — lewa kolumna dotyczy zniekształcenia i głośności, a prawa — częstotliwości każdego kanału.

Jeśli zawartość rejestru AUDCTL jest zerowa — zerowe wartości bitów B0 do B7 — wtedy mogą być używane instrukcje SOUND. Aby je zobaczyć wprowadźmy SWYK, RETURN.

Wyjście „wymuszone” występuje w rejestrach nieparzystych i wytwarza „trzask” w głośniku. Jest on włączany i wyłączany za pomocą WYW, RETURN. Kanały dźwiękowe muszą być włączane indywidualnie za pomocą komendy STA. WYL wyłącza wszystkie kanały. Jeśli dokonamy zmiany i chcemy ją usłyszeć, wprowadzamy numer kanału i STA.

**Tomasz MROWIEC**  
**Ludwik PIELA**

## Oszczędzajmy czas — „ZX Spectrum”

Zadanie było proste: ułożyć z dwudziestu znaków graficznych UDG komputera Spectrum wzór kobierca. W celu uzyskania symetrii należało podzielić zasoby znaków UDG na 4 grupy po 5 znaków, symetrycznych względem osi poziomej i pionowej. Wystarczyło zdefiniować tylko 5 znaków po 8 bajtów (64 punkty), a następnie powtórzyć je w odbiciach zwierciadlanych względem obu osi symetrii.

ABCDE	JIHGF
ONMLK	PQRST

Odwrocenie w stosunku do osi poziomej nie było trudne — wystarczyło wpisać 40 bajtów odpowiadających znakom ABCDE w odwrotnej kolejności. Trudniej było z odwróceniem w stosunku do osi pionowej — tu należało rozbić bajty na poszczególne bity, a następnie złożyć z nich odwrócone „antybajty”. Przypominam, że wartość liczbowa bitów w bajcie odpowiada kolejnym potęgom liczby 2.

**W pierwszej próbie (wiersze 1000—1020)** rozbito generowane przez generator liczb losowych RND bajty o wartości 0—255 na poszczególne bity metodą rachunkową dzieląc je przez kolejne potęgi liczby 2, a następnie złożono w odwrotnej kolejności, mnożąc przez kolejne potęgi dwójki. W końcu wprowadzono obliczone antybajty przez POKE w odpowiednie komórki pamięci UDG. Pozostało tylko wydrukowanie (wiersz 800).

Program jest napisany poprawnie, ale wskutek użycia pętli **FOR—NEXT** i podnoszenia dwójki do kolejnych potęg trwał nieznośnie długo — ponad dwie minuty.

**W drugiej próbie (w. 2000—2020)** wygenerowano od razu bity (zera lub jedynki) i złożono z nich kolejne bajty i antybajty. Przyspieszyło to działanie programu o 28 sekund, ale tak pozostały jeszcze 92 sekundy — o wiele za długo, jak na tak krótki program.

**W trzeciej próbie (wiersze 3000—3020)** ułożono na początku tabelę kolejnych potęg dwójki. Dało to od razu 4-krotne przyspieszenie programu. Program wraz z generacją tabeli trwał 23 sekundy, powtórzenia o sekundę krócej.

Skoro tabela potęg dała tak dobre wyniki, należało spróbować stworzyć tabelę antybajtów (wersja trzecia). Obliczanie takiej tabeli trwało ponad dwie minuty, ale potem kolejne mutacje kobierca powstawały po prawie 3 sekundach. W ten sposób (**wiersze 4000—4030**) wyczerpaliśmy możliwości metod rachunkowych BASIC-a. Szybciej można by w assemblerze, ale to już zupełnie inna historia.

**W piątej wersji (wiersze 5000—5020)** zastosowano trick graficzny: kolejne bajty wprowadzono do pamięci ekranu pod adresem

22432 (lewe dolne naroże ekranu), a następnie odczytywano punkt po punkcie przez POINT (x, y) i sumowano w antybajty wprost bez użycia pętli. Dzięki temu skrócono generację zestawu znaków do 4 sekund.

Ale to jeszcze nie koniec.

Dzięki omawianemu już trickowi graficznemu w ciągu 17 sekund stworzono tablicę antybajtów i zapisano w pamięci pod adresem 60000—60255. Ta wersja programu (**szósta — wiersze 6000—6030**) tworzy kolejne mutacje kobierca co 2,65 sek. Czyba nieźle? A może uda się jeszcze komuś urwać kawałek sekundy? Czekamy na pomysły Czytelników.

W wierszach 9000—9020 znajduje się prosty program „wyplatający” wzór kobierca. Wzór ten składa się z dziewięciu pasów po 64 znaki (a/\$) i z dwóch pasów skrajnych (c/\$). Przez kombinację tych pasów uzyskuje się 704-znakowy wzór d/\$. Oczywiście zawartość a/\$ i c/\$ wprowadza się w trybie graficznym (kursor „G”). Wzór ten w postaci pierwotnej-literowej można zobaczyć tylko raz, bezpośrednio po wygraniu programu z kasy. Potem znaki UDG zamieniają się w mozaikę punktów. Na wszelki wypadek jest w programie wiersz 400, przywracający w razie potrzeby pierwotną postać znaków UDG. Uruchamia się go przez **GO TO 400**.

Pozostaje jeszcze wiersz 9999 zapisujący program na taśmę i sprawdzający poprawność zapisu.

Jeszcze jedna propozycja dla Czytelników: Kto wymyśli prosty sposób, jak każdej czwórce znaków UDG (np. AFOT lub EJKP) przyporządkować jeden z pięciu kolorów?

**Jan RUBINOWICZ**

```
0:REM Jan Rubinowicz 1987
1:REM
2:REM
3:REM
1000:REM KOBIERCE
2000:GO SUB 2000: GO SUB 300
210 LET w2=1: GO SUB 500: GO SU
B 1000: GO SUB 300
220 LET w2=2: GO SUB 500: GO SU
B 2000: GO SUB 300
230 LET w2=3: GO SUB 500: GO SU
B 3000: GO SUB 300
235 BEEP .02,14: FOR r=1 TO 5:
GO SUB 3010: GO SUB 300: GO SUB
600: NEXT r
240 LET w2=4: GO SUB 500: GO SU
```

```

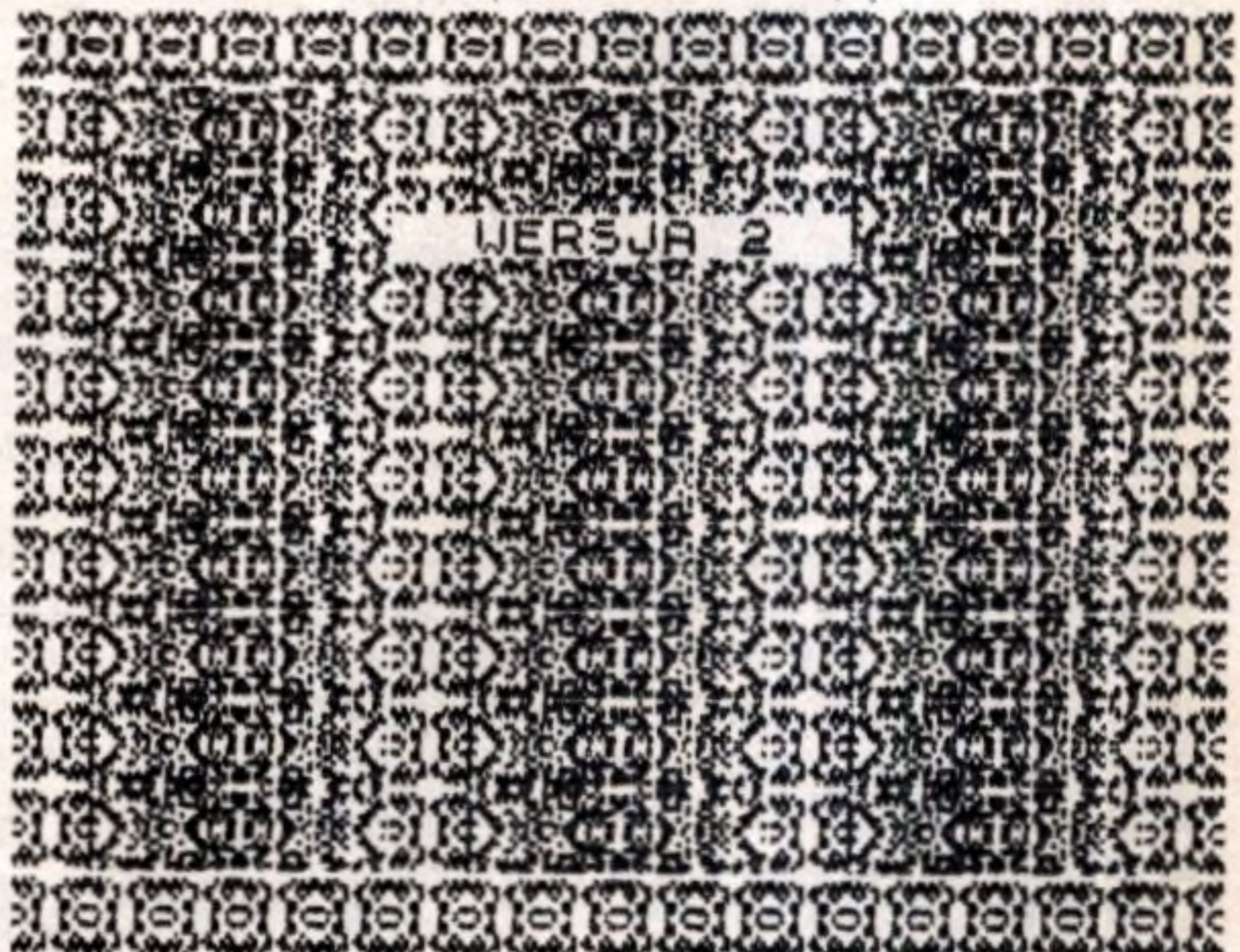
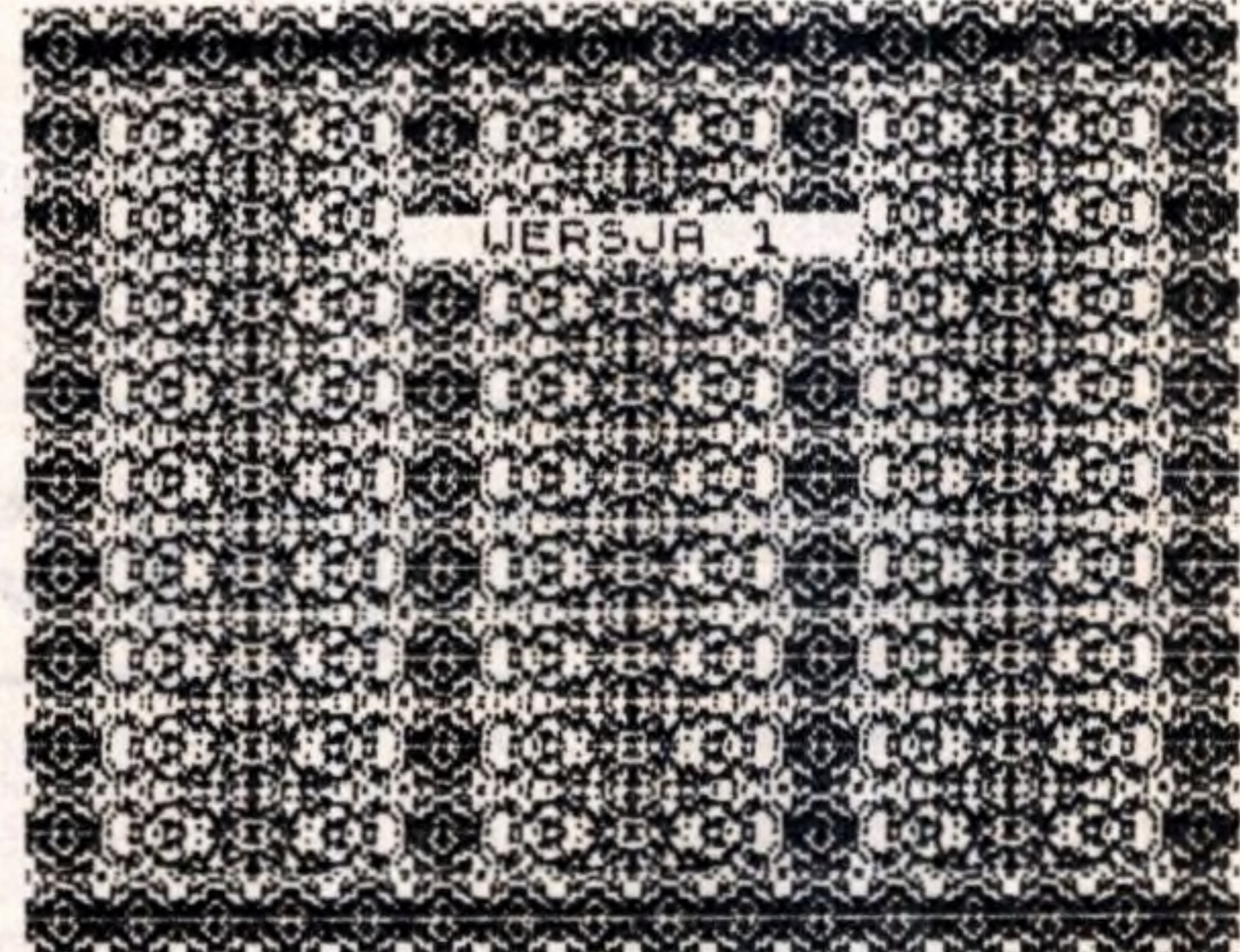
B 4000: GO SUB 300
245 BEEP .02,14: FOR r=1 TO 5:
GO SUB 4020: GO SUB 300: GO SUB
500: NEXT r
250 LET w=5: GO SUB 500: GO SU
B 5000: GO SUB 300
255 BEEP .02,14: FOR r=1 TO 5:
GO SUB 5000: GO SUB 300: GO SUB
500: NEXT r
260 LET w=6: GO SUB 500: GO SU
B 6000: GO SUB 300
265 BEEP .02,14: FOR r=1 TO 5:
GO SUB 5020: GO SUB 300: GO SUB
500: NEXT r
270 GO TO 310
400 FOR n=0 TO 167: POKE 65368+
n,PEEK (15880+n): NEXT n: GO TO
300
500 BEEP .02,14: PRINT AT 5,10;
BRIGHT 1, FLASH 1: " WERSJA ";w
e: " " : RETURN
500 PRINT BRIGHT 1;AT 21,0;r: R
ETURN
800 PRINT AT 0,0; BRIGHT 1;d$:
RETURN
999 REM
wersja 1 - odwracanie bajtow
1000 FOR m=0 TO 39: LET bt=INT (
RND*256): LET abt=0
1010 FOR b=0 TO 7: LET p=(bt/2^b
-INT (bt/2^b)).5: LET abt=abt+2
*p*2^(7-b): NEXT b
1020 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
1030 RETURN
1999 REM
wersja 2 - skladanie z bitow
2000 FOR m=0 TO 39: LET bt=0: LE
T abt=0:
2010 FOR b=0 TO 7: LET a=INT (RN
D*2): LET bit=(2^b)*a: LET bt=bt
+bit: LET abit=(2^(7-b))*a: LET
abt=abt+abit: NEXT b
2020 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
2030 RETURN
2999 REM
wersja 3 - tabela poteg
3000 DIM k(8): FOR n=1 TO 8: LET
k(n)=2^(n-1): NEXT n
3010 FOR m=0 TO 39: LET bt=0: LE
T abt=0: FOR b=1 TO 8: LET a=INT
(RND*2): LET bit=k(b)*a: LET bt
=bt+bit: LET abit=k(9-b)*a: LET
abt=abt+abit: NEXT b
3020 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
3030 RETURN
3999 REM
wersja 4 - tabela antybajtow
4000 DIM k(8): FOR n=1 TO 8: LET
k(n)=2^(n): NEXT n
4010 DIM t(256): DIM p(8): FOR a
=1 TO 256: LET t(a)=0: FOR b=1 T
O 8: LET p(b)=((a-1)/k(b)-INT ((
a-1)/k(b))).5: LET t(a)=t(a)+(.
5*p(b)*k(9-b)): NEXT b: NEXT a
4020 FOR m=0 TO 39: LET bt=INT (
RND*256): LET abt=t(bt+1)
4030 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
4040 RETURN
4999 REM
wersja 5 graf.odwracanie bajtow
5000 FOR m=0 TO 39: LET bt=INT (
RND*256): POKE 22432, bt
5010 LET abt=POINT (0,0)+2*POINT
(1,0)+4*POINT (2,0)+8*POINT (3,
0)+16*POINT (4,0)+32*POINT (5,0)
+64*POINT (6,0)+128*POINT (7,0)
5020 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
5030 RETURN
5999 REM
wersja 6 graf.tabela antybajtow
6000 FOR n=0 TO 255: POKE 22432,
n
6010 LET t=POINT (0,0)+2*POINT (
1,0)+4*POINT (2,0)+8*POINT (3,0)
+16*POINT (4,0)+32*POINT (5,0)+6
4*POINT (6,0)+128*POINT (7,0): P

```

```

OKE 60000+n,t: NEXT n
6020 FOR m=0 TO 39: LET bt=INT (
RND*256): LET abt=PEEK (60000+bt
)
6030 POKE (65368+m),bt: POKE (65
408+m),abt: POKE (65487-m),bt: P
OKE (65527-m),abt: NEXT m
6040 RETURN
8999 REM
wypisanie wzoru
9000 LET a$=
" ABCDEJIHGFABCD EJIHGFABCD EJIHGF
TONMLKQPQRSTONMLKQPQRSTONMLKQPSTO
"
9010 LET b$="": FOR m=1 TO 9: LE
T b$=b$+a$: NEXT m
9020 LET c$=
" FAFAFAFAFAFAFAFAFAFAFAFAFAFAF
TOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTO
": LET d$=c$+b$+c$
9030 RETURN
9999 CLS : SAVE "KOBIERZEC" LINE
100: BEEP 1,2: PRINT "Verify:
KOBIERZEC": VERIFY "KOBIERZEC":
GO TO 100

```



## Arytmetyka modulo liczba całkowita (1)

Mówiąc o arytmetyce mamy zwykle na myśli umiejętność wykonywania podstawowych działań dodawania i mnożenia na dowolnych liczbach. *Arytmetyka modulo liczba całkowita* zajmuje się działaniami na pewnej skończonej grupie liczb. Przykładowo arytmetyka modulo 7 dotyczy liczb 0, 1, 2, 3, 4, 5, 6, a arytmetyka modulo 11 liczb 0, 1, 2, ..., 10. Łatwo zauważyć, że są to wszystkie możliwe reszty otrzymane z dzielenia dowolnej liczby całkowitej przez liczbę wybraną jako moduł. Wynikiem dodawania bądź mnożenia liczb modulo liczba całkowita np. 7, jest reszta z dzielenia sumy bądź iloczynu przez 7, np.  $(2 + 8) \bmod 7 = 3$ ,  $(2 * 4) \bmod 7 = 1$ . Z arytmetyką modulo 24 mamy do czynienia przy odczytywaniu godzin, gdzie dopuszczalne są wyniki z przedziału [0,23]. Dni liczymy modulo 7, gdzie zerowym dniem jest niedziela, pierwszym poniedziałek, a szóstym sobota.

Program nasz podaje przykłady dodawania i mnożenia modulo liczba całkowita, sprawdza umiejętności operatora programu i ocenia, czy może on przejść do programu matematycznie bardziej skomplikowanego, tj. do twierdzenia Fermata.

```
320 INPUT a
330 IF a=FNmp(x) THEN licz=licz+1:PRINT "dobrze"
340 IF a<>FNmp(x) THEN PRINT "zle - spróbuj to : "
350 PRINT n2;" * ";n3;" mod ";n1;" = ?"
360 INPUT a
370 IF a=FNmp(y) THEN licz=licz+1: PRINT "dobrze"
380 IF a<>FNmp(y) THEN PRINT "zle"
390 NEXT i
400 IF licz<6 THEN PRINT "musisz jeszcze potrenowac":GOTO 20
410 PRINT"mozesz przejsc do programu :TWIERDZENIE FERMATA"
420 END
```



A.T.

```
10 REM arytmetyka modulo
20 INPUT "liczymy (t/n)";x$
30 IF x$="t" THEN GOTO 60
40 IF x$<>"n" THEN GOTO 20
50 GOTO 420
60 PRINT "podac przyklady obliczen (t/n) ?"
70 DEF FN mp(x)=x-INT(x/n1)*n1
80 DEF FN los(n)=INT(RND(1)*9)+n
90 INPUT x$
100 IF x$="t" THEN GOTO 130
110 IF x$<>"n" THEN GOTO 60
120 GOTO 220
130 n1=FNlos(2)
140 PRINT "arytmetyka modulo ";n1
150 n2=FNlos(1)
160 n3=FNlos(1)
170 x=n2+n3
180 y=n2*n3
190 PRINT n2;" + ";n3;" mod ";n1;" = ";FNmp(x)
200 PRINT n2;" * ";n3;" mod ";n1;" = ";FNmp(y)
210 GOTO 60
220 PRINT "sprawdze czy tez potrafisz "
230 licz=0
240 FOR i=1 TO 5
250 n1=FNlos(2)
260 n2=FNlos(1)
270 n3=FNlos(1)
280 PRINT "liczymy modulo ";n1
290 PRINT n2;" + ";n3;" mod ";n1;" = "
300 x=n2+n3
310 y=n2*n3
```



Nasz pan profesor ma komputerową głowę.

# Fortran w CP/M

Mimo że od wielu lat PASCAL wypiera z użytkowania język FORTRAN, to jednak w wielu centrach obliczeniowych wyposażonych w komputery ODRA, RIAD itp. ze względu na bogate oprogramowanie w tym języku, wiele programów pisze się nadal w języku FORTRAN.

Przy dużym obciążeniu tych maszyn i wysokim koszcie użytkownika, dostępność do nich jest ograniczona. Warto więc sprawdzić poprawność działania opracowywanego programu na bardziej dostępnym Personal Computerze.

Istnieje kompilator tego języka na komputery pracujące w systemie CP/M o nazwie FORTRAN-80. Jak wiele innych programów tak i ten dostępny jest na giełdach jednakże najczęściej bez opisu. Oto w skrócie podstawowe informacje o sposobie uruchamiania i użytkowania tego kompilatora.

Pakiet oprogramowania FORTRAN-80 zawiera bloki o nazwach:

F80	.COM	L80	.COM
FORLIB	.REL		

Pakiet ten porównywalny jest z kompilatorami tego języka na dużych maszynach cyfrowych i minikomputerach. Posiada on pewne ograniczenia np. brak typu COMPLEX, ograniczenia w stosowaniu tablic itd. Mimo to wydaje się być pomocny w uruchamianiu nawet dużych programów.

## Pisanie i kompilowanie programów:

Przystępując do pracy z kompilatorem radzę zrobić jego dodatkową kopię roboczą na innym dysku. Wszystkie powstające w kolejnych etapach kompilacji zbiory należy zapisać na ten dysk, gdyż kompilator musi komunikować się z podprogramami zawartymi w pliku. Kopia zabezpieczy Cię przed przypadkowym skasowaniem któregoś ze zbiorów kompilatora, a co się z tym wiąże, z jego utratą.

Program w języku FORTRAN piszemy w dowolnym wordprocesorze dającym na dysku blok znaków kodu ASCII (znakomicie do tego celu nadaje się edytor języka TURBO PASCAL. Edytor tekstów TASWORD lub inne).

W stosowaniu musisz pamiętać, aby nazwę programu podać z rozszerzeniem .FOR. oraz zachować właściwy FORTRAN-owski format wiersza.

Kompilator pracuje w systemie CP/M., dlatego przed dalszymi czynnościami należy wprowadzić ten system do komputera.

## Kompilacja przebiega dwuetapowo

W pierwszym etapie przy użyciu kompilatora F80 tekst programu przekształcany jest w blok typu .REL.

Przykładowym zleceniem dla kompilatora jest:

```
F80 = PROG.FOR
```

gdzie: PROG oznacza nazwę kompilowanego programu.

Nakazuje ono: wczytaj kompilator F80, następnie program zawarty w bloku PROG.FOR, po czym skompiluj ten program, wypisz znalezione błędy składniowe i zapisz postać wynikową na dysku w blok o nazwie PROG.REL.

