

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965  
PL ISSN 0860-1674

# Bajtek

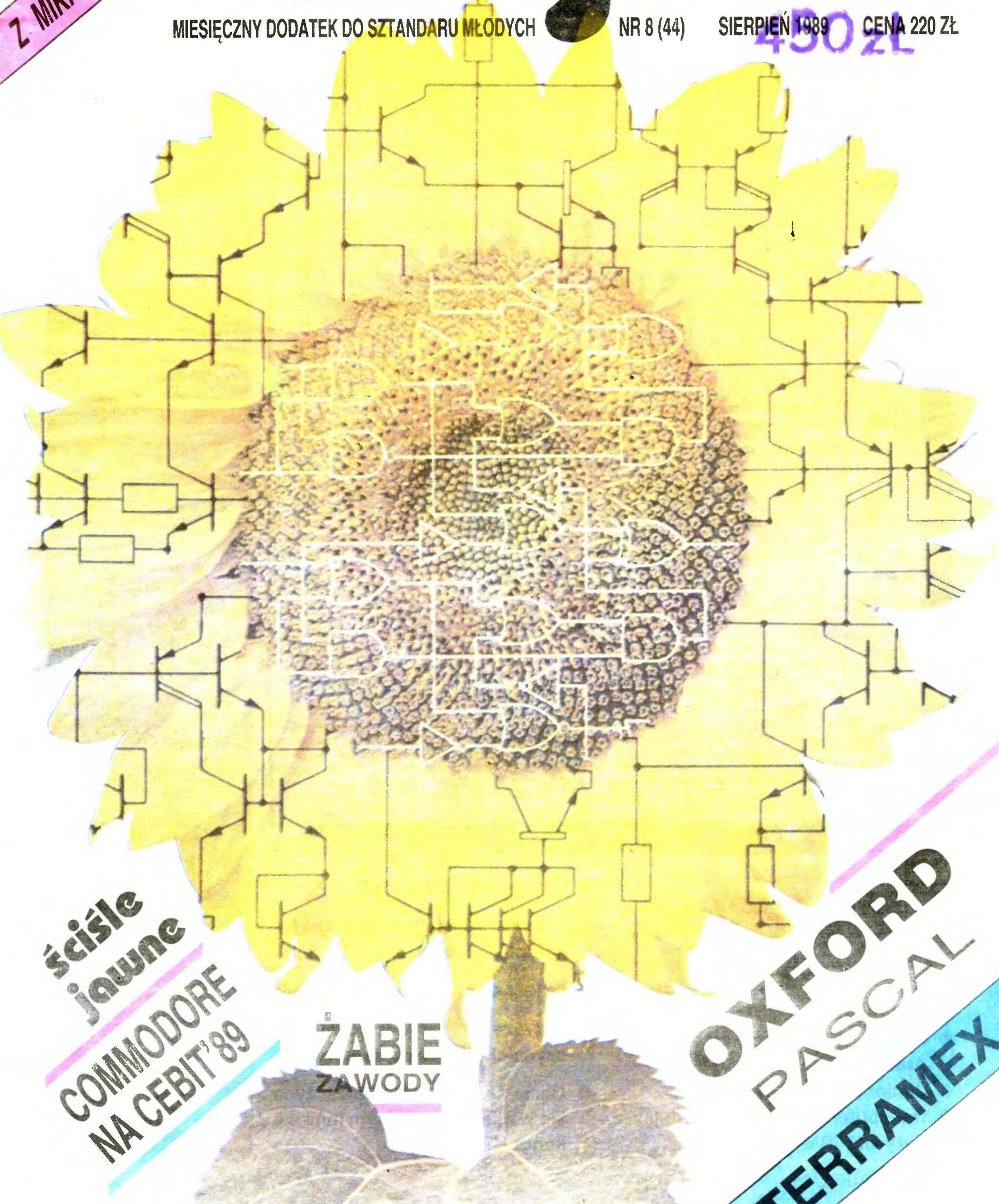
MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 8 (44)

SIERPIEŃ 1989

CENA 220 ZŁ

430 ZŁ



ściśle  
jawne  
COMODORE  
NA CEBIT '89

ZABIE  
ZAWODY

OXFORD  
PASCAL  
TERRAMEX



# — BEZ EMBARGA —

Wydarzeniem miesiąca jest zapowiedź ministra handlu USA o złagodzeniu ograniczeń w handlu komputerami — chodzi oczywiście głównie o handel ze Związkiem Radzieckim i krajami naszego ugrupowania polityczno-wojskowego. Specjaliści pracujący w słynnym COCOM-ie uważają nie analizowali do tej pory każdy wniosek firm zachodnich o zgodę na eksport na Wschód, pod tym kątem, czy nie ma w nim czasem produktów o tzw. podwójnym przeznaczeniu, tzn. takich, które mogłyby „podnieść zdolność obronną” kupującego. Dużo wskazuje na to, że ten relikw zimy wojny przejdzie wkrótce do przeszłości.

W uzasadnieniu zapowiedzi o liberalizacji eksportu komputerów podkreśla się, że analogiczne do amerykańskich urządzenia dostępne są już dość swobodnie na rynkach trzecich, a nawet wytwarzane w niektórych krajach socjalistycznych. Dlatego właśnie relikwem stają się dotychczasowe przepisy, w myśl których firmy amerykańskie muszą przechodzić długą procedurę uzgodnień przy uzyskiwaniu zezwoleń rządowych na dostawy do krajów socjalistycznych takich komputerów, które mocą i szybkością działania odpowiadają modelowi IBM-AT. Już wkrótce specjalne zezwolenia na sprzedaż tej klasy komputerów nie będą potrzebne.

Amerykańskie decyzje zostały natychmiast z uwagą przeanalizowane na Starym Kontynencie i analitycy doszli do wniosku, że powinny one przynieść więcej korzyści Europie Zachodniej, aniżeli USA, a takie firmy jak Siemens, ICL czy Bull najprawdopodobniej uzyskają przewagę w stosunku do amerykańskich rywali.

Najlepszą pozycję wyjściową do tej nowej jakościowo sytuacji znają już wcześniej Siemens, który w czerwcu br. podpisał z ZSRR umowę w sprawie dostarczenia na rynek radziecki co najmniej 200 tys. komputerów PCD-2M, za sumę 1 mld marek. Brytyjska ICL prowadzi już jedno wspólne przedsiębiorstwo z ZSRR, natomiast francuski Bull podpisał wstępne porozumienie w sprawie skomputeryzowania radzieckich centrów sprzedaży hurtowej i nosi się z zamiarem powołania joint venture w zakresie oprogramowania i usług komputerowych. Wymienione trzy firmy są największe odpowiednio w RFN, WIELKIEJ BRYTANII i FRANCJI.

Piszę o tych wydarzeniach ze sfery wielkiego biznesu komputerowego, gdyż są one kolejną ilustracją tezy, że nie da się zatrzymać na

dłuższą metę prawidłowości rozwoju technicznego poprzez decyzje administracyjne. Świat staje się coraz mniejszy i tworzenie sztucznych przegród — typu właśnie ograniczeń eksportowych najnowszych technologii — nie może na dłuższą metę być efektywne. Oczywiście, nie znamy jeszcze wszystkich szczegółów odwołania embarga, ale tendencja do ułatwiania handlu światowego, a nie wykorzystywania go w celach politycznych jest zarysowana bardzo wyraźnie.

A skoro już o wydarzeniach z wielkiego świata mowa, to miło mi odnotować, że Czytelnicy radzieckiego pisma popularno-naukowego „Technika — Molodioży”, mieli okazję zapoznać się bliżej z „Bajtkiem”. Pismo to, wydawane przez Komitet Centralny Komsomołu w niebagatelnym nakładzie 1 miliona 600 tysięcy egzemplarzy, uznało bowiem, że również młodzież radziecka powinna mieć swój własny miesięcznik komputerowy i rozpoczęło dyskusję o tym, jaki ten przyszły miesięcznik powinien być.

Bliższa prezentacja „Bajtki” — i to poczynając już od okładki „T-M” — ma pomóc radzieckim Czytelnikom w udzieleniu odpowiedzi na postawione przez „Technikę — Molodioży” pytania: „Czy potrzebny jest w ogóle miesięcznik komputerowy?” oraz „Jakim chcielibyście go widzieć?” I o ile pierwsze pytanie ma charakter retoryczny, to na drugie redakcja „T-M” zaczyna już otrzymywać setki odpowiedzi.

Redaktor naczelny „Techniki — Molodioży” Swiatosław Czumakow i kierownik działu w tym piśmie Michaił Puchow, znany autor powieści science-fiction, poczynili już wiele starań, aby młodzież radziecka jak najszybciej mogła korzystać z własnego pisma komputerowego. Sporo czasu przegadaliśmy już nawet o wspólnych zamierzeniach „Bajtki” i „Techniki — Molodioży”. Wychodzimy bowiem ze wspólnego założenia, że młodzi entuzjaści informatyki w naszych krajach nie są w niczym gorsi od swoich kolegów z krajów zachodnich i trzeba im tylko stworzyć odpowiednie warunki, aby mogli zmierzyć się w szlachetnej rywalizacji mającej na celu jak najszersze wykorzystywanie osiągnięć informatyki dla dobra społecznego.

Zniesienie embarga na dostawy komputerów do ZSRR i innych krajów socjalistycznych jest czynnikiem, który realizację tych naszych wspólnych planów powinien przyspieszyć.

Waldemar Siwiński

# BAJTEK

**„BAJTEK” — MIESIĘCZNY  
DODATEK DO „SZTANDARU  
MŁODYCH”**

**ADRES:** 00-687 Warszawa, ul. Wspólna 61, Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych”

**ZESPÓŁ REDAKCYJNY:** Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtki”), Grzegorz Onichimowski (sekretarz redakcji „Bajtki”), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Wallgórski, Roman Wojciechowski, Zdjęcia w numerze: Leopold Dzikowski

**Klany redagują:**  
Commodore — Klaudiusz Dybowski, Dominik Falkowski  
Amstrad-Schneider — Jonasz Mayer  
Spectrum — Marcin Przasnyski  
Atari — Wojciech Zientara, Sergiusz Piotrowski

**Fotokład —** Tadeusz Olczak,  
**Montaż offsetowy —** Grażyna Ostaszewska,  
**Korekta —** Maria Krajewska, Zofia Wóltańska

**WYDAWCA:** RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stańków Zjednoczonych 53, 04-028 Warszawa. Telefon: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.

**Cena 150 zł.**  
Skład technika CRT-200, przygotowanie offsetowe i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.  
Nr zlecenia 033259 n. 120000 A-111

## ZA MIESIĄC:

- o komputerach w szkole — czyli pomóż sobie sam
- odzyskiwanie programów na Atari 65 XE!
- Junior i inni — co dalej?
- transputer do XT — czyli jak „ożywić” IBM

- instrumenty z Teksasu
- galeria „Bajtki” — bez grafiki komputerowej
- ile jest na Ziemi kosmodromów?
- koprocesor arytmetyczny — szybciej, więcej, dokładniej
- Game Over II — ta gra zaczyna się w miejscu, w którym inne się kończą!





# Ściślej jawne

## MUSIMY ZACZAĆ PRODUKOWAĆ NA DUŻĄ SKALĘ TANIE ELEMENTY ELEKTRONICZNE



### rozmowa z plk doc. Bolesławem Szafrąńskim informatykiem

— Mikroelektronika przeciętnemu obywatelowi naszego kraju kojarzy się z reklamami sprzętu komputerowego w telewizji, kolejkami przed „skomputeryzowanymi” kasami na Dworcu Centralnym w Warszawie i źle naliczonymi rachunkami za rozmowy telefoniczne.

— Niestety, to prawda. Dziś Polska jest krajem, w którym informatyka zatrzymała się na poziomie chałupnictwa. Mamy oczywiście bardzo wielu zdolnych naukowców i techników. Jeśli jednak zajmują się oni „produkcją” sprzętu, to jest to głównie składanie komputerów z importowanych części. Czy wie pan, że nasze komputery „chorują na żółtaczkę”?

— Nie rozumiem.

— To oczywiście żart. Gdyby pan zajął pod obudowę „produkowanych” u nas komputerów zabaczyłby pan części sprowadzone z Tajwanu, Singapuru, Korei Południowej. Nie jesteśmy w stanie wyprodukować niezawodnych mikroprocesorów — nawet tych najprostszych — ośmio-bitowych. Jest to świadectwem naszego poziomu, a raczej zacofania cywilizacyjnego. Proszę sobie wyobrazić, że w Chinach w ubiegłym roku wyprodukowano 70 tysięcy własnych mikrokomputerów zgodnych z IBM PC XT. Za seryjną produkcję tych urządzeń biorą się dziś takie kraje jak Indie czy Algieria.

— Będą je prawdopodobnie składały z importowanych elementów.

— Prawdopodobnie tak. Jednak na znacznie większą skalę niż my. W krajach wysoko rozwiniętych „kości” stanowią jedynie niewielką część wartości sprzętu elektronicznego. Koszt ich wytwarzania stale maleje. W naszym kraju ich cena jest uzależniona od kursu dolara i dlatego są one bardzo drogie. Mikroelektronika wtedy się opłaca kiedy jest masowa — to znaczy tania. Jeśli nie będziemy produkowali tanich elementów elektronicznych i to na dużą skalę, to nie tylko komputery nie staną się towarem powszechnego użytku, lecz przede wszystkim elektronika nie poprawi jakości, niezawodności i

funkcjonalności dziesiątków rodzajów sprzętu codziennego użytku: telewizorów, pralek, kuchenek mikrofalowych itd. Bo to się po prostu nie opłaci. Wracając do pańskiego pierwszego pytania. Komputeryzacja nie oznacza postawienia iluś tam maszyn na biurkach. To niczego nie zmienia a kosztuje. Potrzebne są działania systemowe. Komputery to nowa jakość, a ich zadaniem nie jest „wspomaganie” np. działania kasy biletowej, ale radykalna zmiana całego systemu obsługi pasażerów. Tylko wtedy ma to sens jeśli bilet będzie można zarezerwować w każdym biurze ORBIS-u czy PTTK, a nie tylko na Dworcu Centralnym.

— Mam wrażenie, że prawdy te nie są oczywiste dla wszystkich.

— Mam nadzieję, że w przyszłości nowoczesne dziedziny przemysłu staną się dla naszej gospodarki wiodące. Kraje Europy Zachodniej dawno już zrozumiały, że ciężka praca górnika nie wystarczy aby podnieść poziom życia społeczeństwa. Ponadto, raz wydobyty węgiel nigdy nie „odrośnie”. Podobnie jak rudy metali i ropa naftowa. Tłumaczenie obywatelom, że elektronika i komputery to wspaniałe rzeczy — mijają się z celem jeśli zwykły magnetofon kasetowy kosztuje trzy średnie pensje. To nie jest rzecz powszechnego użytku, a drogi towar luksusowy.

— Telewizor kosztuje dziesięć takich pensji.

— Niestety. Dopóki te rzeczy i inne nie staną się powszechnie dostępne, nie ma co marzyć o poparciu społecznym dla elektroniki. Tymczasem zdobycie poparcia społecznego dla dokonania radykalnego zwrotu ku nowoczesnym technologiom a nie tylko elektronice jest sprawą kluczową. Nie jest to tylko problem rządu. Warunki dla tej operacji musi tworzyć nowy Sejm. Nie wystarczy bowiem tylko zamykanie nierentownych stoczni, kopalń i hut. Trzeba tworzyć systemowe warunki dla rozwoju nowoczesnego przemysłu. Uświadomienie społeczeństwu takiej konieczności mogą wziąć na siebie siły społeczne, które podpisały ugodę okrągłego stołu. Jeśli się uda, to w ciągu 3—5 lat będą już widoczne efekty.

— Telewizory staniają?

— Nie tylko telewizory ale i komputery. Musimy w Polsce uruchomić seryjną produkcję mikroprocesorów i pamięci. To z pewnością obniży koszty wytwarzania sprzętu komputerowego. Ponadto mikroprocesor może być zamontowany w pracle automatycznej, samochodzie, maszynie do szycia czy odbiorniku radiowym. Jest to rzecz uniwersalna, nie tylko zwiększa możliwości sprzętu ale staje się też źródłem znacznych oszczędności energii. Treść opublikowanego w naszym kraju trzy lata temu Raportu dla Klubu Rzymskiego pt. „Mikroelektronika i społeczeństwo. Na dobre czy na źle” nie pozostawia cienia wątpli-

wości co do znaczenia słowa „mikroelektronika”. Raport dotyczy SKUTKÓW zastosowań urządzeń przetwarzających informację.

Kilka lat temu Japończycy wprowadzili nowe pojęcie „społeczeństwo informatyczne”. To jeszcze jeden dowód tempa zmian zachodzących na świecie. Nie chcę nikogo straszyć, ale o ile w ciągu kilku lat zachodnie stacje radiowe i telewizyjne przejdą na cyfrowe kodowanie sygnału, to po prostu nie będziemy w stanie ich odebrać — nasz poziom techniczny jest zbyt niski.

— Czyli staniemy się skansenem w środku Europy.

— Dlatego musimy jak najszybciej szukać partnerów do współpracy — na Wschodzie i na Zachodzie. Szczególnie owocna powinna być współpraca z sąsiadami. Jednakże aby tak się stało muszą najpierw nastąpić zmiany systemowe w organizacji i zasadach funkcjonowania RWPG. Z Rady Pomocy musi się ona przekształcić w efektywny system przedsięwzięć gospodarczo-kapitałowych, których ocena zależy będzie od zysku.

Nie oznacza to, że powinniśmy czekać aż powstaną takie warunki. Nie zapominajmy, że jeszcze dziesięć lat temu pod względem rozwoju przemysłu informatycznego Polska była na dobrej pozycji wśród krajów socjalistycznych. Dzisiaj wyprzedziła nas Bułgaria, która seryjnie produkuje mikrokomputery PRAWEC, minikomputery VAX, nowocześniejsze od naszych RIAD-y. Czasem śni mi się koncert złożony np. z producentów radzieckich, polskich i bułgarskich, którzy kontrolują dużą część rynku komputerowego od Łaby po Władywostok. Korea Południowa i Malezja udowodniły, że można w krótkim czasie osiągnąć bardzo wiele. Postęp w elektronice może się stać źródłem szokującego skoku cywilizacyjnego. Można się było o tym przekonać oglądając transmisje z olimpiady w Seulu.

— Brzmi to fantastycznie, ale nie mogą wyzbyć się sceptycyzmu. Zbyt wiele szans zostało zmarnowanych w ostatnich dziesięcioleciach.

— To prawda, ale właśnie dlatego musimy zdecydować się na radykalne rozwiązania i bezwzględnie pilnować ich realizacji. Sądzę, że będzie to jedno z zadań Sejmu.

— W jakiej mierze Pańskie poglądy wynikają z tego, że jest Pan zawodowym wojskowym?

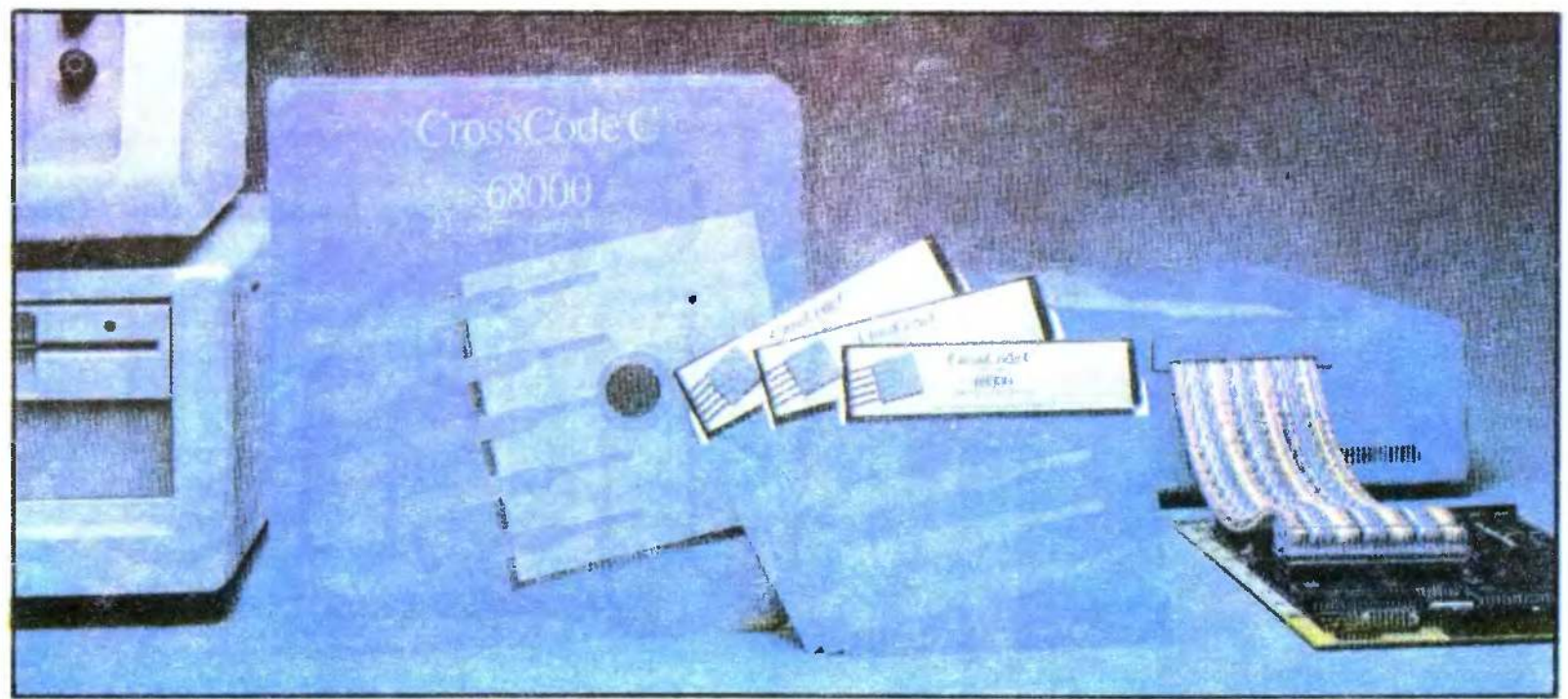
— Napoleon mawiał, że armia maszeruje na brzuchu — dziś, gdyby żył, powiedziałby, że bez „kości” też nie można maszerować. Niech pan nie zapomina, że to właśnie wojsko jest największym odbiorcą części elektronicznych. Nowoczesna armia nie jest sprawą tylko wojskowych — ale wszystkich obywateli.

Marek Czarkowski





**D**odanie Monoputera lub Quadputera, kart do komputera IBM PC, zawierających transputery T800 firmy INMOS, powoduje istotny wzrost mocy obliczeniowej zestawu. Niestety, wąskim gardłem systemu pozostaje tor wizyjny — komputer nie nadąża z wyświetlaniem obliczonych wyników. Problem ten ma duże znaczenie przy korzystaniu z programów typu CAD, które wymagają nie tylko szybkiego koprocatora numerycznego, ale również są często wspomagane przez koprocetry graficzne. Ciekawe rozwiązanie, w formie karty Videoputera, prezentuje amerykańska firma MicroWay. Oferowana karta jest sterownikiem obrazu, zrealizowanym w oparciu o układy T800 (INMOS), 34010 (Texas Instrument), 130MHz DAC (BrookTree). Zastosowanie transputera pozwala na szybką wymianę informacji między pozostałymi transputerami systemu, z pominięciem standardowej szyny IBM'a. Karta zawiera 2MB pamięci RAM, z których połowa wykorzystywana jest do pamiętania obrazu. Możliwa jest praca z monitorami analogowymi z częstotliwością odchylenia poziomego 64 KHz. Karta sprzedawana jest łącznie z podstawowym oprogramowaniem graficznym za 5000\$.  
(JM)



### Cross Code C — narzędzie do programowania sterowników opartych na Motorolii 68 000

**W**iele polskich firm oferuje, w formie dodatkowych kart do IBM PC, systemy uruchomieniowe dla mikroprocesora Z80. Dzięki takim układom możliwe jest projektowanie sterowników opartych na 8-bitowym procesorze firmy Zilog. Sterowniki te wykorzystywane są później w prostych systemach automatyki, w urządzeniach powszechnego użytku (pralki automatyczne, windy itp.). Bardziej skomplikowane układy projektowane są w oparciu o 16-bitowe procesory firmy Motorola. Do ich uruchomienia służy specjalistyczne oprogramowanie. Amerykańska firma Software Development Systems dostarcza do tego celu zestaw o nazwie CrossCode C, zawierający kompilator języka C. Oprócz niego oferowany jest także pełny assembler, konsolidator i pro-

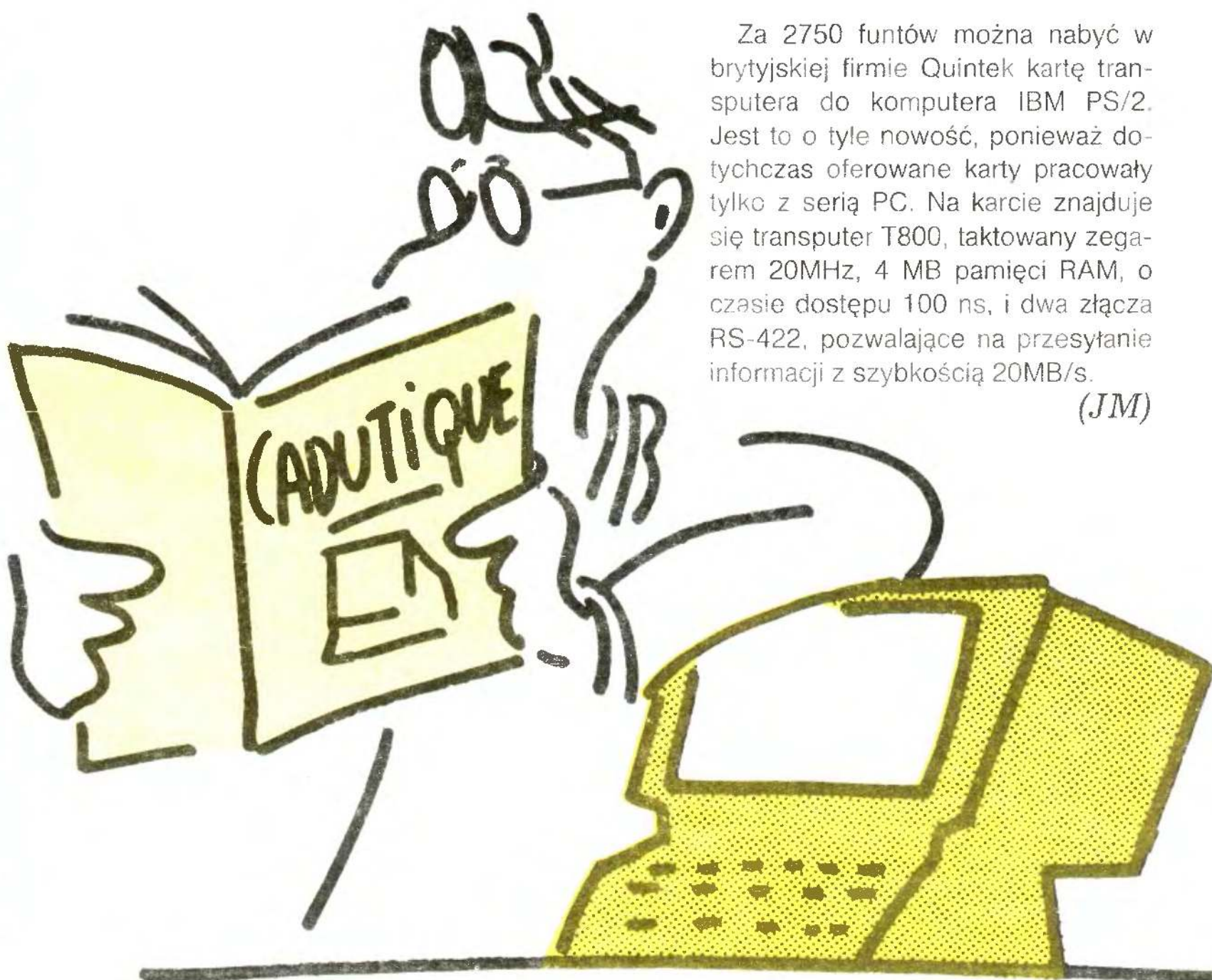
gram typu downloader, który umożliwia bezpośrednią współpracę z programatorami EPROM'ów i sprzętowymi emulatorami. Kod generowany przez kompilator nadaje się do zapisu w pamięci typu ROM, a downloader pozwala na konwersję kodu na jeden z wielu standardowych formatów: Motorola Srecords, Intel Hex, Tek Hex i inne.  
(JM)

### KARTA TRANSPUTERA DO PS/2

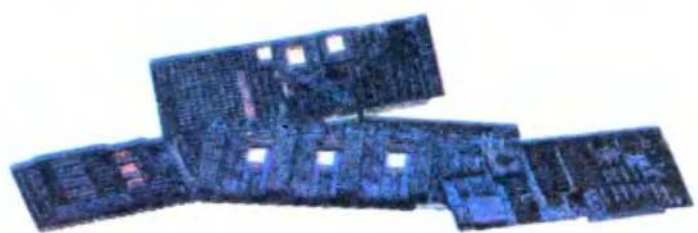
Za 2750 funtów można nabyć w brytyjskiej firmie Quintek kartę transputera do komputera IBM PS/2. Jest to o tyle nowość, ponieważ dotychczas oferowane karty pracowały tylko z serią PC. Na karcie znajduje się transputer T800, taktowany zegarem 20MHz, 4 MB pamięci RAM, o czasie dostępu 100 ns, i dwa złącza RS-422, pozwalające na przesyłanie informacji z szybkością 20MB/s.  
(JM)

**KARTA DO NAPRAW KOMPUTERÓW IBM PC**

Naprawianie komputerów bez dokumentacji i schematów jest w pewnym sensie naszą polską specjalnością. Nie mają takich problemów zachodnie firmy serwisowe. Dodatkowo ich praca może być wspomagana poprzez sprzętowe diagnozowanie uszkodzeń. Za 460 \$ można kupić specjalną kartę do komputera IBM PC razem z zestawem układów pamięci typu ROM, zawierających programy testujące. Producent zapewnia wykrywalność 98 proc. uszkodzeń w czasie mniejszym niż 30 sekund. Karta wyposażona jest w wyświetlacz alfanumeryczny, na którym podawany jest kod uszkodzenia, pozwalający zlokalizować wadliwy element z dokładnością do pojedynczej nogi układu scalonego. Jeśli sprawny jest tor video, pełna diagnoza wyświetlana jest na ekranie komputera. Karta nosi nazwę Postromtem i produkowana jest przez szwajcarską firmę Swiscomp Incorporation.  
(JM)







## SUN NA XT

Sercem systemu SUN 4/110, opisywanego wcześniej przez nas w Bajtku, jest procesor typu SPARC, wykonany w technologii RISC. Ostatnim osiągnięciem firmy Definion International Corporation jest dodatkowa karta do IBM PC wyposażona właśnie w ten układ. Pracuje ona z zegarem 25 MHz, ma 8MB pamięci RAM, a jej moc obliczeniowa sięga 15 MIPS-ów (milion instrukcji na se-

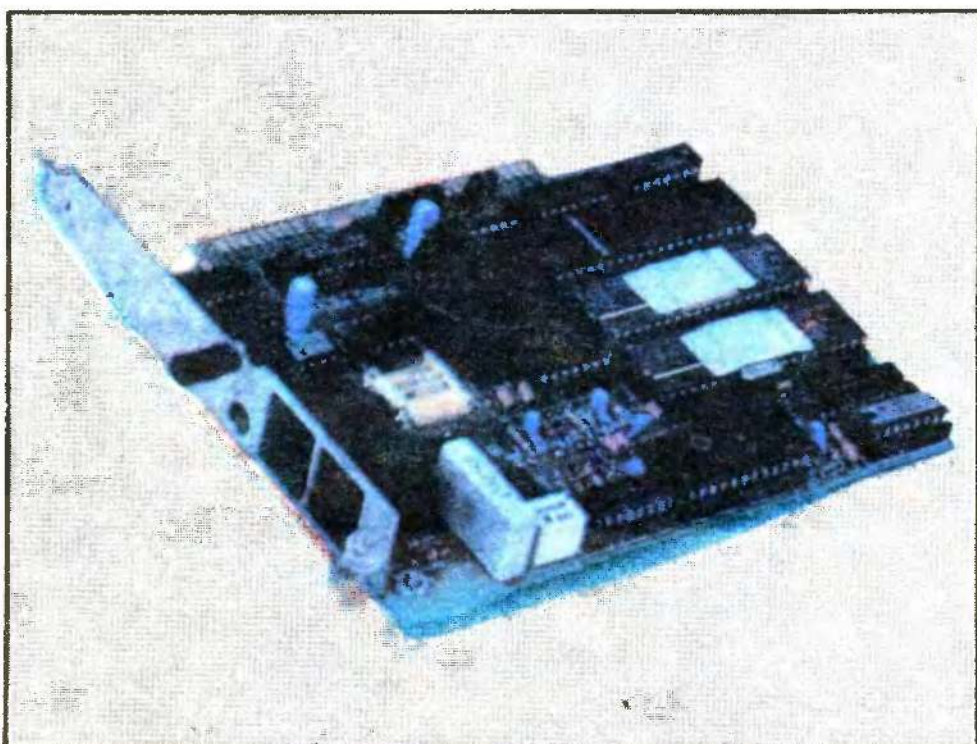
kundę). Podstawową zaletą tej karty, oprócz wielokrotnego przyspieszenia obliczeń, jest udostępnienie na komputerach IBM PC standardu środowiska programistycznego charakterystycznego dla zestawów typu workstation. Wyprodukowane dzięki niej oprogramowanie można uruchomić na komputerach SUN.