Znalezione błędy opisywane są komunikatami w języku angielskim podającym określenie błędu, numer linii, w której wystąpił oraz część tej linii.

Jeśli w pierwszej części kompilacji nie wykryto błędów, można przystąpić do drugiego etapu.

Teraz będziemy używać programu L80. Ma on za zadanie wyszukać wszystkie funkcje i podprogramy zewnętrzne, jakie zostały użyte w programie (należą do nich funkcje standardowe FORTRAN-u procedury drukowania itp.), dołączyć je do kompilowanego bloku, a następnie całość zapisać na dysku w bloku typu .COM. Przykładowym zleceniem jest:

```
L80 PROG.PROG/N.PROG/S/E
```

które nakazuje:

wczytaj z dysku program L80, następnie blok PROG.REL., potrzebne procedury, utwórz w systemie CP/M blok o nazwie PROG.COM i zapamiętaj go na dysku.

Czas przetwarzania pliku programem L80 jest stosunkowo długi, dlatego nie należy się zrażać faktem długiej pracy disc drive'u.

W trakcie tej części kompilacji mogą pojawić się informacje o niezdefiniowanych zmiennych globalnych. Oznacza to, że na dysku nie znaleziono funkcji lub podprogramu, do którego odwoływano się w kompilowanym programie. Tak skompilowany program uruchamia się w systemie CP/M zleceniem:

```
PROG.
```

Podane tutaj zlecenia są zleceniami przykładowymi. Dokładniejsze informacje zarówno o kompilatorze, jego ograniczeniach i rozszerzeniach w stosunku do standardu ANSI FORTRAN X3.9-1966, jak i różnych postaciach zleceń, można znaleźć w DOKUMENTACJI UŻYTKOWEJ KOMPILATORA FORTRANU I MAKRO-ASSEMBLERA MACRO-80 DLA SYSTEMU CP/M dostarczanej wraz z kompilatorem przy zakupie od firmy zajmującej się sprzedażą oprogramowania.

Ceny giełdowe kompilatora są w granicach 1000—2000 zł (bez dokumentacji użytkowej).

(Opracowano na podstawie „Dokumentacji użytkowej kompilatora FORTRAN dla systemu CP/M.”)

**Cezary DOBROWOLSKI**

## Pojazd kosmiczny — CPC 6128

```
10 ' Pojazd kosmiczny
20 INPUT "ILE RAZY POWTORZYC GLOS POJAZDU KOSMI- CZNEGO":I
30 FOR L=1 TO I
40 ENV 1,1,5,1,1,-5,1
50 FOR k = 7 TO 0 STEP -1
60 ENT -1,10,-1,1,9,1,0.5
70 SOUND 1,100,-100,k,1,1
80 NEXT
90 NEXT
10 ' Kukulka
20 INPUT "ILE RAZY POWTORZYC GLOS KUKULKI":I
30 FOR L=1 TO I
40 FOR k = 1 TO 2
50 ENV 1,20,1,2
60 SOUND 1,63,0,2,1
70 SOUND 1,75,0,2,1
80 SOUND 1,0,70
90 NEXT
100 NEXT
```

## Helikopter — CPC 6128

```
10 ' Helikopter
20 INPUT "ILE RAZY POWTORZYC GLOS HELIKOPTERA":I
30 FOR L=1 TO I
40 ENV 1,5,1,0.5,1,-5,0.5,5,3,0.5,1,-15,0.5
50 ENT -1,1,20,50
60 ENV 2,5,1,100,1,0,255,5,-1,100
70 ENT -2,10,1,0.5,10,-1,0.5
80 g=1
90 SOUND 10,758,-3,7,2,2,1
100 SOUND 17,50,-7,g,1,0,1
110 FOR k=1 TO 50
120 IF k<15 AND g<15 THEN g=g+1
130 IF k>35 AND g>0 THEN g=g-1
140 SOUND 1,50,-7,g,1,0,1
150 NEXT
160 NEXT
```

## IBM Personal System

Sześć lat temu firma IBM wprowadziła na rynek komputerowy model Personal Computer, wyposażony w mikroprocesor INTEL 8088 i pamięć zapisywalną RAM 64 KB — mikrokomputer, który stał się standardem światowym, powielanym przez setki firm zachodnioeuropejskich i azjatyckich. W ciągu kolejnych lat pojawiły się modele XT, AT, RT oraz klony tych modeli (kompatybilne z nimi), wyposażone w: mikroprocesory rodziny iAPX 86 (8088, 8086, 80186, 80286, 80386) i koprocesory; pamięć zapisywalną od 640 KB do 16 MB; dyski stałe o pojemności od 10 do 210 MB; stacje dysków elastycznych 5 1/4 cala jednostronne 160/180 KB, dwustronne 320/360 KB i dużej pojemności 1,2 MB; karty graficzne Color Graphics Card (CGA) o rozdzielczości 320 × 200/640 × 200 punktów adresowalnych bezpośrednio (pixeli) przy 4 kolorach, Monochrome Graphics Card (MA) o rozdzielczości 720 × 348 pixeli, Enhanced Graphics Card (EGA) o rozdzielczości 640 × 350 pixeli przy 16 kolorach, Enhanced Graphics Card Plus (EGA Plus) o rozdzielczości 1280 × 1024 pixeli przy 256 kolorach; moduły i porty wejścia/-wyjścia umożliwiające współpracę z różnego typu urządzeniami wejścia/-wyjścia m.in. drukarkami, plotterami, digitizerami oraz pracę w sieci lokalnej (LAN — Local Area Network).

Rok 1987 i modele IBM Personal System/2, to prawdopodobnie kolejny krok w historii mikrokomputerów. Oto opis modeli 30, 50, 60 i 80.

Wymienione cztery modele mają kilka wspólnych cech. Firma IBM zdecydowała się na wprowadzenie mniejszych i trwalszych dyskietek 3 1/2 cala, umożliwiających zapisywanie większej ilości informacji niż na dyskietskach 5 1/4 cala. Zapis może mieć pojemność 720 KB lub 1,44 MB.

Mikrokomputery wyposażono w pięć portów wejścia/wyjścia: dla klawiatury (typu mini-DIN), dla myszki (typu ditto), dla monitora kolorowego (typu DIN) oraz równoległy CENTRONICS i szeregowy RS-232 C.

Klawiatura mikrokomputerów posiada 102 — klawisze.

Modele 50, 60 i 80 wyposażone są w nowe układy scalone b. dużej skali integracji VLSI, wśród nich: VGA (Video Graphics Array) z pamięcią 256 KB; 16-kanalowy sterownik bezpośredniego dostępu do pamięci (DMA). Układ VGA umożliwia grafikę z rozdzielczością 320 × 200/640 × 480 pixeli w 256 kolorach lub monochromatycznie. Po kolejnym rozwiązaniu sprzętowym można będzie uzyskać rozdzielczość 1024 × 768 pixeli w 256 kolorach.

Modele te wyposażone są w pamięć stałą o pojemności 128 KB, zawierającą CBIOS i ABIOŚ (w nich m.in. procedury BIOS-u Basic Input Output System). Dla wymienionych modeli zostanie opracowany OS/2 (Operating System/2), przystosowany specjalnie do możliwości mikroprocesora 80286.

Wszystkie modele mają możliwość dołączenia zewnętrznej stacji dysków elastycznych 5 1/4 cala i stacji z zapisem laserowym WORM (Write-Once Read-Many) o pojemności 200 MB.

Kolejną wspólną cechą począwszy od modelu 30 to zastosowanie dysków stałych 3 1/2 cala. W mikrokomputerach IBM PS/2 wykorzystano PC DOS 3.30.

Model 30 wyposażono w mikroprocesor 8086 pracujący z częstotliwością zegara 8 MHz. Pamięć zapisywalna standardowo ma pojemność 640 KB. W pamięci stałej 64 KB

zapisane są procedury CBIOS (kompatybilne z BIOS). Model 30 ma dwie wersje: 30—002 z dwiema stacjami dysków elastycznych 3 1/2 cala o pojemności 720 KB oraz 30—021 z jedną stacją dysków elastycznych i dyskiem stałym 3 1/2 cala o pojemności 20 MB. Posiada on układ MCGA (Multi-Colour Graphics Array) umożliwiający pracę w trybie graficznym z rozdzielczością 640 × 480 pixeli monochromatycznie i 320 × 200 pixeli w 256 kolorach.

Konfigurację systemu można rozbudować wykorzystując trzy łączówki 8-bitowej magistrali, zamontowane w płycie systemowej. Model 30 jest krokiem w ewolucji IBM PC/XT. Cena wersji 30 — 002 £1106.

Model 50 wyposażony jest w mikroprocesor 80286 pracujący z częstotliwością zegara 10 MHz. Pojemność pamięci zapisywalnej o czasie dostępu 150 ns standardowo 1 MB może być rozszerzona do 7 MB. Pamięć stała 128 KB zawiera CBIOS i ABIOŚ. Model może posiadać jedną lub dwie stacje dysków elastycznych 3 1/2 cala o pojemności 1,44 MB oraz dysk stały 20 MB. Model 50 posiada układ VGA. System może być rozbudowany o trzy

moduły dołączone do łączówek 16-bitowej magistrali mikrokomputera. Cena modelu z I stacją dysków elastycznych i dyskiem stałym 20 MB £2658. Model 50 pozwoli na pewno lepiej wykorzystać możliwości mikroprocesora 80286 niż w IBM PC/AT.

Model 60 różni się od modelu 50 tym, że posiada większą pojemność pamięci zapisywalnej i masowej oraz lepszą możliwość rozbudowy.

Pojemność pamięci zapisywalnej może wynosić 15 MB, mikrokomputer może posiadać dysk stały 44 MB (wersja 60-041) lub 70 MB (wersja 60-071) oraz konfiguracja może być rozszerzona o siedem modułów. Jednostka centralna modelu 60 umieszczona jest w obudowie typu „wieża”, stawianej pionowo na podłodze. Cena wersji 60-071 £ 4464.

Model 80 posiada trzy wersje: 80-041 i 80-071 z mikroprocesorem 80386 pracującym z częstotliwością zegara 16 MHz oraz 80-111 z mikroprocesorem 80386 pracującym z częstotliwością 20 MHz. Pamięć zapisywalna o czasie dostępu 80 ns, standardowo o pojemności 2 MB, może być rozszerzona do 16 MB. Mikrokomputer w wersji 80-041 posiada dysk stały 44 MB, w wersji 80-071 dysk 70 MB, a w wersji 80-111 dysk o pojemności 115 MB. Konfiguracja systemu może być rozbudowana o trzy moduły — dołączone do 32-bitowej magistrali i cztery — dołączone do 16-bitowej magistrali, poprzez łączówki umieszczone w płycie systemowej. Cena wersji 80-041 £ 4916, wersji 80-111 £ 7056.

Z przedstawionymi wyżej modelami mogą pracować monitory kolorowe CD 8513 o przekątnej 12 cali lub CD 8512 o przekątnej 14 cali.

Wśród nowych drukarek, przeznaczonych do współpracy z IBM PS/2, można wymienić: IBM Proprinter II, X24, XL24 lub IBM Quietwriter III Printer.

Mikrokomputery IBM PS/2 to kolejna, nie ostatnia propozycja firmy. Rozwiązania modeli z uwagi na parametry są atrakcyjne i na pewno zadowolą użytkowników profesjonalnych, którym wystarczy proponowana rozdzielczość 1024 × 768 pixeli, pojemność pamięci operacyjnej i masowej oraz szybkość procesora i niezbyt długi czas dostępu do informacji.

Mikrokomputery będą chyba z powodzeniem wykorzystywane do prac inżynierskich w małych i średnich systemach CAD, CAE i CAM, do prac biurowych i innych, których wymienić nie jest łatwo.

**Jacek WOJTALA**



— A co umie robić pański robot?  
— Przychodzi punktualnie po wypłatę pieniędzy.

# Procedury ROM SPECTRUM (2)

Podstawową procedurą wydruku znaków na ekranie (drukarka) jest 16. Można ją wywoływać przez CALL 16, lecz warto pamiętać, że specyfika mikroprocesora Z80 pozwala na stosowanie krótszych, 1-bajtowych rozkazów RST.

4. PRINT	16	#10
CEL: wydruk znaku na ekranie/drukarka:		
WE: A — kod znaku: 0..31 — kody kontrolne; 32..127 — kody znaków drukowalnych ASCII; 128..143 — semigrafika; 144..164 — znaki użytkownika; 165..255 — słowa kluczowe (tokens);		
WY: BC, DE, HL, IX przechowane; używane rejestry pomocnicze (HL' przechowane);		

Procedura 4 uwzględnia aktualny kanał (odczyt CURCHL), atrybuty ekranu (ATTR-T, MASK-T), tryby ekranowe (PFLAG), bieżące zmienne CHARS i UDG. Semigrafiki tworzone są w obszarze pamięci kalkulatora „na gorąco”. Znaki o kodach 0..5, 7, 9..12, 14..15, 24..31 są drukowane jako znak zapytania.

5. PRINT OUT	2548	#09F4
CEL: jak w proc. 4;		
WE: jak w proc. 4;		
WY: IX przechowane; używane AF';		

**Procedura 5** może zastępować procedurę 4 wszędzie tam, gdzie nie jest wskazane używanie rejestrów pomocniczych. Nie odczytuje również zmiennej CURCHL.

6. POSITION FETCH	2819	#0B03
CEL: odczyt współrzędnych pozycji wydruku znaku na ekranie/drukarka;		
WE: nieistotne;		
WY: dla ekranu głównego lub edycyjnego: B — rząd fizyczny; C — kolumna fizyczna; HL — adres pola wydruku w pamięci ekranu (górny z ośmiu bajtów); dla drukarki: C — pozycja w buforze drukarki wskazująca ilość znaków pozostałych do wydruku; HL — adres w buforze zawierający bajt do wydruku; flaga Z = 1 dla ekranu głównego; flaga Z = 0 dla ekranu edycyjnego i drukarki; A, DE, IX przechowane		

**Procedura 6** dokonuje odczytu odpowiednich dla używanego kanału zmiennych systemowych:

- dla ekranu głównego: S-POSN i DF-CC;
- dla ekranu edycyjnego: S-POSNL i DF-CCL;
- dla drukarki: P-POSN i PR-CC;

7. POSITION STORE	2780	#OADC
CEL: ustawienie pozycji wydruku znaku na ekranie lub drukarka;		
WE: dla ekranu: B — rząd fizyczny; C — kolumna fizyczna; HL — adres górnego pola bajtu w pamięci ekranu; dla drukarki: C — pozycja fizyczna w buforze; HL — adres w buforze;		
WY: A, BC, DE, HL, IX przechowane;		

**Procedury 6 i 7** stosują nieco inną konwencję współrzędnych ekranu niż użytkownik w BASIC'u. Przechodzenie jest stosunkowo proste:

rząd fizyczny = 24 — rząd logiczny (użytkownika)  
kolumna fizyczna = 33 — kolumna logiczna.

Przykład rozwieje ewentualne nieporozumienia: chcąc ustawić kursor w pozycji logicznej 4.5 (rząd, kolumna) dokonuje się następujących operacji:

```
LD      B,4
LD      C,5
LD      A,24
SUB     B
LD      B,A
LD      A,33
SUB     C
LD      C,A
```

WY: B,C — współrzędne fizyczne

Ustawiając pozycję należy podać również adres, co może być czasem dużym utrudnieniem, które pomoże pokonać Procedura 8

8. C—ADDR	3739	#0E9B
CEL: obliczenie adresu pola znakowego dla podanego rzędu fizycznego;		
WE: B — rząd fizyczny;		
WY: HL — adres najwyższego z ośmiu bajtów pola wydruku; flaga C = 0; BC, E, IX przechowane;		

Obliczenie adresu dla współrzędnych logicznych 4,5 wygląda następująco:

```
LD      B,4
LD      C,5
PUSH   BC
LD      A,24
SUB     B
LD      B,A
CALL   CL ADDR
LD      B,0
ADD    HL,BC
POP    BC
CALL   POS STORE
```

Ostatnia linia ustawia pozycję kursora według zawartości rejestrów BC (współrzędne) i HL (adres). Jeśli chcemy posługiwać się procedurą 8 podając współrzędną logiczną, to należy ją wywoływać pod adresem #0E9E i wprowadzać rząd w akumulatorze.

9. AT	8202	#200A
CEL: ustawienie pozycji wydruku znaku;		
WE: B — kolumna logiczna (brak konsekwencji); C — rząd logiczny;		
WY: jak w proc. 4		

10. TAB	8220	#201C
CEL: ustawienie kolumny wydruku znaku;		
WE: B — kolumna logiczna;		
WY: jak w proc. 4		

**Procedury 8 i 9** pozwalają przesuwac kursor ekranu tekstowego wedle zasad przyjetych w BASIC'u. Obie uzywaja procedury 4, wiec sa nieco wolniejsze od 7. **TAB** wydrukuje spacje od kolumny ostatniej pozycji do podanej.

11. PRINT CR	2639	#0A4F
CEL: wyslanie kodu 13 (#0D) na aktualny kanal;		
WE: nieistotne;		
WY: jak w proc. 4;		

Na ekranie bedzie to nowa linia, a w drukarce powrot glowicy i przesuniecie papieru o jedna linie (line feed). Jesli bufor byl czesciowo zapełniony, nastapi jego wydruk.

12. PRINT QUEST	2665	#0A69
CEL: wydruk znaku zapytania;		
WE: nieistotne;		
WY: jak w proc. 4;		
13. PRINT COMMA		#0A5F
CEL: przeniesienie kolumny wydruku do nastepnej strefy (czyli do kolumny 0 lub 16) polaczone z dodrukiem spacji;		
WE: nieistotne;		
WY: E, IX przechowane; uzywane rejestry pomocnicze (HL' przechowane);		

Zamiennikiem instrukcji BASIC'a:

**PRINT PAPER 4: INK 1; AT 5.2; „A”, „B”**, bedzie nastepujaca sekwencja instrukcji w jezyku wewnetrznym:

```
LD      A,33
LD      (ATTR ..., T), A
LD      B,2
LD      C,5
CALL    AT
LD      A, „A”
RST     16
CALL    PR COMMA
LD      A, „B”
RST     16
CALL    PRINT ..., CR
```

14. CL—SET	3545	# 0DD9
CEL: obliczenie adresu ekranu dla fizycznych wartosci rzędu i kolumny oraz uaktualnienie pozycji wydruku;		
WE: dla ekranu glownego: bit 1, (FLAGS) = 0; bit 0, (TVFLAG) = 0; B— rząd fizyczny; C— kolumna fizyczna; dla ekranu edycyjnego; bit 0, (TVFLAG) = 1; B i C — jw. dla drukarki: bit 1, (FLAGS) = 1; C— pozycja fiz. w buforze drukarki;		
WY: przez proc. 7; BC, HL, IX przechowane;		
15 PRINT BACK	2595	# 0A23
CEL: przesuniecie pozycji wydruku na ekranie/drukarce o jedna pozycje w lewo;		
WE: B— rząd fizyczny; C— kolumna fizyczna;		
WY: przez proc. 14;		

**Procedura 15**, powodujaca cofniecie pozycji, wydaje sie pomyka, gdyz trudno inaczej wytumaczyc szkolne bledy, jakie zawiera. Brakuje mozliwosci cofania z pozycji (1,0) do (0,31), przeoczony zostal takze szczegolny przypadek cofniecia w pozycji (0,0). Po uruchomieniu programu:

```
10 CLS: PRINT CHR$ B;
20 PRINT PEEK 23688
```

pozycja wydruku znajduje sie w obszarze atrybutow!

Ponizszy program rozwiazuje ten problem:

```
10 : ** BACKSPACE '86
20 : GENS      3M21  assembler
30 FETCH     EQU   # 0B03
40 CL...SET  EQU   # 0DD9
45 :
50           ORG   60000
60 START    CALL  FETCH
70         LD   A, 33
80         CP   C
90         JR   NZ, NZERO
100        LD   A, 24
110        CP   B
120        RET  Z
130 NZERO   INC   C
140        LD   A, 34
150        CP   C
160        JR   NZ, WYJ
170        BIT  1, (IY + 1)
180 :      (IY + 1) = (FLAGS)
190        JR   NZ, NLINIA
200        INC  B
210        LD   C, 2
220        LD   A, 25
230        CP   B
240        JR   NZ, WYJ
250        DEC  B
260 NLINIA  LD   C, 33
270 WYJ     JP   CL...SET
```

Linie 70...120 eliminuja cofniecie z pozycji (0,0). Program mozna uruchamiać z poziomu BASIC'a, zastepujac instrukcje: PRINT CHR\$ 8.

**K. MAMCARZ**



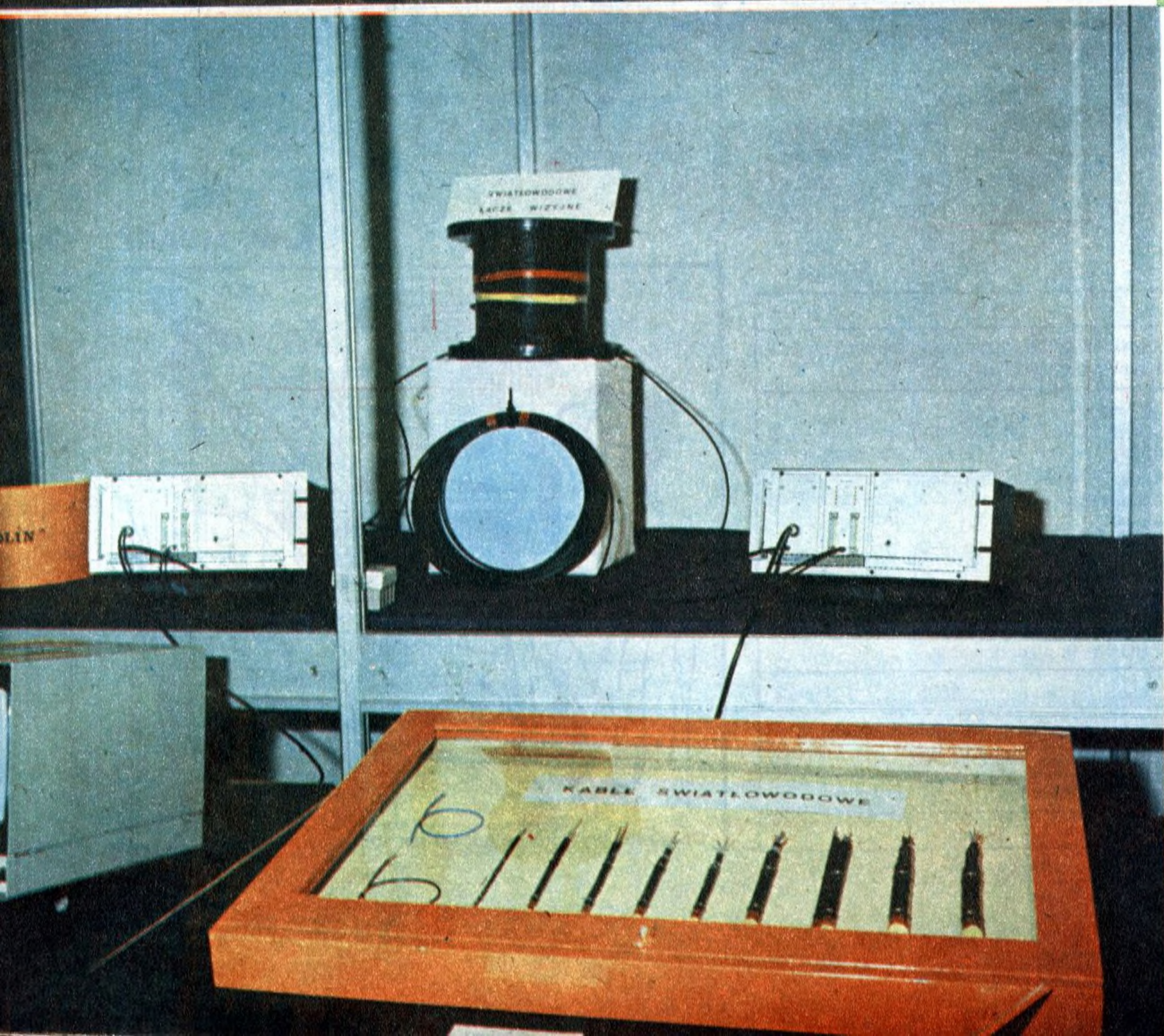
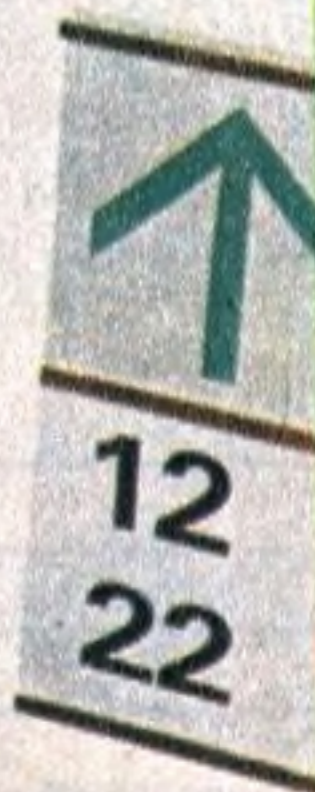
Dasz rade wykolowac go, tak jak kolujemy majstra?