(JM)

## BAZY DANYCH W TURBO PASCALU

Można pisać duże bazy danych w dBase III plus, kompilować je Clipperem, można pracować w języku C pod kontrolą wielodostępnego systemu UNIX, ale dlaczego nie skorzystać z Turbo Pascala? Osobom wahającym się można pomóc w wyborze informacji o najnowszej wersji pakietu firmy Turbo Power Software. B-Tree Filer 5.0, zestaw 100 procedur do Turbo Pascala 5.0, umożliwia tworzenie efektywnych, relacyjnych baz danych pracujących w standardzie sieci Novell. Firma dostarcza wersję źródłową pakietu łącznie z pełną dokumentacją za 175 \$. Maksymalny rozmiar bazy 2 miliardy rekordów o zmiennej długości.

(JM)



## MODEM 2400 BODÓW ZA 95\$

Typowe modemy tej klasy kosztują w granicach 150—350 \$. Pierwszym urządzeniem tego typu, dostępnym w cenie poniżej 100 \$ jest modem amerykańskiej firmy Compu Com Corporation. Pozwala on na pracę z szybkością 300, 1200 i 2400 bodów, z automatyczną selekcją rodzaju transmisji. Modem wyposażony w 40-znakowy bufor poleceń, akceptuje w pełni zbiór komend zgodnych z zestawem Hayes-AT. Dostarczany jest razem z programem komunikacyjnym Boyan (wersja D3), ale pozwala również na pracę z Proceomm Plus. To ostatnie oprogramowanie jest konieczne, jeśli chcemy korzystać z trzeciego lub czwartego portu szeregowego. Modem wykonany jest w formie krótkiej karty dodatkowej do komputera typu IBM PC.

(JM)

## Pakiet komunikacyjny DO Z88

Londyńska firma Dateflex Design oferuje do komputera Cambridge Z88 zestaw złożony z modemem i oprogramowania komunikacyjnego. Dostarczany modem ma niewielkie rozmiary (75×55×25 mm), zasilany jest baterią 9V i pozwala na pracę w trybie full-duplex z szybkością 300 i 1200 bodów. Dodatkowe cechy to zgodność ze standardem Hayes i funkcje auto-dial i auto-answer. Oprogramowanie komunikacyjne Z-Term 2.3, dostarczane w postaci wymiennego cartridge-u, umożliwia współpracę z komercyjnymi sieciami informatycznymi i wymianę zbiorów według standardu XMODEM.

(JM)

## SILICON DISK DO IBM PC

Angielska firma Micro Control Systems, z siedzibą w Nottingham, oferuje krzemowy dysk do komputerów typu IBM PC. System jest kombinacją pakietów zawierających układy typu RAM i EPROM. Baterijne podtrzymywanie zawartości pamięci pozwala na przechowywanie do 12 MB danych przez czas 4 miesięcy. Zestaw, produkowany w formie dodatkowych kart do IBM PC, jest szczególnie użyteczny przy pracy w sieciach. Duża szybkość przesyłania danych (300KB/s) pozwala na znaczne przyspieszenie pracy w systemach typu CAD/CAM. Cena zestawu 3000 funtów.

(JM)

## Napędy dysków optycznych

Amerykańska firma Ricoh Corporation wprowadziła na rynek nowy typ napędów dyskowych z możliwością ponownego zapisu na wymiennym nośniku optycznym. Maksymalna pojemność dysku wynosi 652 MB. Pozostałe parametry odpowiadają osiągom uzyskiwanym przez klasyczne napędy typu Winchester. Czas dostępu do informacji jest rzędu 60 ms, szybkość transmisji wynosi 1.4 MB/s. Dane buforowane są przez wewnętrzną pamięć RAM o pojemności 256KB. Współpraca z komputerem odbywa się przy pomocy wbudowanego w napęd sterownika SCSI. Zastosowano standardowy rozmiar dyskietki 5 i 1/4 cala. Cena napędu 3800\$, dyskietki — 250\$.

(JM)

## Sensor obrazu o rozdzielczości 4 milionów punktów

Standardowe układy detekcji obrazu, stosowane w kamerach video powszechnego użytku, zapewniają rozdzielczość 510 na 389 punktów. W styczniu tego roku firma Eastman Kodak doniosła o wyprodukowaniu nowego układu KAF-2000 Full-Frame Imager, o znacznie lepszych parametrach. Pozwala on uzyskać niemal „fotograficzną” rozdzielczość 2048 × 2048, czyli 4194304 punktów. Dzięki dwóm kanałom wyjściowym o częstotliwości 20MHz umożliwia on przesyłanie 10 obrazów na sekundę. Niewielki uzysk przy wytwarzaniu tego układu i problemy technologiczne są powodem jego wysokiej ceny rzędu 50 tys. \$. Z powodu tego ograniczenia przewiduje się, że będzie on stosowany tylko w bardzo specjalistycznym sprzęcie. Na powszechne zastosowanie musimy jeszcze poczekać. Pocięsza trochę fakt, że nieposprawne układy można nabyć za 1000 \$.

(JM)

CO NOWEGO W SZKOLE?





## PRZESUW PIONOWY

**W wielu programach spotykane jest płynne przesuwanie obrazu, które daje znacznie lepszy efekt niż przesuwanie zawartości ekranu o cały wiersz lub kolumnę.**

Uzyskanie tego ciekawego efektu jest — wbrew pozorom — bardzo proste i może być zrealizowane nawet w Basicu. Konieczne jest w tym celu przede wszystkim właściwe przygotowanie pamięci obrazu.

Przy przesuwie pionowym pamięć obrazu musi zawierać liczbę wierszy większą niż odpowiednia dla danego trybu. Zapis treści obrazu w tych wierszach może być wykonany tylko instrukcjami POKE, gdyż system operacyjny nie przewiduje rozbudowania standardowej pamięci obrazu. Odpowiednią wielkość obszaru pamięci obrazu można uzyskać obniżając RAMTOP przed instrukcją GRAPHICS lub umieszczając nieco niżej w pamięci własny program ANTIC-a. W tym drugim przypadku adres początkowy nowego programu ANTIC-a należy umieścić także w rejestrze MEMTOP (741-742 = \$02E5-\$02E6). Zabezpiecza to przed zniszczeniem obrazu przez interpreter Atari Basic. Ponadto trzeba

pamiętać, aby pamięć obrazu nie przekraczała granicy bloku 4 KB pamięci ze względu na konstrukcyjne ograniczenia ANTIC-a.

Przesuwanie zawartości obrazu o cały wiersz można teraz łatwo uzyskać zmieniając tylko adres pamięci obrazu w programie ANTIC-a. Aby uzyskać dokładnie pionowy przesuw (bez odchylenia na boki) należy zawsze zmieniać adres pamięci obrazu o liczbę bajtów niezbędnych dla utworzenia jednego wiersza programu albo o jej wielokrotności.

W trybach, w których wiersz ma wysokość jednej linii ekranu (8—11, 14 i 15), każda taka zmiana adresu powoduje przesunięcie obrazu tylko o jedną linię. Jednak w pozostałych trybach zawartość obrazu przesuwa się skokowo o kilka linii ekranu (odpowiednio do liczby linii ekranu tworzących jeden wiersz trybu). Do uzyskania przesuwu płynnego (o jedną linię ekranu) trzeba w tych trybach wykorzystać specjalny rejestr ANTIC-a (VSCROL) oraz bit uruchamiający przesuw pionowy w rozkazach ANTIC-a (bit 5).

Ustawienie bitu 5 w rozkazie tworzenia wiersza trybu powoduje przesunięcie tego wiersza w górę. Wielkość przesunięcia jest określona przez zawartość rejestru VSCROL. Praktycznie przebiega to następująco:

*Rozkazy ze skasowanym bitem 5 powodują normalne wyświetlanie wiersza. Po napotkaniu pierwszego rozkazu, który ma ustawiony bit 5, opuszczane są w wierszu linie o numerach mniejszych od zawartości rejestru VSCROL (linie są numerowane od zera). Na przykład w trybie 0 zawartość VSCROL równa 3 spowoduje pominięcie linii 0—2, a wyświetlone zostaną tylko linie 3—7. Następne rozkazy z ustawionym bitem 5 dają na ekranie także normalne wiersze. Dopiero pierwszy rozkaz ze skasowanym bitem 5 spowoduje wyświetlenie w wierszu tylko linii o numerach mniejszych od zawartości*

VSCROL. W podanym wyżej przykładzie wyświetlone zostaną tylko linie 0—2.

Ze sposobu działania ANTIC-a wynika wyraźnie, że obraz przesuwany pionowo ma zawsze o jeden wiersz mniej od obrazu stałego. Gdy niezbędne jest utrzymanie niezmięionej wysokości obrazu, to trzeba w programie ANTIC-a dodać jeden rozkaz tworzenia wiersza.

Korzystając z rejestru VSCROL można przesuwać obraz tylko w obrębie jednego wiersza. Aby umożliwić dalszy ruch obrazu, po całkowitym przesunięciu wiersza należy zmienić zawartość VSCROL oraz adres pamięci obrazu. Połączenie tych dwóch sposobów przesuwu umożliwi płynne przemieszczanie obrazu w górę i w dół w zakresie całego wyznaczonego obszaru pamięci.

Zamieszczony obok program demonstruje realizację pionowego przesuwu w trybie GRAPHICS 0. Po ustaleniu trybu i wypełnieniu obrazu liczbami pętla w

wierszu 130 ustawia bit 5 w części rozkazów ANTIC-a. Następnie pokazywane jest dwukrotnie przesuwanie w obrębie jednego wiersza (za pierwszym razem ze zmniejszoną prędkością). Potem program demonstruje połączenie przesuwu delikatnego (o jedną linię ekranu) i przesuwu zgrubnego (o jeden wiersz obrazu).

UWAGA: Widoczne na ekranie zakłócenia są spowodowane powolnym działaniem interpretera Atari Basic w porównaniu z prędkością pracy ANTIC-a oraz dokonywaniem zmian w rejestrach ANTIC-a podczas tworzenia obrazu, a przede wszystkim przekraczaniem granicy bloku 4 KB.

Wojciech Zientara

Fragment książki „Poradnik programisty Atari” przygotowywanej do druku przez SOETO.

```

YZ 100 POKE 82,0:GRAPHICS 0:FOR I=1 TO 92
   :? I,:NEXT I
FN 110 DL=PEEK(560)+256*PEEK(561)
TV 120 POKE 54277,0
ZZ 130 FOR I=DL+10 TO DL+27:POKE I,PEEK(I)
   )+32:NEXT I
GF 140 FOR K=1 TO 10:FOR J=0 TO 7
ST 150 POKE 54277,J:GOSUB 300
EB 160 NEXT J:NEXT K:J=0
HB 170 FOR K=1 TO 20:FOR J=0 TO 7
KH 180 POKE 54277,J
GJ 190 NEXT J:NEXT K
QO 200 POKE 54277,J:J=J+1
VF 210 IF J<8 THEN 200
RA 220 J=0:L=PEEK(DL+4):H=PEEK(DL+5):L=L+
   40
AH 230 IF L>255 THEN L=L-256:H=H+1
HK 240 IF H>255 THEN POKE 82,2:GRAPHICS 0
   :END
SE 250 POKE DL+4,L:POKE DL+5,H:GOTO 200
TJ 300 FOR I=1 TO 100:NEXT I:RETURN
    
```

## TURBO



**System TURBO 2000 służy do przyspieszenia transmisji do i z magnetofonu. Jest on przystosowany do współpracy zarówno z firmowymi, jak i innymi magnetofonami. Nasza redakcja otrzymała do testowania interfejs TURBO 2000 przeznaczony do pracy z dowolnym magnetofonem.**

Autorami systemu TURBO 2000 są: Mirosław Lazarowicz, który zaprojektował interfejs, oraz Wojciech Zabołotny, który całość oprogramował. Całość można nabyć na giełdzie „Bajtki” lub bezpośrednio w firmie (03-922 Warszawa, ul. Międzynarodowa 60/10).

Kompletny zestaw TURBO 2000 składa się z cartridge'a, kasety z programami oraz interfejsu. Interfejs ten jest wbudowywany do wnętrza magnetofonów firmowych. Dla innych magnetofonów dostarczany jest natomiast w oddzielnej obudowie i należy go przyłączyć pomiędzy magnetofon i gniazdo drugiego joysticka (!).

Obsługa systemu jest prosta: wkładamy cartridge do gniazda i włączamy komputer. Po wyświetleniu pytania należy zdecydować się na pracę z Basicem lub bez.

Przy pracy z Basicem możliwe jest jednoczesne korzystanie z zapisu normalnego i przyspieszonego. Można więc odczytać program zapisany standardowo i zapisać go na innej kasiecie w systemie TURBO. Do zapisu w TURBO służy instrukcja SAVE, a do odczytu — LOAD.

Jeśli Basic zostanie odłączony, co w większości przypadków jest konieczne, to również możliwy jest odczyt programów zapisanych obydwoma sposobami, jednakże do kopiowania z zapisu normalnego na TURBO niezbędne jest użycie programu kopiującego z załączonej kasety.

W obydwu przypadkach nazwą urządzenia dla magnetofonu w systemie TURBO jest "D!". Następująca po tym nazwa programu musi być oddzielona jedną spacją, gdyż system ignoruje pierwszy znak nazwy. Podczas odczytu programu w TURBO można umieścić w nazwie gwiazdkę (\*), która oznacza dowolny ciąg znaków, na przykład: D:\* lub D:SP\*. W pierwszym przypadku zostanie odczytany początek pierwszego napotkanego programu, a w drugim początek pierwszego programu, którego nazwa rozpoczyna się od liter "SP". Pełna nazwa znalezionej w taki sposób programu jest wyświetlana na ekranie razem z pytaniem, czy ma on być odczytany. Jeżeli zamiast gwiazdki w nazwie napiszemy znak "#", to znaleziony program będzie odczytany bez pytania.

Transmisja w systemie TURBO jest sygnalizowana dwoma dodatkowymi buczkami. Przy odczycie programu słychać więc trzy dźwięki, a przy zapisie — cztery. Dźwięk podczas transmisji jest natomiast podobny jak w Spectrum. Także wygląd ekranu jest zbliżony — pokazują się na nim migocące paski.

Najważniejszą informacją jest dla użytkownika wielkość przyspieszenia transmisji. Nie można jednak podać dokładnej jej wartości. Wynika to z faktu, że niektóre programy (na przykład „Arkanoid”, „Zybez”, „Super Soccer”, „Ninja”, „Mouse Trap”) wymagają zapisania na taśmie specjalnego programu ładującego, który oczywiście przedłuża czas odczytu. Szacunkowo określiliśmy, że średnie przyspieszenie odczytu jest ośmiokrotne.

Przy pracy w systemie TURBO znacznie rzadziej występują błędy odczytu, co zwiększa nieco szybkość

pracy, a przede wszystkim poprawia jej komfort. W zasadzie możliwe są jedynie błędy numer 140 i 143. Powodem ich wystąpienia jest zwykle wadliwa taśma w kasiecie. Inne błędy oznaczają nieprawidłowe nagranie programu. Przyczyną może być w tym przypadku także brak odpowiedniego programu ładującego na początku. Czasem taka sytuacja sygnalizowana jest jako „zły format pliku”.

Do pracy z systemem TURBO 2000 niezbędne są więc programy pomocnicze. Znajdująca się w komplecie kaset zawiera sześć takich programów. Są to cztery programy kopiujące — „FCopy „145”, „Universal Copy”, „Turbo Long Copy” i „Super Copy” oraz dwa programy ładujące — „L1” („Super Loader”) i „L2”. Korzystając z tego zestawu można skopiować z formatu normalnego na TURBO niemal każdy program. Kłopoty występują tylko w przypadku programów, które zostały zapisane w formacie standardowym, lecz z zabezpieczeniami. Niektórych zabezpieczeń nie potrafi odczytać żaden z programów kopiujących i trzeba się zadowolić normalną wersją.

System TURBO nie może współpracować ze stacją dysków. Skopiowanie programów z dyskietki wymaga więc skopiowania go najpierw na taśmę w normalnej formie, a później dopiero przepisania na format TURBO. Cenną zaletą jest natomiast zgodność z Atari Super Turbo („Bajtek 4/89). Programy zapisane w AST bez kłopotów można odczytać przy użyciu TURBO 2000. Autorzy obydwu systemów zgodnie twierdzą, że nie jest to spowodowane skopiowaniem ich rozwiązania przez konkurencję. Po prostu forma zapisu jest bardzo podobna, pomimo innej konstrukcji systemu.

TURBO 2000 jest dosyć popularne na giełdzie „Bajtki”. Można nawet zakupić programy zapisane w tym systemie, więc użytkownik nie musi ich samodzielnie kopiować. Trudno jednak ocenić, który z systemów (AST i TURBO 2000) jest lepszy. AST jest, co prawda, droższy, lecz programy pomocnicze (kopiujące i ładujące) są w nim umieszczone na cartridge'ie, a więc nie trzeba ich specjalnie odczytywać. Ponadto AST ma lepiej opracowaną instrukcję. Ostatecznego wyboru musi dokonać sam nabywca.

Robert Rogowski  
Wojciech Zientara



# Pierwsze kroki w asemblerze

# JĘZYK MASZYNOWY (2)

W dalszym ciągu nauki programowania w języku maszynowym zajmiemy się odcinkami 4—6 kursu publikowanego w klanie Commodore. Zamieszczone są one w kolejnych „Bajtkach” od numeru 5/89 do 7/89.

W czwartym odcinku znajduje się rozwiązanie zadania z poprzedniego numeru „Bajtki” oraz opis działania rozkazów operacji logicznych AND, ORA i EOR. Drugie zagadnienie rozwiązuje się w Atari tak samo jak w Commodore. Większy kłopot sprawi nam pierwsza część artykułu, do której odnoszą się trzy listingi.

Pierwszy listing pokazuje program odczytujący znak z klawiatury i umieszczający na ekranie tekst. Powinien on być w zasadzie identyczny, trzeba jedynie zmienić adresy procedur systemu operacyjnego. Zamiast FFD2 (CHROUT) należy wpisać F2B0 (EOUTCH), zaś procedurę GETIN (adres FFE4) trzeba zastąpić znajdującą się w Atari procedurą EGETCH (adres F24A), która jednak jest nieco inna — wpisany znak musi być zatwierdzony przez RETURN. Ale kłopoty zaczynają się dopiero teraz.

Problem polega na tym, że procedura EOUTCH zmienia zawartość obu rejestrów indeksów X i Y. Aby zapewnić poprawne działanie programu, trzeba więc przechować te dwie wartości. Do tego celu można wykorzystać dwie komórki znajdujące się w pozostawionym specjalnie do takich zastosowań obszarze pamięci na stronie zerowej — CB i CC.

Przed wywołaniem EOUTCH umieścimy w nich zawartości X i Y, a później odczytamy je ponownie. Zawartość Y jest przy tym wykorzystywana tylko w drugiej pętli i w pierwszej nie trzeba jej zapisywać. Treść poprawionego programu jest przedstawiona na listingu.

Wykonanie tych zmian spowoduje przedłużenie programu, czyli przesunie nieco w pamięci jego dane. Uwzględniają to zmiany wprowadzone do operandów rozkazów LD nnnn,X. Dwie kolejne zmiany są drugorzędne. Pierwsza to znak czyszczenia ekranu, który ma w Atari kod 7D, zaś w Commodore — 93. Druga wynika z treści wyświetlanego napisu. Nie można zgodzić się na taką „herezję” i trzeba napis zmienić, co spowoduje także zmianę jego długości. Liczbę 27 musimy więc zastąpić przez 23. Zmiana samego tekstu jest prosta i jej opis zostanie pominięty. Trzeba tylko pamiętać, że ostatnim znakiem musi być 9B zamiast 8D (znak końca wiersza — RETURN).

Poprawiony program można teraz zapisać w instrukcjach DATA, pamiętając przy tym o podanych w poprzednim odcinku różnicach między SYS i USR. Dodatkowym utrudnieniem jest niedostępność liczb szesnastkowych w Atari Basic. Wartości w DATA muszą więc być przeliczone na system dziesiętny. Końcowy efekt przedstawiają trzy listingi, które odpowiadają listingom z „Bajtki” 5/89.

Kolejny, piąty odcinek „Języka maszynowego” w klanie Commodore („Bajtek” 6/89) jest poświęcony operacjom dodawania i odejmowania. Można przenieść go do klanu Atari niemal bez żadnych zmian. Konieczne jest tylko pominięcie fragmentu dotyczącego obrazu w wysokiej rozdzielczości dla C-64. Organizacja obrazu jest w Atari całkowicie odmienna, a różne tryby graficzne uzyskuje się z Basica instrukcją GRAPHICS, zaś w języku maszynowym poprzez otwarcie kanału transmisji dla urządzenia "S".

```

270F 68          PLA
2710 A2 00      LDX #$00
2712 86 CB      STX $CB
2714 BD 4E 27   LDA $274E,X
2717 20 B0 F2   JSR $F2B0
271A A6 CB      LDX $CB
271C E8         INX
271D E0 13      CPX #$13
271F D0 F1      BNE $2712
2721 20 4A F2   JSR $F24A
2724 C9 31      CMP #$31
2726 90 F9      BCC $2721
2728 C9 3A      CMP #$3A
272A B0 F5      BCS $2721
272C 85 CC      STA $CC
272E A9 7D      LDA #$7D
2730 20 B0 F2   JSR $F2B0
2733 A2 00      LDX #$00
2735 86 CB      STX $CB
2737 BD 61 27   LDA $2761,X
273A 20 B0 F2   JSR $F2B0
273D A6 CB      LDX $CB
273F E8         INX
2740 E0 23      CPX #$23
2742 D0 F1      BNE $2737
2744 A4 CC      LDY $CC
2746 88         DEY
2747 84 CC      STY $CC
2749 C0 30      CPY #$30
274B D0 E6      BNE $2733
274D 00         BRK
    
```

```

274E 7D 50 4F 44 41 4A 20 43 .PDAJ C
275E 59 46 52 45 20 28 31 2D YFRE (1-
275E 39 29 9B 4E 41 4A 4C 45 9).NAJLE
2766 50 53 5A 59 20 57 20 42 PSZY W B
276E 41 4A 54 4B 55 20 4A 45 AJTKU JE
2776 53 54 20 4B 4C 41 4E 20 ST KLAN
277E 41 54 41 52 49 9B 00 00 ATARI...
    
```

```

BF 10 DATA 104,162,0,134,203,189,78,39
VQ 20 DATA 32,176,242,166,203,232,224,19
CG 30 DATA 208,241,32,74,242,201,49,144
BP 40 DATA 249,201,58,176,245,133,204
XW 50 DATA 169,125,32,176,242,162,0,134
WZ 60 DATA 203,189,97,39,32,176,242,166
XB 70 DATA 203,232,224,35,208,241,164
DB 80 DATA 204,136,132,204,192,48,208
YS 90 DATA 230,0,125,80,79,68,65,74,32
DZ 100 DATA 67,89,70,82,69,32,40,49,45
HT 110 DATA 57,41,155,78,65,74,76,69,80
KN 120 DATA 83,90,89,32,87,32,66,65,74
IN 130 DATA 84,75,85,32,74,69,83,84,32
MV 140 DATA 75,76,65,78,32,65,84,65,82
KW 150 DATA 73,155
UR 160 FOR J=9999 TO 10115:READ Q:POKE J,
Q:NEXT J
    
```

Podobnie następny odcinek („Bajtek 7/89), w którym podane są sposoby mnożenia i dzielenia, można w całości polecić użytkownikom Atari. Pokazane tam programy w języku maszynowym będą działały poprawnie również na tym komputerze. Oczywiście, jeżeli zechcemy zapisać je w Basicu (w instrukcjach DATA), to niezbędne są poprawki opisane w poprzednim odcinku, a pokazane na początku tego.

Na zakończenie mam jeszcze dla Czytelników trochę nieprzyjemną wiadomość. Ze względów technicznych kolejny odcinek poprawek do kursu języka maszynowego ukaże się dopiero w listopadowym numerze „Bajtki”.

Wojciech Zientara

# JBW ASSEMBLER

Do licznej już rodziny asemblerów przeznaczonych dla osmiobitowych komputerów Atari przybył nowy produkt o nazwie JBW ASSEMBLER. Powstał on całkowicie w Polsce i nie jest adaptacją, przeróbką ani naśladownictwem żadnego programu zagranicznego. Tylko ta jedna cecha wystarczy, aby zwrócić na niego uwagę, choć posiada on jeszcze wiele innych zalet.

Przyzwyczajony do polskich realiów użytkownik dozna szoku już przy pierwszym zetknięciu się z tym programem. Nie otrzymuje bowiem „gotę” dyskietki, lecz ładnie oprawioną broszurę z dyskietką umieszczoną na końcu, w specjalnej kopercie. Broszura zawiera natomiast instrukcję obsługi programu, która sama w sobie jest cennym nabytkiem.

Na szesnastu stronach tej instrukcji użytkownik znajdzie wiele cennych informacji i to nie tylko o nabytym programie. Jej początek zajmują opisy działania procesora 6502, jego trybów adresowania oraz wszystkich rozkazów. Jest to więc bardzo krótki podręcznik języka maszynowego.

Właściwa instrukcja zajmuje następujące rozdziały. Kolejno opisane są: składnia wierszy wpisywanego programu, działanie dyrektyw asemblera oraz znaczenie komunikatów informujących o błędach podczas asemblacji. Dalej podane są wszystkie dostępne funkcje programu wraz ze szczegółowymi objaśnieniami. Całość napisana jest poprawną polszczyzną i w bardzo przystępnej formie. Lekturę tej instrukcji należałoby polecić autorom i wydawcom większości „instrukcji” dostępnych na giełdzie i w różnych firmach komputerowych.

Największe zdumienie budzi jednak koniec instrukcji. Znajduje się tam kupon rejestracyjny, który trzeba przesłać do producenta. Zapewnia to wszystkim LEGALNYM nabywcom prawo do nieodpłatnych informacji i porad dotyczących programu oraz do otrzymania kolejnych wersji programu po obniżonej cenie. Takim zwyczajom hołdowali dotychczas tylko „wredni” kapitaliści.

JBW ASSEMBLER składa się z dwóch połączonych programów — edytora tekstu i programu aplikacyjnego. Są one sterowane poprzez menu wyświetlone w dolnej części ekranu, co pozwala na proste operowanie programem bez ciągłego odwoływania się do instrukcji. W menu są ponadto zawarte funkcje umożliwiające komunikację ze stacją dysków.

Edytor umożliwia nie tylko pisanie programów w języku maszynowym, ale także pisanie dowolnych tekstów w dowolnej formie. Podczas pisania nie jest bowiem sprawdzana składnia wprowadzanych instrukcji. Dla osób przyzwyczajonych do pracy w Basicu lub MAC/65 może to stanowić pewne utrudnienie, lecz dzięki temu edytor jest narzędziem bardziej uniwersalnym.

Program aplikacyjny (właściwy asembler) pozwala na asemblację programu w języku maszynowym oraz na jego testowanie. Wykonanie asemblacji ułatwia sześć dodatkowych dyrektyw, które służą między innymi do wyboru wariantu asemblacji, umieszczenia w programie wynikowym różnych danych, dołączenia zewnętrznego pliku z procedurami itd.

Wszystkie opisane wyżej możliwości zostały „upakowane” w obszar o pojemności 8 KB. Pozostała dla użytkownika pamięć może pomieścić do 2000 wierszy programu wraz z jego kodem wynikowym. Pomimo konieczności dwukrotnego przejścia programu szybkość asemblacji dochodzi do 120 wierszy na sekundę (jeśli listing nie jest wyświetlany na ekranie).

Podczas dwutygodniowego testowania nie znalazłem w tym programie żadnych poważnych wad. Niepoważne zaś wynikają raczej z moich przyzwyczajzeń, a nie z błędnych rozwiązań. Naprawdę brakuje mi tylko dodatkowych plików, które zawierałyby najczęściej stosowane fragmenty programów. Myślę, że w jednej z następnych wersji pojawi się, na przykład, biblioteka standardowych operacji wejścia/wyjścia lub innych często używanych funkcji.

Marek Zachar

Autor: Janusz B. Wiśniewski  
 Producent: Harvest, Sp. z o.o., 01-432 Warszawa,  
 ul. Kędzierskiego 8/55  
 Wymagany sprzęt: Atari XL/XE, stacja dysków.





Profesjonalny program wydawniczy — "Desktop Publishing" na komputerze Amiga 2000.

**Tradycyjnie już, pierwszy dzień wystawy firmy Commodore Business Machines na targach w Hanowerze rozpoczął się od konferencji prasowej. W tym roku przedstawiciele Commodore sprawiali wrażenie bardzo zadowolonych z wyników handlowych osiągniętych w ciągu minionego roku. Stwierdzili również, że tak pod względem asortymentu wyrobów, jak i organizacyjnie przygotowani są na nadchodzące lata dziewięćdziesiąte.**

Jak wiadomo do końca 1992 roku zakończy się ostatni etap integracji Europejskiego Wspólnego Rynku co stawia nowe zadania i problemy przed konkurującymi na tym rynku firmami (nie tylko komputerowymi). Wraz z zanikiem granic, żadna z firm nie będzie posiadała specjalnych preferencji odnośnie terytorium, na którym zawierać będzie transakcje handlowe. Nowe struktury organizacyjne Commodore mają zapewnić dostępność tych komputerów oraz oprogramowania i akcesoriów w każdym zakątku Europy (oczywiście z instrukcjami i wersjami komputerów w odpowiednim języku).