Techniczna  
młoc

W Łodzi w dniach 27-28 września odbyła się wystawa, na której uczestnicy prezentowali swoje osiągnięcia. Główna część ekspozycji poświęcona była nowatorskim propozycjom dla rodzin (w tym ciekawych pomysłów dotyczących procesów i operacji produkcyjnych przemysłowych), a także nowym technologiom (światłowodów), które są oczekiwanych na rynku. Wśród wystawców znalazły się firmy komputerowe i przedstawiciele, które promowały swoje najnowsze osiągnięcia, takie jak komputer BOSMAN z...

Ekspozycję informatyczną klubów mikrokomputerowych na wystawie również organizowały. Swój dorobek prezentowali przedstawiciele z Poznania, Klubu Inżynierskiego i jednostki Warszawskiej. Wśród wystawców znalazł się również Klub Mikrokomputerowy przy Ośrodku Obrotu WSOŁ.

Imprezę ubarwiły koncerty muzyki młodzieżowej i wystawy kaset wideo.



## Twórczość młodzi

X—5 XI odbyła się interaktywnej młodzi twórcy pre-  
sięgnięcia innowacyjne.  
ycji to ponad 150 wyn-  
rozwiązań stanowiących  
tego przemysłu. Najwię-  
to automatyzacji proce-  
ywnych (modele robotów  
ze nowoczesnych tech-  
Przedstawiono projekty  
ku urządzeń elektroni-  
plotery). Kilka uznanych  
elektronicznych prezen-  
te produkty (np. mikro-  
UNIMOR-u).

tyczną wzbogacały stoi-  
puterowych. Nie zabrakło  
przedstawicieli wojska.  
owały Klub Wojsk Lotni-  
lub Mikrokomputerowy z  
tego OW oraz Garnizon-  
erowy TELEBIT działają-  
czeniowym zegrzyńskiej

pokazy mody, koncerty  
oraz projekcje filmów z

K. M.



## MŁODZIEŻOWY OŚRODEK INNOWACJI POLIN LUBLIN

ZOWY OŚRODEK  
Z SMP „POLIN”  
WSKA 12 20 071 LUBLIN

UJE:

CYNYCH ROZWIĄZAŃ  
GANIZACYJNYCH

ADZENIA I WYROBY  
SWIATELOWODOWE

ZAMY  
PRACY

**ŚWIATELOWODOWE**

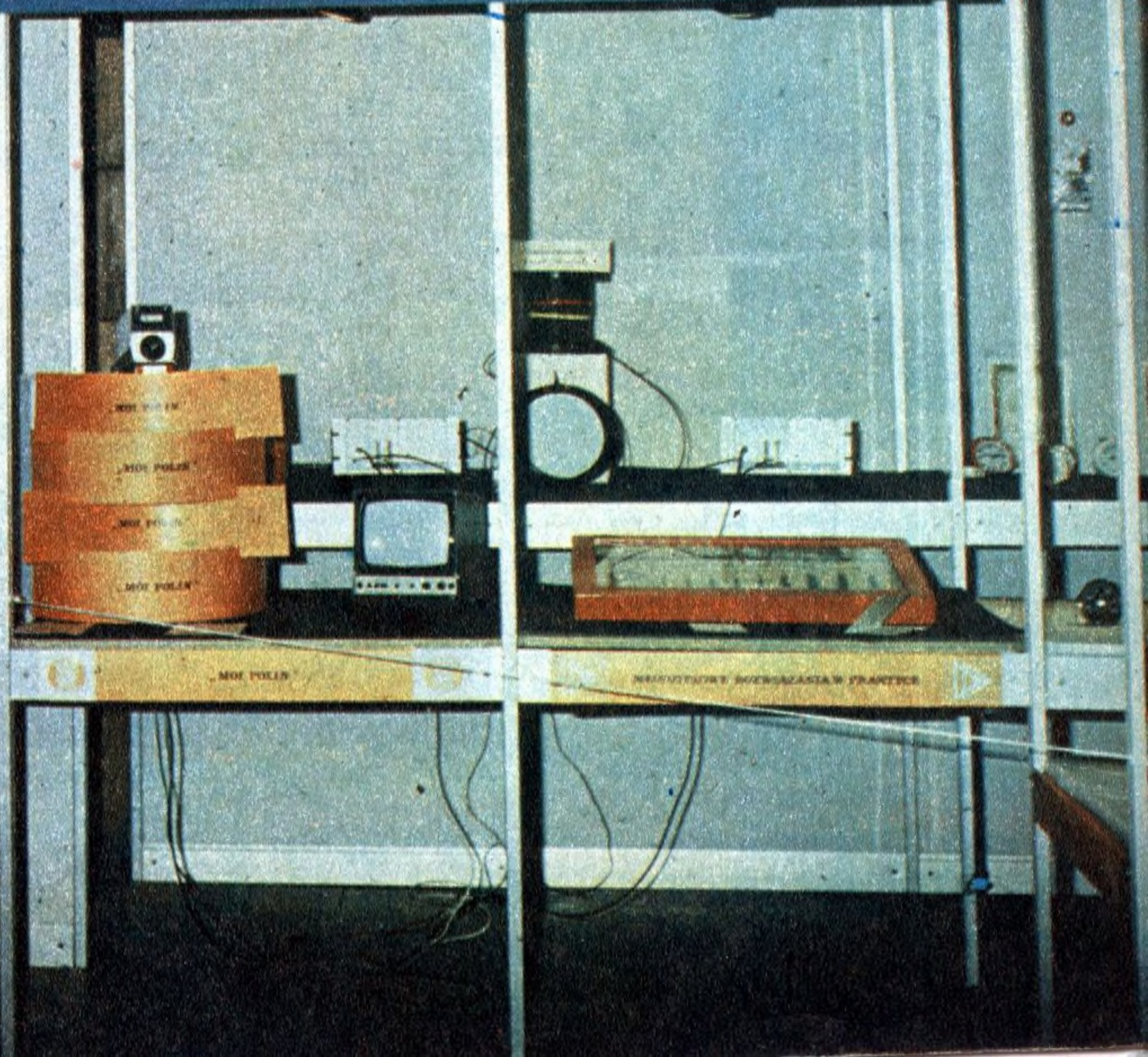
WIZYNE

KOMPUTEROWE RS 232c/124

TRANSMISJI SYGNAŁÓW CYFROWYCH

TELEMTRYCZNE

WIELOKANALOWE O MODULACH PCM

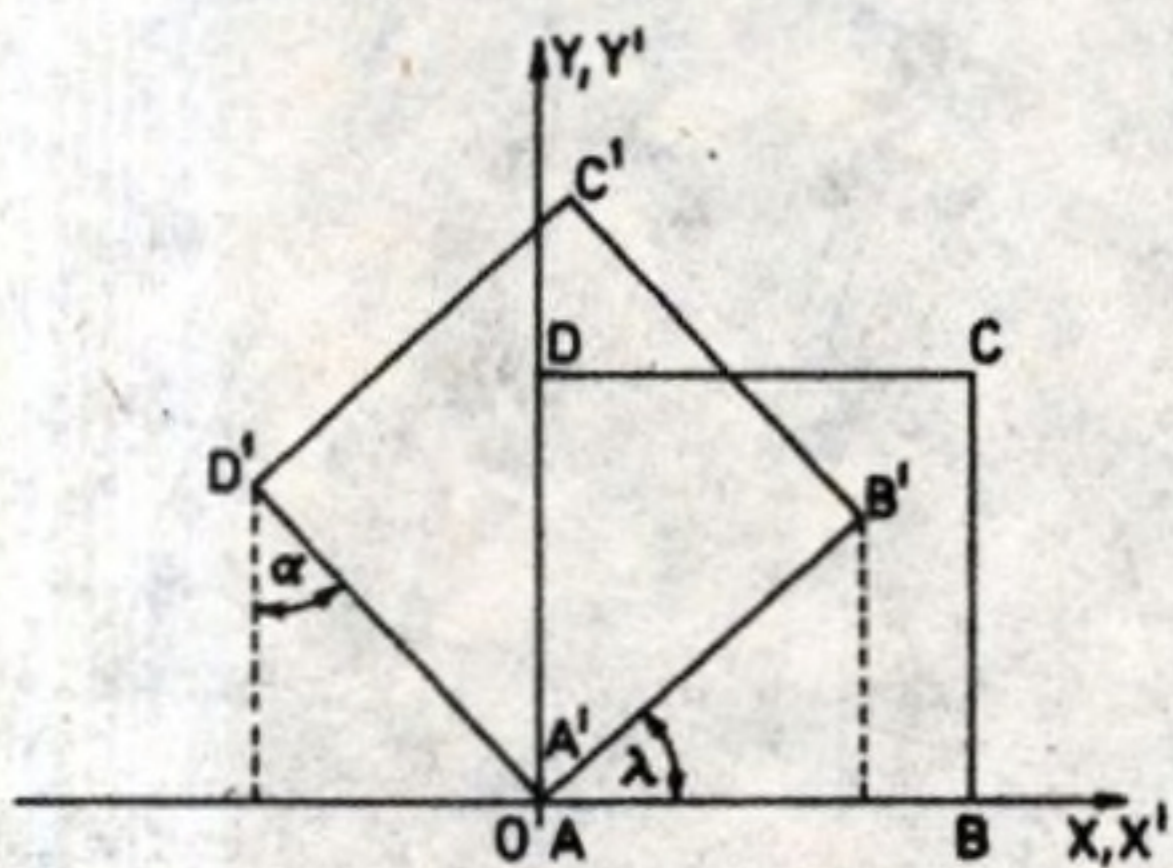


# Grafika na ekranie (2)

## OBRÓT

Istotnym przekształceniem, wykorzystywanym w grafice komputerowej, jest obrót względem wybranego punktu o żądany kąt. Przekształcenie to może być wykorzystywane do obrotu zdefiniowanych figur lub całych obrazów. Jednym z punktów, wokół którego wykonywany jest obrót, może być początek układu współrzędnych.

W numerze tym omówimy obrót na płaszczyźnie względem początku układu współrzędnych z wykorzystaniem rachunku macierzowego. Uogólnioną macierz  $2 \times 2$ , która pozwala realizować omawianą funkcję można wyznaczyć, rozpatrując obrót jednostkowego kwadratu (rys. 1.) w kierunku dodatnim, tzn. przeciwnym do kierunku obrotu wskazówek zegara.



Rys. 1. Obrót jednostkowego kwadratu o kąt  $\alpha$

Jak widać na rysunku, punkt B, mający przed obrotem współrzędne (1,0) przekształcił się w punkt B' o współrzędnych  $(x' = (1) \cdot \cos \alpha, y' = (1) \cdot \sin \alpha)$ , a punkt D o współrzędnych (0,1) w punkt D' o współrzędnych  $(x' = -(1) \cdot \sin \alpha, y' = (1) \cdot \cos \alpha)$ . Po zastosowaniu ogólnego przekształcenia na kwadracie jednostkowym ABCD otrzymujemy:

$$\begin{matrix} A \rightarrow [0 \ 0] \\ B \rightarrow [1 \ 0] \\ C \rightarrow [1 \ 1] \\ D \rightarrow [0 \ 1] \end{matrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix} \begin{matrix} - A' \\ - B' \\ - C' \\ - D' \end{matrix}$$

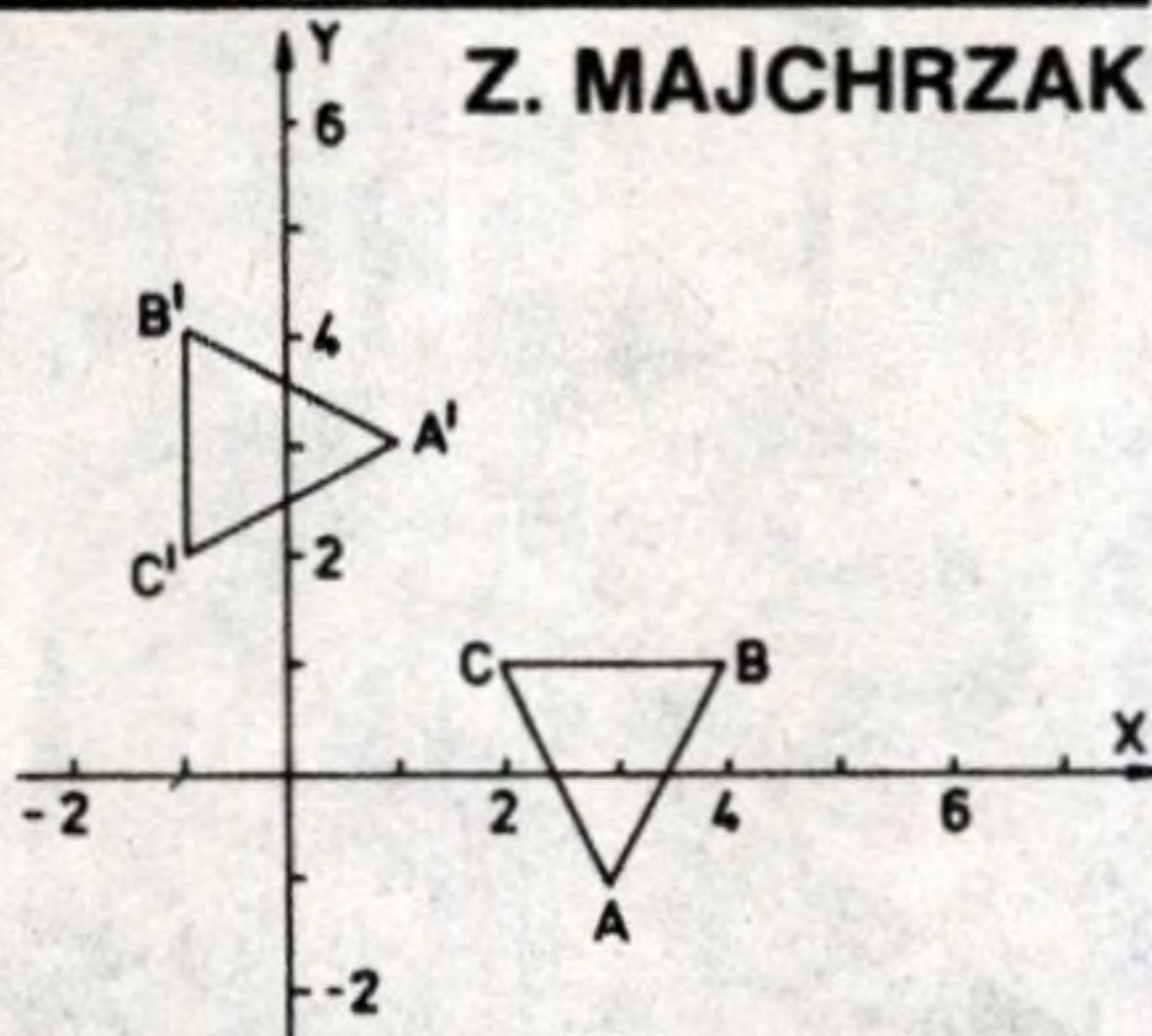
Z powyższego wynika, że współrzędne punktu B' określa się pierwszym wierszem macierzy transformacji, a współrzędne punktu D' drugim.

Z zestawienia powyższych wyników otrzymujemy postać poszukiwanej macierzy transformacji.

$$\begin{bmatrix} \cos & \sin \\ -\sin & \cos \end{bmatrix}$$

Otrzymana macierz transformacji nie powoduje zmiany skali i kształtu figury, lecz zmienia położenie na płaszczyźnie. Dla zilustrowania problemu rozpatrzmy trójkąt ABC, pokazany na rys. 2.

Obrót trójkąta ABC o  $90^\circ$  względem początku układu współrzędnych w kierunku



Rys. 2. Obrót trójkąta

przeciwnym do kierunku obrotu wskazówek zegara otrzymujemy poprzez transformację wierzchołków macierzą

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Otrzymujemy nowe współrzędne wierzchoł-

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix}$$

ków trójkąta ABC, które tworzą trójkąt A'B'C'. Obrót o  $180^\circ$  w kierunku dodatnim uzyskujemy dzięki macierzy

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \text{ a o } 270^\circ \text{ macierzy } \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

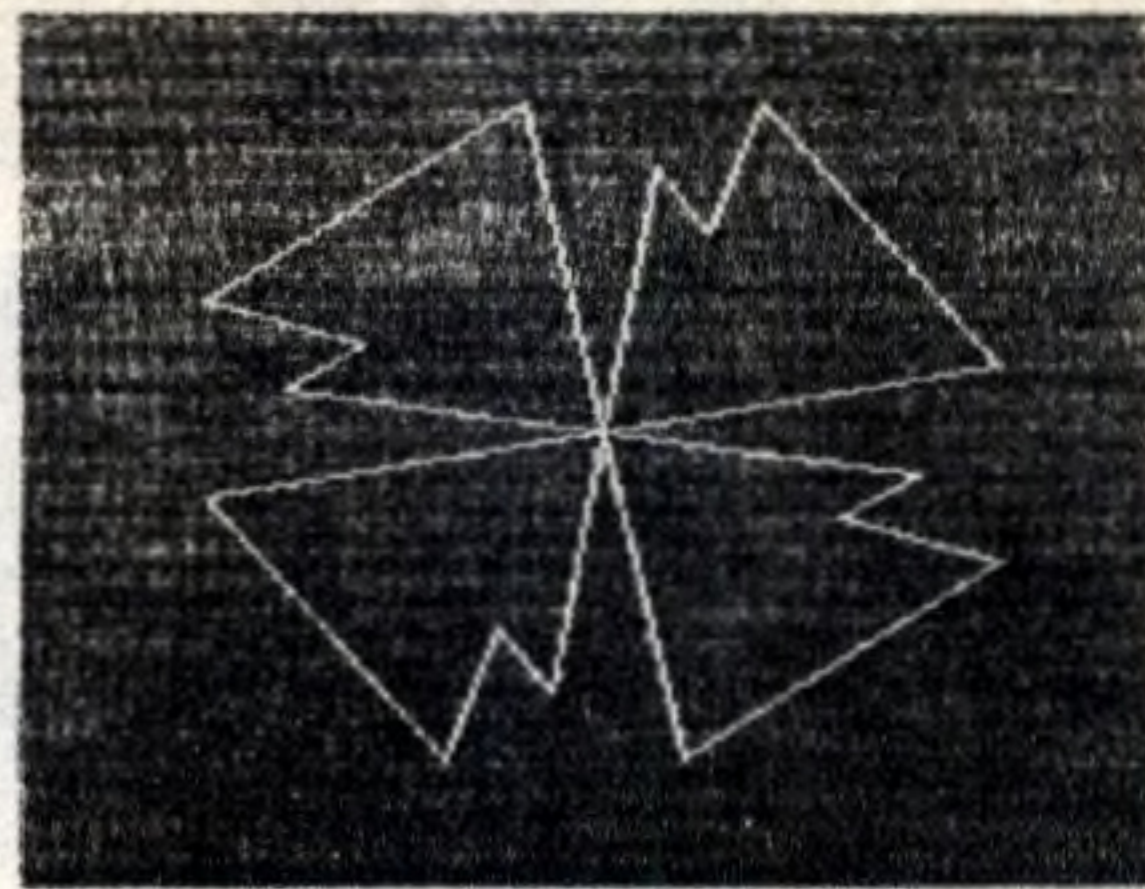
Przedstawiona tematyka została zastosowana w prezentowanym programie reali-



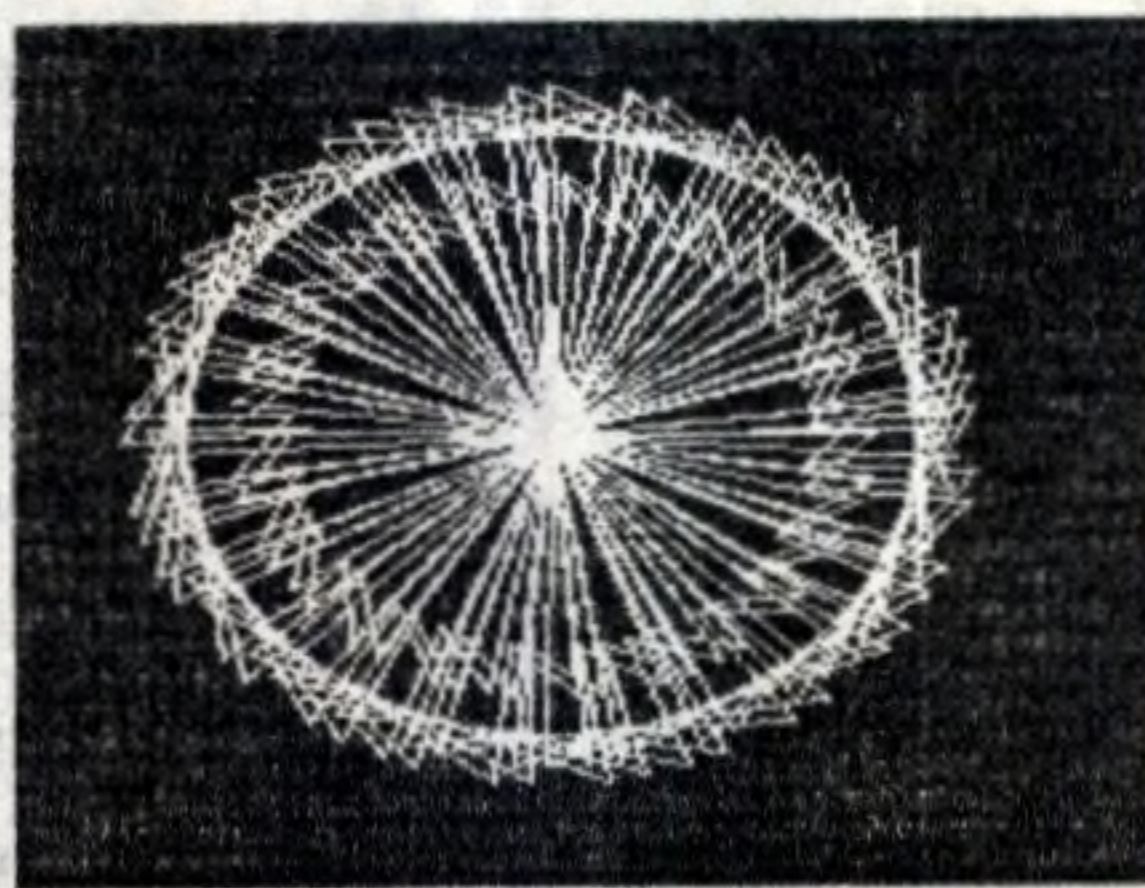
Rys. 3. Postać zdefiniowanej figury



Rys. 4. Obraz powstały w wyniku obrotu, figury o kąt  $180^\circ$



Rys. 5. Obraz powstały w wyniku obrotu figury w 3 krokach o kąt  $90^\circ$



Rys. 6. Obraz powstały w wyniku obrotu figury w 50 krokach o kąt  $43^\circ$

zującym obroty figur płaskich wokół początku układu współrzędnych. Program realizuje obroty jednokrotne o zadany kąt oraz wielokrotne w określony przyrost kąta.