Oferta handlowa składająca się z komputerów Amiga 2000 i MS-DOS PC dała firmie 33% udziału w RFN ugruntowując tym samym pozycję zdecydowanego lidera na tym rynku. Podobne wyniki firma uzyskała i w innych krajach europejskich (m.in. w Wielkiej Brytanii). Najlepiej jednak za siebie mówią dane o liczbie sprzedanych modeli komputerów w ciągu 6 miesięcy (1 lipiec — 31 grudnia) odpowiednio 1987 i 1988 roku i tak:

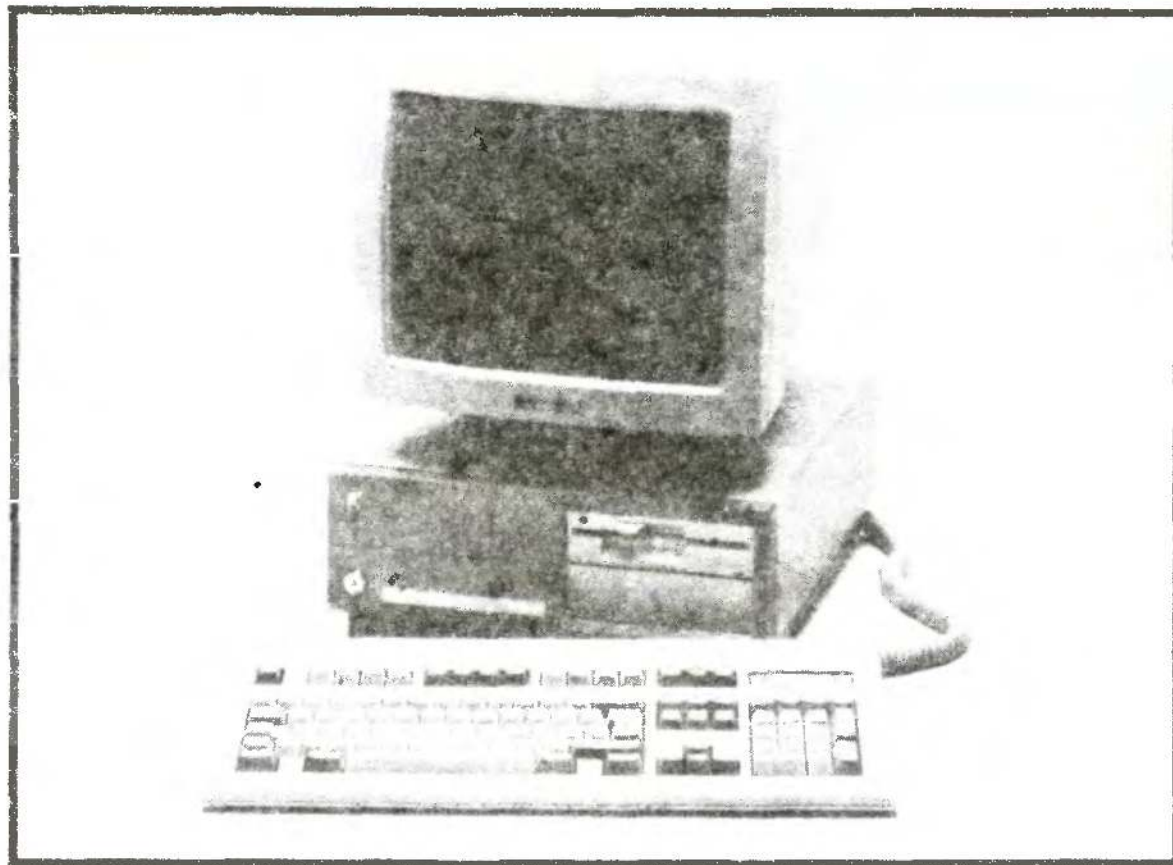
Komputer	1988	1987	Zmiana [%]
MS-DOS PC i A2000	53460	39440	+39.1
A2000	16000	9700	+64.9
C-64, C-128D, A500	326440	188000	+73.6
A500	81440	60000	+35.7
Wszystkie Amigi	94440	69700	+39.8

Na rynku komputerów kompatybilnych z IBM daje się zaobserwować nowy trend. Oprócz typowych zastosowań komputerów o dużych możliwościach w zarządzaniu, przemyśle, handlu rośnie liczba prywatnych użytkowników, którym w domu nie wystarczy już PC-XT. Do pracy w domu tani model AT o znacznie większych możliwościach jest uznawany za coraz powszechniejszy standard. Prognozy firm zajmujących się marketingiem (np. Dataquest z RFN) potwierdzają te tendencje twierdząc, że obecnie AT stanie się w domu tym, czym do tej pory było XT. Dowodem na wyczuwanie tego trendu przez Commodore jest PC-40 III. Jest to właśnie AT o doskonałym wyważeniu stosunku możliwości do ceny. Komputer ten posiada mniejszą od swoich poprzedników obudowę zawierającą więcej układów dających większe możliwości eksploatacyjne. Sercem jest procesor Intel 80286 pracujący z częstotliwościami 6, 8, 12 MHz (przełączalne). Istnieje miejsce na koprocesor 80287. Płyta główna zawiera 1MB pamięci RAM

rozszerzalną do 8MB. Oprócz tego na płycie znajduje się kontroler obrazu dysponujący następującymi trybami pracy: MDA, MGA, CGA, Hercules, EGA oraz rozszerzona VGA. Pamięć masowa to stacja dyskietek 5.25" (360KB/1.2MB) i opcjonalnie 3.5" (720KB/1.44MB). Wbudowany jest szybki twardy dysk o pojemności 40MB. Pojawił się również nowy interfejs, oprócz standardowego równoległego i szeregowego, przeznaczony do podłączenia myszy. System dostarczany jest standardowo z 14" monochromatycznym, analogowym monitorem.

Jednym z zadań stawianych przed producentami komputerów i układów do nich jest przetwarzanie coraz większych ilości danych w coraz krótszym czasie, budowa procesorów taktowanych zegarami o coraz wyższych częstotliwościach itd. W rzeczywistości istnieje jednak pewien ekonomiczny próg powodujący, że krok w kierunku procesora 64-bitowego (już istniejącego) w zastosowaniach niemilitarnych nie będzie wykonany tak szybko, choć w komputerach szybko jest czasami kwestią roku. Alternatywnym rozwiązaniem dla procesora o coraz dłuższym słowie maszynowym jest transputer. Jak wiadomo transputer umożliwia podzielenie wykonywanego zadania na fragmenty wykonywane następnie równoległe przez kilka procesorów i innych specjalizowanych układów. Transputer firmy Commodore był już prezentowany w Hanowerze w ubiegłym roku. Pojawił się również w tym roku na wnętrzu niektórych wersji A2000 i A2500. Szybkość przetwarzania pojedynczego transputera rzędu 10MIPS (milionów instrukcji na sekundę) powoduje, że komputer z tym układem jest 10x szybszy od tradycyjnego AT. Ten układ istnieje już w wersjach z lub bez funkcji zmiennoobrotowych. Duża szybkość przetwarzania jest wynikiem budowy tego transputera. Każdy jego układ, oprócz normalnie stosowanej magistrali (szyny), posiada 4 superszybkie szeregowo połączenia. Jeżeli pierwszy układ transputera stosowany jest jako macierzysty to dodatkowo można do niego podłączyć 4 nowe transputery i dalej do każdego z transputerów po 3 następne itd. W ten sposób tworzy się system transputerowy pracujący w czasie rzeczywistym, umożliwiający wykonywanie współbieżne kilku programów z

Commodore PC PC 60.



Nowy AT Commodore — PC 40-III.

dużymi szybkościami. Transputer do Amigi sprzedawany jest jako karta pracująca pod kontrolą systemu operacyjnego Helios firmy Perihelion. Wszystkie operacje WE/WY obsługiwane są przez Amigę.

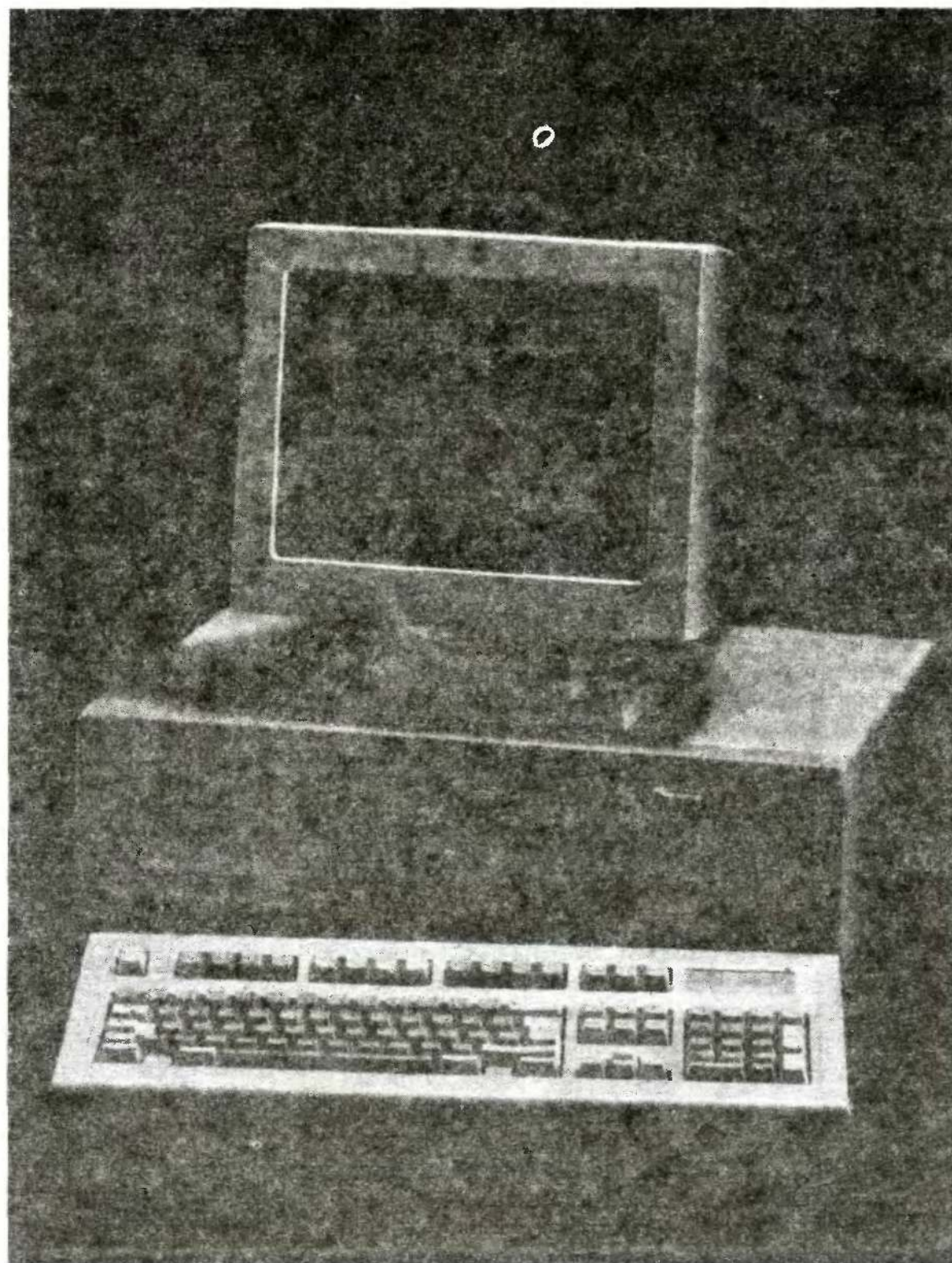
W ciągu minionego roku ugruntowała się pozycja Amigi o czym świadczy nie tylko liczba sprzedanych komputerów. Filozofia systemu o otwartej architekturze przyjęła się w Amidze jako nowy standard. Jako wyjście naprzeciw potrzebom użytkowników, Commodore będzie oferować w przyszłości różne pełne, profesjonalnie ukierunkowane rozszerzenia systemu. Przykładem obecnie może być A2000 DTP (DeskTop Publishing) czy A500 Desktop Video.

W skład zestawu A2000 DTP wchodzi A2000 z 3MB RAM, napęd dyskietek 3.5" 880KB, twardy dysk 40MB, kolorowy skaner Sharp JX300 z rozdzielczością 300dpi (punktów na cal), drukarka laserowa NEC LC890 oraz wymagane

oprogramowanie do DTP z możliwością przenoszenia zbiorów z Deluxe Paint, Aegis Images itp. Zestaw taki umożliwia pełne redagowanie kolorowego pisma. Jego dane wyjściowe mogą być bezpośrednio użyte w profesjonalnych systemach składająco-naświetlających stosowanych w drukarniach np. Linotron 200R.

A500 Desktop Video umożliwia obróbkę filmu video w oparciu o możliwości komputera. Daje to możliwość dodawania napisów, tytułów itp. Możliwa jest również ingerencja w sam obraz wideo: dodawanie trójwymiarowej animacji, zmiana lub usuwanie barw, cieniowanie obrazu, wyświetlanie obrazu w negatywie itd. Cena takiej przyjemności 1595DM.

Na CeBIT zaprezentowana została również Amiga 2500UX — wielozadaniowe, potężne rozwinięcie filozofii systemu A2000. Nowa Amiga pracuje w oparciu o UNIX system V. Ten system operacyjny





został opracowany przez firmę AT&T i zdobywa szybko popularność. Pierwotnie ten system operacyjny opracowany był dla superkomputerów, w których miała być zagwarantowana błyskawiczna obsługa pamięci masowej i wysoka wydajność komputera. Wyglądem A2500UX przypomina A2000. Różni się natomiast wbudowanym twardym dyskiem o pojemności 80MB i czasie dostępu 19ms; posiada wbudowany streamer taśmowy 150MB umożliwiający zgrywanie zbiorów w regularnych odstępach czasu. A2500UX starowana jest procesorem Motorola 68020 (14.2MHz). Do zarządzania pamięcią służy procesor 68851, a jako koprocesor 68881 (do 25MHz). Płyta główna posiada 3MB RAM rozszerzalne do 9MB. Wymienione cechy powodują 4-krotny wzrost możliwości A2500 w porównaniu z A2000. A 2500 UX umożliwia zarówno pracę pod Amiga DOS jak i Amiga Unix. Oba systemy ładowania są z twardego dysku. Twardy dysk może być podzielony na odrębne partycje przeznaczone dla otoczenia programowego pracującego pod jednym z wymienionych systemów.

Nas jednak bardziej interesują nowe peryferia do A500, której ze wszystkich Amig jest u nas na rynku najwięcej. Otóż pojawił się ostatnio w postaci modułu o nazwie A590 twardy dysk z rozszerzeniem pamięci. A590 zawiera 20MB twardy dysk o czasie dostępu 80ms z funkcją autoboot realizowaną każdorazowo przy włączaniu komputera. A 590 daje możliwość rozbudowania pamięci o 2MB krokami co 0.5MB. Moduł ten zawiera specjalny interfejs (SCSI) dla podłączania innych twardego dysków, dysków optycznych... Przesyłanie danych jest optymalizowane przez kontroler DMA specjalnie zaprojektowany dla A590. Stanowi on ważny pomost łączący Motorola 68000 z twardym dyskiem umożliwiając przesyłanie danych z szybkością 2.4MB/s co jest 2-3x szybciej niż operacje WE/WY kontrolowane programowo. Kontroler DMA jest w stanie zarządzać przesyłaniem danych z pamięci do pamięci w dowolnym 16MB obszarze adresowym widzianym przez M68000. Jest to jedna z cech architektury Amigi polegająca na odciążeniu CPU przez specjalne koprocesory.

Ponadto na targach pojawiła się szybka kolorowa drukarka do Amigi i systemów AT pod nazwą Commodore MPS1224C. Drukarka ma zastąpić poprzednie modele 2000 i 2010. Jest to 24 igłowa drukarka mozaikowa drukująca tekst z szybkością 220 cps (znaków na sekundę) lub 72 cps w NLO. Kompatybilna jest z drukarką Epson LQ-1500 i IBM 24XL. Posiada wbudowane złącze równoległe i szeregowo. Parametry drukarki ustawiane są za pomocą menu (nie ma już przetaczników na konsoli) Szerokość stosowanego papieru 102-406 mm. Zaletą tej drukarki jest możliwość szybkiej zmiany kolorów i papieru.

Na koniec tej wyliczanki wspomnę jeszcze o 3 wysławionych komputerach zgodnych z IBM. Są to PC-10 III / PC-20 III (zgodne z XT) i PC-60 (AT). Dwa pierwsze pracują w oparciu o Intel 8088-2 z częstotliwościami 4.77, 7.16, 9.54 MHz, posiadają 640 KB na płycie głównej. Pamięć masowa to:

PC-10 III dwie stacje 5.25" po 360 KB  
PC-20 III jedna stacja 5.25" 360 KB i 3.5" dysk twardy 20 MB  
PC-60 pracuje w oparciu o procesor 80386 (8/16MHz). Posiada 2.5 MB pamięci standardowo rozszerzalne do 16MB, jedną stację dyskietek 5.25" 1.2MB i twardy dysk 40MB.

Użytkownicy komputerów Commodore zastanawiają się pewnie czy ich firma wycofała się z produkcji komputerów 8-bitowych. Otóż oczywiście nie. Świadczą o tym liczby z początku artykułu. Również w tym roku jedynym wystawionym komputerem 8-bitowym był C-64. Firma kontynuuje produkcję C-64 i C-128D i na razie nie myśli o jej zaprzestaniu gdyż w dobie 32-bitowych gigantów te maleństwa sprzedają się nadal znakomicie.

*Dominik Falkowski*



## Dzisiejsza lekcja zawierać będzie jedynie rozwiązanie zadania z ostatniego wykładu, a polegającego na mnożeniu dwóch liczb ośmiobitowych dających wynik 16-bitowy.

W skład podanego programu wchodzi procedura (\$271A-\$273B) dokonująca wspomnianego mnożenia. Można ją wykorzystać jako podprogram w pisanych przez nas programach. Wywołujemy ją wtedy przez JSR \$271A. Przed jej wywołaniem należy zadbać tylko o to aby w komórkach o adresach \$273C i \$273D umieszczone były liczby, które zamierzamy wymnożyć. Po powrocie z procedury komórki te zawierać będą 16-bitowy wynik mnożenia i tak w komórce o adresie \$273C znajdzie się starszy bajt odpowiedzi, a w komórce \$273D młodszy bajt odpowiedzi. Program dla własnych celów wykorzystuje ponadto jeszcze dwie komórki pamięci o adresach \$273E i \$273F. Wspomniane powyżej 4 komórki na listingu zastąpione są instrukcjami NOP (No Operation). Instrukcje te w trakcie wykonywania programu są ignorowane (nie powodują żadnych działań, choć jak powiedziałyby matematyk powodują niedziałanie). Nie jest to instrukcja którą umieszczamy w programach za które płacą nam od bajta, natomiast jeżeli popełniliśmy błąd i chcemy z programu usunąć jedną instrukcję a nie chcemy przepisywać programu, zmieniać adresowania to po prostu w miejscu instrukcji przewidzianej do usunięcia wpisujemy NOP.

Wspomniane 4 komórki przed uruchomieniem programu nie zawierają żadnych konkretnych wartości. Należy do nich wpisać liczby przewidziane do wymnożenia a po wykonaniu programu odczytać wynik. Komórki te i przechowywane przez nie wartości możemy obejrzeć wykonując instrukcje w monitorze M 273C 273F. Podprogram (procedura) pobiera liczby przewidziane do mnożenia z rejestrów .A i .X, wymnaża je, a wynik pozostawia w tych samych rejestrach. Jeżeli nie potrzebujecie aby wynik oraz dane znajdowały się w konkretnych komórkach, mnożenia wykonujecie na bieżąco, to już sami możecie usunąć z programu stosowne instrukcje LDX, LDA, STX, STA. W ten sposób w trakcie wykonywania określonego programu procedura może być wykorzystana parokrotnie. Należy zadbać tylko o to aby nie stracić kolejnych wyników mnożenia.

Program ten może być dowodem na to jak efektywne programy można tworzyć w języku maszynowym. Nie oznacza to, że Wasze programy od razu muszą być takie same, ale z upływem czasu, w miarę uczenia się i ćwiczenia osiągniecie wymaganą wprawę.

Jak działa sama procedura mnożąca? Sumuje ona 8 liczb. Odbywa się to w pętli, przy czym licznikiem jest rejestr .X do którego na początku ładowania jest wartość #\$08. Pod koniec każdej pętli wartość licznika zmniejszana jest o jeden, a wynik porównywany jest z zerem. Mnożenie odbywa się w podwójnym rejestrze: starszy bajt znajduje się w rejestrze .A, a młodszy w komórce o adresie \$273F. Lewe 7+k bitów tego podwójnego rejestru wykorzystuje się gdy dodajemy k-ty składnik (w tym momencie w rejestrze .X jest 9-k). Pozostała część \$273F zawiera wartość, która na wstępie pobrana była z rejestru .A. Dokonujemy mnożenia przez tę wielkość, przy czym za

każdym obiegiem pętli przemieszczamy bity w bajcie z prawej strony w lewą. Za każdym razem gdy bit ma wartość 1 my dodajemy liczbę, która na wstępie zawarta była w rejestrze .X. Jeżeli bit ma wartość 0 dodawać liczby nie musimy. Wynika to z oczywistej zależności, że  $X \times 1 = X$  a  $X \times 0 = 0$ . Podwójne oznaczenie komórki o adresie \$273F konieczne jest dlatego, że bity w obu wielkościach przemieszczane są w prawo, ale podczas gdy jedna z nich narasta to druga ulega zmniejszeniu. W rezultacie akumulator .A i komórka \$273F zawierają pełny 16-bitowy wynik. Po tym do rejestru .X kopiujemy zawartość \$273F i otrzymujemy wynik we wspomnianych na wstępie rejestrach .A i .X.

Program rozpoczyna się od ustawienia wstępnych wartości i tak zawartość wstępnego rejestru .X zapamiętywana jest w komórce \$273E, .A w \$273F jak również zerowany jest znacznik przeniesienia. Zauważcie, że gdy dochodzimy po raz pierwszy do wykonywania instrukcji BCC \$272B (linia \$2725) to znacznik przeniesienia zawiera już prawy bit (najmłodszy) z rejestru .A. To by było na tyle. Z pozostałymi instrukcjami już się zapoznaliście wcześniej więc rozszyfrowanie pracy programu nie powinno Wam nastęrczać problemów. Zachowajcie tylko cierpliwość a zrozumiecie wszystko. Co powiecie na program dzielący dwie dowolne liczby? Myślę, że macie już podstawy do napisania takiego programu własnoręcznie. W najbliższych wykładach czeka nas potężna dawka teorii gdyż przejdziemy do trybów adresowania. Teorię jednak przeplatać będziemy przykładami np. wspomniane niedawno sortowanie w języku maszynowym. Program na sortowanie ostatecznie przekona niedowiarków, że „maszynówka” jest najszybsza.

*Dominik Falkowski*

MONITOR	PC	SR	AC	XR	YR	SP
!	B000	00	00	00	00	FB
F2710	AD	3C	27	LDA	\$273C	
F2713	AE	3D	27	LDX	\$273D	
F2716	20	1A	27	JSR	\$271A	
F2719	00			BRK		
F271A	8E	3E	27	STX	\$273E	
F271D	4A			LSR		
F271E	8D	3F	27	STA	\$273F	
F2721	A2	08		LDX	#\$08	
F2723	A9	00		LDA	#\$00	
F2725	90	04		BCC	\$272B	
F2727	1B			CLC		
F2728	6D	3E	27	ADC	\$273E	
F272B	6A			ROR		
F272C	6E	3F	27	ROR	\$273F	
F272F	CA			DEX		
F2730	D0	F3		BNE	\$2725	
F2732	AE	3F	27	LDX	\$273F	
F2735	8D	3C	27	STA	\$273C	
F2738	8E	3D	27	STX	\$273D	
F273B	60			RTS		
F273C	EA			NOP		
F273D	EA			NOP		
F273E	EA			NOP		
F273F	EA			NOP		



# OXFORD

## P a s c a l

**Niniejszy artykuł oferuje wszystkim tym, którzy zainteresowani nauką i programowaniem w języku Pascal, nie potrafili do tej pory uruchomić czy w pełni wykorzystać najbardziej popularnej implementacji standardu Pascal-a na komputery Commodore 64 jaką jest OXFORD Pascal.**

OXFORD Pascal posiada dwa tryby pracy. W pierwszym i najprostszym trybie kompilator języka oraz tekst źródłowy koegzystują jednocześnie w pamięci komputera. Oznacza to że automatycznie mniej miejsca pozostaje na kod wynikowy, czyli na skomplikowany program. Tryb ten jest idealny dla nauki języka oraz pisanie krótkich programów nie wymagających współpracy z dyskiem. Większość instrukcji języka Pascal jest dostępna w tym trybie. Dla bardziej złożonych i dłuższych programów (nawet do 6000 linii) istnieje tzw. tryb dyskowy z kompilatorem znajdującym się w czasie edycji na dysku. Kompilacji podlega tekst źródłowy programu znajdujący się również na dysku. Nawet w takim wypadku, jeżeli okaże się że program jest jeszcze zbyt duży, istnieje możliwość rozbicia programu na mniejsze „cegiełki”, oddzielnej ich kompilacji oraz włączenia ich do głównego programu jako gotowe procedury.

Na rynku występują dwie wersje omawianego programu: kasetowa i dyskowa. Różnica polega na tym, że wersja kasetowa ogranicza się jedynie do pierwszego trybu pracy, natomiast wersja dyskowa daje pełną możliwość wykorzystania programu jako narzędzia programistycznego.

Słowa kluczowe pokrywają się w 100 proc. ze standardem i nie mogą być przededefiniowane. Standardowe identyfikatory również są takie same jak w standardzie, rozbudowane o szereg nowych opisanych poniżej. Identyfikatory te mogą być przededefiniowane, lecz wtedy przestają być ich pierwotne znaczenie. Czytelników zainteresowanych wyszczególnieniem słów kluczowych i standardowych identyfikatorów języka Pascal odsyłam do książki *M. Iglewski; J. Madey; S. Matwin* pt. „Pascal” wydanej przez WNT w 1986 roku.

**UWAGA!!!** W OXFORD Pascal-u identyfikatory różniące się w oparciu o pierwsze 8 znaków, tzn., że identyfikatory: DługieSłowo 1 i Długiesłowo2 będą uznane za jeden i ten sam identyfikator. Kompilator nie rozróżnia małych i dużych liter tzn. IDENTYFIKATOR i identyfikator uznane będą za tę samą zmienną. Dopuszcza się używanie znaku podkreślenia (klawisz C= i „AT”) na równi z każdym innym znakiem alfanumerycznym.

### INSTRUKCJE EDYTORA

W odróżnieniu od standardu Pascala w OXFORD Pascal-u linie programu są numerowane. Numerowanie linii ma znaczenie wyłącznie dla edytora: linie zaczynające się od numeru traktowane są jako nowe linie i umieszczane są w programie stosownie do numeru linii (tak samo jak w BASIC-u wprowadzenie linii z numerem 5 po linii 10 spowoduje przy listowaniu, że linie te pojawią się w kolejności rosnących numerów).

**Auto** — uruchamia lub wyłącza automatyczną numerację linii. Auto 20 (return) powoduje numerowanie linii ze skokiem 20; auto (return) wyłącza automatyczną numerację linii. Standardowo przyjęte jest auto 10.

**List** — znaczenie i sposób użytkowania taki sam jak w edytorze BASIC-a. Naciśnięcie dowolnego klawisza podczas listowania spowoduje zatrzymanie listowania do

chwili naciśnięcia dowolnego klawisza po raz drugi. Klawisz STOP przerywa listowanie.

**Upper, Lower** — powoduje przełączenie klawiatury do pracy odpowiednio: wyłącznie z dużymi literami (znaczniki graficzne są nadal dostępne) lub z małymi i dużymi literami. Standardowo przyjęty jest tryb pracy z małymi i dużymi literami (lower). Zmiana trybu pracy jest równoznaczna z naciśnięciem jednocześnie klawiszy SHIFT i C=

**Basic** — powrót do BASIC-a.

**New** — kasuje program znajdujący się w pamięci.

**Disk** — uruchamia tryb dyskowy

**Resident** — powoduje ponowne załadowanie kompilatora do pamięci komputera (pierwszy tryb).

**Number** — instrukcja ta służy przenieściowi linii np. number 100,500,25 spowoduje przenieście programu istniejącego w pamięci począwszy od linii 100 (stary numer) z nowym numerem linii 500 i skokiem 25. Niestety instrukcja ta wymaga by nowy numer linii był zawsze równy lub większy od starego numeru tzn. dla instrukcji number a,b,c wymaga się by  $b \geq a$ .

**Find** — znajduje i wyświetla poszukiwany łańcuch alfanumeryczny, jeśli taki występuje w programie. Find (słowo) — znajduje i wyświetla wszystkie miejsca w programie, w których takie słowo występuje podczas gdy find (słowo), 100-150 — powoduje to samo tylko, że w zakresie numerów linii 100-150.

**Change** — działanie jak instrukcja find z tą różnicą, że następuje zastąpienie starego łańcucha przez nowy łańcuch i tak: change (staresłowo/novesłowo), 150 spowoduje zastąpienie łańcucha „staresłowo” nowym łańcuchem „novesłowo” w linii 150.

**Delete** — znaczenie jak w edytorze BASIC-a z tą różnicą, że delete bez żadnego parametru równoważne jest instrukcji New.

**Put** — instrukcja powodująca zapisanie programu na nośniku magnetycznym.

Dla wersji kasetowej instrukcja: put tytuł spowoduje zapamiętanie na kasecie tekstu źródłowego programu z nazwą tytuł. Dla wersji dyskowej możemy stosować instrukcji jak w wersji kasetowej lub możemy wybrać numer stacji w wypadku podwójnej stacji dysków np put0: tytuł. Dopuszcza się również użycie instrukcji put at0: tytuł. W takim wypadku następuje skasowanie starej wersji (jeśli taka występuje) i zastąpienie jej nową o takiej samej nazwie.

**Get** — powoduje wczytanie zbioru. Dla wersji kasetowej nie jest wymagane podanie nazwy zbioru. Wczytywany jest wtedy pierwszy napotkany program. Dla wersji dyskowej należy napisać: get tytuł, co spowoduje wczytanie programu o nazwie tytuł (w razie potrzeby zostaną przeszukane oba napędy dyskowe).

**R lub Run** — (tylko pierwszy tryb pracy) spowoduje skompilowanie programu, a następnie jego uruchomienie.

**L** — (tylko pierwszy tryb pracy) powoduje kompilację programu z równoczesnym wyświetlaniem tekstu i ewentualnych błędów na ekranie monitora. Instrukcja ta nie powoduje uruchomienia programu.

**P** — jak wyżej lecz z wydrukiem na drukarce.

**COMP** — (tryb dyskowy) powoduje kompilację programu zapamiętanego na dysku. Najpierw musi być włożona do napędu dyskowno dyskietka systemowa. Postać instrukcji comp tytuł, opcjonalne parametry.

Opcjonalne parametry:

1 — powoduje wyświetlenie tekstu i błędów na ekranie

p — jak wyżej lecz z wydrukiem na drukarce

n — kompilacja bez tworzenia zbioru wynikowego (ze skompilowanym programem) Zbiór taki na dysku oznaczany jest przez dodanie końcówki. OBJ do tytułu programu

c — kompilacja dająca trochę szybszy i bardziej zwarty kod wynikowy.

**Ex** — (tryb dyskowy) powoduje wykonanie zbioru z końcówką. OBJ. Najpierw do napędu dyskowno musimy włożyć dyskietkę systemową.

**UWAGA!!!** Program może być wykonany jedynie wtedy, gdy podczas kompilacji nie wystąpił żaden błąd. Wszystkie

ewentualne błędy należy poprawić, a następnie poddać program powtórnej kompilacji.

**Hex** — zamiana liczby dziesiętnej na równoważną jej liczbę szesnastkową.

**Decimal** — zamiana liczby szesnastkowej na dziesiętną.

**Dump** — wydruk programu na drukarce.

**Cold** — zimny start C-64.

**Link** (tryb dyskowy) — dla dużych programów jest często wskazane ich rozbicie na mniejsze segmenty, co ułatwia ich kompilację i testowanie. Zbiory wynikowe mogą być następnie połączone w jeden program. Jest to sposób na stworzenie np. biblioteki procedur. link0: nazwa= nazwa1,nazwa2,nazwa3 spowoduje połączenie zbiorów NAZWA1.OBJ;NAZWA2.OBJ;NAZWA3.OBJ w jeden zbiór NAZWA.OBJ.UWAGA: każdy ze zbiorów musi mieć zadeklarowane takie same zmienne globalne; zakłada się, że zbiór o nazwie NAZWA.OBJ zawiera program główny.

Dołączanie zbiorów do kompilacji (tryb dyskowy) — jeżeli w tekście programu pojawi się nazwa programu poprzedzona znakiem # (#NAZWA) oznacza to, że na tym etapie kompilator dołączy do głównego programu, program o nazwie NAZWA 1 skompiluje całość. Jest to niesłychanie wygodne, gdy mamy już przetestowane fragmenty programów/procedur.

**Locate** — locate0:nazwa=wynik spowoduje przekształcenie zbioru NAZWA.OBJ w program o nazwie WYNIK, który może być uruchomiony spod BASIC-a (bez konieczności wgrywania OXFORD Pascal-a). Instrukcja ta spowoduje dołączenie procedur pascalskich do zbioru zawierającego skompilowany program.

**Chain** (nazwa zbioru) (tryb dyskowy) — powoduje wstrzymanie wykonywanego programu i wykonuje program o nazwie „nazwa zbioru”.

### INSTRUKCJE JĘZYKA BASIC dostępne w trybie bezpośrednim (nie w programie)

Podaję jedynie listę tych instrukcji zakładając, że Czytelnicy już się z nimi kiedyś zetknęli. Instrukcje te to: **PRINT** (lub skrót ?); **PRINT**; **OPEN**; **CLOSE**; **CMO**; **POKE**; **PEEK**; **SYS**; **FOR TO STEP**; **LET**; **LOAD**; **FRE O** oraz zmienna ti.

### ROZSZERZENIA W STOSUNKU DO STANDARDU PASCAL-a

Dopuszcza się deklarację liczb szesnastkowych jako stałych. Liczba taka musi być poprzedzona znakiem \$ np. const liczba=\$fe84;

**peek** (x:integer) :0..255 — funkcja działa jak funkcja PEEK w BASIC-u (odczyt zawartości komórki o zadanym adresie x np. d:=peek (145).

**poke** (x:integer;y:0..255) — zmiana zawartości lokacji o adresie x na wartość y np. poke (\$04fc, \$3d).

**origin** (x:jakiśtyp;y:integer) — ustawia wskaźnik x na lokację y w pamięci komputera.

**VDU** (x,y:integer;c:char) — procedura powodująca wyświetlenie znaku typu char w rzędzie określonym przez x i kolumnie określonej przez y np. YDU (5,12,'?'),

**getkey** char — funkcja odczytująca klawiaturę. Otrzymujemy chr(0) gdy żaden klawisz na klawiaturze nie został wciśnięty np. while getkey=chr (0) do; spowoduje wstrzymanie pracy programu do chwili wciśnięcia dowolnego klawisza.

**page** — procedura powodująca wyczyszczenie ekranu

**restore** (true) lub restore (false) powodują odpowiednio włączanie lub odłączanie klawisza restore. Standardowo jest restore (true).

**random** 0..255 — funkcja powoduje wyświetlenie wartości liczby pseudolosowej z zakresu 0..255 Ciąg randomrandom (mod 128) 256 spowoduje wyświetlenie liczb



by losowej z zakresu 0..32767 a ciąg random mod n+1 spowoduje wyświetlenie liczby losowej z przedziału 1..n jeśli n = 32767.