Gdy zadana wartość kąta jest dodatnia mamy obrót przeciwny do kierunku obrotu wskazówek zegara i zgodny, gdy wartość kąta jest ujemna.

Otoczenie i sposób obsługi programu zostały ustalone w podobny sposób, jak w prezentowanym poprzednio, w ramach tego cyklu programie skalowania.

Obraz jest definiowany poprzez wprowadzenie współrzędnych (x,y) punktów wyznaczających kolejno kreślone odcinki. Współrzędne wprowadza się na zaproszenie „podaj wsp. punktu (x,y)”. W celu zakończenia definiowania figury należy wprowadzić punkt o współrzędnej  $x > 998$  oraz dowolnej wartości o współrzędnej y.

Po zakończeniu definiowania figury pojawia się komunikat „pojedynczy — j, wiele — w”, pozwalający wybrać rodzaj obrotu. Wprowadzenie litery „j” oznacza obrót jednokrotny o kąt, który należy wprowadzić po komunikacie: „podaj kąt”. Wprowadzenie litery „w” powoduje obrót figury w określonej ilości kroków o zadany przyrost kąta w jednym kroku.

Wartości te wprowadzamy po ukazaniu się komunikatów: „podaj krok”, „podaj ilość kroków”.

Po wykonaniu przez program obrotu pojawiają się następne komunikaty określające operacje, które możemy wykonać na otrzymanym obrazie.

Wyboru operacji dokonujemy przez wprowadzenie określonej litery, podanej w treści komunikatu.

Dla zilustrowania działania programu zamieszczamy przykładowe wyniki uzyskane na figurze zdefiniowanej punktami: (0,0); (20,120); (40,90); (60,150); (150,30); (0,0).

```

30 REM Obroty
60 DIM wsp(101,2),mat(2,2),nwsp(101,2)
70 MODE 1:INK 0,6:INK 1,13:INK 2,1:INK 3,24
80 BORDER 0:PEN 1:PAPER 2
90 CLS
100 PRINT "Rysowanie figur-poligonow"
110 PRINT "Obrot w jednym kroku lub"
120 PRINT "obrot w dowolnej ilosci krokow"
130 INPUT "Start gdy wcisniesz t"; nk$
140 IF nk$="t" THEN 160
150 GOTO 130
160 GRAPHICS PAPER 1:REM Tlo grafiki
170 ORIGIN 320,208,0,640,16,400
180 CLEAR INPUT
190 WINDOW 1,40,25,25
200 GOSUB 1220:REM Rysowanie ukladu wspol.
210 i=0:REM Licznik punktow
220 INPUT "podaj wsp. punktu x,y";x,y
230 IF x>998 THEN 300
240 i=i+1
250 IF i>100 THEN GOTO 300
260 wsp(i,1)=x
270 wsp(i,2)=y
280 GOSUB 830:REM Rysowanie odcinka
290 GOTO 220
300 INPUT "pojedynczy-j,wielo-w";jk$
310 IF jk$="j" THEN 340
320 IF jk$="w" THEN 650
330 GOTO 300
340 INPUT "podaj kat";ak
350 ak=ak*PI/180
360 GOSUB 900:REM Ustaw macierz przeksztal.
370 GOSUB 1020:GOSUB 1130:GOTO 380
380 INPUT "poczatek-r,koniec-k";nk$
390 IF nk$="r" THEN 420
400 IF nk$="k" THEN 1420
410 GOTO 380
420 INPUT "kasowanie obrazu-k,rysowanie-r";nk$
430 IF nk$="k" THEN 460
440 IF nk$="r" THEN 730
450 GOTO 420
460 INPUT "nowego obrazu-n,starego-s";nk$
470 IF nk$="n" THEN 610
480 IF nk$="s" THEN 500
490 GOTO 460
500 REM Kasowanie starego obrazu
510 GOSUB 1220:REM Rys. ukladu wspolrzednych
520 FOR k=1 TO i
530 wsp(k,1)=nwsp(k,1):wsp(k,2)=nwsp(k,2)
540 NEXT k
550 GRAPHICS PEN 3
560 GOSUB 1320
570 INPUT "obrot-o,dorysowanie-d";nk$
580 IF nk$="o" THEN 300
590 IF nk$="d" THEN 220
600 GOTO 570
610 REM Kasowanie nowego obrazu
620 GOSUB 1220
630 GRAPHICS PEN 3
640 GOSUB 1320:GOTO 570
650 INPUT "podaj krok";kr
660 INPUT "podaj ilosc krokow";nk
670 kr=kr*PI/180
680 FOR ii=1 TO nk
690 ak=ii*kr
700 GOSUB 900:GOSUB 1020:GOSUB 1130
710 NEXT ii
720 GOTO 380
730 FOR n=1 TO i:REM Rysowanie od poczatku

```

```

740 FOR k=1 TO 2
750 wsp(n,k)=0:nwsp(n,k)=0
760 NEXT k
770 NEXT n
780 GOTO 200
800 REM PODPROGRAMY
820 REM
830 REM Rysowanie punktu lub odcinka
840 REM
850 GRAPHICS PEN 3
860 IF i=1 THEN PLOT wsp(i,1),wsp(i,2)
870 IF i>1 THEN DRAW wsp(i,1),wsp(i,2)
880 RETURN
890 REM
900 REM Ustawianie macierzy
910 REM
920 sk=SIN(ak):ck=COS(ak)
930 mat(1,1)=ck:mat(1,2)=sk
940 mat(2,1)=-sk:mat(2,2)=ck
950 FOR k=1 TO i:REM Zerowanie nwsp
960 FOR l=1 TO 2
970 nwsp(k,l)=0
980 NEXT l
990 NEXT k
1000 RETURN
1010 REM
1020 REM Mnozenie macierzy
1030 REM
1040 FOR n=1 TO i
1050 FOR k=1 TO 2
1060 FOR m=1 TO 2
1070 nwsp(n,k)=nwsp(n,k)+wsp(n,m)*mat(m,k)
1080 NEXT m
1090 NEXT k
1100 NEXT n
1110 RETURN
1120 REM
1130 REM Rysowanie nowej figury
1140 REM
1150 GRAPHICS PEN 0:REM nowe wspolrzedne
1160 PLOT nwsp(1,1),nwsp(1,2)
1170 FOR n=2 TO i
1180 DRAW nwsp(n,1),nwsp(n,2)
1190 NEXT n
1200 RETURN
1210 REM
1220 REM Rysowanie ukladu wspolrzednych
1230 REM
1240 CLG 1
1250 GRAPHICS PEN 2:REM Pioro wspolrzednych
1260 PLOT 0,-208
1270 DRAW 0,208
1280 PLOT -320,0
1290 DRAW 320,0
1300 RETURN
1310 REM
1320 REM Rysowanie starej figury
1330 REM
1340 PLOT wsp(1,1),wsp(1,2)
1350 FOR n=2 TO i
1360 DRAW wsp(n,1),wsp(n,2)
1370 NEXT n
1380 RETURN
1390 REM
1400 REM Zakonczenie
1410 REM
1420 WINDOW 1,40,1,25
1430 CLS
1440 CLG 1
1450 END

```

# Gdy wyłączą prąd...

— Rozpoczynamy przesłuchanie Edgara Quaila, podejrzanego o zamordowanie swojej żony. Głos ma oskarżony!

— Wysoki sędzie! W dobie postępującej komputeryzacji i automatyzacji życia nasze dotychczasowe powłoki zaczynają być bezużyteczne, często hamując postęp naszej wspaniałej cywilizacji. Nasze ciało ma takie wygórowane wymagania, jak określona temperatura otoczenia, oddychanie tlenem, zapotrzebowanie na znaczną w skali globalnej ilość materii organicznej jako pokarmu; że nie wspomnę o takich utrudnieniach, jak golenie czy mycie. Już kilkanaście lat temu odkryłem możliwość przeniesienia informacji z mózgu żyjącego organizmu do pamięci dowolnego komputera. Jest to wspaniała szansa dla naszego świata!!! Wyobraźmy sobie...

— Niech oskarżony powie jeszcze sądowi o skutkach ubocznych...

— Hm, no tak... Podczas kopiowania w organizmie dawcy następują nieodwracalne zmiany, spowodowane przez... zresztą, nieważne.. W każdym razie zmiany te mogą doprowadzić do śmierci, ale jedynie w b i o l o g i c z n y m tego słowa znaczeniu! Umysł nadal żyje, a

w dodatku obdarzony jest nowymi, wspaniałymi możliwościami! Wyobraźmy sobie, że zamiast tego zlepku białek mamy poprzez nasz elektroniczny mózg dostęp, i to bezpośredni, do wszystkich wynalazków techniki. Możemy na przykład sterować sondami, badającymi przestrzeń kosmiczną, i czuć się częścią ich orga... przepraszam, mechanizmu. W ten sposób każdy będzie mógł zrealizować swoje dziecięce marzenia o podróżach kosmicznych, nie narażając przy tym swojego istnienia na żadne ryzyko, zostawiwszy duplikat swojego mózgu na przykład w Banku Pamięci.

Każdy wie, że jednym z odwiecznych marzeń człowieka było przedłużenie swojego życia. Dobrze funkcjonujące mechanizmy dadzą nam tę możliwość — mamy szansę stać się nieśmiertelni!

— Czyżby żona oskarżonego stała się rzeczywiście nieśmiertelna?

— To nie była moja wina! Skąd mogłem wiedzieć, że akurat wtedy elektrownia wyłączy mi dopływ prądu?! To był tylko tragiczny wypadek, że cała zawartość mózgu mojej żony została skasowana, zanim zapisałem ją na dyskietkę. Mój komputer jest modelem starszego typu i nie ma własnego, awaryjnego źródła napięcia...



## Charakterystyka zestawu podstawowego

- I. Jednostka centralna:
  - mikroprocesor: Intel 8088-2/NEC V20
  - zegar procesora 4.77/8 MHz
  - koprocesor arytmetyczny 8087 (opcjonalnie)
  - pamięć operacyjna 256KB do 1024KB
  - pamięć stała ROM 8KB do 64KB (BIOS i Basic)
  - pamięć masowa: 2 jednostki dysków elastycznych 360KB, do 2 jednostek dysków typu Winchester 20MB, szybka pamięć taśmowa 20 do 60 MB
  - pakiet wizyjny: O tryb tekstowy — max. 80 znaków w 25 wierszach, O tryb graficzny — rozdzielczość 640 x 200 (czarno-biały), 320 x 200 (kolor), O wyjścia: RGB i sygnał zespolony
  - interfejsy: O równoległy Centronics (drukarka), O szeregowy RS232
  - zegar czasu rzeczywistego z podtrzymaniem baterijnym
  - 8 gniazd do podłączenia dodatkowych pakietów (grafika, specjalizowane kontrolery, karty prototypowe itp.)
  - zasilacz 150 W o wysokiej sprawności

## Sumy kontrolne

W numerze czwartym "X" (1987) opublikowano program pt. „Edytor Basic-a”. Celem jego było ułatwienie bezbłędnego wprowadzenia wydrukowanych programów, a w razie wystąpienia błędu łatwe jego odszukanie. Ten sam cel przyświecał opracowaniu poniższego programu. Otrzymywane za jego pomocą sumy kontrolne są identyczne jak sumy uzyskane dzięki powyżej wspomnianemu programowi.

Po uruchomieniu programu (RUN), ukazuje się nam menu. Mamy do wyboru cztery możliwości: wyświetlanie sum kontrolnych na ekranie, drukowanie ich na drukarce, powrót programu do Basic-a i dla posiadaczy stacji dysków — wywołanie zawartości dyskietki. W przypadku wybrania opcji drukowanie lub wyświetlanie należy: gdy wprowadzamy program z dyskietki, wpisać nazwę programu, a gdy z magnetofonu, po poleceniu wpisania nazwy programu, trzeba wcisnąć klawisz RETURN. Program kolejno wypisuje etykiety oraz przyporządkowane im sumy kontrolne w sposób, jak na załączonym przykładzie.

Na koniec jeszcze jedna uwaga. Przed przystąpieniem do sprawdzania, powinniśmy dany program zapisać na taśmie lub dyskietce instrukcją LIST.

Janusz W. JANIEC

```

100 REM *****
105 REM *
110 REM *   SUMY   KONTROLNE   *
115 REM *
120 REM *****
125 DIM A$(130),B$(130),C$(6),D$(
(10),A(5),N$(15),AN$(14),SUM$(2)
130 DIM ME$(18):N$="D:" :B=1:E=0
135 OPEN #3,8,0,"E:" :SETCOLOR 2,
0,0:SETCOLOR 1,0,10:POKE 752,1
140 REM *** OPCJE PROGRAMU ***
:POKE 82,4:? CHR$(125):POKE 752,
1:SETCOLOR 2,0,0:SETCOLOR 1,0,10
145 POSITION 2,1:? "SUMY KONTROL
NE":POSITION 2,4:? CHR$(177);" W
YSWIETLANIE SUMY KONTROLNEJ"
150 POSITION 2,6:? CHR$(178);" D
RUKOWANIE SUMY KONTROLNEJ":POSIT
ION 2,8:? CHR$(179);" POWROT DO
BASIC-a"
155 POSITION 2,10:? CHR$(180);"
MENU DYSKIETKI":POSITION 9,13:?
">?<";
160 CLOSE #1:OPEN #1,4,0,"K:" :GE
T #1,K:? CHR$(30);CHR$(30);CHR$(
K+128):IF K<49 OR K>52 THEN 160
165 IF K=49 THEN 190
170 IF K=51 THEN 490
172 IF K=52 THEN 500

```

```

175 REM ** WYDRUK NA DRUKARCE **
180 TRAP 450:CLOSE #3:OPEN #3,8,
0,"P:"
185 REM *** OTWARCIE ZBIORU ***
190 POSITION 2,15:? "Podaj nazwe
zbioru do sprawdzenia"
191 POSITION 7,16:? CHR$(17);:FO
R L1=1 TO 15:? CHR$(18);:NEXT L1
:? CHR$(5)
195 POSITION 7,17:? "ID:
I":POSITION 7,18:? CHR$(26)
;:FOR L1=1 TO 15:? CHR$(18);:NEX
T L1:? CHR$(3)
200 POSITION 10,17:? ;:INPUT AN$
205 POKE 752,1
210 IF AN$(1,1)=" " THEN N$="C:"
:POSITION 8,17:? "C:" :GOTO 225
215 SK=LEN(AN$):N$(3,SK+2)=AN$
220 TRAP 235
225 CLOSE #1:OPEN #1,4,0,N$
230 GOTO 240
235 ? :? "Nie ma takiego zbioru
na dyskietce !":GOSUB 460:RUN
240 POKE 82,0:P=0:B=1:E=0:? CHR$(
(125);" TWORZENIE SUM KONTROLNYC
H...":?
255 REM ***** OPIS *****
270 A$=" * SUMA KONTROLNA /
ETYKIETA *":? #3;A$
275 A$=" -----
-----":? #3;A$
280 REM ** POBRANIE ETYKIETY **
285 C=1:TRAP 335
290 GET #1,Z
295 IF Z=32 THEN C$(C,C)=CHR$(32
):GOTO 310
300 C$(C,C)=CHR$(Z):C=C+1:GOTO 2
90
310 IF E=1 THEN 435
315 REM *** POBRANIE LINII ***
320 B$="":B$=C$:T=C+1:C=0
325 GET #1,Z:IF Z=155 THEN 340
330 B$(T+C,T+C)=CHR$(Z):C=C+1:GO
TO 325
335 E=1:GOTO 310:REM * KONIEC *
340 REM ***** SUMA LINII *****
345 C=0:ANS=0
350 FOR A=1 TO LEN(B$)
355 C=C+1:ANS=ANS+(C*ASC(B$(A,A)
)):NEXT A
360 KOD=INT(ANS/676):KOD=ANS-(KO

```

```

D*676)
365 HKOD=INT(KOD/26):LKOD=KOD-(H
KOD*26)+65
370 HKOD=HKOD+65
375 SUM$(1,1)=CHR$(HKOD):SUM$(2,
2)=CHR$(LKOD)
380 REM *** FORMAT ETYKIETY ***
385 D$=" " :L=VAL(C$):A=LEN(STR
R$(L)):D$(6-A)=STR$(L):GOTO 410
395 ? CHR$(125);"BLEDNE DANE !!!
";CHR$(125):END
400 REM ** DRUKOWANIE WYNIKU **
405 REM ***** NA EKRANIE *****
410 IF P=0 THEN CC=1:X=2:P=1:A$="
" :A$(38)=" " :A$(2)=A$:GOTO 425
415 X=X+10:IF X>=39 THEN X=2:Y=Y
+1: ? #3;A$:A$=" " :A$(38)=" " :A$(
2)=A$
425 A$(X,38)=SUM$:A$(X+2,38)=CHR
$(32):A$(X+3,39)=D$
430 FOR LI=10 TO 30 STEP 10:A$(L
I,LI)="I":NEXT LI:GOTO 280
435 ? #3;A$
440 A$=" -----
----- " : ? #3;A$
445 CLOSE #1:GOSUB 460:RUN
450 ? : ? : ? "Drukarka nie dziala
!" : ? : ? :GOSUB 460:RUN
455 REM *** WCISNIJ KLAWISZ ***
460 CLOSE #1:OPEN #1,4,0,"K:"
465 ? " " ;CHR$(17);:FOR L1=1
TO 23: ? CHR$(18);:NEXT L1: ? CHR
$(5)
470 ? " |Wcisnij dowolny kla

```

```

wiszl"
475 ? " " ;CHR$(26);:FOR L1=1
TO 23: ? CHR$(18);:NEXT L1: ? CHR
$(3)
480 GET #1,Z:CLOSE #1:RETURN
485 REM ** POWROT DO BASIC-a **
490 GRAPHICS 0:END :REM *****
500 REM ***** MENU DYSKIETKI *****
505 TRAP 525
510 CLOSE #3:OPEN #3,6,0,"D:*. *"
:FOR KO=0 TO 64:INPUT #3,ME$
520 ? ME$:NEXT KO
525 CLOSE #3
530 GOSUB 455:GOTO 135:REM *****

```

\* SUMA KONTROLNA / ETYKIETA \*

PC	100I	YF	105I	XS	110I	YH	115
PG	120I	YB	125I	OI	130I	YM	135
TS	140I	LV	145I	UJ	150I	RB	155
ND	160I	SZ	165I	RO	170I	MI	172
TF	175I	EM	180I	HJ	185I	RY	190
KF	191I	UE	195I	HK	200I	JV	205
UF	210I	BN	215I	NU	220I	ES	225
NP	230I	LJ	235I	YJ	240I	MK	265
VD	270I	JY	275I	KY	280I	TV	285
NY	290I	ML	295I	WG	300I	YK	310
GS	315I	TK	320I	SP	325I	XI	330
CK	335I	WJ	340I	GN	345I	IG	350
JJ	355I	WD	360I	OE	365I	PY	370
OZ	375I	YC	380I	DS	385I	OO	395
SV	400I	FP	405I	LQ	410I	OH	415
IF	425I	XI	430I	SG	435I	GY	440
QQ	445I	IR	450I	PW	455I	XK	460
PZ	465I	SK	470I	KW	475I	JI	480
AV	485I	TQ	490I	NJ	500I	PB	505
YJ	510I	SY	520I	MS	525I	KJ	530

## Mapa dysku

Program DYSKMAP tworzy i wyświetla na ekranie monitora mapę dysku. Pod pojęciem tym kryje się wykaz wszystkich sektorów dysku z zaznaczeniem, który z nich jest wolny (niezapisany), a który zajęty (zapisany). Program umożliwia tworzenie mapy dysku sformatowanego z pojedynczą gęstością zapisu (DOS 2.0—720 sektorów) oraz z podwójną gęstością (DOS 2.5—1040 sektorów).

Program po uruchomieniu żąda podania liczby sektorów zawartych na dyskietce. Jeżeli nie jesteśmy pewni, z jaką gęstością zapisana jest dyskietka, to podajemy 1040. Program jest tak skonstruowany, że gdy wprowadzimy liczbę sektorów równą 1040, to sprawdza, z jaką gęstością jest rzeczywiście sformatowana dyskietka. Informacja o tym zapisana jest w sektorze o numerze 360, w bajcie pierwszym i drugim (licząc od zera), zwanym tabelą zawartości wolumenu (dyskietki) — ang. Volume Table of Contents. Jeżeli w wyniku sprawdzenia okaże się, że na dyskietce jest 720 sektorów, to program sam poprawi wprowadzoną przez nas liczbę. Wprowadzenie liczby, na przykład 100, spowoduje, że program utworzy i wyświetli mapę zawierającą informacje o pierwszych stu sektorach dyskietki.

Mapa dysku zawierającego 720 sektorów wyświetlana jest na ekranie w postaci 20-wierszowej tablicy zawierającej po 36 znaków w każdym wierszu. Każdy element (znak) z tej tablicy opisuje stan jednego sektora. Numer sektora, któremu odpowiada znak z tablicy, odczytuje się przez dodanie liczby zapisanej na początku wiersza do liczby zapisanej na dole kolumny, zawierających dany znak. Znaczenie znaków z tablicy jest następujące:

Gwiazdka — oznacza sektor zapisany, kropka — oznacza sek-

tor wolny, wykrzyknik — oznacza sektor uszkodzony, tzn. taki, który przy próbie odczytu generuje błąd.

Tę ostatnią cechę programu można wykorzystać do wykrywania i oznaczania tak zwanych bad sektorów, stosowanych przez producentów oprogramowania do jego zabezpieczenia przed kopiowaniem.

Mapa dysku sformatowanego z podwójną gęstością zapisu (1040 sektorów) składa się z dwóch części. Część pierwsza opisuje sektory od 1 do 520, a część druga sektory od 521 do 1040. Każda z części jest również tablicą, analogiczną do opisanej poprzednio, zawierającą 20 wierszy 26-znakowych. Klawisz OPTION przełącza wyświetlanie części pierwszej na drugą i odwrotnie. Po przeanalizowaniu utworzonej mapy można program uruchomić ponownie, wciskając klawisz START.

Program DYSKMAP zawiera kilka procedur napisanych w kodzie maszynowym (linie DATA programu). Szczególną uwagę proponuję zwrócić na procedurę umieszczoną w 40-bajtowej stałej tekstowej o nazwie S — linie 802, 806. Jej funkcją jest odczyt lub zapis sektora o dowolnym numerze. Zwracam uwagę na tę procedurę, ponieważ odczyt i zapis sektora za pomocą instrukcji języka BASIC (NOTE, POINT, PUT, GET) jest dosyć kłopotliwy (wymaga skonstruowania pętli) i trwa znacznie dłużej. Procedurę tę można wykorzystać w każdym programie w języku BASIC. Sposób jej wywołania jest następujący:

X = USR(PAR1,PAR2,PAR3,PAR4,PAR5)

gdzie:

PAR1 — adres pamięci RAM, od którego umieszczona jest procedura,

PAR2 — numer napędu dyskowego (1—4),

PAR3 — numer sektora do odczytu lub zapisu,

PAR4 — adres 128-bajtowego bufora, do którego mają być

wczytane informacje z sektora, a przy zapisie informacje zawarte w tym buforze zostaną zapisane do sektora

PAR5 — kod funkcji do wykonania, 82 — odczyt sektora, 87 — zapis sektora.