**settime** (gg,mm,ss) — powoduje ustawienie wewnętrznego zegara na gg-godzinę; mm-minutę; ss-sekundy. Wartości wprowadzane muszą być typu całkowitego. Odczyt czasu następuje poprzez instrukcję writeln.

## ROZKAZY MUZYCZNE

**ENVEL** V,A,D,S,R; — gdzie

V jest numerem głosu 1—3

A charakteryzuje narastanie dźwięku 0-15

D charakteryzuje zanikanie dźwięku 0-15

S charakteryzuje wytrzymanie dźwięku 0-15

R charakteryzuje opadanie dźwięku do zera 0-15

**VOICE** V,F,W,D ; gdzie V jest numerem głosu 1-3

F jest częstotliwością dźwięku

0-65535

W jest typem obwiedni 0-3 (trójkąt,

piła, pulsacyjny, szum):

D czas trwania 0-65535

**częstotliwość określana jest tak jak w instrukcji obsługi do C-64 f=** (fx 0.059604645)

Instrukcja ENVEL musi zawsze poprzedzać instrukcję VOICE ponieważ ta ostatnia powoduje rozpoczęcie wydawania zaprogramowanego dźwięku.

**VOLUME** (L) ; gdzie L charakteryzuje wzmocnienie dźwięku 0-15

## INSTRUKCJE GRAFICZNE W TRYBIE ZWYKŁEJ ROZDIELCZOŚCI

**BORDER** (C) ; — ustala kolor ramki

**SCREEN** (C) ; — ustala kolor tła

**PEN** (C) ; ustala kolor tekstu

C jest stałą całkowitą zmieniającą się w zakresie 0-15. Kolory odpowiadające danej wartości są identyczne z tymi podanymi w instrukcji obsługi C-64

## INSTRUKCJE GRAFICZNE W TRYBIE WYSOKIEJ ROZDIELCZOŚCI

**PAPER** (C) ; — ustalenie koloru tła C (0-15)

**INK** (C) ; — ustawienie koloru rysunków C (0-15)

**HIRES** (C) ; — włącza C=1 lub wyłącza C=0 tryb wysokiej rozdzielczości

**P:=EXAMINE** (X,Y) ; — stwierdzenie czy punkt o współrzędnych x,y jest zapalony P=1 czy też zgaszony P=0

**PLOT** (F,X,Y,X1,Y1) ; — gdy F=0 to następuje zrównanie koloru tła z kolorem zadeklarowanym

gdy F=1 to następuje wyczyszczenie ekranu graficznego

gdy F=2 to rysowana jest linia od punktu o współrzędnych X,Y do punktu o współrzędnych X1,Y1.

gdy F=3 to kasowana jest linia od X,Y do X1,Y1

gdy F=4 to wypełniany jest obszar wokół punktu X,Y

gdy F=5 to czyszczony jest obszar obrazu wokół punktu X,Y

W celu narysowania (skasowania punktu należy podać X1,Y1=X,Y

**WINDOW** (U) — Aby stworzyć okienko na ekranie wysokiej rozdzielczości powyżej linii o numerze U.WINDOW 25 to cały ekran.

**Kill** — w trybie edytora jest równoważne instrukcji hires O w programie

UWAGA: Wszystkie parametry występujące w wyżej wymienionych procedurach są liczbami stałymi. Dla instrukcji graficznych x zmienia się w zakresie 00-255 a y w zakresie 0-200. Punkt 0,0 znajduje się w lewym dolnym rogu ekranu.

## MANIPULACJA BITAMI

W OXFORD Pascal-u zaimplementowane są znane z BASIC-a instrukcje logiczne działające na 16-bitowych słowach AND, OR, XOR, NOT. Jako uzupełnienie tych funkcji mamy jeszcze:

**SHL** (x,y) — przesunięcie bitów x o y pozycji w lewo. Lewa strona uzupełniana jest zerami np. shl (4,-1) = 4000, shl (4,4) = 40

**SHR** (x,y) — przesunięcie bitów x o y pozycji w prawo. Lewa strona uzupełniana jest zerami np. shr (4444,4) = 444

Podanie instrukcji shl (x,-y) jest równoznaczne z instrukcją shr (x,y).

*Dominik Falkowski  
Klaudiusz Dybowski*

# 3D

## GRAFIKA WEKTOROWA

**Nawiązując do rozpoczętego tematu programowego tworzenia grafiki przedstawiamy rodzaj grafiki określonej jako wektorowa.**

Wprowadzić na ekranie Spectrum uzyskanie czystej grafiki wektorowej jest niemożliwe, gdyż podstawą obrazu są punkty, to jednak można zasymulować ją w prosty sposób. Potrzebne to będzie jedynie do rozważań i eksperymentów, bo grafikę wektorową wykorzystują tak naprawdę specjalne ekrany graficzne i plotery.

Co to jest wektor? Nie jest to ani „odcinek ze strzałką”, ani „strzała”, lecz uporządkowana para punktów. W szkole definicja taka nie jest od razu zrozumiała, lecz przy komputerze nabiera głębszego znaczenia. Dostępna w BASIC-u (i nie tylko, patrz Bajtek 7/89) komenda DRAW powoduje wykreślenie **wektora** o zadanych współrzędnych. Składnia DRAW nie pozostawia chyba wątpliwości — **DRAW x,y**.

Początek rysowanego wektora, czyli inaczej mówiąc, jego punkt przyłożenia (zaczepienia) to ostatni postawiony na ekranie punkt (lub (0,0), gdy ekran jest świeżo wyczyszczony). Przypominam, iż współrzędne tego ostatniego punktu przechowują zmienne systemowe 23677 i 23678.

Teraz zwyczajne, szkolne zadanie: dany jest wektor [3,5] o początku w punkcie (2,2). Jakie są współrzędne końca wektora? Odpowiedź: (2+3,2+5), czyli (5,7).

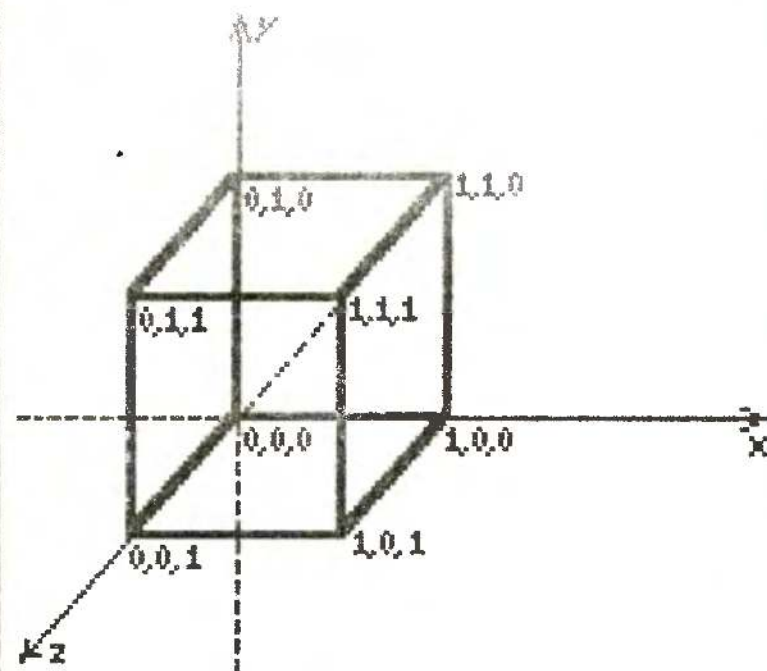
Te wiadomości powinny wyjaśnić niedouczonym programistom istotę wektora na ekranie Spectrum. A po co to wszystko? Zaraz stanie się to jasne.

Niech nawet wrogowie gier komputerowych spojrzą teraz na jedną z gier: Star Wars, The Empire Strikes Back, Elite, Night Raider, Starstrike, Starstrike II, Virus. W każdej z nich wykorzystana jest grafika wektorowa. Co to oznacza? Wszystkie przedmioty (pojazdy, samoloty, góry itp.) przedstawione są jako zarysy krawędzi bryły, w dodatku przezroczystej. Bryła pokazana jest w rzucie, a dokładniej — w aksonometrii ukośnej.

Istota tego rodzaju aksonometrii polega na tym, że krawędzie, które w rzeczywistości są równoległe do patrzącego, zachowują swoje wymiary i położenie, zaś krawędzie prostopadłe do patrzącego ulegają skróceniu o połowę i przedstawiane są w płaszczyźnie równoległej do patrzącego, pod kątem 45°. Trochę to zawiłe, ale w istocie bardzo proste.

Należy się teraz zastanowić, jak uwieńczyć te myśli programem. Jego zadaniem niech będzie umożliwienie obejrzenia ze wszystkich stron zdefiniowanej bryły. Tak działa najlepszy program inżynierski na PC — AutoCAD, tak też działa Spectrumowy VU-3D (opis wkrótce w Bajtku).

Projektujemy naszą bryłę. Niech będzie to sześcian, by uprościć dane i nie tracić miejsca. Sześcian przedstawiony jest na rysunku, w prostokątnym układzie współrzędnych XYZ. Każ-



demu wierzchołkowi przypisane są współrzędne, krawędź ma długość jednostkową.

Pozostaje do rozstrzygnięcia problem, jak kodować bryłę w pamięci. Ja przyjąłem sposób bardzo prosty: kodowane są wszystkie krawędzie po kolei; każdej z nich przyporządkowana jest szóstka liczb — współrzędne początku i współrzędne wektora pokrywającego się z tą krawędzią. Otrzymujemy w ten sposób 72 liczby, lecz są to zera, jedynki lub minus jedynki.

Zakodowaną bryłę należy wpisać w tablicę. W celu przejrzystości posłużyłem się aż sześcioma tablicami: **X, Y, Z** dla początków i **K, L, M** dla współrzędnych wektorów. Teraz odtworzenie bryły na ekranie odbywa się następująco: pobierane są współrzędne początku pierwszego wektora i współrzędne jego samego, a następnie przetwarzane z układu trójwymiarowego na układ płaski.

$$\text{Wzory: } xx=x+\frac{z}{1.4}, \quad yy=y+\frac{z}{1.4}$$

można łatwo wydedukować z twierdzenia Pitagorasa. Liczba 1.4 (w wydruku 1.41 to przybliżona wartość  $\sqrt{2}$ )

Od tej chwili wszystkie operacje związane z obrotem i powiększaniem/pomniejszaniem bryły odbywają się na współrzędnych i po każdej operacji ta sama procedura powoduje odtworzenie bryły na ekranie.

Przeanalizujcie sami, co wykonują linie 210—240, w których następuje obrót bryły w płaszczyznach XY i YZ. Sterowanie: klawisze **Q, A, O, P** oraz **Z, X** — zmiana skali.

Równania opisujące obrót płaszczyzny wprowadza się w drugiej klasie liceum, a wyglądają one następująco:

$$xx=x*\cos \alpha+y*\sin \alpha$$

$$yy=-x*\sin \alpha+y*\cos \alpha$$

Za  $\alpha$  przyjąłem tutaj  $15^\circ$ , czyli  $\frac{\pi}{12}$  radianów.

Liczby 0.2588 i 0.9659 to przybliżone wartości sinusa i cosinusa kąta  $15^\circ$ .

Procedura obrotu działa wolno. Cały program można przyspieszyć za pomocą kompilacji (najlepiej TOBOS-em), lub wskazówek zawartych w Bajtku 3/89. A kto pokusi się o napisanie naszego mini-programu w assemblerze?

*Marcin Przasnyski*

```

10 DIM X(12): DIM Y(12): DIM Z(12): DIM
K(12): DIM L(12): DIM M(12)
99 LET SK=50
100 FOR F=1 TO 12
110 READ X(F),Y(F),Z(F)
120 READ K(F),L(F),M(F)
140 NEXT F
145 GOSUB 1000
149
200 LET A$=INKEY$: IF A$="" THEN GOTO 200
210 IF A$="p" THEN FOR F=1 TO 12: LET A=X
(F)*.9659+Y(F)*.2588: LET B=-X(F)*.2588+Y
(F)*.9659: LET X(F)=A: LET Y(F)=B: LET A=
K(F)*.9659+L(F)*.2588: LET B=-K(F)*.2588+
L(F)*.9659: LET K(F)=A: LET L(F)=B:NEXT F
220 IF A$="o" THEN FOR F=1 TO 12: LET A=X
(F)*.9659-Y(F)*.2588: LET B=X(F)*.2588+Y(
F)*.9659: LET X(F)=A: LET Y(F)=B: LET A=K
(F)*.9659-L(F)*.2588: LET B=K(F)*.2588+L(
F)*.9659: LET K(F)=A: LET L(F)=B:NEXT F
230 IF A$="q" THEN FOR F=1 TO 12: LET A=Y
(F)*.9659+Z(F)*.2588: LET B=-Y(F)*.2588+Z
(F)*.9659: LET Y(F)=A: LET Z(F)=B: LET A=
L(F)*.9659+M(F)*.2588: LET B=-L(F)*.2588+
M(F)*.9659: LET L(F)=A: LET M(F)=B:NEXT F
240 IF A$="a" THEN FOR F=1 TO 12: LET A=Y
(F)*.9659-Z(F)*.2588: LET B=Y(F)*.2588+Z(
F)*.9659: LET Y(F)=A: LET Z(F)=B: LET A=L
(F)*.9659-M(F)*.2588: LET B=L(F)*.2588+M(
F)*.9659: LET L(F)=A: LET M(F)=B:NEXT F
250 IF A$="z" AND SK<75 THEN LET SK=SK+2
260 IF A$="x" AND SK>5 THEN LET SK=SK-1
300 CLS:GOSUB 1000:GOTO 200
999
1000 FOR F=1 TO 12
1010 LET XX=X(F)+Z(F)/1.41: LET dXX=K(F)+
M(F)/1.41
1020 LET YY=Y(F)+Z(F)/1.41: LET dYY=L(F)+
M(F)/1.41
1030 PLOT SK*XX,SK*YY: DRAW SK*dXX,SK*dYY
1040 NEXT F
1050 RETURN
8999
9000 DATA 0.0,0.1,0.0
9010 DATA 1.0,0.0,0.1,0
9020 DATA 1.1,0.-1.0,0
9030 DATA 0.1,0.0.-1.0
9040 DATA 0.1,1.0.-1.0
9050 DATA 0.0,1.0,0.-1
9060 DATA 1.0,1.0,0.-1
9070 DATA 1.0,1.0,1.0
9080 DATA 1.1,1.-1.0,0
9090 DATA 1.1,1.0,0.-1
9100 DATA 1.0,1.-1.0,0
9110 DATA 0.1,1.0,0.-1
    
```



## TRZY KANALY

Kontynuujemy cykl artykułów poświęconych muzyce na Spectrum. Dziś zajmiemy się wykorzystaniem trzech dostępnych kanałów z poziomu Basica, tak jak w 128-Basic. Niestety, potrzebny jest spory program obsługi generatora.

Prezentowana procedura w kodzie maszynowym wyciągnięta z programu „The Music Box 128” została napisana właśnie w tym celu. Umożliwia ona tworzenie muzyki w trzech niezależnych od siebie kanałach dźwiękowych. Jeśli zaświeciły się Wam oczy, to pewnie zgasi ten blask informacja, że program ten jest przeznaczony na ZX Spectrum 48 z dołączonym generatorem dźwięku AY 3-8910. Działa on także na ZX Spectrum 128, pod warunkiem, że wspiszemy go w trybie 48 Basic.

Program jest bardzo prosty w obsłudze. Muszę jednak wyjaśnić zasadę, według której należy pisać muzykę. Każdy dźwięk ma odpowiedni symbol (inny jednak niż w instrukcji PLAY). I tak np. dźwięk **DO** ma symbol średnika ;. Można zapisywać dźwięki o rozpiętości do 4 oktaw. Ich symbolami są znaki ASCII od # do dużej litery **P**. Dźwięki następujących znaków to kody perkusji. Litera **b** oznacza zapętlenie muzyki, czyli po dojściu do końca utworu komputer zaczyna grać go jeszcze raz. Nie będę zdradzać wszystkich tajników tego programu, by zachęcić Czytelników do samodzielnych eksperymentów. Gdy już ułożymy jakiś utwór, wtedy trzeba go wpisać w linii 130, 140 i 150, pamiętając o tym, że podajemy dźwięki dla pierwszego kanału w pierwszej z tych linii, dla drugiego w drugiej itd. Po wpisaniu uruchamiamy program. Odpowiednie kody zostaną umieszczone w adresach od 60000 do 62999.

Jako ciekawostkę mogę podać, że zapisując różne wartości do pamięci od adresu 63000 do 63999 uzyskamy wiele ciekawych efektów, np. zmianę brzmienia dźwięku czy zmianę głośności. Natomiast nie wolno nic zmieniać między adresem 64000 a 65025, ponieważ jakkolwiek zmiana spowoduje restart systemu i, co za tym idzie, utratę programu. Nie należy zmieniać podanych adresów, ponieważ program nie jest relokowalny.

Skomponowany utwór możemy nagrać na taśmę instrukcją **SAVE** nazwa **CODE** 60000,5024. Tak zapisaną kompozycję wczytujemy przez **LOAD** „nazwa” **CODE** i uruchamiamy zleceniem **RANDOMIZE USR 64000**. Odpowiednie tempo dobieramy poleceniem **POKE 64062, tempo**.

Mimo, że program jest dość długi, to jednak zachęcam do jego wpisania. Naprawdę warto, tym bardziej, że planujemy podanie w przyszłych numerach „Bajtki” kodów kilku melodii (np. z filmu „Gliniarz z Beverly Hills”). Pozostaje mi tylko życzyć Wam powodzenia i owocnej pracy z programem.

Michał Sobieszuk

```

2 LET S=0: FOR I=64000 TO 65019
4 READ A:LET S=S+A: POKE I,A: NEXT I
6 IF S>99864 THEN PRINT 'BLAD':STOP
10 DATA 195,165,251,33,19,194,17,18,25
3,1,44,1,237,176,201,15,15,15,168,1,159,
6,61,1,30,251,0,0,15,0,0,0,16,0,0,0,0,0,
0,0,0,0,0,0,0,108,234,97,234,84,238,73
,238,60,242,49,242,36,246,25,246,230,8,7
,6,5,6,7,8,9,10,11,11,11,11,11,11,15,
14,13,12,11,10,9,8,7,6,5,4,3,2,1,0,15,14
,13,12,11,10,9,8,8,9,10,11,12,13,14,15,4
,6,8,10,11,12,13,14,15,14,13,12,11,10,7,
3,8
20 DATA 12,12,12,12,12,12,12,12,12,12
,12,12,11,9,8,11,11,11,11,11,11,11,8,
5,2,2,3,3,3,3,8,11,13,14,15,15,14,13,12,
11,10,9,9,9,9,9,15,0,15,0,15,0,15,0,15,0
,15,0,15,0,15,8,159,6,64,6,230,5,145,5,6
5,5,246,4,174,4,107,4,44,4,240,3,183,3,1
30,3,79,3,32,3,243,2,200,2,161,2,123,2,8
7,2,54,2,22,2,248,1,220,1,193,1,168,1,14
4,1,121,1,100,1,80,1,61,1,44,1,27,1,11
30 DATA 1,252,0,238,0,224,0,212,0,200
,0,189,0,178,0,168,0,159,0,150,0,141,0,1
33,0,126,0,119,0,112,0,106,0,100,0,94,0,
89,0,84,0,255,255,8,9,24,9,0,128,28,128,
4,16,12,16,20,16,28,128,30,128,0,0,0,0,0
,0,0,0,205,81,251,205,142,2,28,40,247,19
5,199,251,243,33,25,250,6,255,112,35,4,2
05,217,252,33,32,250,112,35,205,217,252,
33,39,250,112,35,205,217,252,35,205,217,
252,33,58,250,205,208,251,205,157,252,33
,46,250,205,208,251
40 DATA 62,1,205,244,251,33,50,250,20
5,208,251,62,2,205,244,251,33,54,250,205
,208,251,62,3,205,244,251,205,21,253,205
,198,253,205,114,252,201,33,96,234,17,46
,250,1,232,3,205,189,251,205,189,251,205
,189,251,205,189,251,195,69,251,125,18,1
9,124,18,19,9,19,19,201,22,7,30,255,205,
184,253,251,201,94,35,86,19,26,254,63,20
2,235,251,71,114,43,115,135,95,22,0,33,1
91,250,25,86,35,94,235,201,35,94,35,86,4
3,43,195,212,251,79,120,254,54,48,63,62,
255,188,200,121,229
50 DATA 135,61,33,16,250,22,0,95,25,2
09,115,43,114,33,36,250,22,0,89,25,43,12
6,43,43,43,119,33,15,250,25,43,126,33,25
,250,25,119,33,25,250,13,194,46,252,203,
134,13,194,52,252,203,142,13,192,203,150
,201,124,50,24,250,125,230,240,254,0,40,
31,15,15,15,15,33,33,250,22,0,89,25,43,1
19,33,25,250,13,32,3,203,158,201,13,32,3
,203,166,201,203,174,201,125,230,15,33,2
6,250,22,0,89,25,43,119,175,24,214,62,1,
205,213,253,62,2,205,213,253,62,3,205,21
3
60 DATA 253,205,198,253,58,62,250,6,5
0,0,16,253,60,32,248,205,149,253,33,32,2
50,52,126,254,16,200,195,114,252,120,230
,3,79,203,80,32,6,203,88,194,223,252,201
,203,88,194,245,252,62,240,160,203,15,20
3,15,203,15,203,15,71,175,185,40,14,33,1
5,250,22,0,89,25,43,112,6,0,195,5,253,33
,15,250,205,217,252,6,0,195,15,253,112,3
5,112,35,112,201,121,50,63,251,62,240,16
0,7,7,7,203,96,71,40,3,238,127,60,50,39,
250,201,203,96,192,62,224,160,7,203,7,20
3,7
70 DATA 71,175,185,40,10,22,0,89,33,3
6,250,25,43,112,201,33,36,250,195,217,25
2,33,39,250,175,190,200,70,58,63,251,254
,0,40,32,221,33,18,250,33,46,250,61,202,
94,253,33,50,250,221,35,221,35,61,202,94
,253,33,54,250,221,35,221,35,195,94,253,
33,46,250,221,33,18,250,205,94,253,221,3
5,221,35,33,50,250,205,94,253,221,35,221
,35,33,54,250,94,35,86,26,128,254,53,208
,22,0,135,95,33,191,250,25,94,35,86,221,
110,0,221,102,1,235,175,237,82,125,108,1
03,175,197
80 DATA 6,4,238,0,203,28,203,29,203,3
1,16,246,180,221,119,22,221,117,23,193,2
01,33,18,250,17,40,250,6,3,126,229,35,10
2,111,235,126,35,229,102,111,25,48,1,35,
76,125,209,19,225,119,35,113,35,16,230,2
01,197,1,253,255,237,81,1,253,191,237,89
,193,20,201,33,18,250,22,0,6,14,94,35,20
5,184,253,16,249,201,33,26,250,61,95,22,
0,25,235,33,7,0,25,126,254,0,200,213,33,
63,250,61,23,23,23,23,22,0,95,25,58,32,2
50,95,25,126,225,119,201
100 READ a$:FOR i=1TO 999:POKE (59999+i
),(CODE a$(i))-35:NEXT i
110 READ b$:FOR i=1 TO 999: POKE (6099
9+i),(CODE b$(i))-35: NEXT i
120 READ c$:FOR i=1 TO 999: POKE (6199
9+i),(CODE c$(i))-35: NEXT i
130 DATA " ; "
140 DATA " ; "
150 DATA " ; "
200 RANDOMIZE USR 64000

```

Ten program to naprawdę pchła. Nie z tych, które szkodzą, jak wirusy, lecz jeden z użytecznych mini-programów.

Po wpisaniu krótkiej procedury i uruchomieniu jej, w głośniczkach swych Spectrumów usłyszycie pseudo-perkusyjne dźwięki. Skąd się one biorą? Proponuję dokładne przeanalizowanie tych kilkunastu bajtów, gdyż na ich podstawie utworzyć można cały plik procedur perkusyjnych. Ta jedna powinna stać się dla Was inspiracją.

Jeśli nie macie jeszcze generatora AY 3-8910, kupcie lub wykonajcie go jak najszybciej. Zaden bowiem zastęp pcheł nie zastąpi prawdziwej, 3-kanałowej muzyki stereo!

```

10 REM Taka sobie pchła
20 FOR f=60000 TO 60020: READ a: POKE f,a
30 NEXT f
40 DATA 1,160,0,17,1,0,10,38,0,111,197
50 DATA 205,181,3,193,11,120,177,200,
24,241
60 RANDOMIZE USR 60000: PAUSE 20: GO
TO 40

```

Gen Martinez

## GWIAZDY

Czytelnikom „Bajtki” interesującym się grafiką komputerową polecam niniejszy program napisany w BASIC-u ZX Spectrum.

Program ów generuje w sposób losowy najróżniejsze grafiki umownie nazwane przeze mnie gwiazdami.

Po wklepaniu tych paru linii programu i uruchomieniu przez RUN, obsługa sprowadza się do zadania, w odpowiedzi na pytanie „Podaj ilość warstw”, ilości nałożonych na siebie rysunków. Najciekawsze efekty uzyskuje się przy zmiennej **ilg** z przedziału od 1 do 3. Przy większych wartościach rysunek może stawać się nieczytelny.

Jeżeli już nasycimy się wielością generowanych rysunków, możemy wprowadzić pewne modyfikacje w programie. Spróbujmy w linii 90 zamiast odcinków rysować łuki poprzez dodanie trzeciego parametru w instrukcji DRAW (wartości od 0.1 do 1). Analogicznie łuki możemy rysować także w liniach 50, 60, 70 i 80. Na przykład zadajmy następujące parametry łuków: linia 50-1, linia 60-(-1), 70-1, 80-(-1). W linii 90 rysujemy tylko odcinki.

Zadajmy parametry jeszcze inaczej — odpowiednio w liniach: 50-2, 60-(-2), 70-2, 80-(-2), a w liniach 90 łuk o parametrach 0.5. Zamiast gwiazd program w tym przypadku kreśli coś na kształt pąków kwiatów. Ciekawą zmianę uzyskuje się wprowadzając w linii 5 instrukcję OVER 1.

Po wnikliwym przeanalizowaniu programu z pewnością odkryjecie jeszcze inne możliwości modyfikacji kreowanej przez komputer rysunku.

Grzegorz Galiński

```

5 BORDER 0: PAPER 0: INK 9: C
LS : LET g=1: INPUT "Podaj ilosc
warstw ";ilg
10 DIM b(15,2): DIM a(50,2): L
ET m=0: LET n=0
20 LET i=INT (1+RAND*14): LET f
=PI/(2*i)
25 FOR k=0 TO i
30 LET dl=INT(150): LET x=INT (
dl*COS (f*k)): LET y=INT (dl*SIN
(f*k)): IF x>127 OR y>87 THEN G
O TO 30
35 LET b(k+1,1)=x: LET b(k+1,2
)=y: NEXT k
40 GO SUB 500
50 FOR n=1 TO i+1: PLOT 128,88
: DRAW b(n,1),b(n,2): GO SUB 100
: NEXT n
60 FOR n=i+1 TO 1 STEP -1: PLO
T 128,88: DRAW -b(n,1),b(n,2): G
O SUB 100: NEXT n
70 FOR n=1 TO i+1: PLOT 128,88
: DRAW -b(n,1),-b(n,2): GO SUB 1
00: NEXT n
80 FOR n=i+1 TO 1 STEP -1: PLO
T 128,88: DRAW b(n,1),-b(n,2): G
O SUB 100: NEXT n
90 PLOT a(1,1),a(1,2): FOR k=1
TO m-1: DRAW a(k+1,1)-a(k,1),a(
k+1,2)-a(k,2): NEXT k
99 GO TO 10
100 LET m=m+1: LET a(m,1)=PEEK
23677: LET a(m,2)=PEEK 23678: RE
TURN
500 IF g=ilg+1 THEN PAUSE 100:
LET n=INT(7): BORDER n: PAPER n:
INK 9: CLS : LET g=1
510 LET g=g+1: RETURN

```



# DODATKOWE INTERFEJSY

## DO PCW...

W swej typowej konfiguracji, Amstrad PCW 8256 nie jest przystosowany do podłączenia joysticka lub myszy, a jego możliwości dźwiękowe są na poziomie ZX Spectrum (brzęczyk). Ograniczenia te wynikają z filozofii tego systemu, przeznaczonego głównie do pracy profesjonalnej. Możliwe jest jednak nabycie dodatkowych modułów, które pozwalają na bardziej rekreacyjne wykorzystanie sprzętu.

Brytyjska firma DK'TRONICS, znana ze swoich rozszerzeń do ZX Spectrum i Amstrada CPC, oferuje kilka ciekawych urządzeń także do PCW. Za 40 funtów można kupić programowalny sterownik joysticka razem z 3-kanalowym syntetyzerem dźwięku opartym na układzie AY-3-8912. Dostarczone oprogramowanie umożliwia przypisanie joystickowi pięciu dowolnych klawiszy, stosowanych w typowych grach. Położenie dźwigni może być również odczytane przez własne programy pracujące pod kontrolą systemu CP/M Plus. Możliwość muzyczne układu oferuje także 5 dodatkowych wejść/wyjść cyfrowych pozwalających na sterowanie i kontrolę pięciu urządzeń zewnętrznych.

Drugim interesującym produktem firmy DK'TRONICS jest przystawka wyposażona w podtrzymywany bakteryjnie zegar czasu rzeczywistego. Dzięki przełotowemu złączu krawędziowemu współpracuje ona ze standardowym interfejsem CPS 8256, oferowanym przez firmę Amstrad. Wewnątrz modułu znajduje się niewielka pamięć RAM (50 bajtów), dostępna przez programy użytkownika. Zniechęca dość wysoka cena — 35 funtów.

Znaczne ułatwienie pracy z systemem operacyjnym CP/M Plus, dzięki myszy i odpowiedniemu oprogramowaniu, zapewnia pakiet firmy Advanced Memory Systems Ltd. Oferowana jest specjalna nakładka na system, zbliżona w działaniu do znanego na IBM PC programu XTREE. Użytkownicy Amstrada PCW 8256 otrzymują także kilka typowych programów pomocniczych podobnych do borlandowskiego Sidekicka, a więc kalkulator, notes i mały edytor, sterowany myszą. Komplet (mysz, interfejs i oprogramowanie) kosztuje 80 funtów.

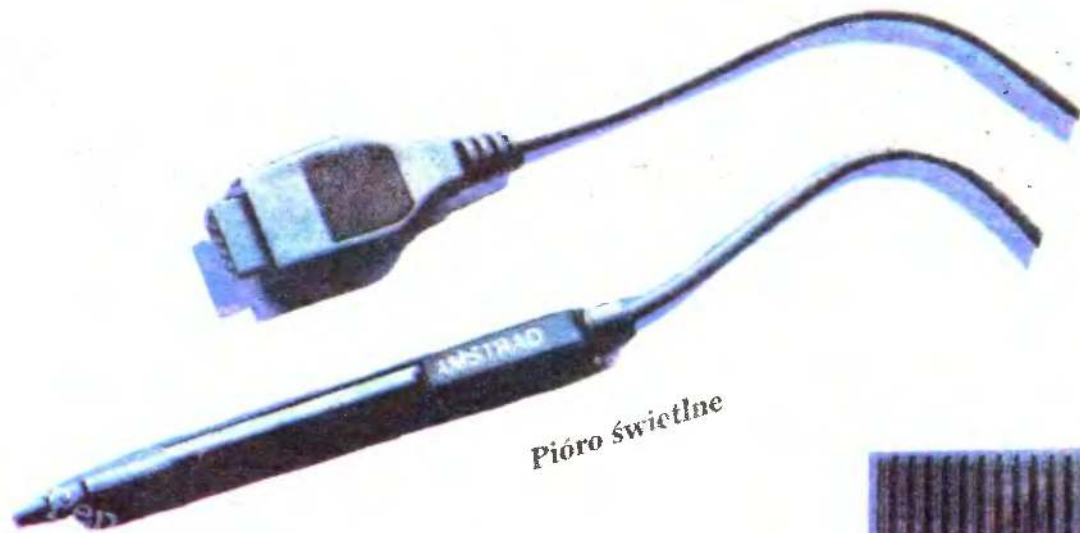
Posiadacze Amstradów CPC mogą skorzystać z bogatej oferty dodatkowych urządzeń produkowanych przez wiele angielskich firm.