Przykładowo, jeżeli procedurę umieściliśmy w tablicy PROCES i chcemy wczytać sektor o numerze 360 do tablicy BUFS, to procedurę należy wywołać w sposób poniższy:

```

5 ? CHR$(125):? "Proszę chwile p
oczekac..."
6 GOTO 100
10 ? CHR$(125):? "    ** DYSKMA
P (C) 1987 ***"
20 ? :? :? "Włoz dyskietke do st
acji dyskow"
30 ? "Nacisnij [RETURN]"
31 ? "CSPACJAJ Wyjście z program
u"
34 CLOSE #1:OPEN #1,4,0,"K:"
35 GET #1,K
39 IF K=32 THEN POKE 752,0:END
40 IF K<>155 THEN 35
45 RETURN
100 DIM L$(19),DMAP$(117),MAP$(
1041),OKNO$(107),L$(60),L2$(60),
L3$(80),SE$(128),S$(40)
110 RESTORE 200
120 FOR I=1 TO 19:READ A:L$(I)=
CHR$(A):NEXT I:GOTO 300
200 DATA 104,104,133,204,104,133
,203,160,0,169,0,145,203,200,192
,240,208,249,96
300 RESTORE 350
310 FOR I=1 TO 117:READ A:DMAP$(
I)=CHR$(A):NEXT I:GOTO 400
350 DATA 104,104,133,204,104,133
,203,104,104,141,1,3,104,133,207
,104,133,206,169,4
352 DATA 141,5,3,169,0,141,4,3,1
69,82,141,2,3,160,1,162,0,169,1,
141,10,3,142,11,3,138,72,152,72,
32
354 DATA 83,228,160,0,173,3,3,20
1,1,208,24,190,0,4,208,12,200,19
2,128,208,246,104,168,169,14,56
356 DATA 176,11,104,168,169,10,5
6,176,4,104,168,169,1,145
358 DATA 203,104,170,200,208,3,2
32,230,204,228,207,208,4,196,206
,240,9,140,10,3,142,11,3,56,176,
185,96
400 RESTORE 450
410 FOR I=1 TO 107:READ A:OKNO$(
I)=CHR$(A):NEXT I:GOTO 500
450 DATA 104,104,133,206,104,133
,205,104,133,118,104,133,117,104
,133,208,104,133,207,104,104
452 DATA 133,209,104,133,120,104
,133,119,165,88,133,203,165,89,1
33,204,165,203,101,117,133,203,1
65,204
454 DATA 101,118,133,204,160,0,1
77,203,170,177,205,145,203,138,1
45,205,200,196,209,208,241,169,0
,198,207
456 DATA 197,207,208,6,197,208,2
40,28,198,208,24,152,101,205,133
,205,144,3,230,206
458 DATA 24,165,203,101,119,133,
203,165,204,101,120,133,204,24,1
44,199,96
500 RESTORE 550
510 FOR I=1 TO 60:READ A:L$(I)=C
HR$(A):NEXT I:GOTO 600
550 DATA 0,0,16,0,19,22,0,23,18,
17,16,24,17,20,20,17,24,16,18,17
,22,18,21,18,18,24,24,19,18,20,1
9,22,16,19
552 DATA 25,22,20,19,18,20,22,24
,21,16,20,21,19,16,21,23,22,22,1
7,18,22,20,24,22,24,20
600 RESTORE 650
610 FOR I=1 TO 60:READ A:L2$(I)=
CHR$(A):NEXT I:GOTO 700
650 DATA 0,0,16,0,18,22,0,21,18,

```

```

0,23,24,17,16,20,17,19,16,17,21,
22,17,24,18,18,16,24,18,19,20,18
,22,16,18
552 DATA 24,22,19,17,18,19,19,24
,19,22,20,19,25,16,20,17,22,20,2
0,18,20,22,24,20,25,20
700 RESTORE 750
710 FOR I=1 TO 80:READ A:L3$(I)=
CHR$(A):NEXT I:GOTO 800
750 DATA 0,21,18,16,0,21,20,22,0,
,21,23,18,0,21,25,24,0,22,18,20,
0,22,21,16,0,22,23,22,0,23,16,18
,0,23,18,24
752 DATA 0,23,21,20,0,23,24,16,0
,24,16,22,0,24,19,18,0,24,21,24,
0,24,24,20,0,25,17,16,0,25,19,22
,0,25,22,18
754 DATA 0,25,24,24,17,16,17,20
800 SE$(1)=CHR$(0):SE$(128)=SE$
SE$(2)=SE$
802 RESTORE 805:FOR I=1 TO 40:RE
AD A:S$(I)=CHR$(A):NEXT I:GOTO 8
10
805 DATA 104,104,104,141,1,3,104
,141,11,3,104,141,10,3,104,141,5
,3,104,141,4,3,104,104,141,2,3,3
2,83,228
806 DATA 173,3,3,133,212,169,0,1
33,213,96
810 GOSUB 10:?:? "Podaj liczbe
sektorow [720] lub [1040] "
820 INPUT D0
830 IF D0=1040 THEN GOTO 900
840 X=USR(ADR(L0$),40000+17*40):
GOTO 5510
900 SEK=360:X=USR(ADR(S$),1,SEK,
ADR(SE$),82):ALA=ADR(SE$):LICZS=
PEEK(ALA+1)+256*PEEK(ALA+2)
901 IF LICZS=1010 THEN GOTO 840
902 ? :? "Dyskietka o pojedyncze
j sestosci":? "Liczba sektorow w
ynosi 720":D0=720:GOTO 840
5510 ? :? :? "Czytanie dysku "
5520 ? :? "Czekaj..."
5530 MAP$="":MAP$(D0,DO)=CHR$(0)
5540 X=USR(ADR(DMAP$),ADR(MAP$)-
1,1,DO+1)
5550 IF D0=1040 THEN 5700
5560 POKE 752,1:?:CHR$(125):POSI
TION 0,20:?:CHR$(30):CHR$(31);
5570 X=USR(ADR(OKNO$),ADR(MAP$),
4,20,36,40):X=USR(ADR(OKNO$),ADR
(L$),0,20,3,40)
5580 POSITION 4,20:?"0000000001
11111111222222222233333333 1234
56789012345678901234567890123456
"
5590 POSITION 15,23:?"[START] =
NOWA MAPA";
5600 IF PEEK(53279)=6 THEN 5620
5610 GOTO 5600
5620 X=USR(ADR(OKNO$),ADR(L$),0,
20,3,40):GOTO 810
5700 POKE 752,1:?:CHR$(125):POSI
TION 0,20:?:CHR$(30):CHR$(31);
5710 POSITION 9,20:?"0000000001
1111111122222222 1234
5678901234567890123456"
5715 X=USR(ADR(OKNO$),ADR(L2$),5
,20,3,40)
5720 X=USR(ADR(OKNO$),ADR(MAP$),
9,20,26,40)
5730 POSITION 3,23:?"[START]= N
OWA MAPA [OPTION]= 2 CZESC";
5740 IF PEEK(53279)=6 THEN 5770
5750 IF PEEK(53279)=3 THEN 5800
5760 GOTO 5740

```

X = USR(ADR(PROCS),1,360,ADR(BUFS),82)  
Odczyt nastąpi ze stacji dysków o numerze 1.  
Po wykonaniu procedury zmienna X zawierać będzie informacje o tym, czy operacja odczytu bądź zapisu zakończyła się pomyślnie. Wartość zmiennej X różna od jedności oznacza, że przy odczycie lub zapisie wystąpił błąd.

**H. KRASUSKI**

```

5770 X=USR(ADR(OKNO$),ADR(L2$),5
,20,3,40):GOTO 810
5800 X=USR(ADR(OKNO$),ADR(MAP$),
9,20,26,40)
5802 X=USR(ADR(OKNO$),ADR(L2$),5
,20,3,40)
5805 X=USR(ADR(OKNO$),ADR(L3$),4
,20,4,40)
5810 X=USR(ADR(OKNO$),ADR(MAP$)+
520,9,20,26,40)
5820 POSITION 3,23:?"[START]= N
OWA MAPA [OPTION]= 1 CZESC";
5830 IF PEEK(53279)=6 THEN 5860
5840 IF PEEK(53279)<>3 THEN 5830
5845 X=USR(ADR(OKNO$),ADR(L3$),4
,20,4,40)
5850 X=USR(ADR(OKNO$),ADR(MAP$)+
520,9,20,26,40):GOTO 5715
5860 X=USR(ADR(OKNO$),ADR(L3$),4
,20,4,40):GOTO 810

```

\* SUMA KONTROLNA / ETYKIETA \*

CK	51	08	61	EA	101	MZ	20
GB	301	SS	311	OM	341	II	35
YX	391	VT	401	MM	451	PR	100
RK	1101	WQ	1201	OL	2001	UP	300
PP	3101	AJ	3501	OB	3521	XV	354
XY	3561	EI	3581	VD	4001	NT	410
DI	4501	GO	4521	YA	4541	RI	456
ME	4581	VR	5001	JA	5101	GZ	550
JC	5521	WF	6001	XZ	6101	EL	650
UP	6521	WT	7001	CI	7101	DB	750
XO	7521	JQ	7541	JN	8001	NI	802
ML	8051	IF	8061	UU	8101	UV	820
TE	8301	TT	8401	CO	9001	SW	901
SI	9021	VH	55101	IH	55201	JW	5530
GE	55401	WB	55501	AF	55601	DM	5570
IA	55801	YD	55901	FE	56001	RT	5610
BE	56201	ZR	57001	BG	57101	OE	5715
AP	57201	LI	57301	MP	57401	DI	5750
UW	57601	BY	57701	AL	58001	NR	5802
PK	58051	BJ	58101	JG	58201	MN	5830

```

JANUSZ MILLER
SHARP PC-1500
R E F L E K S

500:"A":RANDOM:
PAUSE "POMIAR
REFLEKSU"
510:N=0:WAIT 0:FOR
K=1TO 30
520:A$=CHR$(64+
RND 26):BEEP 1
530:B$=INKEY$:N=N
+1
540:CLS:PRINT " *
";N;" * ";A$
550:IF A$<>B$GOTO
530
560:NEXT K:CLS:
WAIT
570:PRINT "CZAS=";
STR$ N;" JEDNO
STEK"
580:INPUT "JESZCZE
RAZ ? T=1 N=0
";T:IF T=1
GOTO 510
590:END

```

Grafika komputerowa jest bardzo ważną dziedziną informatyki, rozwijaną w wielu liczących się ośrodkach. Jej celem jest wypracowanie efektywnych pakietów programowych (standardów) umożliwiających łatwe i szybkie generowanie obrazów figur i brył z uwzględnieniem punktu widzenia i napotykanym przeszkód.

W tym kursie nie jest możliwe przedstawienie żadnego znanego standardu graficznego. Ograniczymy się wyłącznie do zaprezentowania koncepcji kreślenia kolorowych obrazów w komputerze AMSTRAD CPC 6128 na podstawie pakietu graficznego CPC 6128 — Grafik Version 1.0 firmy Heimsoeth Software Munich.

Aby włączyć pakiet do poszczególnych programów, wykorzystano specjalną dyrektywę komputera

{ \$ I Nazwa zbioru }

Umożliwia ona włączenie do tekstu programu, w miejsce wskazane dyrektywą, przygotowanego wcześniej tekstu programu przechowywanego w zbiorze o nazwie podanej w dyrektywie. Należy pamiętać, że tekst całego programu (tzn. z włączonymi zbiorami) musi mieć strukturę akceptowaną przez kompilator.

Ekran monitora dzieli się na dwie odrębnie wysterowane i niezależne od siebie części:

— obszar główny, wykorzystywany do zapisu tekstów i kreślenia obrazów, zwany kartką, — pobocze.

Pobocze nie jest wykorzystywane do zapisu tekstów i kreślenia obrazów, można natomiast zmieniać jego kolor. Kolor pobocza jest tłem dla kartki. Do zmiany koloru pobocza wykorzystywana jest procedura

**border** (*i*, *j*)

której parametrami są identyfikatory kolorów. Programista dysponuje dwudziestomasiedmioma kolorami, oznaczonymi numerami od 0 do 26. Kolory przyporządkowane poszczególnym numerom zamieszczono w tabeli 1.

TABELA 1

numer kałamarza	kolor	numer kałamarza	kolor	numer kałamarza	kolor
0	czarny	9	zielony	18	jasnoziel.
1	niebieski	10	cyjanowy	19	morski ziel.
2	jasnonieb.	11	kolor nieba	20	jasnocyjan.
3	czerwony	12	żółty	21	zieleń wapna
4	karmazynowy	13	biały	22	zieleń pastel
5	lila-róż	14	nieb. pastel	23	cyjan pastel
6	jasnoczerw.	15	pomarańcz	24	jasnożółty
7	purpurowy	16	różowy	25	żółty pastel
8	jasnopurp.	17	purp. pastel	26	jasnobiały

W procedurze border podaje się numery dwóch kolorów zmienianych naprzemiennie kilka razy w ciągu sekundy. Jeśli oba numery są jednakowe, to efekt migotania nie jest dostrzegalny. Program umożliwiający zmianę koloru pobocza (bez migotania) zamieszczono w przykładzie 22.

**Przykład 22**

Program ten zmienia kolor pobocza na odpowiadający kolejnemu numerowi, po każdorazowym naciśnięciu dowolnego klawisza. Do sprawdzenia, czy wciśnięto klawisz, wykorzystano funkcję standardową **keypressed** przyjmującą wartość **true** po wciśnięciu klawisza i **false** po odczytaniu kodu wciśniętego klawisza (**READ (kbd...)**). Po wyczerpaniu wszystkich możliwych kolorów, kolor pobocza jest zmieniany na niebieski (przyjmowany standardowo przez system). Efekt migotania można uzyskać realizując program z przykładu 23, w którym kolejne pary kolorów pobocza są ustalone przez użytkownika.

```
PROGRAM prog21;
VAR
  i:integer;
  ch:char;
  {$i grafik3.inc}
BEGIN
  FOR i:=0 TO 26 DO
  BEGIN
    border(i,i);
    REPEAT UNTIL keypressed;
    READ(kbd,ch);
  END;
  border(1,1);
END.
```

Bardziej skomplikowane są zasady wykorzystywania kartki. Na kartce można zapisać teksty piórem tekstowym oraz kreślić obrazy piórem graficznym. Przed użyciem pióra należy wskazać kałamarz, z którego czer-

```
PROGRAM prog22;
VAR
  k1,k2:integer;
  ch:char;
  {$i grafik3.inc}
BEGIN
  REPEAT
  WRITELN('zmiana koloru ranki (T/N)?');
  READ(kbd,ch);
  IF (ch='T') OR (ch='t') THEN
  BEGIN
    WRITELN('podaj nr pierwszego koloru');
    READLN(k1);
    WRITELN('podaj nr drugiego koloru');
    READLN(k2);
    border(k1,k2);
  END;
  UNTIL (ch='N') OR (ch='n');
  border(1,1);
END.
```

piany będzie tusz. W każdym kałamarzu można umieścić jednocześnie — w dwóch odrębnych jego częściach — dwa tusze (różnokolorowe lub tego samego koloru). Jeżeli tusz będzie czerpany z kałamarza, w którym umieszczono różnokolorowe tusze, uzyskamy efekt migotania, jak w przypadku pobocza. Liczba dostępnych kałamarzy uzależniona jest od trybu wykorzystania ekranu:

- 16, w trybie 0 (numerowanych od 0 do 15),
- 4, w trybie 1 (numerowanych od 0 do 3),
- 2, w trybie 2 (numerowanych od 0 do 1).

W miarę wzrostu numeru trybu wykorzystania ekranu poprawia się jakość kreślonego obrazu (rozdzielczość) kosztem zmniejszającej się liczby równocześnie możliwych do wykorzystania kałamarzy, a tym samym — liczby kolorów. Do napełniania kałamarzy wykorzystuje się procedurę

**ink** (*nrk*, *i*, *j*)

gdzie: *nrk* — numer kałamarza (który powinien uwzględnić tryb wykorzystania ekranu), *i*, *j* — numery kolorów tuszu (wg tabeli 1).

W przykładzie 24 zamieszczono procedurę napełniania kałamarzy wg żądań użytkownika. Procedura ta wypełnia tylko tyle kałamarzy, ile wynika z trybu wykorzystania ekranu.

**Przykład 24**

Jak już podano, do zapisu tekstów oraz kreślenia obrazów wykorzystywane są dwa różne pióra: tekstowe i graficzne. Z każdym z nich związane są oddzielne procedury. Ich omawianie rozpoczniemy od procedur związanych z piórem tekstowym.

Użycie pióra tekstowego powinno być poprzedzone wyborem kałamarza, którego tusze będą wykorzystywane do zapisu tekstu. Służy do tego procedura.

**pen** (*nrk*)

gdzie: *nrk* — numer kałamarza. Wszystkie litery zapisywane są na tle, którego

```

PROCEDURE napelnij(tryb:integer);
VAR
  il,k1,k2,
  i:integer;
  ch:char;
BEGIN
  IF tryb=0 THEN il:=16
  ELSE
    IF tryb=1 THEN il:=4
    ELSE il:=2;
  i:=0;
  REPEAT
    WRITELN('zmiana tuszu w ');
    WRITELN('kalamarzu nr',i);
    WRITELN('(T/N)?');
    READ(kbd,ch);
    IF (ch='T') OR (ch='t') THEN
      BEGIN
        WRITELN('nr pierwszego koloru');
        READLN(k1);
        WRITELN('nr drugiego koloru');
        READLN(k2);
        ink(i,k1,k2);
      END;
      i:=i+1;
    UNTIL i=il;
  END;

```

kolor wybierany jest przez wykonanie procedury.

paper (nrk)

gdzie: nrk — numer kalamarza.

Jeżeli procedury te nie zostaną użyte przed wykorzystaniem pióra, to dla tła przydzielany jest standardowo kalamarz 0, a dla pióra — kalamarz 1.

Dla ilustracji działania procedur **paper** i **pen** w przykładzie 25 po ustaleniu trybu wykorzystania ekranu i napełnieniu kalamarza dla tła, a w przykładzie 26 — po czynnościach wstępnych — następuje wybieranie kolejnych kalamarzy dla pióra (przyjęto, że procedura uzupełniania kalamarza znajduje się w zbiorze PROCNAP. PAS).

#### Przykład 25

```

PROGRAM prog23;
VAR
  tryb,ilosc,nrk:integer;
  ch:char;
  ($i grafik3.inc)
  ($i procnap.pas)
BEGIN
  WRITELN('podaj tryb');
  READLN(tryb);
  napelnij(tryb);
  IF tryb=0 THEN ilosc:=16
  ELSE
    IF tryb=1 THEN ilosc:=4
    ELSE ilosc:=2;
  mode(tryb);
  FOR nrk:=0 TO ilosc-1 DO
    BEGIN
      paper(nrk);
      WRITELN('01234567890123456789');
      REPEAT UNTIL keypressed;
      READ(kbd,ch);
    END;
    mode(2);
    ink(0,1,1);
    paper(0);
    ink(1,26,26);
    pen(1);
  END.

```

```

PROGRAM prog24;
VAR
  tryb,ilosc,nrk:integer;
  ch:char;
  ($i grafik3.inc)
  ($i procnap.pas)
BEGIN
  WRITELN('podaj tryb');
  READLN(tryb);
  napelnij(tryb);
  IF tryb=0 THEN ilosc:=16
  ELSE
    IF tryb=1 THEN ilosc:=4
    ELSE ilosc:=2;
  mode(tryb);
  FOR nrk:=0 TO ilosc-1 DO
    BEGIN
      pen(nrk);
      WRITELN('01234567890123456789');
      REPEAT UNTIL keypressed;
      READ(kbd,ch);
    END;
    mode(2);
    ink(0,1,1);
    paper(0);
    ink(1,26,26);
    pen(1);
  END.

```

#### Przykład 26

W obu podanych przykładach do ustalenia żadanego trybu wykorzystania ekranu zastosowano procedurę.

modele (nrtryb)

gdzie:

nrtryb — przyjmuje wartość 0,1,2.

Należy tutaj wyjaśnić, że z trybem wykorzystania ekranu wiąże się również liczba znaków tekstu, które mogą być wyświetlone w jednym wierszu.

Wynosi ona:

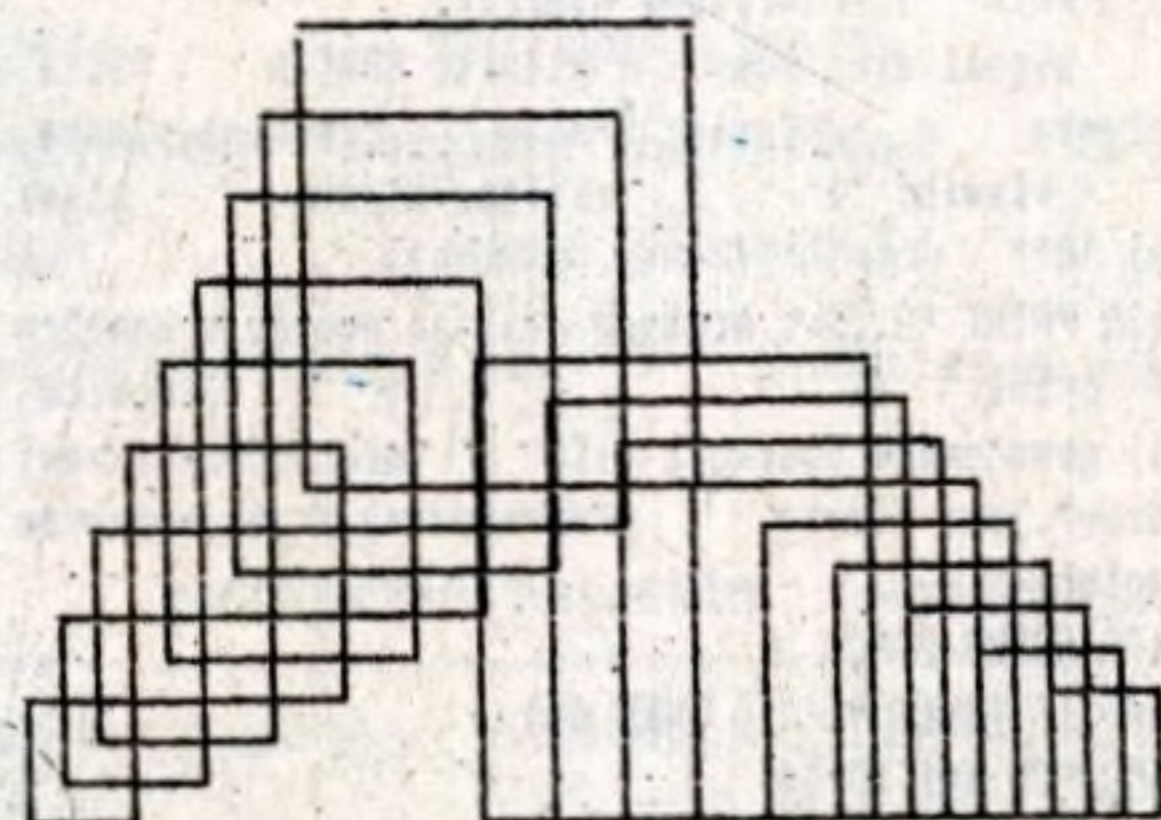
- 20, dla trybu 0,
- 40, dla trybu 1,
- 80, dla trybu 2.