Dołączenie telewizora kolorowego, pracującego w systemie PAL jest możliwe dzięki modulatorom MP1 i MP2 sprzedawanym do CPC 464 i 6128 w cenie 15 i 30 funtów, odpowiednio. Dodatkowa stacja dysków 3-calowych firmy Amstrad kosztuje 100 funtów dla komputera CPC 6128. Wyposażenie 464 w podobne urządzenie możliwe jest po nabyciu specjalnego interfejsu. Jednak w tym przypadku nie możemy pracować z

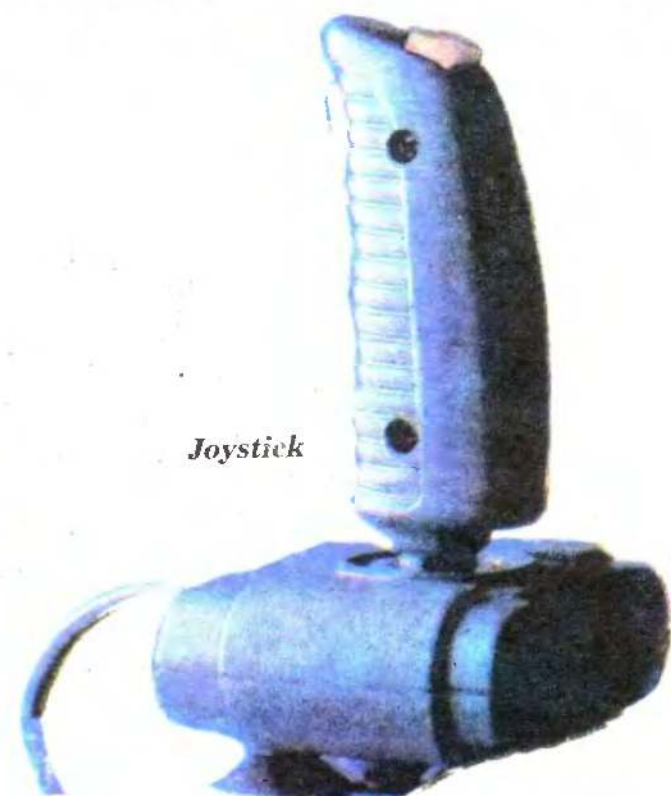
CP/M'em Plus, a tylko z jego wcześniejszą wersją 2.2. Większy obszar na programy użytkownika daje karta firmy DK'TRONICS zawierająca 64 KB pamięci RAM. Przybliżona jej cena 50 funtów. Zamiast kupować drugą stację do CPC 464 lub 6128 możemy rozważyć nabycie za 100 funtów modułu ramdysku o pojemności 256 KB. Rozwiązanie to jest bardzo wygodne przy pracy z dużymi programami, nie mieszczącymi się w pamięci operacyjnej widzianej przez CP/M.

Wydanie 30 funtów na syntetyzer dźwięku ze wzmacniaczem stereofonicznym, zamienia CPC w dość dobry instrument muzyczny. Pióro świetlne razem z oprogramowaniem graficznym oferowane jest przez kilka firm w cenie od 20 do 50 funtów.

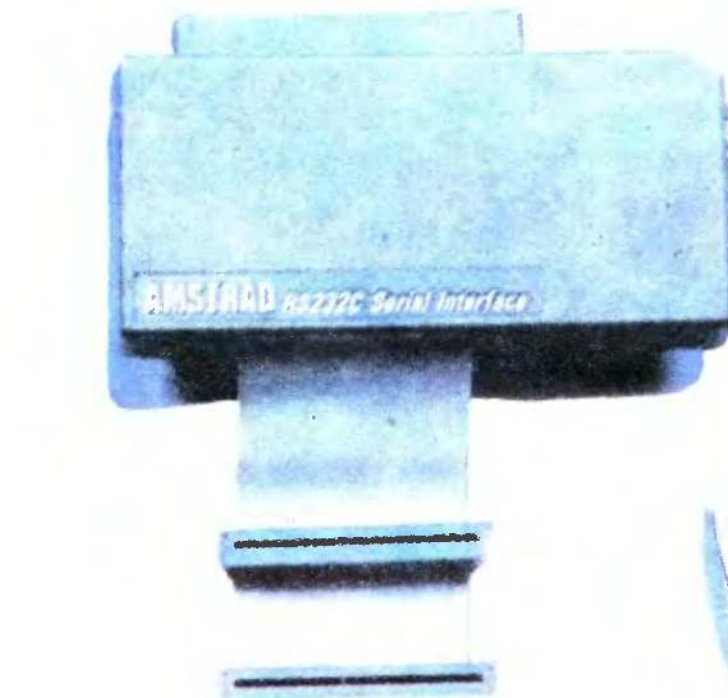
Podobnie jak w przypadku Amstrada PCW 8256, mysz z odpowiednim oprogramowaniem dostarczana jest przez firmę Advanced Memory Systems Ltd. Interfejsy RS232 produkowane są przez



Pióro świetlne



Joystick



Interface RS 232C do CPC



Syntezytor mowy i wzmacniacz stereofoniczny



Stacja dysków i interfejs do CPC 464



Modulator TV do CPC 6128

Amstrada i firmę Pace. Cena od 50 do 100 funtów. Najtańszy modem akustyczny oferowany przez firmę CirKit kosztuje 30 funtów i współpracuje z komputerem poprzez złącze szeregowe. Oprogramowanie dostarczone jest na taśmie. Modemy galwaniczne są kilkakrotnie droższe (120-150 funtów).

O ile monopolistą w dziedzinie pakietów pamięci RAM na CPC jest firma DK'TRONICS, to podobną rolę dla modułów zawierających ROMy odgrywa firma Micopower. Produkowana przez nią karta pozwala na umieszczenie 7 układów ROM o pojemności 8 lub 16KB każdy. Mniejsze karty na 4 układy oferowane są w cenie 30 funtów przez firmę KDS, która oprócz tego dostarcza pakiety pozwalające na sterowanie prostych robotów lub innych urządzeń zewnętrznych przez CPC.

Jonasz Mayer

# SZYBSZY DYSK

Stacja dysków Amstrada 6128 jest szybka. Nic nie stoi jednak na przeszkodzie by była jeszcze szybsza. Umożliwia to poniższy program. Jest on przydatny zwłaszcza przy użytkowaniu programów na bieżąco korzystających z dysku. Amatorom gier polecam go specjalnie dla Sorcery+.

Jacek Kunowski

```

1 *****
2 *
3 * Zmiana parametrow czasowych *
4 *
5 * sterownika dyskowego v-1.1 *
6 *
7 * JACEK KUNOWSKI 1989 r *
8 *
9 *****
    
```

```

10 MEMORY &BFFF:MODE 1:PRINT
20 DATA &21,&1d,&90,&cd,&d4,&bc,&22,&1e
30 DATA &90,&79,&32,&20,&90,&1e,&00,&16
40 DATA &00,&0e,&00,&21,&30,&90,&df,&1e
50 DATA &90,&32,&21,&50,&c9,&82,&23,&00
60 DATA &c3,&00,&01,&01,&0a,&00,&03,&50
70 DATA &00,&fa,&00,&af,&0f,&0c,&01,&03
80 FOR i=&9000 TO &9000+29
    
```

```

90 READ zm:FOKE i,zm:NEXT
100 PRINT" ZMIANA PARAMETROW CZASOWYCH"
110 PRINT" STEROWNIKA DYSKU wersja 1.1"
120 PRINT:FOR k=&9030 TO &9030+17
130 READ zm:FOKE k,zm:NEXT:PRINT" ";
140 PRINT" |Wyjście z programu - 0"
150 PRINT:PRINT" ";
160 PRINT" |Przyspieszony Dysk - 1"
170 PRINT:a$="":PRINT" ";
180 PRINT" Normalny Dysk - 2"
190 WHILE NOT(a$="1"OR a$="2"OR a$="0")
200 a$=INKEY$:WEND:IF a$="0" THEN END
210 PRINT:IF a$<>"1" THEN 240
230 FOKE &9014,&30:PRINT" Szybki ";
240 IF a$<>"2" THEN 260
250 FOKE &9014,&37:PRINT" Normalny ";
260 CALL &9000:a$="":PRINT"DYSK G.K."
270 PRINT CHR$(&7);:GOTO 190
    
```



# WYKORZYSTANIE PRZERWAŃ CPC 6128

## PRZERWANIA

Przerwanie jest to zaniechanie wykonywania przez procesor aktualnego programu i rozpoczęcie wykonywania tzw. procedury obsługi przerwania. Procedura ta wykonuje różne niezbędne dla komputera czynności np.: odświeżanie pamięci dynamicznych RAM, odczyt klawiatury, współpraca z urządzeniami peryferyjnymi. Po zakończeniu procedury obsługi procesor wznawia wykonywanie przerwanej programu od miejsca jego przerwania.

Mikrokomputer AMSTRAD może przyjmować przerwania z dwóch źródeł jako:

— przerwania zewnętrzne, pochodzące od urządzeń dołączonych do szyny rozszerzającej,

— przerwania wewnętrzne, które są regulowanymi przerwaniem czasowymi generowanymi przez układy zegarowe mikrokomputera. Układy te generują sygnały przerwania co 1/300 sekundy. System operacyjny (AMSDOS) przyjmuje je jako przerwania od szybkiego zegara. Co trzeci sygnał traktowany jest jako przerwanie dla generatora dźwięku. Steruje on oprogramowaniem generatora i nie jest w inny sposób dostępny dla systemu. Co szósty sygnał uważany jest za przerwanie od powrotu plamki obrazu (ang. frame flyback) i od zwykłego zegara. Przykładowo w czasie obsługi przerwania od powrotu plamki obrazu testowany jest bufor klawiatury.

System operacyjny AMSTRADA przekształca przerwania w ich odpowiedniki programowe tzw. zdarzenia (ang. events). Standardowo mikroprocesor przyjmuje przerwania w trybie 1 (IM 1), co oznacza, że po pojawieniu się sygnału przerwania generuje on rozkaz RST 7. Obsługa przerwania polega zatem na wykonaniu przez procesor stałej procedury obsługi, która po wykonaniu pewnych niezbędnych dla systemu czynności przekazuje sterowanie do procedury obsługi zdarzenia (uaktywnienie zdarzenia). Obsługa zdarzeń jest bardziej elastyczna niż obsługa przerwania sprzętowych. Nie ma np. ograniczeń na miejsce umieszczenia przerwania sprzętowych. Nie ma np. ograniczeń na miejsce umieszczenia procedur obsługi, konfigurację pamięci ROM, czas ich realizacji.

## ZDARZENIA

Zdarzenie definiowane jest poprzez blok zdarzenia (rys. 1), który

jest obszarem 7 bajtów pamięci i zawiera następujące pola:

- adres systemowy,
- licznik zdarzenia,
- klasę zdarzenia,
- adres procedury obsługi,
- konfigurację pamięci ROM.

Adres systemowy przeznaczony jest do zapamiętania bloku w kolejce zdarzeń i nie może być zmieniany przez użytkownika. Licznik zdarzenia służy do śledzenia różnicy pomiędzy liczbą uaktywnień zdarzenia a liczbą zakończonych jego obsług. Licznik jest zwiększany o jeden w chwili zakończenia procedury obsługi. Zapobiega to traceniu przerwania pojawiających się w czasie obsługi zdarzeń. Klasa zdarzenia określa jego rodzaj. Zdarzenia mogą być:

- asynchroniczne — obsługiwane są natychmiast lub prawie natychmiast po ich uaktywnieniu,
- synchroniczne — obsługą steruje wykonywany program, który powinien regularnie przeglądać ich kolejkę i jeśli nie jest pusta informować o tym system,
- ekspresowe — procedura obsługi wywoływana jest jeszcze w stanie przerwania, nie może włączać przerwania (rozkazem EI) ani niszczyć rejestrów IX, IY oraz używać przerwania (rozkazem EI) ani niszczyć rejestrów IX, IY oraz używać dodatkowego zestawu rejestrów,
- zwykłe — w czasie obsługi zdarzenia system pracuje z włączonymi przerwaniem i „uważa”, że nie jest już w stanie przerwania.

Procedura obsługi może znajdować się w środkowych 32 kB pamięci RAM lub gdziekolwiek w pamięci ROM. Zależy to od zawartości bajtu „klasa zdarzenia” i może być to adres „daleki” lub „bliiski”. Poszczególne bity w tym bajcie mają następujące znaczenie:

- 0 : 1 — adres „bliiski”, 0 — adres „daleki”.
- 1.. 4 : priorytet dla zdarzeń synchronicznych,
- 5 : zawsze 0,
- 6 : 1 — zdarzenie ekspresowe, 0 — zdarzenie zwykłe,
- 7 : 1 — zdarzenie asynchroniczne, 0 — zdarzenie synchroniczne,

Na adres „daleki” składa się 16 bitowy adres procedury bajt o nazwie ROM. W przypadku adresu „bliskiego” bajt ten jest pomijany. Licznik zdarzenia zlicza ilość uaktywnień w następujący sposób:

- zwiększanie o 1

-128..-2: licznik nie jest zmieniany, zdarzenie jest ignorowane,

-1: wartość niedopuszczalna,

0: licznik jest zwiększany o jeden i zdarzenie jest uaktywniane,

1..126 : licznik jest zwiększany o jeden lecz nie jest wykonywane dalsze działanie i zdarzenie czeka na koniec obsługi poprzedniego,

127 : licznik nie jest zwiększany — zdarzenie ignorowane,

b) zmniejszanie o 1

128 : wartość niedopuszczalna, -127..0: licznik nie zmienia się, zdarzenie pozostaje nieobsługiwane,

1: licznik jest zmniejszany o jeden i obsługa zdarzenia kończy się.

2..127 : licznik jest zmniejszany o 1. Procedura obsługi może uczynić zdarzenie nieaktywnym poprzez ustawienie licznika na wartość ujemną (według przyjętej konwencji na -64).

System grupuje bloki zdarzeń w kolejki trzech rodzajów:

- zwykłego zegara,
- powrotu plamki obrazu,
- szybkiego zegara.

Obok przedstawione są schematy wymienionych bloków.

Adres zegarowy (rys. 2) jest obszarem wykorzystywanym przez system i nie może być zmieniany przez użytkownika. Licznik zegarowy służy do zliczania impulsów pojawiających się co 1/50 sekundy. Zliczanie odbywa się od wartości, nadanej mu początkowo przez użytkownika, do zera. Zdarzenie może być uaktywnione dopiero po wyzerowaniu się licznika zegarowego. Następnie licznikowi temu nadawana jest wartość z pola o nazwie „odtworzenie licznika”. Jeśli to pole jest równe zero, to po jednym uaktywnieniu blok stanie się „drzemliwym” (ang. dormant) i żadne nowe uaktywnienie nie będzie przyjmowane.

Adres systemowy (rys. 3) jest wykorzystywany przez system do zapamiętania bloku w kolejce i nie może być zmieniany przez użytkownika.

## WYKORZYSTANIE PRZERWAŃ

System pozwala użytkownikowi na dołączanie definiowanych przez niego bloków zdarzeń z własnoręcznie napisanymi procedurami ich obsługi. Użytkownik uzyskuje zatem bardzo wygodne narzędzie programowe, za pomocą którego może tworzyć takie specyficzne funkcje, których nie dałoby się inaczej zaprogramować, bądź byłoby to niezmiernie kłopotliwe. Przykładowo można w ten sposób w dowolnym momencie działania programu (np. po wciśnięciu dwóch wybranych klawiszy) uzyskać kopię ekranu na drukarce (Hard Copy), można też utworzyć tzw. help, który byłby wyświetlany na żądanie użytkownika (może to być np. instrukcja obsługi uruchamianego programu). Poniżej umieszczono trzy przykłady

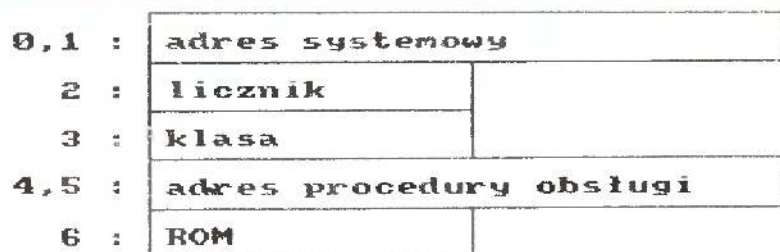
### Przykład 1

```

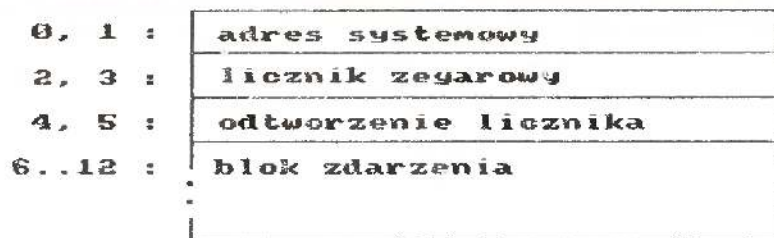
10      org 40960                ;adres umieszczenia kodu
20
30 testkl: equ #bb1e           ;adr. proc. testow. klawisza
40 włącz: equ #bcd7           ;adr. proc. inicj. i dol. bloku
50 wyl:   equ #bcd8           ;adr. proc. wyl. bloku z kol.
60 dod:   equ #bcda           ;adr. proc. dol. bloku do kol.
70
80 init:
90      ld hl,blok              ;adres bloku zdarzenia
100     ld bc,#81ff            ;klasa zdarzenia i nr ROMu
110     ld de,obsl             ;adres proc. obsługi
120     jp włącz               ;inicj. i dolacz. bloku zdarzenia
130 dodaj:
140     ld hl,blfrm            ;adres bloku powrotu plamki
150     jp dod                 ;dolaczenie bloku do kolejki
160 blfrm: defs 2              ;blok powrotu plamki
170 blzd:  defs 7              ;blok zdarzenia
180 obsl:
190     ld a,68                 ;procedura obsługi
200     call testkl            ;testowanie naciśnięcia
210     ret a
220     ld a,20                 ;
230     call testkl            ;klawiszy CTRL i TAB
240     ret a                   ;wyjście, gdy nie naciś.
250     ld hl,blfrm            ;wylaczenie bloku z kolejki
260     call wyl               ;powrotu plamki na czas obsługi
270     call proc              ;wywołanie procedury
280     jp dodaj               ;dolaczenie bloku do kolejki
290 proc:
300 ;*****
310 ;tu powinien być umieszczony kod procedury wykonującej
320 ;potrzebne czynności (może to być np. HardCopy)
330 ;*****
340     ret

```

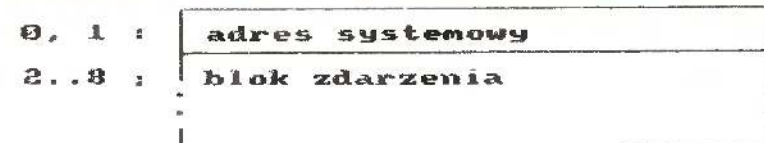




Rys. 1. Bok zdarzenia.



Rys. 2. Blok kolejki zwykłego zegara.



Rys. 3. Bok kolejki szybkiego zegara i kolejki powrotu plamki obrazu.

szkieletów programów dołączających bloki zwykłych zdarzeń asynchronicznych do trzech rodzajów kolejek oraz przykład 4 zawierający program napisany w języku BASIC, po uruchomieniu którego w lewym górnym rogu ekranu monitora będzie wyświetlany zegar czasu rzeczywistego. W przykładach 1 — 3 procedury obsługi zdarzeń muszą znajdować się w środkowych 32 kB pamięci RAM. Mogą one wywoływać procedury systemowe zapamiętane w dolnym ROM'ie z wyjątkiem obsługujących pamięci zewnętrzne.

Przykład ten prezentuje sposób dołączenia bloku zdarzenia do kolejki powrotu plamki obrazu. Może z powodzeniem służyć do wykonywania w dowolnym momencie Hard Copy ekranu na drukarkę. W miejscach komentarzy po etykiecie „proc:” należy umieścić kod procedury

kopiowania ekranu. Program po załadowaniu edytorem GENA31 uruchamia się z poziomu języka BASIC instrukcją CALL 42352. Od tego momentu będzie on działał w „tle” programu, który będzie później uruchomiony a kopię ekranu można uzyskiwać po jednoczesnym wciśnięciu klawiszy CTRL i TAB.

Przykład powyższy pokazuje jak dołączyć blok zdarzenia do kolejki zegara. Można w ten sposób uzyskać powtarzanie jakichś operacji w regularnych odstępach czasu z zadanym opóźnieniem od wybranego momentu. Odpowiednie wartości (w jednostkach równych 1/50 s.) należy podać w momencie dołączania bloku do kolejki (w przykładzie linie 120 i 130). Można w oparciu o tę kolejkę zaprogramować zegar czasu rzeczywistego. Okazało się jednak, że nie jest on wystarczająco

dokładny. Wynika to z faktu nie dość częstego wywoływania procedury obsługi zdarzenia (co 1/50 sekundy) oraz niedokładności układów zegarowych mikrokomputera.

Przykład trzeci podaje sposób dołączenia bloku do kolejki szybkiego zegara. W oparciu o tę kolejkę został zaprogramowany zegar czasu rzeczywistego, który znajduje się poniżej. Należy ostrzec, że procedura obsługi zdarzeń z tej kolejki powinna być możliwie najkrótsza pod względem czasu działania, gdyż wywoływana jest co 1/300 sekundy i w związku z tym pochłania sumarycznie bardzo dużo czasu procesora.

W liniach od 10 do 310 w postaci instrukcji DATA znajduje się kod binarny programu dołączającego blok zdarzenia do kolejki szybkiego zegara oraz kod procedury aktualizującej i wyświetlającej zegar. Linie 320 do 430 wczytują dane do pamięci RAM, a linie 450 do 460 ustawiają początkowe wskazanie zegara i uruchamiają go. Zegar wskazuje bardzo dokładnie czas niezależnie od wykonywania innych czynności przez komputer pod warunkiem, że nie korzysta się z dyskiety. Wynika to z tego, że na czas transmisji dyskowych blokowane są przez system przerwanie i zegar będzie się opóźniał.

Roman Sawicki

## Przykład 2

```

10      org 40960          ;adres umieszczenia kodu
20
30      init: equ #bcef    ;adr. proc. inicj. bloku zdarzenia
40      ad:   equ #bce5    ;adr. proc. dol. bloku do kolejki
50
60      ld hl,blok        ;adres bloku zdarzenia
70      ld bc,#B1ff       ;klasa zdarzenia i nr ROM
80      ld de,obsl        ;adres proc. obsługi
90      call init         ;inic. bloku zdarzenia
100     ld hl,blzeg
110     jp ad             ;dotaczenie bloku do kolejki
120     blzeg:
130     defs 2
140     blok:
150     defs 7
160     obsl:              ;procedura obsługi
170     ;*****
180     ;tu powinien byc umieszczony kod procedury wykonujacej potrzebne
190     ;czynnosci (moze to byc np. zegar czasu rzeczywistego)
200     ;*****
210     ret

```

## Przykład 3

```

10      org 40960          ;adres umieszczenia kodu
20
30      init: equ #bcef    ;adr. proc. inicjacji bloku
40      dod:  equ #bce9    ;adr. proc. dol. bloku do kolejki
50
60
70      ld hl,blok        ;adres bloku zdarzenia
80      ld bc,#B1ff       ;klasa zdarzenia i nr ROMu
90      ld de,proc        ;adres proc. obsługi
100     call init         ;inic. bloku zdarzenia
110     ld hl,blzeg
120     ld de,50           ;pocz. wartosc liczn. zegarowego
130     ld bc,100         ;wartosc odtworzenia licznika
140     jp dod            ;dolaczenie bloku do kolejki zeg.
150     blzeg: defs 6      ;blok zegara
160     blok:
170     defs 7             ;blok zdarzenia
180     proc:              ;procedura obsługi
190     ;*****
200     ;tu powinien byc umieszczony kod procedury wykonujacej
210     ;potrzebne czynnosci (moze to byc np. zegar czasu rzeczywistego)
220     ;*****
230     ret

```

## Przykład 4

```

10 DATA &A100,&012B
20 DATA &21,&27,&A1,&01,&FF,&81,&11,&2F,&A1,&CD,&041B
30 DATA &EF,&BC,&21,&25,&A1,&C3,&E3,&BC,&00,&00,&04F4
40 DATA &00,&01,&00,&00,&2C,&01,&52,&2E,&53,&2E,&012F
50 DATA &20,&46,&33,&20,&57,&49,&49,&00,&00,&00,&01A2
60 DATA &00,&00,&00,&00,&00,&00,&00,&2A,&1B,&A1,&00E3
70 DATA &2B,&22,&18,&A1,&7D,&B4,&C0,&21,&2C,&01,&0345
80 DATA &22,&18,&A1,&3E,&17,&CD,&1E,&BB,&2B,&0F,&030D
90 DATA &3E,&44,&CD,&1E,&BB,&2B,&0B,&3A,&15,&A1,&034B
100 DATA &EE,&01,&32,&15,&A1,&3A,&14,&A1,&3C,&FE,&0400
110 DATA &3C,&3B,&2F,&AF,&32,&14,&A1,&11,&20,&00,&026A
120 DATA &CD,&21,&A2,&3A,&13,&A1,&3C,&FE,&3C,&3B,&042C
130 DATA &18,&AF,&32,&13,&A1,&11,&1E,&00,&CD,&21,&02CA
140 DATA &A2,&3A,&12,&A1,&3C,&FE,&18,&3B,&01,&AF,&03C9
150 DATA &32,&12,&A1,&1B,&0B,&32,&13,&A1,&1B,&03,&0206
160 DATA &32,&14,&A1,&3A,&15,&A1,&FE,&00,&CB,&2A,&03C7
170 DATA &29,&B7,&22,&16,&A1,&11,&00,&00,&ED,&53,&030A
180 DATA &29,&B7,&3A,&33,&B7,&F5,&AF,&32,&33,&B7,&04C4
190 DATA &2A,&26,&B7,&E5,&ED,&53,&26,&B7,&3A,&2E,&0471
200 DATA &B7,&F5,&3E,&02,&21,&2E,&B7,&B6,&77,&CD,&04EC
210 DATA &17,&A2,&3E,&3C,&CD,&5D,&BB,&3A,&12,&A1,&0405
220 DATA &CD,&F8,&A1,&CD,&11,&A2,&3A,&13,&A1,&CD,&05A1
230 DATA &F8,&A1,&CD,&11,&A2,&3A,&14,&A1,&CD,&F8,&05CD
240 DATA &A1,&3E,&3E,&CD,&5D,&BB,&CD,&17,&A2,&F1,&0579
250 DATA &32,&2E,&B7,&E1,&22,&26,&B7,&F1,&32,&33,&044D
260 DATA &B7,&2A,&16,&A1,&22,&29,&B7,&C9,&2E,&00,&0391
270 DATA &FE,&0A,&3B,&05,&2C,&DE,&0A,&1B,&F7,&C6,&042E
280 DATA &30,&F5,&7D,&C6,&30,&CD,&5D,&BB,&F1,&CD,&063B
290 DATA &5D,&BB,&C9,&3E,&3A,&CD,&5D,&BB,&C9,&2A,&0531
300 DATA &2F,&B7,&7C,&65,&6F,&22,&2F,&B7,&C9,&F3,&04FA
310 DATA &2A,&1B,&A1,&19,&22,&1B,&A1,&FB,&C9,&00,&039B
320 MODE 2:RESTORE
330 READ adrespc,dlugosc:MEMORY adrespc-1
340 FOR i=0 TO dlugosc-1 STEP 10
350   adres=adrespc+i:sum=0
360   FOR j=adres TO adres+9
370     READ bajt$:POKE j,VAL(bajt$)
380     sum=sum+VAL(bajt$)
390   NEXT j
400 READ ksum:IF sum<>ksum THEN PRINT"Blad w ";20+i;"
   linii DATA ":END
410 PRINT"Linia nr. ";20+i;"Ok.":LOCATE 1,1
420 NEXT i
430 PRINT"Poprawny wpis danych"
440 PRINT"Podaj czas gg,mm,ss ";;INPUT gg,mm,ss
450 POKE &A112,gg:POKE &A113,mm:POKE &A114,ss
460 CALL &A100
470 MODE 2:DELETE

```



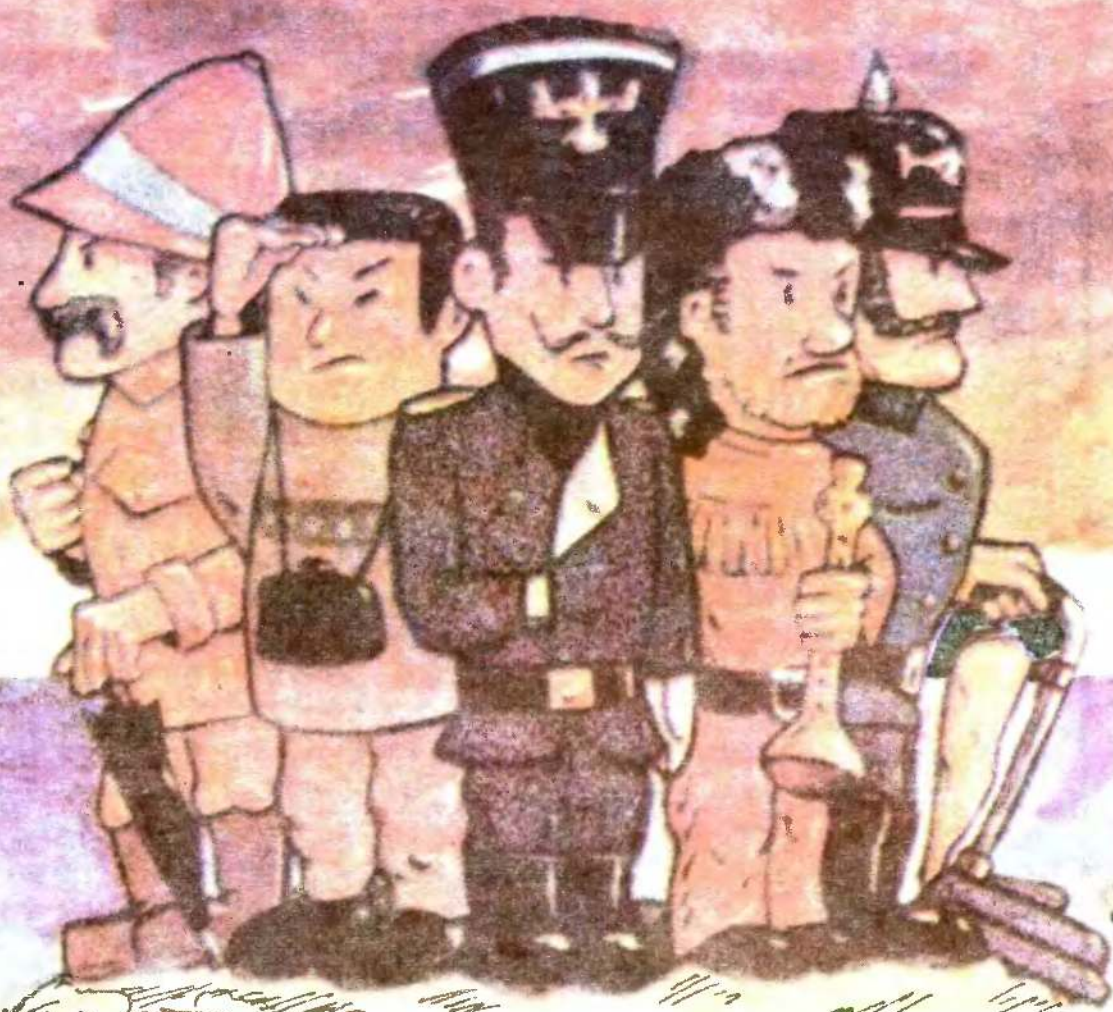
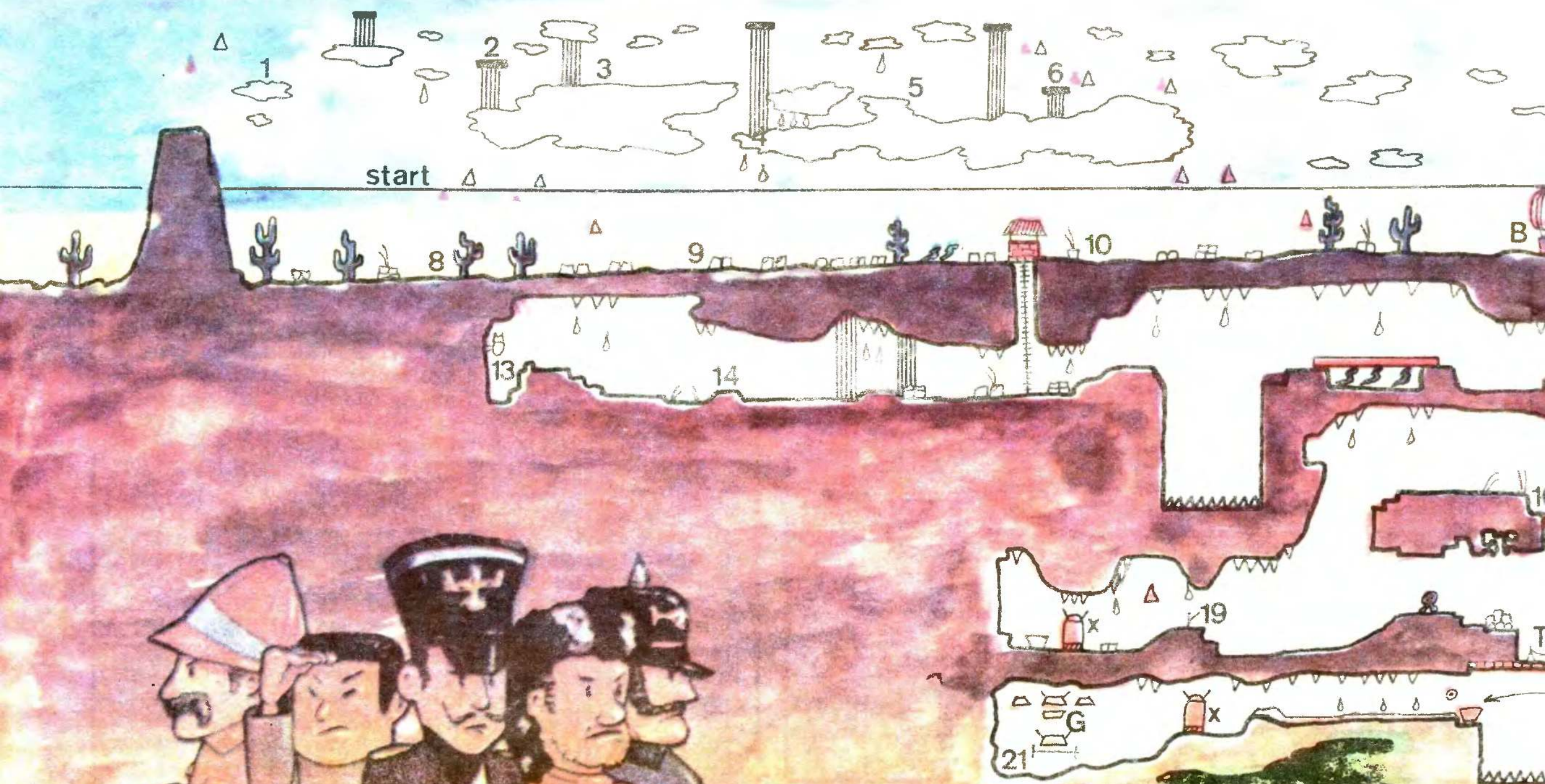
- 1 — Proch śr. mocy
- 2 — Drabinka sznurowa
- 3 — Ogłoszenie
- 4 — Srebrna łuska
- 5 — Rower jednośladowy
- 6 — Parasolka

- 7 — Tabletki antyradiacyjne
- 8 — Odkurzacz
- 9 — Flet
- 10 — Kula krykiotowa

- 11 — Wrzaski
- 12 — Ostrogi
- 13 — Proch małej mocy
- 14 — Pistolet
- 15 — Proch dużej mocy

- 16 — Przełącznik
- 17 — Baryłka piwa
- 18 — Kryształ energetyczny
- 19 — Wieszak

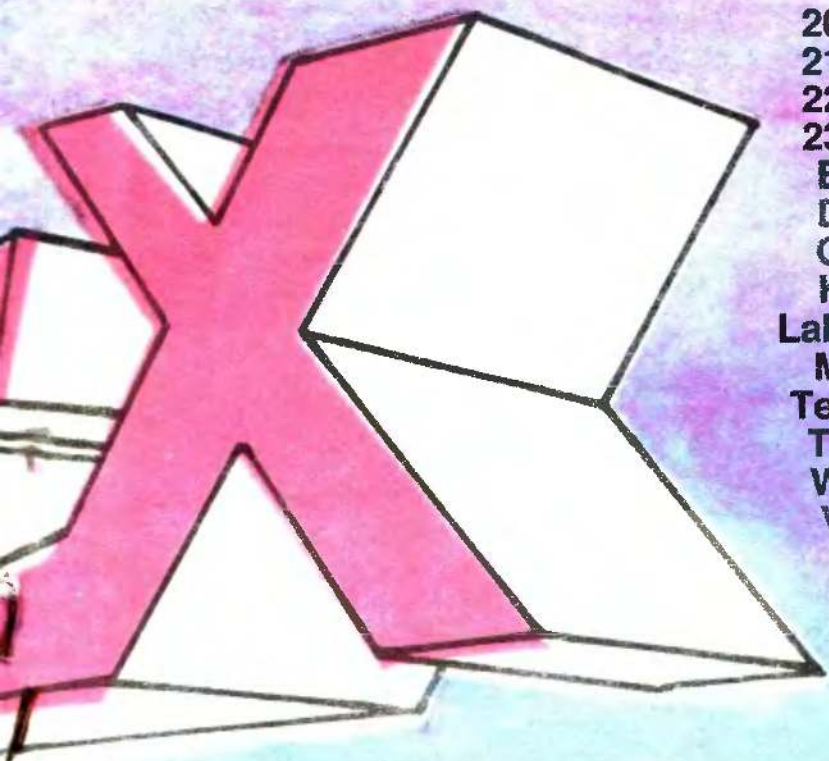
# TERRA ME



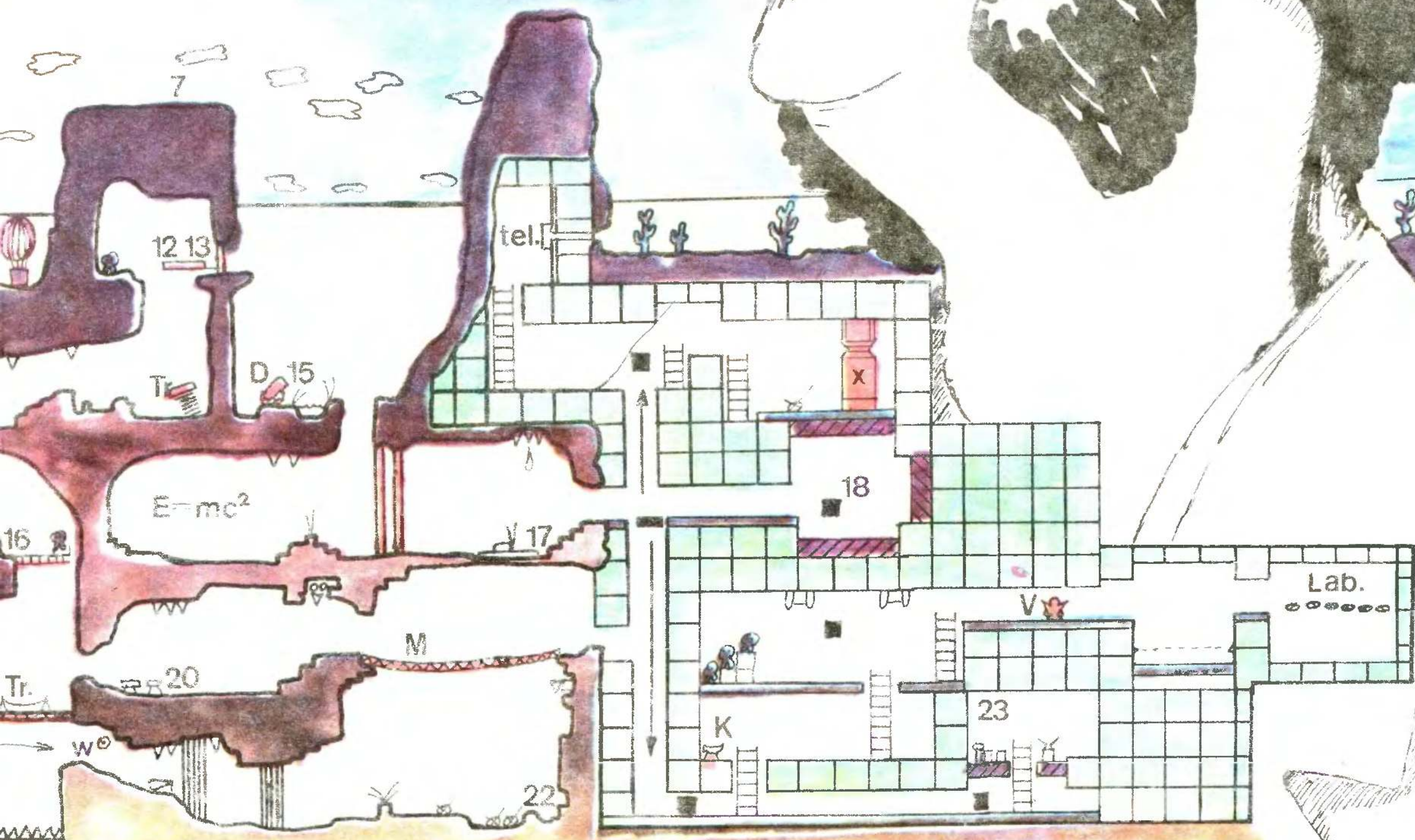
**Bajtek**







- 20 — Akumulator
- 21 — Stos atomowy
- 22 — Niespodzianka
- 23 — Filiżanka herbaty
- B — balon
- D — Działo
- G — Generator
- K — Kowadło
- Lab — Laboratorium
- M — most linowy
- Tel — Teleskop
- Tr — Trampolina
- W — Przełącznik kolejki
- V — Wampir



*Regaz Ass*

**P**rofesor Eyestrain już od dawna ostrzegał o grożącym Ziemi niebezpieczeństwie. Zbliża się wielki meteoryt, który po upływie tygodnia uderzy w Ziemię. Spowoduje to jeszcze większe skrzywienie osi ziemskiej, a w rezultacie stopienie się lodów na biegunach i powódź, która zaleje cały glob. Istnieje też możliwość, że lecąc przez atmosferę meteoryt rozgrzeje się tak, że nastąpi wybuch zgromadzonej na Ziemi broni atomowej.

O tym wszystkim profesor dowiedział się na podstawie swoich obliczeń i obserwacji Kosmosu. Nikt jednak nie dał wiary prorocztwom profesora. Urażony profesor ukrył się i buduje sobie kapsułę ratunkową, którą opuści Ziemię.

To też nie poruszyło ludzi. Jest jednak pięciu podróżników i odkrywców, którym nieobcy jest los Ziemi. Postanowili oni wybrać spośród siebie najwszechstronniejszego, który odnajdzie profesora i skłoni go do zmiany toru meteorytu.

Jeden z podróżników wyrusza więc na poszukiwania. Ma tylko tydzień i to nie cały. Okazuje się, że profesor Eyestrain ukrył się w stworzonym przez siebie świecie zwanym Terramex. Światek składa się z nieba, pustyni, jaskiń i miasta. Na samym końcu drogi, w

najdalszym zakątku miasta, ukryty jest profesor.

Wędrując po Terramexie napotkać można wiele dziwnych przedmiotów. Są to wynalazki profesora, które pomagają dotrzeć do jego kryjówki. Oprócz tego podróżnikowi potrzebne będą elementy do budowy specjalnego urządzenia, które pozwoli na zmianę toru lotu meteorytu.

Razem z bohaterem wędruje kilku tragarzy, którzy niosą znalezione przedmioty, bowiem bohater może naraz nieść tylko jeden. Wymiana przedmiotu z najbliższym tragarzem następuje po wciśnięciu klawisza S, zaś przesuwanie rzędu tragarzy (widocznych w okienku w prawym dolnym rogu) możliwe jest za pomocą klawiszy 1 i 2.

Na drodze czyha wiele niebezpieczeństw. Ze szczelin skalnych wysuwają się jadowite węże, podobnie jak z wiklinowych koszy. Te drugie można jednak chwilowo uspić grając na znalezionym flecie.

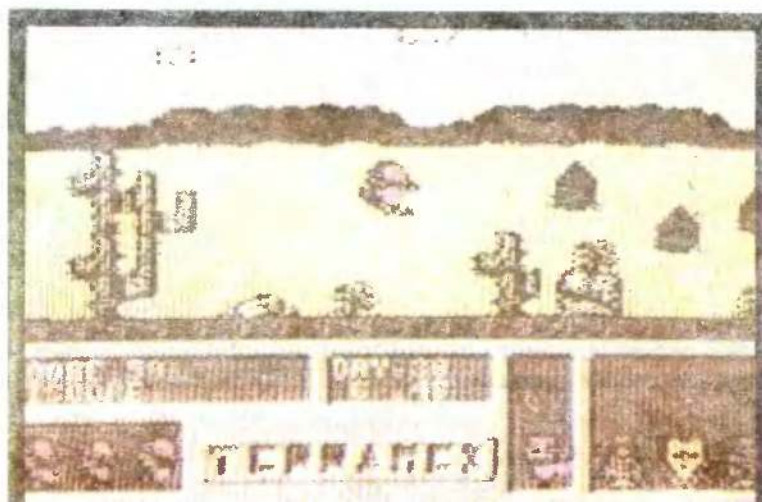
Każdy z napotkanych przedmiotów służy do czegoś i na pewno będzie przydatny. Po wciśnięciu klawisza T podróżnik zastanawia się chwilę i podpowiada grającemu, jaki przedmiot powinien teraz wykorzystać. Bardzo przydatny będzie parasol, działający jak spadochron przy skoku w przepaść. Bez pa-

rasola bohater odmawia skoku, kręcąc niechętnie głową.

Po skompletowaniu wszystkich przedmiotów i dotarciu do profesora zbuduje on urządzenie, które uratuje ludzkość. Do grającego należy teraz jego uruchomienie i odchylenie toru meteorytu. Przy odrobinie szczęścia wszystko powinno się udać.

**Komputer: ZX Spectrum 48/+, Amstrad/Schneider, Commodore 64, Atari ST**

*Gen*





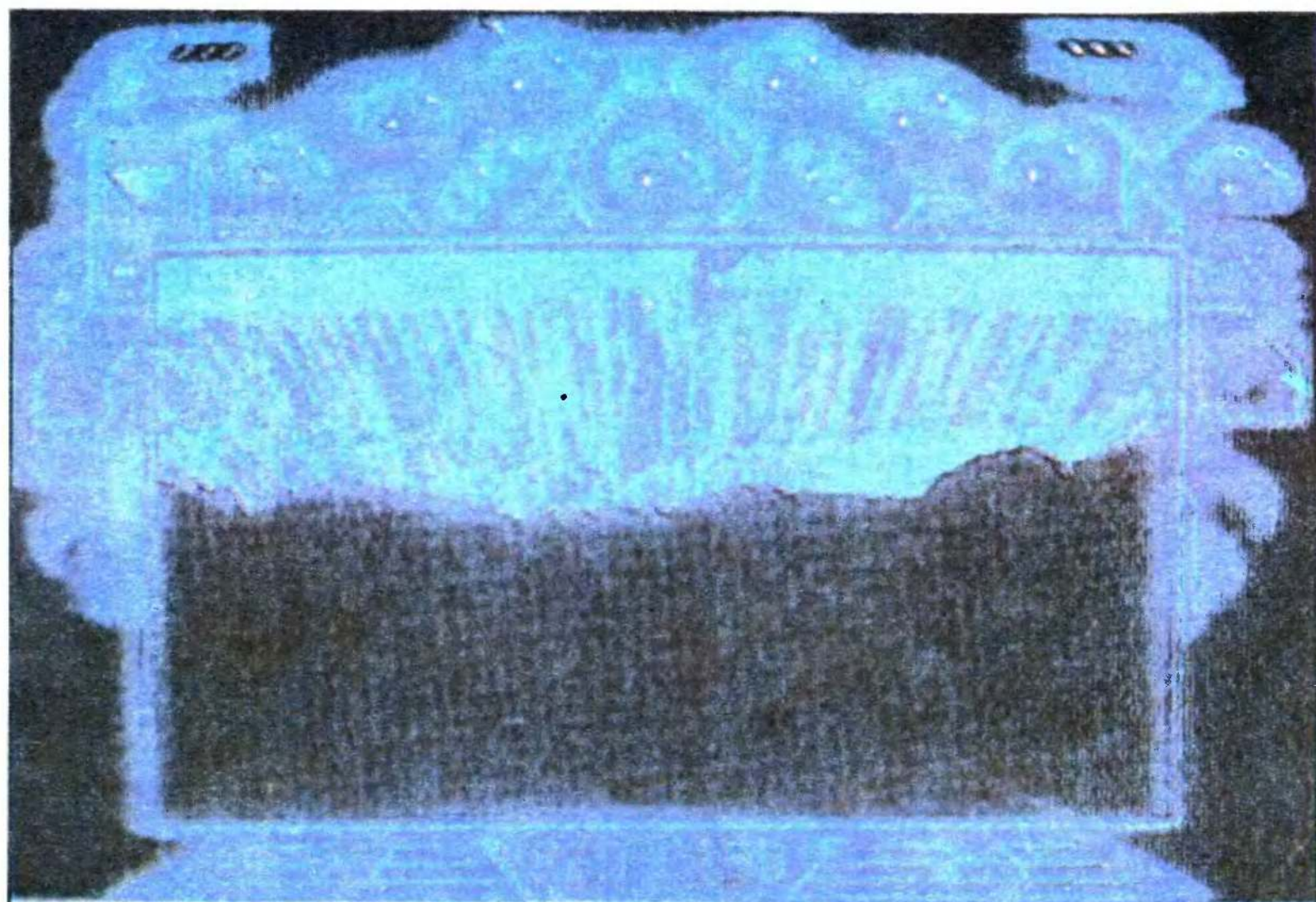


O nowych informacjach zawartych w dzisiejszej Liście (które będą oczywiście konsekwentnie prowadzone) możecie dowiedzieć się z artykułu „Innowacje”. A jak przedstawia się dzisiejsze notowanie?

Króluje Operation Wolf — wyladowywacz energii i emocji, skutkuje nawet po klasówce z matematyki. Zaraz za nim gra tego samego typu — Cybernoid, lecz nie aż tak relaksująca. Dalej bez niespodzianek, jeśli nie liczyć Kapitana Blooda, który w swej arce ze wspaniałą grafiką gości obok. Police Quest spada, już najwyższy czas.

Na ósme tegoroczne notowanie nadeszło 2977 propozycji. Czytelnicy głosowali na 96 tytułów.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 OPERATION WOLF	>		x	<
2 CYBERNOID	!	<	<	x
3 ROCKET RANGER	↑		x	
4 DRACONUS	↑	<	<	<
5 BMX SIMULATOR +	!	x	<	x
6 HOT SHOT	↓	>	<	x
7 CAPTAIN BLOOD	!	>	x	<
8 STREET FIGHTER	↓	>	<	x
9 GUNSHIP	↓	>	x	<
10 POLICE QUEST	↓	>		



## CAPTAIN BLOOD

**Wielki**, stalowy kolos mknie przez kosmiczną pustkę, z oszołamiącą prędkością oddalając się od Ziemi. Rozżarzone dysze silników drgają z powodu niesamowitej temperatury, którą zdolne są wytrzymać tylko stopy tytanu i wolframu. W pancernej szybie bocznego włazu odbijają się okrągłe kształty gwiazd. We wnętrzu wypukłych wieżyczek rysują się podłużne sylwetki najnowszych laserów HZH-2. Kosmiczne torpedy umocowane na wysięgnikach w dolnej części statku drgają wraz z całym kadłubem. Kilka radarów kręci się w sąsiedztwie niezliczonej ilości czujników wystających z wnętrza. Żaden statek nie zbliży się do tej latającej fortelicy, zdolnej w kilka sekund unicestwić największą nawet planetę.

A jednak właśnie z takiego krążownika porwano prezydenta galaktyki, lecącego na międzyplanetarnej wizje lokalną. W pewnej chwili stracono łączność ze statkiem. Wystąpiła ekipa ratunkowa oraz grupa komandosów DELTA FORCE, po kilkunastu dniach poszukiwań odnalazła transgalaktyk, lecz cała załoga była zabita nieznanym promieniowaniem, a sam prezydent zniknął. Wyznaczono nagrodę dla każdego, kto znajdzie miejsce uwięzienia prezydenta.

Każdy, kto załaduje tę grę na swój komputer, zmuszony jest odszukać zaginionego polityka. Wcielamy się więc w postać kapitana niewielkiego niszczyciela, o średnim standardzie wygody i zerowej liczbie przyjemności podczas podróży.

Zaczynamy w pobliżu planety, gdzie spotkamy pierwszego z informatorów. Jest to z reguły jej mieszkaniec, który w przypływie złego humoru powie: „Kapitanie Blood, ja Cię nie wyrzucam, ale wynoś się stąd.” Po tej milej wypowiedzi, wszyscy

z reguły rozwalają tę planetkę, nie martwiąc się o nic, a najmniej o ludzi. Niestety, ale takie bezmyślne, niepotrzebne zabijanie, nie wchodzi w tej grze w rachubę.

Porozumiewanie się z innymi postaciami jest proste i wymowne. Pod ekranem narysowane są tabliczki oraz głośnik-tłumacz. Każdy rysunek na płytce symbolizuje jedno słowo w jednym z trzech języków (angielskim, francuskim lub niemieckim). Gdy ułożysz już zdanie, wciśnij FIRE na głośniku i możesz czekać na odpowiedź. Niektóre słowa (zapisane w nawiasie) symbolizują zachowanie, np.: śmiech, wściekłość, przekleństwo itp. Naszym głównym celem (oprócz zabawy) jest wyciągnięcie od rozmówcy współrzędnych planet, na których mieszkają inni informatorzy. Aby to osiągnąć, musisz spodobać się stworkowi — wtedy on sam udzieli Ci niezbędnych rad. Potem możesz ulżyć sobie, zamieniając wszystko w gwiazdny pył.

Nasz niszczyciel wyposażony jest w wiele niezbędnych urządzeń. Znajdziesz tam rakiety anihilacyjne, teleskop, dzięki któremu możesz dokładnie obejrzeć planetę, lądownik, bez którego nie można rozmawiać w ogóle z nikim. Lądując na planecie musisz uważać na wysokie szczyty oraz rakiety.

Twój niszczyciel posiada także luki towarowe, w których możesz przewozić postacie, prosząc Cię o przeniesienie na rodzinną planetę. Pamiętaj jednak, że gdy zawieszysz go na niewłaściwą, to równie dobrze mógłbyś mu strzelić w głowę. Nie bądź więc sadystą i nie zabijaj bezbronnych.

**Firma:** Informatique

**Komputer:** ZX Spectrum 48, Commodore 64, Atari ST, Amiga.

Luke

## KRÓL I KRÓLOWA GIER



**Agnieszka Neuman** — VII klasa szkoły nr 309 w Warszawie — komputer ATARI, ulubiona gra: „Bruce Lee”, hobby: fotografia.

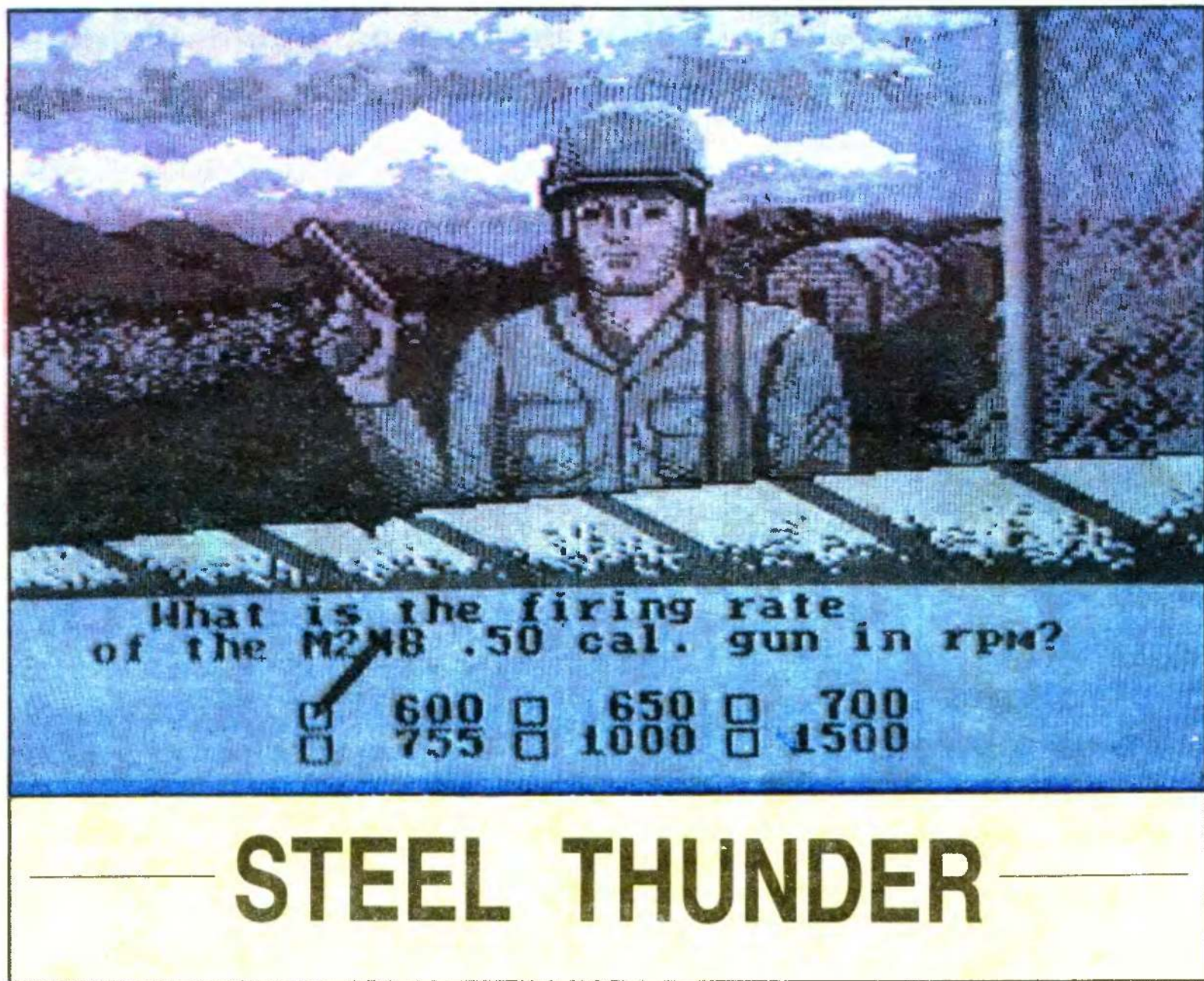


**Marcin Nakoneczny** — Szk. Podst. 289 w Warszawie, Klasa IV<sup>B</sup> — komp. SPECTRUM, gry: West Bank, Gryzor, hobby: tenis, motoryzacja.





S.O.S.



STEEL THUNDER

**Czy chcesz,** by na dźwięk grzmotu Twoje ciało drętwiało, a mózg zamieniał się w wodę? A może wolisz, aby dwa słowa — „Żelazny grzmot” przemieniły Cię w krwiożerczego szaleńca? Jeśli tak, wgraj „STEEL THUNDER” — program, którym bawią się tylko najwięksi mocarze.

Grę rozpoczynasz, wybierając czołg (jeden z czterech), którym będziesz sterował. Decyzja jest trudna i wiąże się zwykle z pogryzioną dyskietką lub skopanym kotem, więc podam ich ogólną charakterystykę:

**M1A1 ABRAMS**

Masa: 60 ton  
Działo: 120 mm  
Prędkość: 150 km/h  
Zasięg: 440 km  
Pancerz: 12 cm

**M60A3**

Masa: 54 tony  
Działo: 105 mm  
Prędkość: 80 km/h  
Zasięg: 480 km  
Pancerz: 15 cm

**M48A5**

Masa: 54 tony  
Działo: 105 mm  
Prędkość: 50 km/h  
Zasięg: 480 km  
Pancerz: 15 cm

**M3 BRADLEY**

Masa: 25 ton  
Działo: 25 mm  
Rakiety: TOW-2, zasięg 5 km  
Prędkość: 100 km/h  
Zasięg: 480 km  
Pancerz: 3 cm

Gdy wreszcie znajdziesz się za sterami maszyny, doznasz wrażenia porównywalnego z pobudką na szczycie wieży Eiffla. Przed oczami ukażą Ci się rzędy

lampek, liczników, przycisków itp., itd. Podanie ich funkcji, z pewnością zapobiegnie potłamanym palcom.

- 4 — kabina kierowcy, a w niej:
    - włącznik zasilania
  - 1 — rozrusznik silnika
  - 2 — noktowizor
  - CLR — gaśnice
  - 5 — pierwsze działo, a w środku:
    - przybliżenie
  - C — wysunięcie rakiet
  - U — stabilizator
  - M — opcje celownika (filtr, czysty widok)
    - podświetlanie celownika
  - N — włączenie komputera
  - 6 — drugie działo:
  - Z — aktywizacja granatów dymnych
  - C= — włącznik zasilania
- Inne funkcje:
- SHIFT — dokładność obrotu wieżyczką
  - RETURN — identyfikacja celu
  - CTRL — powrót do bazy
  - SPACE — automatyczne śledzenie celu
  - F1 — rzucenie zasłony dymnej
  - F3 — przełącznik działo-karabin
  - F5 — zasięg pocisków
  - F7 — rodzaj pocisków
  - 7 — zniszczenia
  - 8 — ilość amunicji, zniszczone cele
  - 9 — mapa

Walkę prowadzisz na poligonach, rozrzuconych po całym świecie. Na każdym z nich spotkasz odmienne warunki — ukształtowanie terenu, pogodę, nawierzchnię oraz uzbrojenie nie zawsze symulowanego przeciwnika.

Pamiętaj, że Twoje wyniki są zapisywane w służbowej karcie, bądź więc ostrożny. Życzę celnych trafień i potłamania działa.

**Firma: Accolade**  
**Komputer: Commodore 64**

Luke

INNOWACJE

Czy zauważyliście, że Lista Przebojów jest trochę inna, niż zwykle? Została nieco zmodyfikowana tabelka, zamieszczana przy tytułach gier z pierwszej dziesiątki. Wykorzystując stary standard wprowadziliśmy nowe informacje. O czym? Zaraz się dowiecie.

Komputery szesnastobitowe rozwijają się i powstaje na nie coraz więcej gier. Dlatego też uwzględniliśmy to w Liście Przebojów. Przyjrzyjcie się uważnie poniższemu kluczowi, by zapamiętać, który znaczek za co odpowiada:

- ATARI: znak „>” — ST, znak „<” — XL/XE, „x” — oba
- AMSTRAD: „>” — PC (głównie IBM PC), „<” — CPC, „x” — oba

- COMMODORE: „>” — AMIGA, „<” — 64, „x” — oba (wraz z 128)
- SPECTRUM: „>” — 128 (lub muzyka w wersji na 48), „<” — 48/+, „x” — oba (wersje gry i na 128 i na 48)

Najwięcej kłopotu będzie zapewne ze Spectrum, gdyż lewą częścią krzyżyka oznaczamy grę na 48, jeżeli wyposażona została w obsługę generatora dźwięku, ale również i grę, która istnieje tylko w wersji na 128 (np. „Where Time Stood Still” w pierwszym wydaniu).

Gen

Jestem posiadaczem mikrokomputera Atari 800XL. Mam kłopoty z grami AIR SUPPORT, THEATRE EUROPE, ARKANOID — nie wiem co zrobić w 33 komnacie. Liczę na pomoc czytelników.

**Krystian Baran os. Wysokie 10/39**  
**31-819 Kraków**

Proszę o pomoc w grach EASTERN FRONT, FORT APOCALYPSE, AIR SUPPORT. W zamian dam opisy gier BRUCE LEE, ROAD RACE, ZORRO, DROP ZONE i MINER 1049. Czekam z niecierpliwością.

**Marcin Szymanowicz ul. Manifestu**  
**Lipcowego 1b/33 44-244 Knurów**

Mam Atari 65XE. Poszukuję gier SPY vs SPY II, SPIKY HAROLD, GUN FRIGHT, THE ROCKY HORROR SHOW, URBAN UPSTART i GHOST CHASER w wersji kasetowej.

**Adrian Śliwiński ul. Lwowska 31/17**  
**53-515 Wrocław**

Szukam opisów gier PYJAMARAMA, BARBARIAN i ANTIRIAD. Posiadam COMMODORE 64.

**Leszek Krause ul. Herkulesa 12/3 67-200**  
**Głogów woj. legnickie**

Posiadam Atari 800XL. Poszukuję kodów do KENNEDY APPROACH i gry ASTEROIDS. Zapłacę lub wymienię na inne gry.

**Adrian Czabanowski ul. Jaskółcza 14**  
**48-300 Nysa**

Proszę o dokładne opisy programów WHAM! THE MUSIC BOX, TASWORD TWO (polska wersja). Szukam też POKE'ów do gier COMMANDO, GILLIGAN'S GOLD, PANAMA JOE i THE GOONIES. Odwdzięczę się ułatwieniami do innych gier.

**Jacek Wajda ul. Szekspira 2/26**  
**01-913 Warszawa**

Poszukuję opisów gier CHIMERA, FORBIDDEN FOREST, SPINDIZZY, ZENI. Posiadam Atari 800XL.

**Łukasz Majewski 55-050 Sobótka**  
**ul. Wrocławska 14e**

Nawiążę kontakt listowy z posiadaczami gry telewizyjnej COLECO VISION firmy CBS ELECTRONICS. Nie mam do niej kaset z grami. Mogę je odkupić lub pożyczyć od kogoś, oczywiście za zapłatą.