Przed zakończeniem każdego z programów przywracany jest standardowy tryb wykorzystania ekranu (2), standardowe numery kalamarzy tła (0) i pióra (1) oraz standardowe numery tuszów w tych kalamarzach.

```

10 FOR x=0 TO 70 STEP 8
30 PLOT x,x
40 DRAW 24+x,0
50 DRAW 0,24+x
60 DRAW -(24+x),0
70 DRAW 0,-(24+x)
80 NEXT x
110 FOR x=0 TO 70 STEP 8
120 PLOT 255-x,0
130 DRAW 0,24+x
140 DRAW -(24+x),0
150 DRAW 0,-(24+x)
160 DRAW 24+x,0
170 NEXT x

```



Program z przykładu 27 umożliwia prześledzenie wszystkich przedstawionych właściwości pióra tekstowego

#### Przykład 27

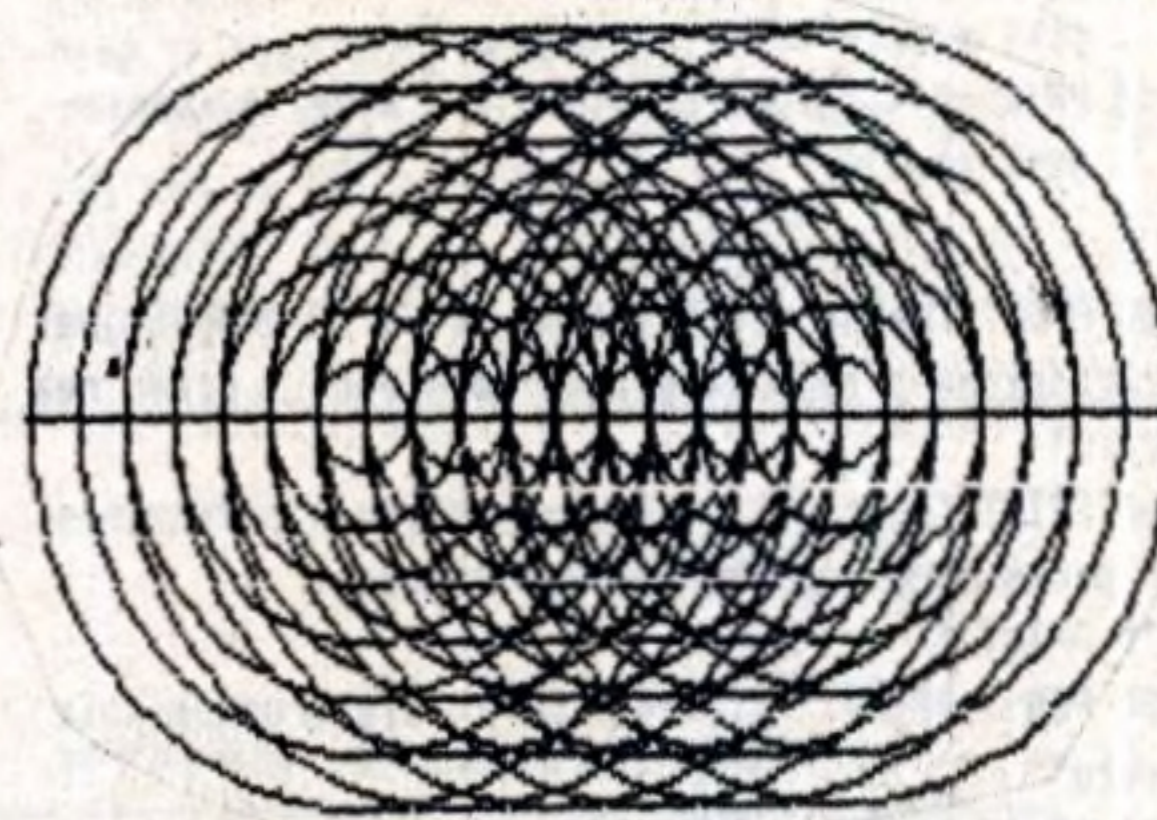
```

PROGRAM prog25;
VAR
  tryb,nrk:integer;
  ch:char;
  ($i grafik3.inc)
  ($i procnap.pas)
BEGIN
  WRITELN('podaj tryb');
  READLN(tryb);
  mode(tryb);
  WRITELN;
  REPEAT
    WRITELN('zmiana tuszu w ');
    WRITELN('kalamarzach (T/N)?');
    READ(kbd,ch);
    IF (ch='T') OR (ch='t') THEN
      napelnij(tryb);
    WRITELN('zmiana tła (T/N)?');
    READ(kbd,ch);
    IF (ch='T') OR (ch='t') THEN
      BEGIN
        WRITELN('podaj nr kalamarza');
        READLN(nrk);
        paper(nrk);
      END;
    WRITELN('zmiana pióra (T/N)?');
    READ(kbd,ch);
    IF (ch='T') OR (ch='t') THEN
      BEGIN
        WRITELN('podaj nr kalamarza');
        READLN(nrk);
        pen(nrk);
      END;
    UNTIL ch=' ';
  mode(2);
  ink(0,1,1);
  paper(0);
  ink(1,26,26);
  pen(1);
  END.

```

W kolejnym odcinku zajmiemy się piórem graficznym.

Stefan ROZMUS



```

5 PAPER 4: INK 0
10 PLOT 0,85
20 DRAW 240,0
30 LET x=70
35 LET y=85
40 FOR j=1 TO 5
50 FOR i=10 TO 70 STEP 10
60 CIRCLE x,y,i
70 NEXT i
80 NEXT j
90 LET x=x+20
100 NEXT j

```

ZW

# Komputerowy test dla kierowców

Ten program może się przydać wszystkim, którzy swój refleks i manualną sprawność muszą utrzymywać w stałej kondycji. Dokładny opis działania znajduje się w programie.

Program powstał w Wyższej Szkole Oficerskiej Wojsk Pancernych w Poznaniu.

```
10 INPUT "Czy wynik testu na drukarce ? ( t - tak/
n -nie )";druk$:IF druk$(">" AND druk$("<")" AND
druk$(">" AND druk$("<")" THEN 10
20 ilbl(1)=2:ilbl(2)=2:ilbl(3)=4:ilbl(4)=5:ocepr$=
TEST NIE ZALICZONY":ocezw$=ocepr$
30 kt=0:IF druk$="t" OR druk$="T" THEN PRINT #8,CH
R$(27) CHR$(64)
40 DIM tt(10),c$(10),t(20),blad(10)
50 GOSUB 2630:CLS:MODE 1:BORDER 4:INK 0,26:INK 1,
6:INK 2,2:INK 3,0:PAPER 0:PEN 1
60 NA=10:ZLY=6:CZAS=11:ZA=13:GOTO 90
70 LOAD "scr"
80 IF INKEY$="" THEN 80
90 czesc=1:czesctestu=1
100 '
110 ' ekran 1
120 '
130 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0
140 FOR e=1 TO 4:blad(e)=0:NEXT e:czesc=1:czesctes
tu=1:IF druk$="T" OR druk$="t" THEN INPUT "Podaj i
mię i nazwisko";imie$
150 ON czesctestu GOTO 160,400,480,580,730
```

```
160 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0:PRINT "Program jest prze
znaczony do sprawdzaniarefleksu kierowc5w wojskow
ych pojazdz5w mechanicznych.Czas reakcji jest zlicz
a="
```

```
170 PRINT "ny przy 2 rodzajach bod;cz5w:"
180 PRINT " - dla oczu (zmysl wzroku)"
190 PRINT " - dla uszu (zmysl sluchu)"
200 PRINT "Program sklada się z 4 cz[7ci,sekwency-
jnie sprawdzaj;cych czas reakcji testo-
wanego ki
erowcy:"
210 PRINT " 1.Test oczu - reakcja sygnalizowana do
- wolni: r[ki."
220 PRINT " 2.Test uszu - reakcja sygnalizowana do
- wolni: r[ki."
230 PRINT " 3.Test oczu - reakcja lewi lub prawi:
r[ki w zale9no7ci od koloru."
240 PRINT " 4.Losowa kombinacja 3 wy9ej wymienio-
nych test5w."
```

```
250 PRINT "Ka9da cz[7* sklada się z 10 pr5b i jest
zakoczona przedstawieniem 7redniego czasu rea
kcji."
```

```
260 PRINT "Po cz[7ci 4 testu nast[puje podsumowani
ei ocena badanego kierowcy wojskowego."
```

```
270 PRINT " "
```

```
280 PEN 1:PRINT " klawisz COPY dla uruchomienia
testu"
```

```
290 IF INKEY(9)<>0 THEN 290
```

```
300 PEN 1
```

```
310 CLS
```

```
320 PRINT " ":PRINT " CZ08] 1 - TEST OCZU"
:PRINT " ":PEN 0
```

```
330 PRINT " Istot! tej cz[7ci testu jest zbadanie
czasu reakcji na bod;ce odbierane przez zm
ysl wzroku.Po odebraniu bod;ca prawdziwa reakcja
polega na naci7ni[ - ciu klawisza SPACJA.Bodziec
polega na"
```

```
340 PRINT " wy7wietleniu 95itego kola na
niebieskim polu roboczym,w dowolnym miejscu
ekranu.W dolnej cz[7ci ekranu ukazuje się czas r
```

```
ee[7ci na bod;ec. Zbyt wczesna reakcja (przed
ukazaniem się kola) powoduje powt5rni: pr5b[."
```

```
350 GOSUB 2590
```

```
360 IF INKEY(9)<>0 THEN 360
```

```
370 FOR i=1 TO 10:GOSUB 1010:t(i)=tb:NEXT i
```

```
380 GOSUB 1310
```

```
390 czesctestu=2:czesc=2:GOTO 150
```

```
400 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0
```

```
410 PRINT " ": PEN 1:PRINT " CZ08] 2 - TE
ST USZU":PRINT " ":PEN 0
```

```
420 PRINT " Istot! tej cz[7ci testu jest zbadanie
czasu reakcji na bod;ce odbierane przez zm
ysl sluchu.Po odebraniu bod;ca prawdziwa reakcja
polega na naci7ni[ - ciu klawisza SPACJA.Bodziec
polega na"
```

```
430 PRINT " wygenerowaniu przez mikrokomputer sy-
gnalu d;wi[kowego.W dolnej cz[7ci ekranu u
kazuje się czas reakcji na bod; - dziec.Zbyt wczesna
reakcja (przed uslyszaniem sygnalu d;wi[kow
ego) powo-
duje powt5rni: pr5b[."
```

```
440 GOSUB 2590
```

```
450 IF INKEY(9)<>0 THEN 450
```

```
460 FOR i=1 TO 10:GOSUB 1420:t(i)=tb:NEXT i:GOSUB
1310
```

```
470 czesctestu=3:czesc=3:GOTO 150
```

```
480 GOTO 490
```

```
490 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0
```

```
500 CLS:PEN 1:PRINT " CZ08] 3 - TEST OCZU"
:PRINT " ":PEN 0
```

```
510 PRINT " Istot! tej cz[7ci testu jest zbadanie
czasu reakcji na bod;ce odbierane przez zmysl wzr
oku.Bodziec polega na wy7wie-
tleniu czerwonego l
ub zielonego kola na niebieskim polu roboczym w
dowo-
lnym miejscu ekranu.Prawdziwa reakcja"
```

```
520 PRINT "polega na:
- naci7ni[ciw lewi: r[ki: klawisza 1 po poka
zaniu się koloru czerwonego - naci7ni[ciw prawi:
r[ki: klawisza 0 po pokazaniu się koloru zie
lonego"
```

```
530 PRINT "W dolnej cz[7ci ekranu ukazuje się
czas reakcji na bod;ec.Zbyt wczesna reakcja (
przed ukazaniem się kola) lub niewla7ciwa (naci7n
i[cie nieodpowiednie-go do koloru klawisza) powodu
je"
```

```
540 PRINT "powt5rni: pr5b[."
```

```
550 GOSUB 2590
```

```
560 IF INKEY(9)<>0 THEN 560
```

```
570 FOR i=1 TO 10:GOSUB 1010:t(i)=tb:NEXT i:GOSUB
1310:czesctestu=4:czesc=4
```

```
580 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0
```

```
590 CLS:PEN 1:PRINT " ":PRINT " CZ0
8] 4
KOMBINACJA TEST6W 1,2
,3":PEN 0:PRINT " "
```

```
600 PRINT " Istot! tej cz[7ci jest zbadanie czasu
reakcji na dowolny z przedtem przedsta-
wionych b
od;cz5w.Obowiazuji dotychczasowezasady:"
```

```
610 PRINT " 1.Prawdziwe reakcje:
sygnal d;wi[kowy - klawisz SPACJA 95ita
plamka - klawisz SPACJA czerwona plamka
- klawisz '1' zielona plamka - klawi
sz '0'"
```

```
620 PRINT " 2.Zbyt wczesna reakcja powoduje powt5rn
i: pr5b[. 3.Bod;ce
si generowane losowo,a mi[fc nie mo9esz przewi
dzie~,jaki b[dzie prezentowany przez mikrokom
puter."
```

```
630 GOSUB 2590
```

```
640 IF INKEY(9)<>0 THEN 640
```

```
650 FOR I=1 TO 10
```

```
660 xx=INT(RND(1)*35)+1:yy=INT(RND(1)*15)+1
```

```
670 xxx=xx MOD 2:yyy=yy MOD 2
```

```
680 IF xxx=0 AND yyy=0 THEN czesctestu=1:GOSUB 101
0:t(i)=tb
```

```
690 IF xxx=0 AND yyy=1 THEN czesctestu=2:GOSUB 142
0:t(i)=tb
```

```
700 IF xxx=1 THEN czesctestu=3:GOSUB 1010:t(i)=tb
```

```
710 NEXT i
```

```
720 GOSUB 1310
```

```
730 CLS:MODE 1:BORDER 4:INK 0,26:INK 1,6:INK 2,9:I
NK 3,11:PAPER 2:CLS:PEN 0
```

```
740 CLS:PEN 1:PRINT " ":PRINT " ":PRINT "
PODSUMOWANIE TESTU":PEN 0:PRINT " ":PRINT " W pos
zczeg5lnych cz[7ciach testu uzyskale7 nast[
puj;ce 7rednie czasy reakcji:"
```

```
750 pr=(tt(1)+tt(2))/2:zw=(tt(3)+tt(4))/2
```

```
760 pr=(ROUND(pr*100))/100:zw=(ROUND(zw*100))/100
```

```
770 IF blad(1)=1 THEN 790
```

```
780 PRINT " CZ08] 1 - TEST OCZU - ";USING "##.##
":tt(1):GOTO 800
```

```
790 PRINT " CZ08] 1 - TEST OCZU - NIE ZALICZONA"
```

```
800 IF blad(2)=1 THEN 820
```

```
810 PRINT " CZ08] 2 - TEST USZU - ";USING "##.##
":tt(2):GOTO 830
```

```
820 PRINT " CZ08] 2 - TEST USZU - NIE ZALICZONA"
```

```
830 IF blad(3)=1 THEN 850
```

```
840 PRINT " CZ08] 3 - TEST OCZU - ";USING "##.##
":tt(3):GOTO 860
```

```
850 PRINT " CZ08] 3 - TEST OCZU - NIE ZALICZONA"
```

```
860 IF blad(4)=1 THEN 880
```

```
870 PRINT " CZ08] 4 - KOMBINACJA - ";USING "##.##
":tt(4):GOTO 890
```

```
880 PRINT " CZ08] 4 - KOMBINACJA - NIE ZALICZONA"
```

```
890 IF blad(1)=1 OR blad(2)=1 THEN 910 ELSE 930
```

```
900 IF blad(3)=1 OR blad(4)=1 THEN 920 ELSE 940
```

```
910 PRINT " ":PRINT " BADANIE REAKCJI PROSTEJ - NI
E ZALICZONE":GOTO 900
```

```
920 PRINT " BADANIE REAKCJI Z WYB0REM - NIE ZALI-
CZONE":GOTO 950
```

```
930 PRINT " ":PRINT " 8R. CZAS REAKCJI PROSTEJ:
";USING "##.##";pr
```

```
940 PRINT " 8R. CZAS REAKCJI Z WYB0REM: ";USING "##
.##";zw
```

```
950 IF druk$="T" OR druk$="t" THEN GOSUB 2060
```

```
960 LOCATE 1,20:PRINT " Je9eli chcesz uzyska~ ocen
[ testu naci7nij klawisz COPY,je7li nie - ENTER"
```

```
970 IF INKEY(9)<>0 AND INKEY(6)<>0 THEN 960
```

```
980 IF INKEY(9)=0 THEN 1660
```

```
980 LOCATE 1,20:PRINT " Je9eli chcesz powt5rzy~ te
st - naci7nij klawisz COPY,je7li nie - ENTER
":IF INKEY(5)<>0 AND INKEY(9)<>0 THEN 980
```

```
990 IF INKEY(9)=0 THEN 90
```

```
1000 END
```

```
1010 CLS #0:CLS #1:BORDER 9:INK 0,11:INK 1,24:INK
2,6:INK 3,9:PAPER #0,2:PEN #0,1:PAPER #1,0:PEN #1,
1
```

```
1020 WINDOW #0,1,40,20,25:WINDOW #1,1,40,1,19
```

```
1030 ' plamka
```

```
1040 a$=CHR$(143)+CHR$(144)+CHR$(10)+CHR$(8)+CHR$(
9)+CHR$(145)+CHR$(146)
```

```
1050 CLS
```

```
1060 'randomize - czas od 1 do 11 sek.
```

```
1070 q=INT(RND(1)*10)+1:g1=TIME
```

```
1080 g2=TIME:j$=INKEY$
```

```
1090 IF INKEY(47)=0 OR INKEY(32)=0 OR INKEY(64)=0
THEN LOCATE #0,ZA,2:PRINT #0,"Za wczętnie !!!":FO
R ii=1 TO 1000:NEXT:CLS #0:blady=blady+1:CLEAR INP
UT:IF blady > ilbl(czesc) THEN 1610 ELSE 1070
```

```
1100 IF (g2-g1)/300<g THEN 1080
```

```
1110 x=INT(RND(1)*35)+1:y=INT(RND(1)*15)+1
```

```
1120 IF czesctestu<3 THEN 1170
```

```

1130 ster1=x MOD 2
1140 IF ster1 = 0 THEN 1150 ELSE 1160
1150 PEN #1,3:GOTO 1170
1160 PEN #1,2
1170 LOCATE #1,x,y: CLEAR INPUT:PRINT#1,A#;t:TIME

1180 IF czesc<3 THEN 1230
1190 IF INKEY="" THEN 1190
1200 IF INKEY(32)=0 AND ster1=0 THEN 1250
1210 IF INKEY(64)=0 AND ster1=1 THEN 1250
1220 LOCATE #0,NA,2:PRINT #0,"Nacisnij 7 zly klawi
sz !!!":FOR ii=1 TO 1000:NEXT:CLS #0:CLS #1:bled
=bledy+1: CLEAR INPUT:IF bledy > 10 THEN 1450
610 ELSE 1070
1230 IF INKEY="" THEN 1230
1240 IF INKEY(47) <> 0 THEN 1220
1250 tb=(TIME-t)/300
1260 LOCATE #0,CIAS,2:PRINT "CIAS REAKCJI:" USING
"##.##";tb
1270 LOCATE #0,NA,6:PRINT #0,"Nacisnij klawisz COP
Y"
1280 IF INKEY(9)<>0 THEN 1280
1290 CLS#1
1300 CLEAR INPUT:RETURN

1310 CLS:REM OBLICZANIE SREDNIEGO CZASU REAKCJI
I
1320 tijd=0
1330 FOR x=1 TO 10:tijd=t(x)+t: NEXT x
1340 tt(czesc)=t:ON CZESC GOTO 1350,1360,137
0,1380
1350 LOCATE #0,1,2:PRINT " SREDNI CZAS REAKCJI TWO
ICH OCZU:"GOTO 1390
1360 LOCATE #0,1,2:PRINT " SREDNI CZAS REAKCJI TWO
ICH USZU:"GOTO 1390
1370 LOCATE #0,1,2:PRINT " SREDNI CZAS REAKCJI TWO
ICH OCZU:"GOTO 1390
1380 LOCATE #0,1,2:PRINT " SREDNI CZAS REAKCJI OCZ
U I USZU:"
1390 LOCATE #0,35,2:PRINT USING "##.##";tt(czesc):
LOCATE #0,NA,6:PRINT #0,"Nacisnij klawisz COPY"
1400 IF INKEY(9)<>0 THEN 1400
1410 RETURN
1420 CLS #0:CLS #1: BORDER 9: INK 0,11: INK 1,24: INK
2,6: INK 3,9: PAPER #0,2: PEN #0,1: PAPER #1,0: PEN #1,
1
1430 WINDOW #0,1,40,20,25: WINDOW #1,1,40,1,19
1440 CLS
1450 g=INT(RND(1)*10)+1:g1=TIME
1460 g2=TIME:IF INKEY(47)=0 THEN LOCATE #0,ZA,2:PR
INT "Za wznie!!!":FOR ii=1 TO 1000:NEXT:CLS #0
:bledy=bledy+1: CLEAR INPUT:IF bledy > 10 THEN 1450
THEN 1410 ELSE 1450
1470 IF((g2-g1)/300)<g THEN 1460
1480 CLEAR INPUT:PRINT#1,CHR$(7):t:TIME
1490 i=INKEY:IF i="" THEN 1490
1500 IF INKEY(47)=0 THEN 1530
1510 LOCATE #0,NA,2:PRINT #0,"Nacisnij 7 zly klawi
sz !!!":FOR ii=1 TO 1000:NEXT:CLS #0:CLS #1:bledy
=bledy+1: CLEAR INPUT:IF bledy > 10 THEN 1450
610 ELSE 1450
1520 CLS #1:GOTO 1450
1530 tb=(TIME-t)/300
1540 LOCATE #0,CIAS,2:PRINT "CIAS REAKCJI:" USING
"##.##";tb
1550 LOCATE #0,NA,6:PRINT #0,"Nacisnij klawisz COP
Y"
1560 IF INKEY(9)<>0 THEN 1560
1570 CLS
1580 CLEAR INPUT:RETURN
1590 PRINT "JESZCZE RAZ ? ...TO NACISKAJ KLUCZ"
1600 IF INKEY="" THEN 1600 ELSE RUN
1610 * OBSZUGA BLEDOW BADANEGO
1620 blad(czesc)=1:tt(czesc)=0:czesc=czesc+1
1630 CLS:MODE 1: BORDER 4: INK 0,26: INK 1,6: INK 2,9:
INK 3,11: PAPER 2: CLS: PEN 0
1640 LOCATE 2,11:PRINT " POPEZNIENZ ZA DU:O B20
D6W !!
PR68A NIEZALICZONA":FOR I=1 TO 1700:
NEXT
1650 bledy=0:IF czesc > 4 THEN 730 ELSE 150
1660 REM OCENA WYNIKOW TESTU
1670 IF blad(1)=1 OR blad(2)=1 THEN ocpr=0:GOTO 17
30
1680 IF pr >= 0.27 THEN ocenapr=2
1690 IF pr < 0.27 THEN ocenapr=3
1700 IF pr > 0.32 THEN ocenapr=1
1710 IF pr < 0.23 THEN ocenapr=4
1720 ocpr=1
1730 IF blad(3)=1 OR blad(4)=1 THEN ocpr=0:GOTO 17
30
1740 IF zw >= 0.52 THEN ocenazw=2
1750 IF zw < 0.52 THEN ocenazw=3
1760 IF zw > 0.59 THEN ocenazw=1
1770 IF zw < 0.47 THEN ocenazw=4
1780 ocpr=1
1790 CLS:MODE 1: BORDER 4: INK 0,26: INK 1,6: INK 2,9:
INK 3,11: PAPER 2: CLS: PEN 1
1800 PRINT " ":PRINT " ### OCENA REAKCJI PROST
EJ ###":PRINT " ":PEN 0:IF ocpr=0 THEN PRINT ocep
r:GOTO 1930
1810 PEN 0:ON ocenapr GOSUB 1830,1860,1880,1910
1820 GOTO 1930
1830 PRINT " TWOJE WYNIKI S) BARDZO SZABE.
CIAS REAKCJI PROSTEJ DYSKWALIFIKUJE CIO JAK
O KIEROWCO WOJSKOWEGO.ZDECYDOWANIE NIE MUSISZ JWCZ
YJ CZAS REAKCJI PROSTEJ NA BOD<CE WZROKOWE
I SZUCHOWE."
1840 PRINT " UZYSKAZEB OCENO : 2"
1850 RETURN
1860 PRINT " TWOJE WYNIKI S) SZABE.CZAS REAKCJI
PROSTEJ KWALIFIKUJE CIO JAKO KIEROWCO WOJSKOW
EGO,ALE MUSISZ ZDECYDOWANIE JWCZYJ CZAS REAK
CJI PROSTEJ NA BOD<CE WZROKOWE I SZUCHOWE.
UZYSKAZEB OCENO : 3"
1870 RETURN
1880 PRINT " TWOJE WYNIKI S) DOBRE.
CIAS REAKCJI PROSTEJ KWALIFIKUJE CIO JAKO KI
EROWCO WOJSKOWEGO,NIE ZASZKODZI JEDNAK JAK BODZIE
SZ JWCZYJ CZAS REA- KCJI PROSTEJ NA BOD<CE WZRO
KOWE I SZUCHOWE."
1890 PRINT " UZYSKAZEB OCENO : 4"
1900 RETURN
1910 PRINT " TWOJE WYNIKI S) BARDZO DOBRE.
CIAS REAKCJI PROSTEJ KWALIFIKUJE CIO JAKO KI
EROWCO WOJSKOWEGO.NIE MUSISZ JWCZYJ CZASU REA
KCJI PROSTEJ NA BOD<CE WZROKOWE I SZUCHOWE.
UZYSKAZEB OCENO : 5"
1920 RETURN
1930 PEN 1:PRINT " ":PRINT " ### OCENA REAKCJI
Z WYBOREM ###":PRINT " ":PEN 0:IF ocpr=0 THEN PRI
NT ocep:GOTO 1950
1940 ON ocenazw GOSUB 1960,1990,2010,2040
1950 IF druk="T" OR druk="t" THEN 2310 ELSE 980
1960 PRINT " TWOJE WYNIKI S) BARDZO SZABE.
CIAS REAKCJI Z WYBOREM DYSKWALIFIKUJE CIO JAK
O KIEROWCO WOJSKOWEGO.ZDECYDOWANIE NIE MUSISZ JWCZ
YJ CZAS REAKCJI Z WYBOREM NA BOD<CE WZROKOWE
I SZUCHOWE"
1970 PRINT " UZYSKAZEB OCENO : 2"
1980 RETURN
1990 PRINT " TWOJE WYNIKI S) SZABE.CZAS REAKCJI
Z WYBOREM KWALIFIKUJE CIO JAKO KIEROWCO WOJSKOW
EGO,ALE MUSISZ ZDECYDOWANIE JWCZYJ CZAS REAK
CJI Z WYBOREM NA BOD<CE WZROKOWE I SZUCHOWE
UZYSKAZEB OCENO : 3"
2000 RETURN
2010 PRINT " TWOJE WYNIKI S) DOBRE.
CIAS REAKCJI Z WYBOREM KWALIFIKUJE CIO JAKO KI
EROWCO WOJSKOWEGO.NIE ZASZKODZI JEDNAK JAK BODZIE
SZ JWCZYJ CZAS REA- KCJI Z WYBOREM NA BOD<CE WZ
ROKOWE I SZUCHOWE."
2020 PRINT " UZYSKAZEB OCENO : 4"
2030 RETURN
2040 PRINT " TWOJE WYNIKI S) BARDZO DOBRE.
CIAS REAKCJI Z WYBOREM KWALIFIKUJE CIO JAKO KI
EROWCO WOJSKOWEGO.NIE MUSISZ JWCZYJ CZASU REA
KCJI Z WYBOREM NA BOD<CE WZROKOWE I SZUCHOWE.
UZYSKAZEB OCENO : 5"
2050 RETURN
2060 * wyjscie na drukarke
2070 kt=kt+1:PRINT #8," ":PRINT #8," ":PRINT #8,CH
R$(27) CHR$(87) CHR$(1) CHR$(27) CHR$(71):IF kt=1
THEN PRINT #8," WYNIKI TESTOWANIA REAKCJI"
2080 IF kt=1 THEN PRINT #8," PROSTEJ I Z
WYBOREM":PRINT #8,CHR$(27) CHR$(72)
2090 PRINT #8,kt;".;imie$
2100 PRINT #8,CHR$(27) CHR$(87) CHR$(0)
2110 IF blad(1)=1 THEN 2130
2120 PRINT #8," CZESC 1 - TEST OCZU - ";US
ING "##.##";tt(1):GOTO 2140
2130 PRINT #8," CZESC 1 - TEST OCZU - NIE
ZALICZONA"
2140 IF blad(2)=1 THEN 2160
2150 PRINT #8," CZESC 2 - TEST USZU - ";US
ING "##.##";tt(2):GOTO 2170
2160 PRINT #8," CZESC 2 - TEST USZU - NIE
ZALICZONA"

```

7 6 5 4 3 2 1 0

Uwaga Atari!