**Sebastian Karbowski 82-439 Waplewo**  
**Wielkie woj. elbląskie**

Jestem posiadaczem Atari 130XE. Proszę o udostępnienie mi gier COMMANDO, ARKANOID, DROP ZONE. Poszukuję też opisów do gier MOLECULE MAN, GREMLINS, FROGGER II i CAVERN OF KHAFKA.

**Bartosz Wyspiański ul. Mickiewicza**  
**107h/6 64-920 Piła**

Nie wiem jak skończyć grę MONTEZUMA'S REVENGE. Proszę o pomoc.

**Jakub Winnicki ul. Akacjowa 18a/3**  
**47-330 Zdzieszowice**

Jestem użytkownikiem Atari 65XE z magnetofonem XC12. Uprzejmie proszę o dokładny opis gry TOMAHAWK. Nie mogę sobie także poradzić z kolejnymi etapami gry RAID OVER MOSCOW. Poszukuję też gier BARBARIAN, POP EYE, GHOSTBUSTERS, SPY vs SPY i STARQUAKE. W zamian oferuję wiele innych gier. Liczę na waszą pomoc.

**Cezary Jakmik os. Centrum 19/6**  
**16-100 Sokółka**

Posiadam Atari 800XL. Mam trudności z grami STARQUAKE, MONTEZUMA'S REVENGE, EASTERN FRONT, SUBMARINE COMMANDER, SPY HUNTER.

**Dawid Kwaśniewski ul. Ogrodowa 8/10**  
**57-540 Łądek Zdrój**

Chciałbym posiadać gry IKARI WARRIORS, COMMANDO, ELEVATOR ACTION. Ostatnia nadzieja w czytelnikach Bajtka. Posiadam Atari 130XE. Za okazaną pomoc będę bardzo wdzięczny.

**Adam Bilan ul. Żwirki 1a/8**  
**63-400 Ostrów Wlkp.**



## POLSKIE LITERY NA DRUKARCE I PLOTERZE SONY

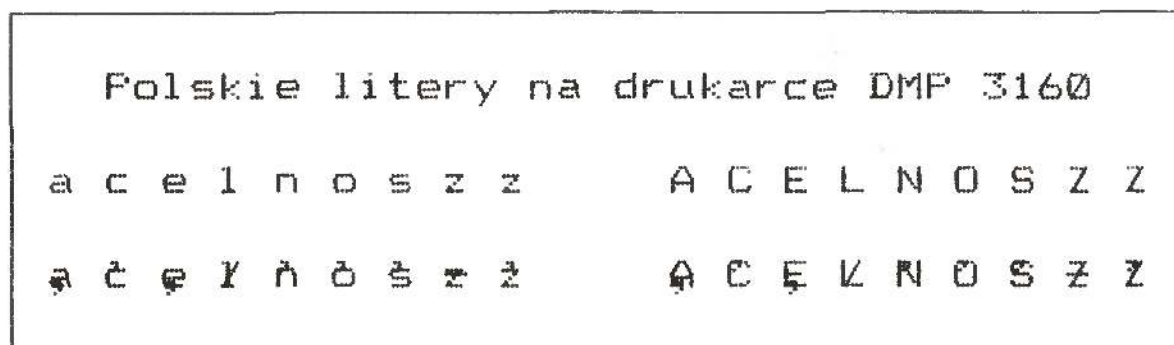
**Chociaż większość nowoczesnych drukarek pozwala na definiowanie dodatkowych alfabetów, to praktycznie na każdej z nich wygląda to inaczej. Z tego powodu wiele edytorów tekstu posiada własne programy instalacyjne dostosowane do kilku najbardziej typowych drukarek.**

Innym często stosowanym rozwiązaniem tego problemu jest wymiana całego generatora znaków znajdującego się w pamięci ROM. Czasami jednak, gdy nie dysponujemy dokumentacją sprzętu lub drukarka nie pozwala na definicję nowych znaków, trzeba posłużyć się inną metodą. Polskie litery można otrzymać ze złożenia litery łacińskiej i odpowiedniego znaku diakrycznego, np. a i przecinek dają ą, l i ukośna kreska ł. Nie jest to rozwiązanie najbardziej eleganckie, ale skutecznie wyróżnia w tekście polskie litery i działa na każdej drukarce, a także na ploterze Sony.

Bezpośrednie zastosowanie tego sposobu przy korzystaniu z edytora byłoby mało wygodne, gdyż każda polska litera byłaby reprezentowana na ekranie przez trzy znaki: literę łacińską, kod powrotu karetki o jeden znak (CTRL H) i symbol diakryczny. Znacznie wygodniej jest zmienić generator znaków ekranu, przyporządkowując odpowiednie kody osiemnastu polskim literom, a drukowanie tekstu z zamianą tych kodów na opisaną sekwencję powierzyć programowi. Na listingu 1 przedstawiono taki program napisany w Turbo Pascalu na komputerze Amstrad PCW. Bez żadnych zmian można z niego korzystać także na CPC 6128 (w CP/M'ie) i na IBM PC. Bardzo ważnym problemem jest wybór kodów dla polskich liter. W tej chwili obowiązuje wiele niezgodnych

wzajemnie standardów. Kody zaproponowane w tekście programu nie pokrywają się z żadnymi z nim. Można je zmienić wstawiając odpowiednie wartości stałych \_a..\_x, \_A..\_X. Uruchamiając program musimy podać dwa parametry — nazwy zbiorów wejściowego i wyjściowego. Pierwszy ze zbiorów zawiera tekst utworzony przy pomocy edytora, natomiast w drugim, generowanym przez program, kody polskich liter zastąpione są przez ciąg trzech znaków. Jeśli w miejsce drugiego parametru podamy LST:, to przetłumaczony tekst zostanie wysłany od razu na drukarkę. Na rysunkach 1 i 2 podano przykładowe wydruki demonstrujące działanie programu na drukarce Amstrad DMP 3160 i ploterze PRN C41 firmy Sony. Niewielka modyfikacja funkcji PLChar i zdefiniowanie alternatywnego zestawu polskich liter pozwala otrzymać program tłumaczący tekst z jednego wariantu na drugi. Zmiany te przedstawiono na listingu 1. Osoby posługujące się językiem Basic mogą skorzystać z programu znajdującego się na listingu 2. Na początku zdefiniowano używane macierze, a podstawową częścią programu jest procedura kopiowania ciągu znaków (string) A\$ na B\$ z zamianą kodów odpowiadającym polskim literom na sekwencje trzech znaków: litera łacińska, CTRL H i znak diakryczny. Opisane w artykule rozwiązania dotyczą wyłącznie drukarki lub plotera. Problem zmiany generatora znaków ekranu i samej klawiatury, tak aby uwzględniała polskie litery pozostawiono Czytelnikowi. W przypadku Amstrada PCW 8256/8512 lub CPC 6128 potrzebne informacje można znaleźć w poprzednich numerach Bajtek: J. Mayer, „Własne znaki na ekranie Joyce'a” nr 5/89, J. Mayer, „Rekonfiguracja klawiatury” nr 10/88, J. Mayer, „Symbol RSX dla CPC 6128”, nr 11/88.

*Jonasz Mayer*



Rys. 1

```

Program PL Print;
{*****}
{ Program kopiuje zbior wejsciowy na wyjsciowy zamieniajac kody }
{ polskich liter przez sekwencje trzech znakow: kod odpowiedniej }
{ litery (bez znakow diakrytycznych), BackSpace, i znak diakry- }
{ tyczny (przecinek, ukośna kreska, apostrof). }
{ Wwołanie: PLR zbior_wejsciowy zbior_wyjsciowy }
{ np. PLR tekst prn - wydruk zbioru 'tekst' na drukarce }
{ Kody polskich liter, zalezne od komputera, nalezy podac w pro- }
{ gramie w deklaracjach stałych a.._x, _A.._X. }
{ (C) JM Styczen 1989 }
{*****}
type
  Str3 = String[3];  Str20 = String[20];  CharSet = Set of Char;
const
  ( ----- POLSKIE LITERY ----- )
  _a=#224;  _c=#245;  _e=#225;  _l=#226;  _n=#249;
  _o=#227;  _s=#246;  _z=#233;  _x=#228;          { MALE }
  _A=#192;  _C=#213;  _E=#193;  _L=#194;  _N=#217;
  _O=#195;  _S=#214;  _Z=#201;  _X=#196;          { DUZE }

  PL      :   charset = [ _a, _c, _e, _l, _n, _o, _s, _z, _x,
                        _A, _C, _E, _L, _N, _O, _S, _Z, _X ];
  BSp = #08; { BACKSPACE }
var
  Source, Dest : text;   SourceName, DestName : Str20;
  PLstr : Str3;         ch : char;

Function PLChar (ch : Char) : Str3;
var
  T : Str3;
begin
  T := '';
  case ch
  of
    _a : T := 'a' + BSp + ' ';
    _c : T := 'c' + BSp + ' ';
    _e : T := 'e' + BSp + ' ';
    _l : T := 'l' + BSp + ' ';
    _n : T := 'n' + BSp + ' ';
    _o : T := 'o' + BSp + ' ';
    _s : T := 's' + BSp + ' ';
    _z : T := 'z' + BSp + ' ';
    _x : T := 'x' + BSp + ' ';
    _A : T := 'A' + BSp + ' ';
    _C : T := 'C' + BSp + ' ';
    _E : T := 'E' + BSp + ' ';
    _L : T := 'L' + BSp + ' ';
    _N : T := 'N' + BSp + ' ';
    _O : T := 'O' + BSp + ' ';
    _S : T := 'S' + BSp + ' ';
    _Z : T := 'Z' + BSp + ' ';
    _X : T := 'X' + BSp + ' ';
  end;
  PLChar := T;
end; { keyPL }

begin
  if ParamCount = 2
  then begin
    SourceName := ParamStr(1);   DestName := ParamStr(2);
    assign (Source, SourceName);  assign (Dest, DestName);
    reset (Source);               rewrite (Dest);

    while not eof (Source)
    do begin
      while not eoln (Source)
      do begin
        read (Source, ch);
        if ch in PL
        then plstr := PLChar(ch)
        else plstr := ch;

        write (Dest, plstr);
      end;
      readln (Source);
      writeln (Dest);
    end;
    close (Source);
  end
  else writeln ('Brak parametrow. Skladnia: PLR Source Dest ');
end.
  
```

Listing 1. Program PL\_Print

```

const
  ( ----- POLSKIE LITERY ZESTAW II ----- )
  _qa=#134;  _qc=#149;  _qe=#130;  _ql=#145;  _qn=#164;
  _qo=#148;  _qs=#132;  _qz=#129;  _qx=#168;          { MALE }
  _qA=#143;  _qC=#151;  _qE=#144;  _qL=#146;  _qN=#165;
  _qO=#153;  _qS=#142;  _qZ=#154;  _qX=#173;          { DUZE }

Function PLChar (ch : Char) : Str3;
{*****}
{# Zamiana jednego zestawu polskich liter na drugi zestaw #}
{*****}
var
  T : Str3;
begin
  T := '';
  case ch
  of
    _a : T := _qa;
    _c : T := _qc;
    _e : T := _qe;
    _A : T := _qA;
    _C : T := _qC;
    _E : T := _qE;
  end;
  PLChar := T;
end.
  
```



```

_l : T := _ql;      _L : T := _qL;
_n : T := _qn;      _N : T := _qN;
_o : T := _qo;      _O : T := _qO;
_s : T := _qs;      _S : T := _qS;
_z : T := _qz;      _Z : T := _qZ;
_x : T := _qx;      _X : T := _qX;
end;
PLChar := T;
end; { keyPL }

```

Listing 1'. Nowa funkcja PLChar z drugim zestawem polskich liter

```

1 REM *****
2 REM Deklaracja macierzy
3 REM *****
5 DIM PL$(18), PLB$(9), PLC$(9)
6 REM *****
7 REM polskie litery bez znakow diakrytycznych
8 REM *****
10 PLB$(1) = "a": PLB$(2) = "c": PLB$(3) = "e":
20 PLB$(4) = "i": PLB$(5) = "n": PLB$(6) = "o":
30 PLB$(7) = "s": PLB$(8) = "z": PLB$(9) = "z":
38 REM *****
40 REM znaki diakrytyczne
48 REM *****
50 PLC$(1) = " ", " ": PLC$(2) = " ' ", " ' ":
60 PLC$(4) = " / ", " / ": PLC$(5) = " ' ", " ' ":
70 PLC$(7) = " - ", " - ": PLC$(8) = " ' ", " ' ":
78 REM *****
80 REM kody malych polskich liter
88 REM *****
110 PL$(1)=CHR$(134): PL$(2)=CHR$(149): PL$(3)=CHR$(130)
120 PL$(4)=CHR$(145): PL$(5)=CHR$(164): PL$(6)=CHR$(148)
130 PL$(7)=CHR$(132): PL$(8)=CHR$(129): PL$(9)=CHR$(168)
132 REM *****
135 REM kody duzych polskich liter
138 REM *****
140 PL$(10)=CHR$(143): PL$(11)=CHR$(151): PL$(12)=CHR$(144)
150 PL$(13)=CHR$(146): PL$(14)=CHR$(165): PL$(15)=CHR$(153)
160 PL$(16)=CHR$(142): PL$(17)=CHR$(154): PL$(18)=CHR$(173)
165 REM *****
170 GOTO 1000 : REM skok na poczatek programu
175 REM *****
198 REM *****
200 REM procedura zamiany A$ na B$. Kazda polska
202 REM litera w A$ zastepowana jest litera bez
203 REM znakow diakrytycznych, BackSpace'm
204 REM (chr$(08)) i odpowiednim znakiem diakr.
208 REM *****
210 B$ = ""
220 FOR I=1 TO LEN(A$)
225   Z$=LEFT$(A$,1) : A$ = MID$(A$,2,255)
230   FOR J=1 TO 9 : REM male p. litery
240     IF Z$<>PL$(J) THEN GOTO 270
250     B$ = B$ + PLB$(J) + CHR$(8) + PLC$(J)
260     GOTO 350
270   NEXT J
280   FOR J=1 TO 9 : REM duze p. litery
290     IF Z$<>PL$(9+J) THEN GOTO 330
310     B$ = B$ + UPPER$(PLB$(J)) + CHR$(8) + PLC$(J)
320     GOTO 350
330   NEXT J
340   B$ = B$ + Z$ : REM POZOSTALE LITERY
350 NEXT I
360 RETURN
1000 REM *****
1010 REM poczatek programu
1020 REM *****
1030 INPUT A$
1040 GOSUB 200
1050 LPRINT B$
1060 IF B$="" THEN STOP
1070 GOTO 1030

```

Listing 2. Program w Basic'u pozwalajacy drukowac polskie litery na ploterze lub drukarce.

### Polskie litery na ploterze SONY

```

a c e l n o s z z   A C E L N O S Z Z
a c e l n o s z z   A C E L N O S Z Z

```

Rys. 2

# CO?

JEST TRUDNE

JEST PROSTE

W wielu dyskusjach w redakcji spotygam się z zarzutami swoich kolegów, że artykuły prezentowane na łamach klanu Amstrada są zbyt poważne i mało zrozumiałe dla przeciętnego użytkownika. Panuje opinia, że jedynym językiem programowania jakim można zwracać się do czytelnika Bajtka jest wyłącznie BASIC. Dopuszczalne są wstawki w kodzie maszynowym, najlepiej bez listingu źródłowego, prezentowane w formie długiego ciągu instrukcji DATA. Natomiast system operacyjny CP/M, albo Turbo-Pascal ma być domeną tylko profesjonalistów.

Niestety nawet osoba pisząca proste programy w Basic'u oprócz znajomości tego języka musi wykazać się pewną wiedzą na temat szcztkowego systemu operacyjnego, jakim dysponują najprostsze komputery domowe takie jak Spectrum, Atari, czy Commodore. Nie zdajemy sobie często sprawy, że napisanie komendy SAVE lub LOAD, jest właśnie odwołaniem do systemu.

Z kolei wprowadzenie nowego programu w Basic'u wymaga pewnych możliwości edycyjnych, takich jak: wpisanie nowej linii tekstu, ewentualne poprawienie go, powielenie fragmentu programu itp. We wspomnianych wyżej komputerach wszystkie te funkcje, a więc interpreter języka, edytor i system operacyjny są zintegrowane w jedną całość. Pytanie czy jest to proste i wygodne rozwiązanie?

Użytkownicy Spectrum, a zapewne i innych komputerów, wiedzą, że nawet tak prymitywna opcja jak przenumerowanie wierszy programu nie jest dostępna w standartowym Basic'u. Kopiowanie zbiorów, albo ich przeglądanie, czy to w postaci tekstu, czy traktowanych jako kolejne bajty, wymaga zastosowania specjalnego programu. Napisanie dowolnego tekstu przy pomocy edytora Basic'a jest też kłopotliwe.

Dochodzi do tego, że jeśli chcemy zrobić coś konkretnego to musimy uruchomić jakiś program, który na każdym komputerze wygląda inaczej, choć często pozwala wykonywać te same czynności. Z tego powodu posiadacz Spectrum doskonale posługujący się swoim komputerem i znający większość użytecznego nań oprogramowania, staje bezradny przed sprzętem innej firmy. Oczywiście posiadacze Commodorów, czy Atari też doznają podobnych wrażeń.

Nie mamy takich problemów jeśli korzystamy z uniwersalnego systemu operacyjnego, który dzięki swej popularności, zainstalowany jest na dużej liczbie zupełnie różnych komputerów. Takim systemem jest właśnie CP/M, dostępny w Polsce głównie na 8-bitowym sprzęcie firmy AMSTRAD.

Oprócz tego można pracować z CP/M'em na Commodorze 128. Portugalskie przystawki Timex'a zawierające dwie stacje dysków także pozwalają na korzystanie z tego systemu operacyjnego na Spectrum. CP/J na polskiego Juniora jest też praktycznie CP/M-em. Do C 64 można dokupić interfejs z procesorem Z80 dającym podobne możliwości.

Na IBM PC zwyciężył inny system operacyjny MS-DOS, ale jest on bardzo podobny do CP/M i wywodzi się wręcz z niego. Nie jest to oczywiście jedyny system tego komputera, a oprócz UNIX'a, bardzo popularnego na dużych maszynach, można również korzystać z 16-bitowych wersji CP/M'u 86.

Komputery wyposażone w uniwersalny system operacyjny nie zgłaszają się interpreterem BASICA po włączeniu. Oczywiście jest on w dalszym ciągu dostępny, ale nie jest szczególnie wyróżniony. Traktuje się go na równi z innymi programami, wywołując z poziomu systemu operacyjnego tak samo jak woła się edytor tekstu, kompilator Pascala, albo inne narzędzia.

Oprócz tego niektóre, ogólnie użyteczne programy, nie muszą być po wywołaniu ładowane z dysku, a są dostępne z pamięci operacyjnej. Należą do nich następujące komendy: DIR — wyświetlenie katalogu dyskietki, TYPE nazwa zbioru — wyświetlenie na ekranie tekstowej zawartości zbioru dyskowego, ERASE nazwa zbioru — usunięcie zbioru, RENAME nowa\_nazwa=stara\_nazwa — zmiana nazwy zbioru. Kopiowanie zbiorów odbywa się w CP/M'ie przy pomocy programu PIP (ang. Peripheral Interchange Program). W MS-DOS'ie służy do tego celu wbudowane w system polecenie COPY.

Komputery, dla których Basic jest systemem operacyjnym sygnalizują najczęściej gotowość przyjęcia następnego polecenia słowem READY. Napisanie komendy zrozumiałej przez interpreter i naciśnięcie klawisza ENTER (RETURN, ewentualnie CR) powoduje wykonanie rozkazu.



W przypadku CP/M'u lub MS-DOS'u system zwraca się do użytkownika znakiem zachęty (ang. prompt) określającym aktualnie wykonywany napęd dyskowy np.: A>. Uruchomienie dowolnego programu polega na wpisaniu jego nazwy zakończonej znakiem powrotu karetki (klawisz ENTER).

Co jest prostsze — napisanie RUN "TASWORD" na Spectrum lub Amstradzie (w AMSDOS'ie), celem uruchomienia edytora, czy napisanie WORDSTAR w CP/M'ie? Sądzę, że większość osób zgodzi się z poglądem o tym samym stopniu trudności obu przypadków.

Dlaczego Basic stał się najczęstszym językiem mikrokomputerów? Jak zwykle zadecydowały względy finansowe. Po prostu interpretery Basic'a były najtańsze, najłatwiej się je pisało i wymagały najmniej drogiej i trudno dostępnej pamięci operacyjnej.

Lubię przeczytać od czasu do czasu jakiś kryminał, ale ograniczenie się tylko do tego kręgu literatury uważałbym za skłonność zbyt silną. Podobnie jest z moim stosunkiem do Basic'a. Znam ten język i potrafię się nim efektywnie posługiwać na wielu różnych komputerach, nie tylko domowych, ale nigdy nie chciałbym pisać w nim dużego programu, np. jakiejś bazy danych.

W takich przypadkach łatwo napisać wiele tysięcy linii kodu źródłowego (tekst programu), ale znacznie trudniej utrzymać porządek, jeśli nie mamy możliwości podziału programu na mniejsze fragmenty poprzez zdefiniowanie wołanych przez nazwę procedur z parametrami i zmiennymi lokalnymi, a struktury danych ograniczone są do macierzy liczbowych, ewentualnie znakowych. Brak instrukcji strukturalnych typu IF THEN ELSE, REPEAT UNTIL, WHILE, CASE, czy innych zmusza do stosowania skoków tam gdzie nie są one naprawdę niezbędne. Instrukcja GOTO używana w nieumarkowany sposób prowadzi do istotnego zaburzenia struktury programu i utrudnia jego pisanie i ewentualną analizę.

Wiele osób skarży się, że nie jest w stanie napisać programu bez instrukcji skoku. Przyznam się, że niekiedy sam mam podobne kłopoty przy formułowaniu algorytmu i w pierwszej wersji brudnopisu nie waham się napisać GOTO. Niemniej staranne przemyślenie o co naprawdę chodzi w programie pozwala mi później wyeliminować tę niepożądaną instrukcję.

Zajmując się jakimś problemem, staram się przedstawić jego rozwiązanie w formie programu pascalowego, posługując się świadomie wszystkimi dostępnymi instrukcjami strukturalnymi. Jeśli Pascal nie jest językiem docelowym, dokonuję tłumaczenia na inny język przy zastosowaniu swoistego słownika. W Pascalu instrukcja warunkowa IF THEN ELSE wygląda następująco:

```
IF warunek
THEN      instrukcja_wykonywana_
przy_nie_spełnieniu_warunku
ELSE      instrukcja_wykonywana_
przy_spełnieniu_warunku
W przykładzie na Basic mamy kilka linii:
10 IF NOT (warunek) THEN GOTO
50
20 REM (THEN)
30      instrukcje_wykonywane_
przy_spełnieniu_warunku
40      GOTO 70
50 REM (ELSE)
60      instrukcje_wykonywane_
przy_nie_spełnieniu_warunku
70 REM (end of if)
```

Przyglądając się obu przykładom widzimy, że konstrukcja w Basic'u wymaga zaprzeczenia testowanego warunku i zastosowania dwóch instrukcji skoku. Tłumaczenie instrukcji wyboru CASE jest jeszcze bardziej skomplikowane:

CASE n \* INSTRUKCJA CASE W PASCALU \*)

```
OF 1 : instrukcja wykonywana dla n=1;
   2 : instrukcja wykonywana dla n=2,
   2 : instrukcja wykonywana dla n=3;
ELSE instrukcja wykonywana dla n różnego od 1, 2, 3;
END;
```

5 REM INSTRUKCJA CASE W BASIC'U

```
10 IF NOT N=1) THEN GOTO
40
20 REM instrukcja wykonywana dla n=1;
30 GOTO 110
40 IF NOT (N=2) THEN GOTO
70
50 REM instrukcja wykonywana dla n=2;
60 GOTO 110
70 IF NOT (N=3) THEN GOTO
100
80 REM instrukcja wykonywana dla n=3;
90 GOTO 110
100 REM instrukcja wykonywana dla n różnego od 1, 2, 3
110 REM KONIEC
```

W tym przypadku jest jeszcze „gorzej” — mamy 6 zbytecznych skoków. Oczywiście podobnie wygląda sprawa przy innych instrukcjach strukturalnych. Przy pomocy tych przykładów chciałem pokazać, że można i w Basic'u pisać porządne programy (strukturalne), ale jest to znacznie bardziej kłopotliwe. Nie możemy skupić się na algorytmie zadania, które rozwiązujemy, jeśli musimy ciągle pamiętać jak sobie poradzić z podstawowymi konstrukcjami programowania.

Chociaż programowanie jako takie jest niezależne od języka i komputera, to jak każda działalność ludzka rozwija się z czasem. Początkowo pisano programy jako ciągi zer i jedynek. Było to „wygodne” dla komputera, ale znacznie mniej dla człowieka. Pewnym krokiem naprzód było wprowadzenie rozkazów symbolicznych, np. ADD (dodawanie), JP (instrukcja skoku) itd. Język ten nazwano assemblerem i choć pozwalał on łatwiej pisać i analizować programy, to w dalszym ciągu obliczanie nawet nieskomplikowanych wyrażeń arytmetycznych wymagało stosowania bardzo wielu rozkazów.

W połowie lat pięćdziesiątych wprowadzono FORTRAN (ang. FORMuła TRANslator), używany do tej pory. Programowanie stawalo się coraz łat-

wiejsze, ale ciągle było bliskie języka maszynowego jeśli chodzi o instrukcje sterujące. Podobnie było z językiem BASIC wymyślonym w 1964 roku. Istotny przełom spowodowało dopiero pojawienie się w 1971 roku raportu N. Wirtha opisującego standard Pascala. Język ten od razu zdobył sobie olbrzymią popularność ugruntowaną następnie przez produkty firmy Borland — Turbo Pascal wersja 3.0 dla CP/M'u i MS-DOS'a, oraz kolejne modyfikacje 4.0, 5.0 dla komputerów klasy IBM PC. W literaturze dotyczącej programowania Pascal stał się językiem publikacyjnym, w którym formułowano opisy algorytmów.

Chociaż w każdym języku można napisać dowolny program to w jednym języku może to być prostsze niż w innym. Basic jest bardzo poręczny jako kalkulator, jako narzędzie do pisania krótkich programów, ale do realizacji większych projektów lub nauki programowania jako takiego, po prostu się nie nadaje.

Jonasz Mayer

Dlaczego podjąłem ten temat?

Z ankiety przeprowadzonej przez naszą redakcję wynika, że tylko 3% czytelników Bajtka posiada sprzęt firmy Amstrad, a przeciętny wiek tych osób to 14—15 lat. Faktem jest mała ilość listów do Klanu Amstrada, ale moim zdaniem użytkownikami tych komputerów są osoby trochę starsze, mniej chętnie uprawiające tego rodzaju korespondencję. Sądzę, że między formułami pism typu Komputer, a Bajtkiem istnieje pewna luka, doskonale wypełniana przez Amstrady, będące komputerami domowymi o pewnych możliwościach profesjonalnych.

Chciałbym jednak odwołać się do opinii Czytelników na ten temat. Prosiłbym o informacje od zainteresowanych osób, czy Klan Amstrada nadal jest potrzebny i jaki ma być jego kształt.



## AGENCJA WYSYŁKOWA ELEKTRONICZNA A.S. SERWICE ELEKTRONICS

40-003 Katowice Teatralna 9

- szukasz schematu Twojego komputera, stacji dysków lub innego urządzenia
- informacji katalogowej, względnie aplikacji elementów
- a może podzespołów elektronicznych, układów scalonych

Napisz, pomożemy Ci.

(SB-35)

## ZX SPECTRUM

Polskie programy do zabawy, eksperymentów oraz ciekawych zastosowań:

- \* zestaw TOTO (DL, SL, Ex, ZS)
- \* LIGA POLSKA
- \* LITERKI — dla przedszkolaków, rodziców i dziadków!
- \* oraz inne atrakcyjne programy

Informacje kopertą zwrotną

MASTER BIT

61-660 Poznań 31, skr.p. 56

G-124

Największy wybór oprogramowania do ATARI ST, XE, XL, COM-MODORE 64, AMIGA, na najkorzystniejszych warunkach udostępnia HOOZY 81-706 SOPOT, BOH.MONTE CASSINO 21 D.

Prosimy o kopertę i znaczek.

G-127

## ZX Spectrum

Ekspresowe naprawy klawiatur

NiKUE ul. Meissnera 14 m 1, 03-982 Warszawa tel. 15-93-38 wieczorem

D-73

MIKROSERVICE COMMODORE — 64/128, AMIGA, ATARI, SPECTRUM, IBM — PC/XT/AT.

INTERFEJSY — CENTRONICS, RS 232, DIGITIZER, FINAL, SPEEDDOS, PROGRAMATORY.

EUROKARTY — Z80, 6502, 68000, A/C, CP/M.

01-911 WARSZAWA, ANDERSENA 3/103.

D-52

MONO LIGHT PEN. Oryginalny program graficzny dla Atari 8-bit współpracujący z piórem świetlnym.

Krzysztof Stus ul. Garbarska 30/20, Piotrków Trybunalski 97-300.

(SB 53)

Oprogramowanie do Twojego ATARI XL/XE. Studio „BIT” 11-035 Unieszewo 85d/13. Najtańszej, najszybciej!

(SB 57)

Naprawa mikrokomputerów mgr Izdebski 47-47-87 w. 308, Warszawa.

(SB 250)

Naprawy komputerów SPECTRUM i TIMEX oraz drukarek SEIKOSMA GP50 wykonuje Zakład Usług Elektronicznych „Spectrum-Service” — Kraków mgr inż. Romuald Rykiert tel.

22-40-55 w. 205 (od 7.30 do 14.30)

(SB 51)

W Krakowie: PRECISION, BASF, 3M, TAYTEL, BONUS, DYSAN, VERBATIM — najtańszej od dostawcy z U.S.A. Kraków: centrum: „Elektronika” Kraków, ul. Krakowska 2 tel. 34 19-10.

D-117

Programy użytkowe, opisy na dyskietkach oferuje „MIKROFAN” — ul. Wesoła 1 skr. poczt. 158. Informacje za załączeniem znaczka

D-87



# STRUKTURY DANYCH

## cz. IV



Dziś lista — struktura danych, która pozwala programom w sposób bardzo elastyczny gospodarować pamięcią operacyjną komputera.

Listę opisałismy miesiąc temu przy pomocy rysunku:



na którym opisane literami prostokąty oznaczają porcje danych, a strzałki stanowią powiązania między nimi. Dzięki tym strzałkom możemy przeglądać dane — od A możemy przejść do B, stąd do C itd. Kolejność w jakiej przeglądamy dane zależy od ustawienia strzałek a nie od fizycznego położenia danych. Tak więc, jeśli z jakichś względów chcemy aby nasze dane tworzyły sekwencję B,C,A,D to musimy jedynie pozmienić strzałki tak aby otrzymać:



Równie łatwo jest dołączyć do listy nowy element lub usunąć z niej zapis (oglądaliśmy to na rysunkach w poprzednim odcinku). Łatwość ta jest w wielu przypadkach zaletą decydującą o wyborze tej metody zapisu danych, zanim jednak zaczniemy dyskutować o wadach i zaletach list zajmijmy się praktyczną stroną ich realizacji.

Niektóre języki programowania, np. Pascal lub mniej znana wśród miłośników mikrokomputerów Simula/67, pozwalają deklorować grupy danych tworzących jedną całość, tzw. rekord, oraz przewidują istnienie zmiennych służących do wskazywania na rekordy. W Pascalu są to zmienne typu WSKAŹNIK, tworzone przy użyciu w ich nazwach strzałki. Nie będziemy szczegółowo omawiać programowania w Pascalu, więc tylko zasygnalizuję, że jeśli deklarujemy rekord, którego wystąpienia mają słać się elementami listy, to w jego zawartości musimy przewidzieć zmienną typu wskaźnikowego, której wartość w trakcie pracy programu będzie wskazywać następny element.

O znaczeniu list w wielu zastosowaniach może świadczyć fakt, że powstały (i uzyskały sporą popularność) języki programowania służące tylko do przelzwarzania danych zapisanych w postaci rozbudowanych list, np. LISP. Także popularne na mikrokomputerach LOGO oparte jest o struktury listowe.

Czy jednak mając do dyspozycji tylko BASIC jesteśmy definitywnie pozbawieni możliwości korzystania z omówionych struktur? Na szczęście nie, mając do dyspozycji tablice możemy łatwo zorganizować listowy dostęp do zapisywanych w tych tablicach danych.