AUDC 1-4

niskie częstotliwości      średnie częstotliwości      wysokie częstotliwości

0 0 0	licznik Geigera	szalejący ogień	prąd powietrza	wrzenie
0 X 1	karabin maszynowy	stojący samochód	silnik elektryczny	transformator mocy
0 1 0	spokojny ogień	jadący samochód		
1 0 0		zawalenie budynku	dudnienie radiowe	wodospad
1 X 1	czyste tony			
1 0 0	samolot	strzyżenie trawnika	elektryczna maszynka	do golenia

Odpowiednio programując rejestry niezależnego sterowania kanałami AUDC 1-4 otrzymać można takie oto efekty dźwiękowe.

```

2170 IF blad(3)=1 THEN 2190
2180 PRINT #8," CZESC 3 - TEST OCZU - ";US
ING "##.##";tt(3):GOTO 2200
2190 PRINT #8," CZESC 3 - TEST OCZU - NIE
ZALICZONA"
2200 IF blad(4)=1 THEN 2220
2210 PRINT #8," CZESC 4 - KOMBINACJA - ";US
ING "##.##";tt(4):GOTO 2230
2220 PRINT #8," CZESC 4 - KOMBINACJA - NIE
ZALICZONA"
2230 IF blad(1)=1 OR blad(2)=1 THEN 2250 ELSE 2280

2240 IF blad(3)=1 OR blad(4)=1 THEN 2270 ELSE 2290
2250 PRINT #8," "
2260 PRINT #8," BADANIE REAKCJI PROSTEJ -
NIE ZALICZONE":GOTO 2240
2270 PRINT #8," BADANIE REAKCJI Z WYBOREM -
NIE ZALICZONE":GOTO 2300
2280 PRINT #8," ":PRINT #8," SR. CZAS REAKCJ
I PROSTEJ: ";USING "##.##";(tt(1)+tt(2))/2:GOTO
2240
2290 PRINT #8," SR. CZAS REAKCJI Z WYBOREM:
";USING "##.##";(tt(3)+tt(4))/2
2300 RETURN
2310 PRINT #8," ":PRINT #8," ***** OC
ENA REAKCJI PROSTEJ *****:PRINT #8," ":IF ocpr=0
THEN PRINT #8,occzw#:GOTO 2450
2320 ON ocenapr 60SUB 2340,2370,2390,2420
2330 GOTO 2450
2340 PRINT #8," TWOJE WYNIKI SA BARDZO SLAB
E.CZAS REAKCJI PROSTEJ DYSKWALIFIKUJE":PRINT #8,"
CIE JAKO KIEROWCE WOJSKOWEGO.ZDECYDOWANIE
MUSISZ CWICZYC CZAS":PRINT #8," REAKCJI PRO
STEJ NA BODZCE WZROKOWE I SLUCHOWE."
2350 PRINT #8," UZYSKALES OCENE : 2"
2360 RETURN
2370 PRINT #8," TWOJE WYNIKI SA SLABE.CZAS
REAKCJI PROSTEJ KWALIFIKUJE CIE":PRINT #8,"
JAKO KIEROWCE WOJSKOWEGO,ALE MUSISZ ZDECYDOWANIE
CWICZYC CZAS"
2380 PRINT #8," REAKCJI PROSTEJ NA BODZCE W
ZROKOWE I SLUCHOWE":PRINT #8," UZYSKALES OC
ENE : 3":RETURN
2390 PRINT #8," TWOJE WYNIKI SA DOBRE.CZAS
REAKCJI PROSTEJ KWALIFIKUJE CIE JAKO":PRINT #8,"
KIEROWCE WOJSKOWEGO.NIE ZASZKODZI JEDNAK JA
K BEDZIESZ CWICZYL":PRINT #8," CZAS REAKCJI
PROSTEJ NA BODZCE WZROKOWE I SLUCHOWE."
2400 PRINT #8," UZYSKALES OCENE : 4"
2410 RETURN
2420 PRINT #8," TWOJE WYNIKI SA BARDZO DOBR
E.CZAS REAKCJI PROSTEJ KWALIFIKUJE CIE":PRINT #8,"
JAKO KIEROWCE WOJSKOWEGO.NIE MUSISZ CWICZY
C CZASU REAKCJI PROSTEJ":PRINT #8," NA BODZ
CE WZROKOWE I SLUCHOWE."
2430 PRINT #8," UZYSKAZES OCENE : 5"
2440 RETURN
2450 PRINT #8," ":PRINT #8," ***** OCE
NA REAKCJI Z WYBOREM *****:PRINT #8," ":IF oczw=0
THEN PRINT #8,occzw#:GOTO 980
2460 ON ocenazw 60SUB 2480,2510,2530,2560
2470 GOTO 980
2480 PRINT #8," TWOJE WYNIKI SA BARDZO SLAB
E.CZAS REAKCJI Z WYBOREM DYSKWALIFIKUJE":PRINT #8,"
CIE JAKO KIEROWCE WOJSKOWEGO.ZDECYDOWANIE
MUSISZ CWICZYC CZAS":PRINT #8," REAKCJI Z
WYBOREM NA BODZCE WZROKOWE I SLUCHOWE."
2490 PRINT #8," UZYSKALES OCENE : 2"
2500 RETURN
2510 PRINT #8," TWOJE WYNIKI SA SLABE.CZAS
REAKCJI Z WYBOREM KWALIFIKUJE CIE":PRINT #8,"
JAKO KIEROWCE WOJSKOWEGO,ALE MUSISZ ZDECYDOWANI
E CWICZYC CZAS":PRINT #8," REAKCJI Z WYBORE
M NA BODZCE WZROKOWE I SLUCHOWE."
2520 PRINT #8," UZYSKALES OCENE : 3":RETURN
2530 PRINT #8," TWOJE WYNIKI SA DOBRE.CZAS
REAKCJI Z WYBOREM KWALIFIKUJE CIE":PRINT #8,"
JAKO KIEROWCE WOJSKOWEGO.NIE ZASZKODZI JEDNAK J
AK BEDZIESZ":PRINT #8," CWICZYL CZAS REAKCJ

```

```

I Z WYBOREM NA BODZCE WZROKOWE I SLUCHOWE."
2540 PRINT #8," UZYSKALES OCENE : 4"

2550 RETURN
2560 PRINT #8," TWOJE WYNIKI SA BARDZO DOBR
E.CZAS REAKCJI Z WYBOREM KWALIFIKUJE":PRINT #8,"
CIE JAKO KIEROWCE WOJSKOWEGO.NIE MUSISZ CWIC
ZYC CZASU REAKCJI":PRINT #8," Z WYBOREM NA
BODZCE WZROKOWE I SLUCHOWE. "
2570 PRINT #8," UZYSKALES OCENE : 5"
2580 RETURN
2590 PRINT " Ta cz[7~ testu sklada si[ z 10 pr5b,
na jej zako[czenie wy7wietli si[ 7re- dni cza
s reakcji.
Uwaga:
Gdy jeste7 got5w naci7nij k
lawisz COPY"
2600 RETURN
2610 IF INKEYS="" THEN 2610
2620 GOTO 130
2630 SYMBOL AFTER 142
2640 SYMBOL 171,0,0,120,12,124,204,118,3
2650 SYMBOL 172,24,60,102,102,126,102,102,3
2660 SYMBOL 173,24,0,60,102,96,102,60,0
2670 SYMBOL 174,24,60,102,192,192,102,60,0
2680 SYMBOL 175,0,0,60,102,126,96,60,6
2690 SYMBOL 176,254,98,104,120,104,98,254,3

```

```

2700 SYMBOL 177,56,24,28,24,56,24,60,0
2710 SYMBOL 178,240,96,112,96,226,102,254,0
2720 SYMBOL 179,24,0,220,102,102,102,0
2730 SYMBOL 180,24,198,230,246,222,206,198,0
2740 SYMBOL 181,24,0,60,102,102,102,60,0
2750 SYMBOL 182,12,56,108,198,198,108,56,0
2760 SYMBOL 183,24,0,60,96,60,6,124,0
2770 SYMBOL 184,24,60,96,60,6,102,60,0
2780 SYMBOL 185,24,0,126,76,24,48,126,0
2790 SYMBOL 186,254,198,140,126,48,102,254,0
2800 SYMBOL 187,12,24,126,76,24,48,126,0
2810 SYMBOL 188,48,254,172,24,50,102,254,0
2820 KEY DEF 10,1,55,171,172
2830 KEY DEF 11,1,56,173,174
2840 KEY DEF 3,1,57,175,176
2850 KEY DEF 20,1,52,177,178
2860 KEY DEF 12,1,53,179,180
2870 KEY DEF 4,1,54,181,182
2880 KEY DEF 13,1,49,183,184
2890 KEY DEF 14,1,50,185,186
2900 KEY DEF 5,1,51,187,188
2910 SYMBOL 143,3,31,63,127,127,127,255,255:SYMBOL
144,192,248,252,254,254,254,255,255:SYMBOL 145,25
5,255,127,127,127,63,31,3:SYMBOL 146,255,255,254,2
54,254,252,248,192
2920 RETURN

```



APPLICATION - wersja użytkowa, zastosowanie, także: sposób użycia,  
APPLICATION LAYER - warstwa aplikacyjna,  
APPLICATION PROGRAM - program aplikacyjny (realizujący określone zadanie), program użytkowy,  
APPLICATION SOFTWARE - oprogramowanie użytkowe,  
APPLICATION SYSTEM ADMINISTRATOR - administrator zastosowań,  
APPLICATIONS ENGINEERING - projektowanie zastosowania, projektowanie systemu informatycznego,  
APPLY - zastosować,  
APPROACH - metoda, podejście, rozwiązanie, sposób,  
APPROPRIATE - odpowiedni, stosowny,  
APPROXIMATE - przybliżony,  
APPROXIMATE COMPUTATION - rachunek przybliżony,  
APPROXIMATELY - w przybliżeniu,  
APPROXIMATION - aproksymacja, przybliżenie,  
APPROXIMATION ERROR - błąd przybliżenia,  
APT - patrz: AUTOMATIC PROGRAMMING FOR TOOLS,  
ARABIC NOTATIONS - cyfry arabskie,  
ARABIC NUMBERS - cyfry arabskie,  
ARABIC NUMERALS - cyfry arabskie,  
ARBITRARY ACCESS - dostęp dowolny,  
ARBITRARY FORMAT - format dowolny,  
ARBITRARY SEQUENCE COMPUTER - komputer sekwencyjny z dowolnym wykonywaniem rozkazów,  
ARCADE GAME - gra rozrywkowa,  
ARCTANGENT - arcus tangens,  
AREA - obszar,  
AREA DEFINITION - definiowanie obszaru,  
AREA SEARCH - przeszukiwanie obszaru (pamięci),  
ARGUMENT - argument, parametr, zmienna niezależna,  
ARITHMETIC CHECK - kontrola arytmetyczna,  
ARITHMETIC INSTRUCTION - rozkaz arytmetyczny,  
ARITHMETIC-LOGIC UNIT (ALU) - jednostka arytmetyczno-logiczna,  
ARITHMETIC REGISTER - rejestr arytmetyczny,  
ARITHMETIC SECTION - moduł arytmetyczny,  
ARITHMETIC SHIFT - przesunięcie arytmetyczne,  
ARITHMETIC UNIT - arytmometr, jednostka arytmetyczno-logiczna,  
ARITHMETICAL COMPUTATION - działania arytmetyczne,  
ARITHMETICAL PROGRESSION - postęp arytmetyczny,  
ARQ - patrz: AUTOMATIC REQUEST FOR REPEATS,  
ARRANGE IN ORDER OF... - uszeregować w kolejności...,  
ARRANGEMENT - rozmieszczenie, układ, także: projektowanie,  
ARRAY - tablica (zbiór zmiennych o wspólnej nazwie),  
ARRAY ELEMENT - element tablicy,  
ARRAY OF CORES - matryca rdzeni,  
ARRAY PROCESSOR - procesor macierzowy,  
ARROW - patrz: ARROWHEAD,  
ARROWHEAD - strzałka wskazująca (np. na wykresie),  
ARTIFICIAL INTELLIGENCE - sztuczna inteligencja,  
ARTIFICIAL DATA - dane sztuczne, dane zmyślone,  
ASA FONT - pismo ASA (pismo stylizowane dostosowane do odczytywania optycznego),  
ASCENDING - rosnący, wstępujący, wzrastający,  
ASCERTAIN - ustalić, upewnić się, stwierdzić dokładnie,  
ASPECT RATIO - współczynnik kształtu obrazu (stosunek długości osi pionowej do poziomej obrazu),  
ASSEMBLER - program assembler, translator języka symbolicznego,  
ASSEMBLER LANGUAGE - język typu assembler, język assemblerowy,  
ASSEMBLY - tłumaczenie języka symbolicznego,  
ASSEMBLY INSTRUCTION - instrukcja assemblera,  
ASSEMBLY LANGUAGE - język symboliczny, język adresów symbolicznych,  
ASSEMBLY LINE ROBOTS - roboty montażowe,  
ASSEMBLY PROGRAM - translator języka symbolicznego,  
ASSEMBLY ROUTINE - translator języka symbolicznego,  
ASSESS - oceniać, szacować,  
ASSESSMENT - ocena, oszacowanie,  
ASSIGN - przyporządkować, przydzielić, także: określić,  
ASSIGN PRIORITIES - wyznaczyć pierwszeństwo, określić priorytet,  
ASSIGNMENT - przydział, wyznaczenie, przyporządkowanie,  
ASSIGNMENT PROBLEM - zagadnienie przydziału, problem przyporządkowania,  
ASSIGNMENT STATEMENT - instrukcja przydziału,  
ASSOCIATE - dołączony, sprzężony, stowarzyszony,  
ASSOCIATED DESCRIPTOR - deskryptor skojarzeniowy,  
ASSOCIATED GROUP - grupa stowarzyszona,  
ASSOCIATED MATRIX OF LINEAR MAP - macierz odpowiadająca odwzorowaniu liniowemu,  
ASSOCIATIVE MEMORY - pamięć asocjacyjna, pamięć skojarzeniowa,  
ASSOCIATIVE STORAGE - pamięć asocjacyjna,  
ASSOCIATOR - urządzenie kojarzące (elementy podobne)  
ASSORT - sortować, segregować, dobierać,  
ASSORTMENT - dobór, zestaw, także: sortowanie, segregowanie,  
ASSUMED DECIMAL POINT - przecinek obliczeniowy,  
ASSUMED MEAN - średnia dowolnie przyjęta,  
ASSUMPTION - założenie, przypuszczenie,  
ASSURANCE - pewność,  
ASTERISK - gwiazdka jako znak graficzny, odsyłacz,  
ASYNCHRONOUS - asynchroniczny,

ASYNCHRONOUS COMPUTER - komputer asynchroniczny,  
ASYNCHRONOUS METHOD OF OPERATION - praca w trybie asynchronicznym, praca asynchroniczna,  
ASYNCHRONOUS OPERATION - praca w trybie asynchronicznym, praca asynchroniczna,  
ASYNCHRONOUS SERIAL PROTOCOL - asynchroniczny protokół szeregowy,  
ASYNCHRONOUS TRANSMISSION - transmisja asynchroniczna (każdy znak poprzedza bit startu a po znaku występuje bit stopu),  
AT DIFFERENT LEVELS - na różnych poziomach,  
ATTACHMENT - urządzenie dodatkowe (dołączane), uzupełnienie,  
ATTACHMENT UNIT INTERFACE - sprzężenie z jednostką połączeniową tj. kabel, złącza i odpowiednia elektronika,  
ATTENTION - uwaga,  
ATTENUATION - tłumienie, osłabienie,  
ATTENUATION BAND - pasmo tłumienia,  
ATTENUATION COEFFICIENT - współczynnik tłumienia,  
ATTENUATION CONSTANT - stała tłumienia, tłumienność jednostkowa,  
ATTENUATION DIAGRAM - charakterystyka tłumienia, także: charakterystyka amplitudowa,  
ATTENUATION DISTORTION - zniekształcenie tłumieniowe (sygnału),  
ATTENUATION EQUALIZER - korektor tłumieniowy,  
ATTENUATION RATE - współczynnik tłumienia,  
ATTRIBUTE - atrybut, cecha, także: cecha elementu obrazu, np. wykrywalność, poziom intensywności świecenia, itd.,  
AU - patrz: ARITHMETIC UNIT,  
AUDIBLE ALARM - wskaźnik akustyczny,  
AUDIBLE SIGNAL - wskaźnik akustyczny,  
AUDIO RESPONSE DEVICE - wyjście foniczne, wyjście akustyczne,  
AUGMENT - zwiększać,  
AUGMENTER - wartość dodawana,  
AUTO-SHUTDOWN - automatyczne wyłączenie systemu,  
AUTOCODE - autokod, język autokodowy, język automatycznego programowania,  
AUTOINDEXING - indeksowanie automatyczne,  
AUTOLOAD - ładowanie automatyczne,  
AUTOMATA THEORY - teoria automatów,  
AUTOMATE - zautomatyzować,  
AUTOMATIC - automatyczny,  
AUTOMATIC AMPLITUDE CONTROL - automatyczna regulacja amplitudy,  
AUTOMATIC ASSEMBLY RECORDING SYSTEM - system zapisu z automatycznym łączeniem,  
AUTOMATIC CALLING UNIT - autowzywak,  
AUTOMATIC CARD PUNCH - automatyczna dziurkarka kart,  
AUTOMATIC CARRIAGE - przesuw automatyczny, karetki automatyczne,  
AUTOMATIC CHARACTER RECOGNITION - automatyczne czytanie pisma,  
AUTOMATIC CODING SYSTEM - system automatycznego kodowania,  
AUTOMATIC CONTROL - sterowanie automatyczne,  
AUTOMATIC CONTROL SYSTEM - układ automatycznej regulacji,  
AUTOMATIC DATA PROCESSING - automatyczne przetwarzanie danych,  
AUTOMATIC NEED-TO-REPAIR INDICATOR - automatyczny wskaźnik konieczności wykonania naprawy,  
AUTOMATIC PROCESSING - przetwarzanie automatyczne,  
AUTOMATIC PROGRAM INTERRUPT - automatyczne przerwanie realizacji programu,  
AUTOMATIC PROGRAMMING - programowanie automatyczne,  
AUTOMATIC PROGRAMMING FOR TOOLS - język programowania do sterowania numerycznego obrabiarek,  
AUTOMATIC REQUEST FOR REPEATS - system z odpowiedzią zwrotną (umożliwia ochronę przed błędami),  
AUTOMATIC RESPONSE FILE - plik (zbiór) przygotowanych odpowiedzi,  
AUTOMATIC SCAN - przeglądanie automatyczne,  
AUTOMATIC STORAGE ALLOCATION - automatyczny przydział pamięci,  
AUTOMATIC STROBE PULSE - bramka śledząca,  
AUTOMATIC TYPEWRITER - automat piszący,  
AUTOMATICS - automatyka (nauka),  
AUTOMATION - automatyzacja,  
AUTOMATIZATION - automatyzacja,  
AUTOMATON - automat, także: komputer z oprogramowaniem,  
AUTONOMOUS WORKING - działanie niezależne,  
AUTOPLOTTER - pisak X-Y,  
AUTOPOLLING - samoczynne zbieranie danych (poprzez przeglądanie),  
AUXILIARY CARRY - przeniesienie pomocnicze,  
AUXILIARY CONTROL AREA - obszar zapisów zniekształconych, także: obszar dodatkowy,  
AUXILIARY MEMORY - pomocnicza pamięć wewnętrzna,  
AUXILIARY STORAGE - pamięć uzupełniająca, pamięć pomocnicza, pamięć wspomagająca,  
AUXILIARY STORE - pamięć pomocnicza,  
AVAILABILITY - dyspozycyjność, dostępność, zapas,  
AVERAGE - wartość przeciętna, wartość średnia,  
AVERAGE DEVIATION - odchylenie średnie,  
AVERAGING DEVICE - układ uśredniający,  
AVERAGING TIME - czas uśredniania,

AVOIDANCE - unikanie (np. zbędnych danych),  
AX (ACCUMULATOR) - akumulator,  
AXIS - oś,  
AXIS OF ABSCISSAE - oś odciętych,  
AXIS OF ORDINATES - oś rzędnych,

# B

BACK BONE - wspólna oś,  
BACK CIRCUIT - układ sprzężenia zwrotnego,  
BACK SLASH - odwrócony ukośnik,  
BACK-UP MEMORY - pamięć rezerwowa,  
BACKGROUND - tło,  
BACKGROUND COLOR - kolor tła (obszaru otaczającego przedmiot),  
BACKGROUND IMAGE - tło (podstawa) obrazu,  
BACKGROUND PROGRAM - program drugoplanowy,  
BACKING STORAGE - pamięć uzupełniająca, pamięć pomocnicza, pamięć wspomagająca,  
BACKSPACE - cofacz,  
BACKSPACE CHARACTER - znak cofania, cofacz,  
BACKSPACE KEY - klawisz cofający, cofacz,  
BACKSPACING - cofanie (np. taśmy),  
BACKTALK - przekazywanie informacji z komputera rezerwowego do komputera czynnego,  
BACKUP - składowanie, żelazna rezerwa,  
BACKUS NORMAL FORM - notacja BACKUSA-NAURA,  
BACKWARD CHAINING - wnioskowanie wsteczne (w systemach ekspertowych),  
BACKWARD COUNTER - licznik odejmujący,  
BALANCE - bilans, pozostałość, równowaga, saldo, także: uzgadniać, zestawiać, równoważyć,  
BALANCED ERROR - błąd zrównoważony,  
BALANCING - uzgadnianie (np. wyników uzyskanych na maszynie), także: sprowadzanie do stanu zerowego,  
BALL PRINTING HEAD - głowica pisząca, głowica czcionkowa, głowica kuliasta,  
BALLOT BOX - kratka na formularzu dla zamieszczenia w niej np. krzyżyka lub kreski,  
BAND - pasmo (np. częstotliwości), strefa, zakres,  
BANDWIDTH - szerokość pasma,  
BANK - bank, także: górna lub dolna połowa karty 90-kolumnowej,  
BAR - cecha, także: kreska, linia,  
BAR CODE - kod kreskowy, kod paskowy,  
BAR-CODE READER - pióro świetlne do odczytywania kodu kreskowego,  
BAR-CODE SCANNER - pióro świetlne do odczytywania kodu kreskowego,  
BAR CODING - kodowanie kreskowe (wyrobów),  
BAR DISPLAY - wskaźnik słupkowy, wskaźnik kolumnowy,  
BAR PRINTER - drukarka dźwigniowa,  
BARREL DISTORSION - zniekształcenia beczkowate (nie-doskonałość kineskopu),  
BASE - baza, także: podstawa liczenia,  
BASE ADDRESS - adres bazowy, adres początkowy, adres podstawowy,  
BASE ADDRESS REGISTER - rejestr adresu bazowego,  
BASE NOTATION - oznaczenie podstawy liczenia,  
BASE NUMBER - podstawa systemu liczenia,  
BASE OF FLOATING POINT NUMBER - podstawa liczby zmiennoprzecinkowej,  
BASE OF THE COUNTING SYSTEM - podstawa systemu liczenia,  
BASE PAGE - strona bazowa,  
BASE POINT - punkt bazowy,  
BASE SPACE - przestrzeń bazowa,  
BASEBAND COAXIAL SYSTEM

BASIC - język instrukcji symbolicznych ogólnego zastosowania dla początkujących (wysokopoziomowy język programowania),  
BASIC CIRCUIT - układ podstawowy,  
BASIC DISC OPERATING SYSTEM (BDOS) - podstawowy dyskowy system operacyjny,  
BASIC ELEMENT - element podstawowy,  
BASIC INDISSOLUBLE INFORMATION UNIT (BIT) - jednostka miary określająca zawartość informacyjną wiadomości,  
BASIC INPUT/OUTPUT SYSTEM (BIOS) - podstawowy system WEJŚCIA/WYJŚCIA,  
BASIC MACHINE CYCLE - cykl podstawowy,  
BASIC OPERATING SYSTEM (BOS) - podstawowy system operacyjny,  
BASIC SET - zbiór bazowy,  
BATCH - grupa, partia, seria, wsad, także: grupować, gromadzić, paczkować (np. dokumenty),  
BATCH DATA PROCESSING - wsadowe (partiotwe) przetwarzanie danych,  
BATCH OF DATA - partia danych,  
BATCH PROCESSING - przetwarzanie wsadowe (partiotwe),  
BATCH STATION - stacja przetwarzania wsadowego (partiotwego),  
BATCHING - grupowanie programów we wsady (w partie),  
BAUD - bod, jednostka miary szybkości transmisji da-

nych cyfrowych,  
BAUD RATE - szybkość transmisji danych cyfrowych (bitów/sekundę),  
B-BOX - rejestr B, rejestr modyfikacji,  
BCD - patrz: BINARY CODED DECIMAL,  
BCD - patrz: BLANK COLUMN DETECTION,  
BEAM - promień światła, pęczek promieni,  
BEAM ACCESSIBLE MEMORY - pamięć z dostępem strumieniowym,  
BEAM LEAD - wyprowadzenie belkowe (w układach scalonych),  
BEAM-LEAD BONDING - mikromontaż metodą wyprowadzeń belkowych,  
BEEP - sygnał akustyczny modulowany,  
BEFORE-IMAGE PROTECTION - zabezpieczenie obrazu pierwotnego,  
BEGINNING FLAG - znacznik początku,  
BEGINNING OF THE TAPE (BOT) - znacznik początku taśmy,  
BEGINNERS ALL-PURPOSE SYMBOLIC INSTRUCTION CODE - patrz: BASIC,  
BEGINNING OF THE FILE - początek zbioru,  
BEGINNING OF THE FILE AREA - początek obszaru pliku (zbioru),  
BEGINNING OF THE SEEK AREA - początek obszaru przeszukiwania,  
BELIEVED CORRECTNESS - "wiarygodność poprawności",  
BELL CHARACTER - znak wywołujący sygnał akustyczny,  
BENCHMARK - standardowe zadanie rozwiązywane przez komputer w celach porównawczych,  
BENCHMARK PROBLEM - problem wzorcowy (dot. porównywania wydajności różnych typów komputerów),  
BENCHMARKING - komparatystyka,  
BEND - wygięcie (np. karty),  
BER - patrz: BIT ERROR RATE,  
BIAS - polaryzacja stałą liczbą, zwiększyć o stałą wartość, wartość odchylenia, obciążenie (estymatora), podkład,  
BIASED BY... - zwiększone o wartość ...

BIASED NUMBER - liczba nieskalowana (przesunięta),  
BIAX - biaks, także: element ferrytowy pamięciowy lub logiczny,  
BIDIRECTIONAL - dwukierunkowy,  
BIDIRECTIONAL BUS - magistrała dwukierunkowa,  
BIFURCATION - rozgałęzienie, rozwidlenie,  
BILATERAL - dwustronny,  
BILLI - przedrostek oznaczający miliard (w nomenklaturze amerykańskiej),  
BILLIBIT - miliard bitów (kilomegabit),  
BILLICYCLE - miliard cykli, gigacykl,  
BILLION - bilion (w Wielkiej Brytanii) także: miliard (w USA),  
BIMAG CORE - pamięciowy rdzeń magnetyczny mający dwa stany namagnesowania,  
BIN - zbiornik (dla kart), pojemnik na taśmę,  
BINARY - binarny, także: system dwójkowy,  
BINARY ARITHMETIC - arytmetyka binarna,  
BINARY CARRY - przeniesienie dwójkowe,  
BINARY CELL - komórka dwójkowa (dla przechowywania jednego znaku w zapisie dwójkowym), także: urządzenie dwustanowe,  
BINARY CODE - kod binarny, kod dwójkowy,  
BINARY CODED DECIMAL (BCD) - system dziesiętny kodowany dwójkowo, kod dwójkowo-dziesiętny,  
BINARY COMPLEMENT - uzupełnienie do dwóch, uzupełnienie binarne,  
BINARY COUNTER - licznik binarny, licznik dwójkowy,  
BINARY DATA - dane binarne,  
BINARY DIGIT - cyfra binarna,  
BINARY DIGITAL VARIABLE - zmienna binarna,  
BINARY ELEMENT - element dwójkowy,  
BINARY ERROR CORRECTING CODE - dwójkowy kod poprawiania błędów,  
BINARY ERROR DETECTING CODE - dwójkowy kod wykrywający błąd,  
BINARY LOGIC - logika dwójkowa, logika dwuwartościowa,  
BINARY LOGICAL FUNCTION - funkcja logiczna (dwuwartościowa),  
BINARY NOTATION - zapis dwójkowy,  
BINARY NUMBER - liczba dwójkowa,  
BINARY NUMBER SYSTEM - układ zero-jedynkowy,  
BINARY POINT - przecinek dwójkowy,  
BINARY PROGRAM - program binarny,  
BINARY REPRESENTATION - reprezentacja dwójkowa,  
BINARY SCALER - przelicznik dwójkowy,  
BINARY SEARCH - przeszukiwanie metodą połowienia,  
BINARY-DECIMAL - dwójkowo-dziesiętny,  
BINARY-HEXADECIMAL - dwójkowo-szesnastkowy,  
BINARY-OCTAL - dwójkowo-ósemkowy,  
BINARY-QUATERNARY - dwójkowo-czwórkowy,  
BINARY-TO-DECIMAL CONVERSION - konwersja dwójkowo-dziesiętna,  
BINDER - urządzenie do oprawiania (np. tabulogramów),  
BINNING - sortowanie struktur półprzewodnikowych,  
BINOMIAL - dwuczłonowy, dwumienny,  
BIOS - moduł zawierający podstawowe procedury obsługi urządzeń zewnętrznych,  
BIPOLAR - dwubiegunowy,  
BIPOLAR JUNCTION TRANSISTOR - tranzystor bipolarny,  
BIQUINARY - dwupiętkowy,

# GIEŁDA POMYSŁÓW

## Tylko dla szczęściarzy

Przysłowia są mądrością narodów. Jedno z nich mówi, że nadzieja jest matką głupich. Jest jednak drugie, które zapewnia, że matka swoich dzieci nie opuszcza. W tym drugim należy chyba upatrywać tak dużej popularności gry w „Toto Lotka”.

Z grą tą związane są co tydzień emocje dwojakiego rodzaju. Najpierw osoba, która zdecydowała się w nią zagrać przeżywa rozterkę związaną z wyborem liczb. Później gracz doprowadza się do stanów przedzawałowych, tkwiąc przed ekranem telewizora w niecierpliwym oczekiwaniu na wyniki losowania.

Prezentowany tu program pozbawia graczy pierwszego z tych stresów. Wystarczy tylko, zgodnie z poleceniem wyświetlonym na ekranie, wybrać jakiś wariant „lotka”, chwilę poczekać, po czym przepisać słupki liczb na kupon.

Na wszelki wypadek, radzę przygotować sobie odpowiedź na pytanie: „Co zrobić z tak dużą ilością wygranych pieniędzy?” Życzę wszystkim rozbicia banku „Toto Lotka”

```

10 CLS : RANDOMIZE : DIM a(2,3) : DIM b(3,15) : DIM c(5,7)
20 LET t(1)="Express-lotek" : LET t(2)="Super-lotek" : LET t(3)="Duzy-lotek"
30 LET a(1,1)=5 : LET a(1,2)=7 : LET a(1,3)=6 : LET a(2,1)=42 : LET a(2,2)=49 : LET
a(2,3)=49
40 PRINT "Możesz zagrać w Toto-lotka" "Który z wariantów wybierasz? (wcisnij
) odpowiedni klawisz)" " e - express-lotek" " s - super-lotek" " d -
duzy-lotek" "Wcisnięcie innego klawisza powoduje zakończenie pracy"
50 IF INKEY="" THEN GO TO 50
60 IF INKEY="e" THEN LET i=1 : GO TO 200
70 IF INKEY="s" THEN LET i=2 : GO TO 200
80 IF INKEY="d" THEN LET i=3 : GO TO 200
90 PRINT "Odpowiedz spora repertuaru s/s/d" "Koniec pracy" : STOP
200 PRINT "FLASH !! Poczekaj" : FLASH 0
210 FOR k=1 TO 5
220 GO SUB 1000 : LET a(k,1)=k
230 FOR l=2 TO a(1,1)
240 GO SUB 1000
250 LET n=1
260 IF a(k,n)=k THEN GO TO 240
270 IF n(1-1) THEN LET n=n+1 : GO TO 260
280 LET a(k,1)=k
290 NEXT l
300 FOR n=2 TO a(1,1)
310 LET l=n-1 : LET x=a(k,n)
320 IF x>a(l,1) THEN LET a(l,1)=x : GO TO 360
330 LET a(k,1)=a(k,1)
340 IF l=1 THEN LET l=1 : GO TO 320
350 LET a(k,1)=x
360 NEXT n : NEXT k
400 CLS : PRINT TAB 5; "REFLEKS" : PRINT
410 FOR l=1 TO a(1,1)
420 FOR k=1 TO 5
430 IF a(k,1)=l THEN PRINT " "
440 PRINT " " : "fa(k,1)"
450 NEXT k : PRINT : NEXT l : GO TO 40
1000 LET x=INT (RND*(a(2,1)+1))
1010 IF x>=5 OR x>a(2,1) THEN GO TO 1000
1020 RETURN
  
```

## Refleks

„...Jestem stałym czytelnikiem „IKS-a” i bardzo sobie cenię Wasze pismo, w szczególności za umieszczanie w nim programów komputerowych.

Obecnie jestem użytkownikiem mikrokomputera COMMODORE 128, lecz kilka lat temu miałem SHARP-a...

...Gdyby Państwo byli zainteresowani moimi programami, bardzo proszę o odpowiedź!...”

Rafał PLUTECKI

CZEKAMY! „IKS” JEST WASZ!

Nadesłany program REFLEKS ma dwie cenne zalety: jest *krótki* oraz *użyteczny* (pomaga w opanowaniu „ślepego zapisu”).

Po wywołaniu RUN 500 ukazuje się napis POMIAR REFLEKSU, a zaraz potem generator liczb losowych rozpoczyna „zręcznościówkę”: wyświetla jedną z liter (od A do Z) — wybraną przypadkowo. Należy SZYBKO nacisnąć WŁAŚCIWY klawisz. Wtedy pokaże się kolejna. Tak „palcujemy” trzydzieści liter.

```

PG 500 REM *****
YA 502 REM *
HX 504 REM * POMIAR REFLEKSU *
QC 506 REM * * Atari * *
YS 508 REM *
PI 510 REM *****
PI 520 OPEN #1,4,0,"K:"
JL 522 DIM A$(1),B$(1)
UY 526 PRINT CHR$(125)
IL 528 ? :? "POMIAR REFLEKSU"
GB 530 GOSUB 556:M=N
TQ 532 FOR K=1 TO 31
CQ 533 B=INT(RND(0)*27)
IX 534 A$=CHR$(64+B):B=B+64
  
```

# LIGA MYŚLĄCYCH

## Zadanie 1

Kierowca samochodu przebył drogę w ciągu 5 godzin przy przeciętnej prędkości p. Dwie piąte drogi przejechał z prędkością o 8 km/h mniejszą, a resztę z prędkością o 5 km/h większą. Do celu podróży dotarł z opóźnieniem 10 min. Z jaką przeciętną prędkością P przebył drogę kierowca samochodu?

## Zadanie 2

Podczas trwania I wojny światowej wojska oblegały stary zamek. Podczas oblężenia kula z armaty uszkodziła posąg, stojący na dziedzińcu zamkowym. Dzień ataku był ostatnim dniem miesiąca, a liczba lat stojącego na dziedzińcu posągu i liczba lat dowódcy wojsk oblegających zamek były liczbami parzystymi. Liczba, wyrażająca dzień miesiąca, w którym nastąpił atak, pomnożona przez numer miesiąca ataku, pomnożona przez podaną w stopach długość piki posągu, pomnożona przez połowę wieku dowódcy wojsk oblegających zamek, pomnożona przez połowę czasu stania posągu daje w iloczynie 451 066. Należy podać rok, w którym wznieziono posąg na placu zamkowym. Odpowiedź proszę uzasadnić.

## Zadanie 3

Połowa pasażerów, wchodząca na przystanku początkowym do autobusu zajęła miejsca siedzące. Na kolejnym przystanku do autobusu wsiadło 8 proc. dotychczasowej liczby pasażerów. Ile osób usiadło na początkowym przystanku, jeżeli wiadomo, że autobus pomieści nie więcej niż 70 osób?

## Zadanie 4

Ile zer na końcu ma liczba, którą otrzymamy mnożąc 1000 kolejnych liczb naturalnych tzn.  $1 \cdot 2 \cdot 3 \cdot \dots \cdot 998 \cdot 999 \cdot 1000$ ? Odpowiedź proszę uzasadnić.

## Zadanie 5

Mamy nierzetelną wagę dwuramienną (tzn. ramiona wagi nie są równe). Chcąc odważyć 2 kg cukru postępujemy w następujący sposób: Kładziemy kilogramowy odważnik na lewej szalce, a na prawą sypiemy cukier aż do zrównoważenia. Opróżniwszy obie szalki wagi, kładziemy kilogramowy odważnik na prawej szalce, a na lewą sypiemy cukier aż do zrównoważenia. Czy łączny ciężar odważonego w ten sposób cukru jest równy 2 kg?

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

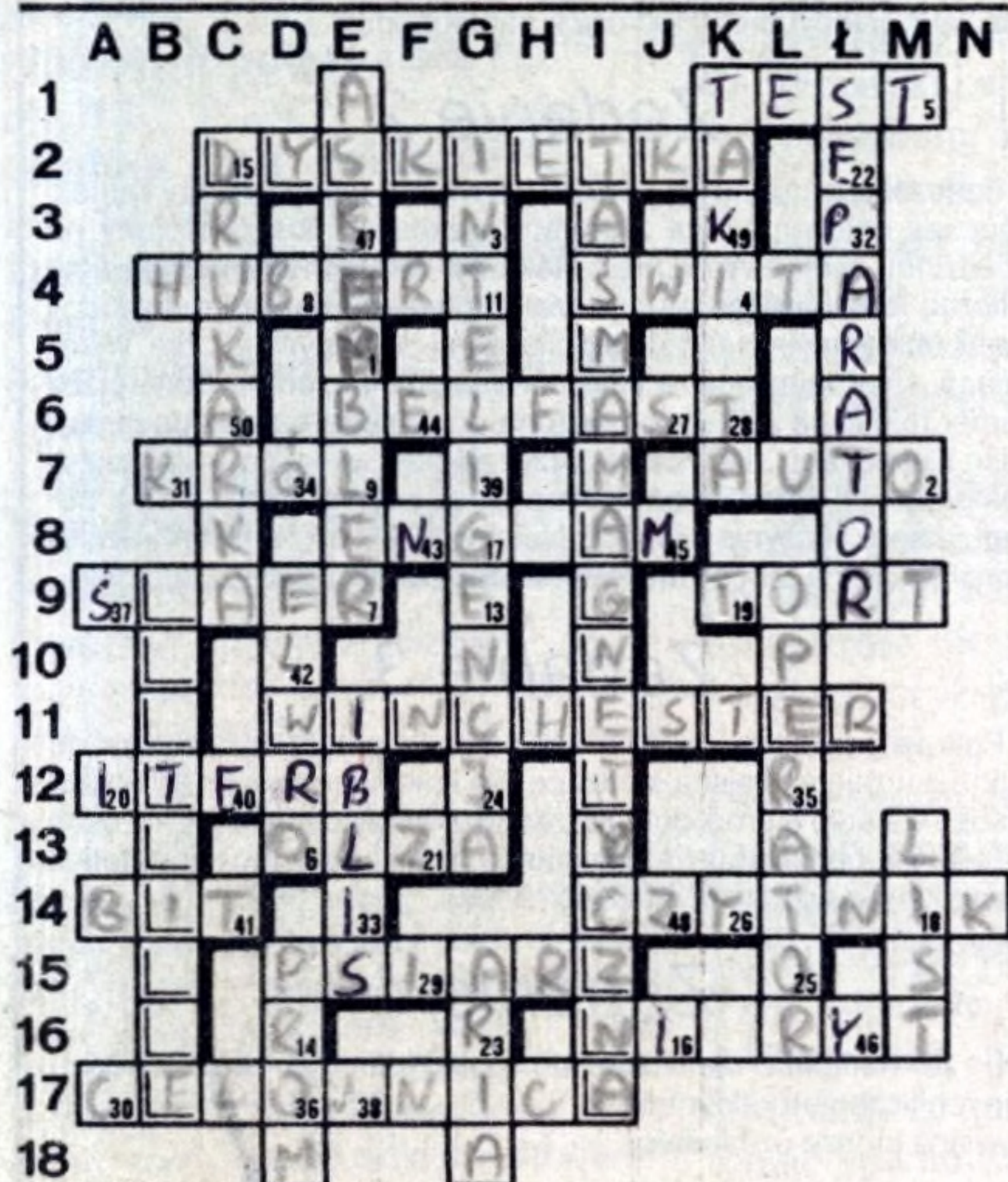
```

YX 536 FOR C=0 TO 10 : SOUND 0,100,10
,10 : NEXT C : SOUND 0,0,0
CU 540 PRINT CHR$(125) : POKE 752,1
AT 541 ? :? " * ";N-M;" * ";A$
UB 542 IF K>30 THEN 548
HL 543 GET #1,Z : Z=PEEK(763)
YK 544 GOSUB 556
EA 545 IF Z<>B THEN 540
HP 546 NEXT K
MR 548 ? :? " CZAS=" ;N-M;" JEDNOSTEK
"
DJ 550 ? :? " JESZCZE RAZ " ;CHR$(212)
);"/";CHR$(206); : INPUT A$
CI 552 IF A$=CHR$(84) THEN 530
GB 554 END : REM *****
*
FQ 556 N=PEEK(18)*65536+PEEK(19)*25
6+PEEK(20) : N=INT(N/10)
AJ 558 RETURN
  
```

# KUPON ►



## KRZYŻÓWKA NR 10



### POZIOMO:

1K) instrukcja pozwalająca na sprawdzenie, czy określony punkt ekranu świeci się, czy nie lub... próba, rodzaj sprawdzianu, 4B) rzadkie imię męskie, 4I) asysta, orszak, 6E) stolica Irlandii Pn., 7B) Bolesław Chrobry lub Stefan Batory, 7K) samochód, 8E) ślad pamięciowy, 9A) półprodukt do wyrobu papieru, otrzymany w wyniku mechanicznego rozwołkniania drewna, 9K) urodzinowe ciasto, 12A) lantanowic o liczbie atomowej 70, 13D) przepływa przez Cieszyn, 14A) 0 lub 1, 14I) urządzenie wejściowe komputera do przyjmowania informacji z taśm lub kart dziurkowanych, 15D) miłośnik psów, 16I) saletra potasowa, saletra indyjska, 17A) pomocnicza lunetka, używana zwłaszcza przy fotografowaniu

### PIONOWO:

C2) np. MPS 801, Seikosha GP500, Star Gemini 10X, D9) polski mikrokomputer z Wrocławia, D15) programowana pamięć tylko do odczytu lub... na trasie Ystad—Swinoujście, E1) język programowania, E11) diabeł, zły duch w wierzeniach muzułmanów, G2) brak jej niestety komputerom, G15) operowa pieśń solowa, K1) tyczka lekkoatlety, L9) pracownik obsługujący komputer, Ł1) średnik, spacja, dwukropki jako oddzielniki, M13) instrukcja wyświetlenia tekstu programu lub... korespondencja

Oznaczone pola (od 2C i 11D poziomo oraz B9 i I2 pionowo) — rodzaje pamięci zewnętrznych mikrokomputerów.

Jako rozwiązanie należy podać cztery nazwy pamięci wpisane w oznaczone pola diagramu krzyżówki oraz część nazw urządzeń we-wy, które otrzymamy po uszeregowaniu liter z pól oznaczonych w prawym dolnym rogu liczbami od 1 do 50.

Rozwiązania krzyżówki na kartach pocztowych należy nadsyłać pod adresem redakcji do końca grudnia 1987 r., naklejając kupon „IKS-a”. Wśród czytelników rozlosujemy bony pieniężne i nagrody książkowe.

## MIKROCIEKAWOSTKI

# IBM

## Personal System/2



### TOWARZYSTWO WYNAŁAZCÓW

Projektowanie systemów informatycznych, prace konstrukcyjne, ekspertyzy techniczne, ekonomiczne i prawne

**UWAGA:** użytkownicy mikrokomputerów

AMSTRAD COMODORE ZX SPECTRUM  
ATARI IBM

- instrukcje, opisy, programy użytkowe  
- programy narzędziowe, dla rzemiosła  
- gry

- tworzymy oprogramowanie według zlecenia klienta

Dowolny nośnik (również dostarczony przez klienta). Prowadzimy sprzedaż wysyłkową.

Informacji żądajcie za pośrednictwem poczty pod adresem:

"RATIO" TOWARZYSTWO WYNAŁAZCÓW  
03-318 NARSZANA ul. OGIŃSKIEGO 1

„IKS” — dodatek „Żołnierza Wolności”. Redaguje: Wiesław Cetera (kierownik zespołu); stali współpracownicy: Włodzimierz Gogolek, Janusz Janiec, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotokład i druk — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 9332. Nr ind. 361682.

K-96