Przypomnijmy sobie przykład, w którym dane zapisane w jednym wierszu tablicy opisywały jeden towar:

Numer	Ilość	Data dostawy	Nr. regału	Nr. półki
5				
11				
34				
43				

Tak więc, wiersz tablicy jest tu odpowiednikiem naszego rekordu — porcji danych. Umieszczenie danych w tablicy ma być takie aby numery towarów (zapisane w pierwszej kolumnie) były uporządkowane, gdyż przyspiesza to wy-

szukiwanie. Pamięamy również, że jakiegokolwiek przemieszczenia danych wymagają przepisywania całych wierszy, gdyż w przeciwnym przypadku nastąpi przemieszanie informacji. Na razie kolejność przeglądania wynika z kolejności umieszczenia wierszy w tablicy. Spróbujmy zapisać ją używając strzałek (dla oszczędności miejsca rysujemy tylko lewy fragment tablicy, resztę zostawiając wyobraźni Czytelników):

Numer	Ilość	D
5		
11		
34		
43		

O ile przedtem przeszukiwania dokonywaliśmy pętlą: FOR I=1 TO N to teraz musimy to robić zgodnie z następującym schematem:

```
weź pierwszy element listy
dopóki nie doszliśmy do końca listy wykonuj
  sprawdź czy to szukany numer,
  jeśli tak to koniec,
  jeśli nie przejdź do następnego
  (wskazywanego przez strzałkę) elementu
koniec pętli dopóki.
```

Spróbujmy teraz dopisać do tablicy nowy wiersz danych, o wartości numeru 10. Powinien on zająć miejsce między wierszem z wartością 5 a wierszem z 11. Jednak zgodnie z metodą dopisywania do listy wpisujemy wiersz tam gdzie jest wolne miejsce (w tym przypadku na końcu) a właściwą kolejność zachowamy przemieszczając strzałki.

Numer	Ilość	D
5		
11		
34		
43		
10		

Pozostaje nam jeszcze do rozwiązania poważny problem: jak mianowicie mamy zrealizować te strzałki? W tym celu dodamy do tablicy jeszcze jedną kolumnę liczb, o nazwie „Nast” co jest skrótem od „Następny”. W kolumnie tej będziemy trzymali numer wiersza, który jest następnym elementem naszej listy. Obejrzyjmy to na przykładzie (dla ułatwienia z lewej strony podajemy numery fizycznych wierszy tablicy):

	Nast	Numer	Ilość	D
1	5	5		
2	3	11		
3	4	34		
4	-1	43		
5	2	10		

Przykład ten warto porównać z poprzednim, w którym powiązania przedstawione były w formie graficznej.

Zwróćmy też uwagę na znacznik końca listy. W naszym przypadku jest nim liczba -1. Jest to dość naturalny wybór, gdyż w naszej tablicy nie ma wierszy o ujemnych numerach.

Jeszcze jedna ważna rzecz uszła jak dotąd naszej uwadze (może dlatego, że rzeczy oczywiste najłatwiej jest przegapić), musimy wiedzieć gdzie szukać początku listy, przecież nie zawsze musi nim być pierwszy wiersz tablicy.

Przy reprezentacji listy przy pomocy zmiennych wskaźnikowych (Pascal) początek ten zapamiętamy bardzo łatwo — w zmiennej typu wskaźnikowego. Przy realizacji listy w oparciu o tablicę rolę wskaźników pełnią numery wierszy czyli liczby, tak więc początek listy (nuemr fizycznego wiersza tablicy w którym jest zapisany pierwszy element) będziemy przechowywać w zmiennej typu całkowitego.

Spróbujmy naszkicować fragment programu szukającego na liście elementu o numerze podanym w zmiennej X. Musimy wprowadzić trochę oznaczeń. Początek listy jest zapisany w zmiennej POCZ, tablica nazywa się L, do jej elementów odwołujemy się zgodnie z rysunkiem:

	Nast	Numer	Ilość	D
1	L(1,1)	L(1,2)	L(1,3)	
2	L(2,1)	L(2,2)	L(2,3)	
3	L(3,1)	L(3,2)		
4	L(4,1)			
5				

Nasz program może wyglądać tak:

```
1 I=POCZ
weź pierwszy element listy
2 IF I=-1 GOTO 9
dopóki nie doszliśmy do końca listy wykonuj
2 IF L(I,2)=X
  sprawdź czy to szukany numer.
  GOTO 8
  jeśli tak to koniec,
4 I=L(I,1)
  jeśli nie przejdź do następnego
5 GOTO 2
  (wskazywanego przez strzałkę)
  elementu
```

koniec pętli dopóki.

8 tu wskakujemy gdy znaleziono element numer X  
9 tu wskakujemy gdy doszliśmy do końca listy, czyli elementu nie ma.

Proponuję uważnie prześledzić działanie tego programu dla naszego poprzedniego przykładu, zwracając uwagę jak zmienia się wartość zmiennej I, gdyż to właśnie ona wskazuje na kolejne elementy listy (zapisywane w wierszach tablicy).

Skoro już umiemy tworzyć listy nawet w językach nie dopuszczających zmiennych wskaźnikowych, przejdźmy do problemów ogólniejszych. Aby tablica od której zaczęliśmy zamieniać się w listę musieliśmy dodać do niej jedną kolumnę. Nie jest to niestety jednostkowy zbieg okoliczności lecz smutna reguła: organizacja struktur okoliczności zawsze (niezależnie od metody realizacji) pochłonie pewną ilość komórek pamięci na reprezentację powiązań między danymi (tych rysowanych przez nas w postaci strzałek). W szczególności więc, jeśli przy rozwiązywaniu jakiegoś problemu brakuje nam pamięci, to być może należy jednak dane przechowywać w tablicach.

Realizować listę możemy także w asemblerze, odwołując się bezpośrednio do komórek pamięci — wartością odpowiadającą strzałce jest wtedy po prostu adres pierwszej komórki z następnego elementu. Takie rozwiązanie daje nam dużo większą elastyczność, np. elementy listy nie muszą mieć wtedy takiej samej budowy, mogą nawet być różnej długości. Dokładnie to samo możemy osiągnąć w BASIC-u stosując tablicę jednowymiarową, w której na zapis jednego rekordu użyjemy odpowiedniej liczby sąsiadujących ze sobą komórek. Oto przykład listy mającej cztery elementy różnej długości i zorganizowanej w omówiony sposób. Przedstawiamy ją za pomocą strzałek:



DD DD DD DD

oraz w postaci maszynowej — z adresami następnych elementów. Dla ułatwienia nad każdą komórką pamięci podany jej numer (czyli adres). Pierwszy element listy dla łatwiejszej identyfikacji wypełniony jest literami A, drugi B itd.

1 2 3 4 5 6 7 8 9 10



11 12 13 14 15 16 17 18

DDDDDDDD

I tu pojawia się drugi ważny problem: wprowadzając elementy o różnej długości straciliśmy rachubę początków tych elementów. W tej chwili do elementu listy możemy dojść **tylko** przez strzałkę z poprzedniego, a do poprzedniego też tylko przez strzałkę, itd. Krótko mówiąc, jeśli chcemy znaleźć coś na liście, to zawsze musimy zaczynać od jej początku i przechodzić przez wszystkie elementy. Nie jesteśmy w stanie wziąć sobie jednego elementu gdzieś ze środka, tak jak to ma miejsce przy przechowywaniu danych w tablicy. I o tym należy pamiętać wybierając strukturę danych dla konkretnego programu.

Na zakończenie jeszcze jeden problem: jeśli z listy usuwamy elementy, to co dzieje się ze zwalnianym miejscem, jak postępować aby móc znowu je wykorzystać przy dopisywaniu nowych elementów. Proponuję abyście sami spróbowali zaproponować rozwiązanie tego zadania. Za miesiąc ja też przedstawię pewne rozwiązanie, poza tym podsumujemy i pożegnamy listy oraz zapoznamy się z inną bardzo często używaną metodą dostępu do danych, nazywaną stosem.



# Przemysłowe drukarki firmy Star

## Tam gdzie liczy się niezawodność



Drukarki przemysłowe firmy Star odpowiadają potrzebom zastosowań w przemyśle. Są przystosowane do pracy ciągłej pod dużym obciążeniem, są bardzo szybkie, charakteryzują się konstrukcją odporną na czynniki zewnętrzne. Wyróżniają się łatwością obsługi.

9-igiel  
8 krojów pisma  
300 zn/sek  
Funkcja „Paper park”  
Opcja kolor

**star**  
Twoja drukarka.



Star FR-10

cena 900 DM

Star FR-15

cena 1100 DM

ABC Data

Bonn  
Augustastr.40  
5300 Bonn 2

telex:

88 55 66

fax:

(0228) 35 56 35

tel.

(0228) 35 44 80/90

Hamburg

Dietmar-Koel-Str. 13  
2000 Hamburg 11

21 66 002

(040) 31 91 783

(040) 31 40 03

Berlin

Alt-Moabit 80  
1000 Berlin 21

18 13 65

(030) 39 36 483

(030) 39 150 90/99

Serwis: Enter

Warszawa ul. Żurawia 4a

81 25 48

21 75 08





## NIE POZWÓL, BY CIĘ ZABRAKŁO!

II MIĘDZYNARODOWE TARGI INFORMACJA'89

10—14 październik 1989 r.

Hala Widowiskowo-Sportowa „Spodek” Katowice

Serdecznie zapraszamy Waszą Firmę  
na II Międzynarodowe Targi INFORMACJA'89

Wytnij i wyślij.



Czy będziecie chcieli wziąć udział w którymś z seminariów zorganizowanych podczas tegorocznych Targów. INFORMAD — informatyka w medycynie.

Desktop Publishing — wspomaganie prac inżynierskich.

Badania naukowe — zapoznanie się z najnowszymi dokonaniem informatycznymi w tej dziedzinie. Wspomaganie projektowania procesu produkcyjnego.

Salon „Prezentacje'89”

Czy Przedsiębiorstwo, Instytucja Wasza pragnie zaprezentować swój dorobek w jednym z wymienionych tematów, a tym samym włączyć się do konkursu, w którym nagrody ufundują: Ministerstwo Przemysłu, Urząd Postępu Technicznego, Redakcja Teleexpressu, NOT-SIGMA, Urząd Wojewódzki w Katowicach, ABC — Data, Interpegro — Pregrotour, PRO-INFO.

Nagrody: magnetowidy, telewizory kolorowe, wyjazd do Moskwy i RFN, Targi Cebit.

Czy zamawiacie karty wstępu, koszt karnetu: 1.500,- zł — 1 szt.

tak ..... (ilość sztuk) ..... nie

Czy będziecie korzystać z bazy hotelowej?

tak ..... nie .....

Jesteśmy zainteresowani w zwiedzeniu wystawy .....

— w uczestnictwie w jednym z wymienionych spotkań seminarnych

— w dokonaniu zakupu:

komputerów ..... tak ..... nie

oprogramowania ..... tak ..... nie

dokumentacji ..... tak ..... nie

Imię i Nazwisko .....

Stanowisko .....

Przedsiębiorstwo (nazwa) .....

Adres .....

Telefon .....

Telex .....

Prosimy o przesłanie zgłoszenia na adres PRO-INFO



Przedsiębiorstwo Techniczno-Usługowo-Promocyjne

40-001 Katowice, ul. Sikorskiego 18/38

skr. poczt. 1347; tlx: 0312401

(SB 50)

## MIKRY WACUŚ!

Na oskarżenia Rolanda Waclawka pod adresem „PRO-INFO” zadają pytania do firmy „Mikro-Wacusia”, której właścicielem jest Roland Waclawek:

1. Czy firma Mikro Wacusia — Roland Waclawka jest autorem oprogramowania, za które pobiera pieniądze, w którym to oprogramowaniu dokonuje manipulacji?
2. Czy Roland Waclawek w swoich artykułach jest wiarygodny, czy może być wiarygodny jeżeli „wiarygodność” jest zależna od ochrony własnych interesów?
3. Czy firma Rolanda Waclawka ma spokojne sumienie dokonując zwykłej kradzieży, przywłaszczając obce prace i przypinając do obcych prac swoje imię i nazwisko?
4. Czy Pan Roland Waclawek utrzymać będzie, że nigdy nie sprzedawał i nie wykonywał prac dla „PRO-INFO”, czy nadal będzie zaprzeczał, gdy już wielokrotnie były dokumentowane fakty płatności naszej firmy dla Pana Rolanda Waclawka, a może to nie były honoraria za otrzymane prace, a jakieś łapówki, które przyjął Roland Waclawek? Być może jest inna prawdy, że Roland Waclawek sprzedaje prace sygnując je swoim nazwiskiem i imieniem mając pełną świadomość, że te prace nie są jego pracami.
5. Dlaczego Rolanda Waclawka grzecznie wyproszono podczas targów Cebit'89 ze stoiska Microsoft i Mark-Technik? Nie będę odpowiadał, odpowiedzi są w moich zapytaniach.

Prezes PRO-INFO

Janusz Gotuch

(SB 52)

## KLUB KSIĄŻKI — SOETO

przy Stołecznym Ośrodku Elektronicznej  
Techniki Obliczeniowej „SOETO”

Nazwisko i imię .....

Dokładny adres .....

Zgłaszam swoje przystąpienie do Klubu Książki — SOETO i zamawiam następujące książki:

autor	tytuł	cena	ilość egz.
W. Zientara	Mapa pamięci — Podstawowe procedury systemu operacyjnego	1500	.....
W. Zientara	Mapa pamięci — Procedury wejścia-wyjścia	2000	.....
W. Zientara	Mapa pamięci — Dyskowe systemy operacyjne	2060	.....

W. Zientara	Mapa pamięci — Procedury interpretera BASIC'a	2000	.....
W. Zientara	Poradnik programisty ATARI XL/XE	2000	.....
W. Zientara	Języki programowania ATARI XL/XE cz.I	2000	.....
W. Zientara	Języki programowania ATARI XL/XE cz.II	2000	.....
B. Radziszewski K. Gajewski	Jak rozbudować interpreter C64	2500	.....
M. Przasnyski	Obsługa mikrokomputera IBM-PC — System operacyjny DOS 3.20	2000	.....
M. Czurak	ASSEMBLER Z-80	2000	.....
Podpis zamawiającego lub opiekuna .....			



## COMPUTER SERVICE

MS elektronik  
naprawy komputerów:  
Spectrum 48k. +, 128, +2,  
+3  
Amstrad- Schneider  
Sharp  
Drukarki, Interfejsy  
Wyjścia monitorowe  
Czynne: od 9.00—16.00  
MSelektroinik Legionowa 23,  
00-343 Warszawa  
Dojazd: 105, 305, F (Jelenki)  
tel. 37-76-65.

(K-118)

## IBM XT, AT/386 COMMODORE ATARI SPECTRUM (TIMEX)

Sprzęt, programy, opisy  
ATUT Sp. z o.o.  
ul. Weteranów 38  
20-045 Lublin

K-133

OTV RADZIECKIE  
JUNOST ELEKTRONIKA SILELIS  
oraz STACJONARE:  
NAPRAWA ● KINESKOPIY ●  
DEKODERY PAL ● FONIA CCIR  
WEJŚCIA MONITOROWE  
INTER-SERWIS,  
ul. Rutkowskiego 10/12  
00-020 Warszawa, tel. 27-47-72

D-76

Naprawa komputerów ATARI, COM-  
MODORE, IBM, SPECTRUM oraz  
urządzeń peryferyjnych. Warszawa tel.  
22-07-85

D-84

## CZĘŚCI ELEKTRONICZNE

— pochodzenia zagranicznego  
za złotówki układy TTL, LS,  
CMOS, stabilizatory, mikro-  
procesory, pamięci.

Krótkie terminy dostaw, atrak-  
cyjne ceny.

Pośredniczymy także przy  
sprzedaży.

„MARIiTEX” tel. 22-02-89

Gdynia, ul. Batalionów

Chłopskich 3

tlx 054622

K-190

## ATARI

- nowości programowe
- instalowanie systemu TURBO we wszystkich typach magnetofonów firmowych i interfejsach
- instrukcje i opisy

## IBM PC XT AT

- pełne oprogramowanie
  - instrukcje i tłumaczenia
  - pisanie programów na zamówienie
- Dla instytucji i osoby prywatnych oferuje:

BRODATY SOFTWARE  
firma „INFO-CELL”

Zapraszamy do stoiska mieszczącego się w S.D.H

„Feniks” IV p. ul. Szewska 75  
oraz wysyłkowo: 50-951 WROCŁAW 3  
skr. pocztowa 1094

D-104

Agencyjny Zakład Usługowy SPHW  
poleca usługi w zakresie:

- serwis komputerów SPECTRUM, C-64, TIMEX, XT, AT, rozszerzanie pamięci
  - przestrajanie PAL-SECAM
  - wejścia monitorowe TV
- Warszawa ul. Puławska 102  
tel. 44-87-89  
czynny 12.00—19.00  
Rachunki, gwarancja  
ZAPRASZAMY

D-114

Odsprzedam tanio wiele najno-  
wszych gier na ATARI  
41-400 Mysłowice Wielka Skotni-  
ca 20/40.

G-163

## SERWIS KOMPUTERÓW

# TEST

40-164 Katowice ul. Modrzewiowa 24/33

poleca naprawy:

- ATARI 600, 800, 65, 130 XL, XE
- COMMODORE 16, 116, +4, 64, 128
- DISC DRIVE 1541, 1570, 1571, 1050
- MAGNETOFONY COMMODORE
- DRUKARKI

godz. 9-11, 16-18

(SB 30)

## ATARI TURBO 2000F

Nowy system transmisji danych z magnetofonem przyspieszony do 6700 bodów

Komplet:

- cartridge
- kasecia z 5 programami kopiującymi
- przeróbka magnetofonu
- 12 m-cy gwarancji za 29 000 zł oraz interfejs do standardowego magnetofonu za 22 000 zł

oferuje firma  
**MUEL**  
ul. Czastkowska 30  
01-678 Warszawa

D-108



## REGULAMIN

1. Członkiem „Klubu Książki” — SOETO może zostać każdy, kto zamówi co najmniej 3 z przedstawionych do wyboru książek.  
Kartę uczestnika czytelnie wypełnioną i podpisaną (w przypadku dzieci — podpisaną przez rodziców lub opiekunów) należy wysłać pod adresem: STOLECZNY OŚRODEK ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ „SOETO” 00-682 Warszawa, ul. Hoża 50
2. Książki do KLUBU KSIĄŻKI — SOETO zostały wybrane z planów wydawniczych na rok 1989. Uczestnicy Klubu będą je otrzymywać w miarę ich ukazywania się, czyli w ciągu całego 1989 r. (a jak podpowiada praktyka — również i w 1990 roku).

3. Książki będą wysyłane członkom Klubu za tzw. pobraniem pocztowym.
4. Należność za zamówione książki oraz koszt pobrania pocztowego płaci się listonoszowi przy odbiorze przesyłki z książkami, nie trzeba więc wnosić żadnych przedpłat z góry.
5. Obowiązkiem odbiorcy książek jest sprawdzenie, czy paczka jest nienaruszona. W przypadku wątpliwości należy odmówić przyjęcia. Po odbiór przesyłki należy wtedy zgłosić się do UPT, w którym otwarcia jej dokona pracownik poczty i ewentualnie sporządzi protokół. Tylko wtedy reklamacja może być uwzględniona.
6. Informacje dotyczące Klubu Książki — SOETO można uzyskać pod tel. 29-18-64 lub co pewien czas będą ukazywały się na łamach „Bajtka”.

(SB 58)



# CONSOFT

Przedsiębiorstwo Wdrożeń  
Informatycznych  
spółka z o.o. (j.g.u.)

58-300 Wałbrzych  
ul. Moniuszki 13/15/3  
skr. poczt. 251 tel. 246-74

tlx 745615 CSOFT PL

(TRZY TOMY, 719 STRON)

- OPIS BUDOWY I ZASADY DZIAŁANIA POSZCZEGÓLNYCH ELEMENTÓW
- INFORMACJE O BIOS-IE I OPROGRAMOWANIU
- LISTA ROZKAZÓW 80286 i 80287 Z KOMENTARZEM
- BIBLIOGRAFIA I SŁOWNIK TERMINÓW INFORMATYCZNYCH

CENA: 85 tys. zł.

ZAMÓWIONĄ POZYCJĘ PRZEŚLEMY NA PODANY ADRES  
W TERMINIE 7 DNI OD DATY POTWIERDZENIA WPŁATY

## W PRZYGOTOWANIU LISTING BIOS

----- Tu naciąć -----

Pieczęć zakładu:

PWI „Consoft” sp. z o.o. (j.g.u.)  
58-300 Wałbrzych skr. poczt. 251  
ul. Moniuszki 13/15/3

Zamawiamy ..... egzemplarzy „Dokumentacji technicznej do AT”.

Nazwa zakładu: .....

Dokładny adres: .....

Należność w wysokości ..... zł x ..... egzemplarzy = ..... zł

przekazano na konto: Bank Zachodni O/Wałbrzych nr 388805-2017.

Gł. Księgowy:

Dyrektor:

SB 56



# GIEŁDA NA DZIEŃ 4.08.1989

	Giełda Bajtka	Sklep Bajtka	Komis	Pewex	RFN	Baltona	CSH i inne
	tys. zł	tys. zł	tys. zł	\$	DM	\$	tys. zł

## SINCLAIR

ZX 81	120	120	—	—	—	—	—
ZX Spectrum 48	400	350	550	—	80	—	—
ZX Spectrum +	450	550	700	—	90	—	—
Timex 2048	400	400	400	—	—	—	240
ZX Spectrum 128+	550	—	—	—	—	—	—
ZX Spectrum 128+2	—	—	—	—	140	—	—
ZX Spectrum 128+3	—	—	—	—	280	—	—
drukarka Seikosha GP 50s	—	280	—	—	—	—	—
Interface Kempston	25	20	—	—	10	—	9.3

## COMMODORE

Commodore 64	900	650	710	199	290	155	950
VC 20	200	180	—	—	—	—	—
C 16	300	250	—	—	80	—	—
C 116	350	240	—	—	70	—	—
C Plus 4	450	450	380	—	150	—	—
C 128	1100	1100	—	—	399	—	—
C 128 D	1900	—	—	450	820	—	—
Amiga 500	2300	2900	—	—	899	—	—
Magnetofon 1531	150	170	170	48	30	225	—
Stacja dysków Oceanic	700	700	—	—	320	170	—
Stacja dysków 1571	1000	1000	—	199	460	—	—
Drukarka LCIOC	1300	1300	—	—	260	230	—

## ATARI

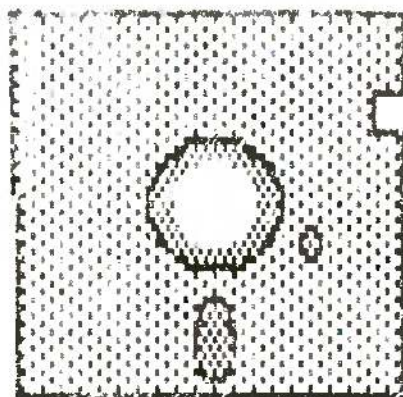
Atari 800 XL	600	700	470	—	160	—	—
Atari 65 XE	700	650	520	127	—	—	—
Atari 130 XE	800	850	—	199	220	—	—
Atari 520 ST	2000	2000	—	—	—	—	—
Atari 1040 ST	—	—	—	—	1140	—	—
Magnetofon XC 12	140	140	—	36	40	—	—
Stacja dysków 1050	900	900	750	185	300	—	—
Stacja dysków 520 STM	—	700	—	—	—	—	—
Drukarka 1029	400	650	—	—	—	—	—

## AMSTRAD

Amstrad 464 mono.	900	900	—	—	350	—	—
Amstrad 664 mono.	—	1200	—	—	—	—	—
Amstrad 6128 mono.	—	—	—	—	670	—	—
Amstrad PCW 8256	—	—	—	—	—	—	—
Amstrad PCW 8512	—	—	—	—	—	—	—
Amstrad PCW 9512	—	—	—	—	—	—	—
Stacja dysków do 464	—	—	—	—	380	—	—

## SHARP

Sharp MZ 700	—	360	—	—	—	—	—
Sharp MZ 800	—	420	—	—	—	—	—
Dyskietki 5.25 cala	2	2-2.5	3	1	0.7	11	4-9
Dyskietki 3.5 cala	7.5	7.5-9	7-9	—	5	2.5	9-10
Dyskietki 3 cala	11	—	—	—	6	3	10
Joystick	21	16-26	20-30	5	10	10	14
Monitor Neptun	60	80	—	—	—	—	68



INDYWIDUALNY  
**BANK**  
DANYCH

**Damian Weymann**, jest posiadaczem ZX Spectrum+ oraz około 300 gier i programów użytkowych. Proponuje wymianę programów oraz literatury. Adres: 62-001 Chludowo, Gołęczewo 44, woj. poznańskie.

**Bartłomiej Kras**, lat 17. Posiada Amstrada CPC 464. Interesuje się muzyką, lotnictwem oraz informatyką. Proponuje wymianę programów użytkowych oraz gier. Adres: 26-900 Kozienice, UPT-3, skr. poczt. 99.

**Zbigniew Wilk** prosi o kontakt posiadaczy Amstrada CPC 464. Adres: 48-200 Prudnik, ul. Nowotki 12/25.

**Piotr Wyszynski**, lat 14. Posiada mikrokomputer SHARP-MZ 700. Pragnie nawiązać kontakt z posiadaczami tego mikrokomputera, a także proponuje wymianę programów użytkowych i gier. Adres: 39-200 Dębica, ul. Sportowa 135.

**Wojciech Wrzaskala**, lat 18. Posiada komputer SPECTRAVIDEO SVI-738 X'PRESS standardu MSX. Jest zainteresowany wymianą literatury i programów. Adres: 59-220 Legnica, ul. Orzeszkowej 18/24/B/4.

**Zdzisław Frąckowiak** nawiąże współpracę i kontakt z posiadaczami AMSTRADA CPC 6128 oraz plotera „SONY” PRN C41. Proponuje również wymianę literatury dotyczącej ATARI na inną. Adres: 88-100 Inowrocław, Aleja Okrężna 81/111.

**Jacek Spruch** uczeń 13 lat. Posiada mikrokomputer TIMEX 2048 z magnetofonem SVI-767 TP. Proponuje wymianę programów użytkowych oraz gier. Adres: 63-700 Krotoszyn, Os. J. Korczaka 1/27.

**Kazimierz Stopyra**, nauczyciel, lat 31. Mikrokomputer AMSTRAD CPC-128, dodatkowa stacja dysków 5 1/4", układ uk, oprogramowanie systemowe, elektronika, sterowniki uk, kompilatory języków. Może udzielić wielu informacji dotyczących komputerów typu CPC — sprzętu i oprogramowania firmowego. Poszukuje pomocy przy projektowaniu przystawki do komputera odbierającej program telegazety. Adres: 48-367 Trzeboszowice, Meszno 63.

**Piotr Kreglicki**, lat 13. Posiada Schneidera CPC 6128 z monitorem GT65. W swoich zbiorach posiada Turbo Pascala, Hisoft C, Forth, Dbase II oraz kilkanaście gier. Proponuje wymianę gier, programów użytkowych oraz literatury. Adres: 21-040 Świdnik, ul. Skarżyńskiego 5/51.

**Artur Nowak**, uczeń LO, lat 18. Posiada AMIGĘ 500. Zainteresowania: muzyka alternatywna i awangardowa. Proponuje wymianę oprogramowania i doświadczenia w tworzeniu dźwięku na AMIDZE 500. Adres: 42-500 Będzin, ul. Skalskiego 7/50.

**Waldemar Gąsior** prosi o kontakt posiadaczy francuskiego mikrokomputera THOMSON MO5. Adres: 50-205 Wrocław 17, skr. poczt. 1610.

**MÓZGOPROCESOR!** — rewelacyjną polską grę przygodową dla ZX Spectrum (program + kaseeta + opis 3100 zł) otrzymasz pisząc: COMPUTER ADVENTURE STUDIO, Bochnia 32-700, ul. Kazimierza Wielkiego 37/45. (SB 20)

## Masz Atari XL/XE

Chcesz powiększyć bibliotekę swoich programów? skorzystaj z usług DATA VISION Sp. z o.o.

Oferujemy:

- szeroki wybór gier i programów użytkowych
- instrukcję i literaturę

Prowadzimy także:

- pośrednictwo w zakresie kupna-sprzedaży sprzętu komputerowego i muzycznego
- oraz NOWOŚĆ

— wysyłkowe!!! nagrywanie na kasety muzyki z PŁYT CD Najwyższa jakość usług! Krótkie terminy realizacji!

Bonifikaty! Dla instytucji rachunki!

Katalogi programów i płyt CD otrzymasz (załączając zaadresowaną kopertę i znaczek) pisząc na adres:

**DATA VISION Sp. z o.o.** 20-606 LUBLIN 17

skr. poczt. 21

tel. 71-42-73

Nie zwlekaj!

Napisz już dzisiaj!!!

(SB-54)

## SYSTEM TURBO 2600 ATARI XL. XE

- pięciokrotne skrócenie czasu trwania nagrania
- oszczędność czasu i taśm
- możliwość współpracy z dowolnym magnetofonem

### INTERFEJS TURBO 2600

do nabycia:

Zakład Elektroniczny  
>SZOK<

ul. Sikorskiego 42 c  
66-200 Swiebodzin

oraz VADIM-SOFT  
al. Niepodległości 11  
65-048 Zielona Góra

SB-55



# Drogi Bajtku!

Na listy czytelników odpowiada  
**DOMINIK FALKOWSKI**

**Na podwórku jestem najlepszym „POKERzystą”. Umiem wpisywać ułatwienia do gier nie używając COPY-COPY. Ale co z tego, gdy nie umiem wyszukiwać POKEów. Czy mógłbyś mi to, Bajtku, w miarę przejrzysto wyjaśnić. I jeszcze jedno: jak bajty bez nagłówka przerobić na bajty z nagłówkiem (...)**

**Dawid Żak, Wrocław**

Jak Ci zapewne wiadomo, gra to normalny program, nawet bardziej skomplikowany, niż niektóre programy użytkowe. W uproszczeniu schemat działania gry można przedstawić następująco: „porusz wrogiem w prawo, sprawdź stan joysticka i ewentualnie przesun bohatera, wydaj sygnał dźwiękowy — kolejną nutę melodii, sprawdź stan przycisku FIRE i ew. strzel wydając dźwięk, sprawdź, czy pocisk trafił we wroga, ... itp.”.

„Zapokowanie” gry polega więc na takiej zmianie programu, by ułatwić życie grającemu. Jednym ze sposobów jest odszukanie instrukcji zmniejszania liczby „żyć” o jeden i usunięcie jej. Inny sposób to usunięcie sprawdzania, czy stracone zostały wszystkie „życia”. Metod jest wiele, a wybór najefektywniejszej należy do POKERzysty.

Dobry POKERzysta musi więc znać doskonale język maszynowy i umieć analizować program gry. Potem, gdy znajdzie odpowiedni adres, wystarczy wpisać tam od-

powiednią liczbę i gramy do znużenia!

Co do drugiego pytania, dotyczącego ZX Spectrum, to „przeróbka” pliku bez nagłówka na plik z nagłówkiem w przypadku bajtów nie jest prosta. Należy przeanalizować procedurę ładującą ten plik i odnaleźć adres, od którego dane umieszczane będą w pamięci. Potem wgrać dowolny nagłówek na COPY-COPY, za nim blok bez nagłówka i zmienić komendę LET start i długość na odczytane z procedury ładującej. Nie jest to oczywiście sposób jedyny, odsyłam Cię do cyklu artykułów „Od środka” pp. Roberta Dudzika i Tomasza Surmacza, Bajtek nr 1 — 8/88.

**Mam 13 lat i zamierzam kupić komputer. Mam do wyboru dwa rodzaje: ZX Spectrum +3 i Atari 520 ST. W związku z tym chciałbym zadać kilka pytań:**

1. Czy ZX +3 może współpracować z drukarką (w +3 Basic) Star LC-10 Colour?
2. Czy ZX +3 posiada stację dyskieta 3" czy 3.5"? Czy 3" dyskietki są w Polsce łatwo dostępne? (...)
3. Jak ma się sprawa z joystickami do ZX +3; czy można stosować zwyczajne Quickshoty we wbudowanych gniazdkach?
4. Jaka jest aktualna cena obydwu komputerów w pełnej konfiguracji w RFN?

**Ryszard Rymon, Skoczów**

Spectrum +3 może współpracować z każdą drukarką za pośrednictwem interfejsu RS 232, lub Centronics, wbudowanych fabrycznie. Możliwa jest również współpraca z LC-10 Colour.

ZX +3 wyposażony jest w stację dyskieta 3", dyskietki takie są w Polsce dostępne, gdyż stosowane są również w komputerach Amstrad CPC i Amstrad PCW. Ich cena jest jednak dość wysoka — nawet do 10.000 zł za sztukę.

Niestety, ZX +3 posiada tę samą wadę, co ZX +2. Współpracować mogą z nimi bowiem tylko joysticki Sinclair II i SJS1. Dlatego Quickshoty i inne standardowe joysticki należy poddać nieznaczącej przeróbce (lub przelutować kable wewnątrz komputera).

Cena ZX Spectrum +3 z dwoma joystickami wynosi w RFN ok. 350 DM, zaś ATARI 520 ST z monitorem kolorowym, wbudowaną stacją i myszą — ok. 2000 DM, z modulatorem TV zamiast monitorem — 800 DM.

**Jako użytkownik ZX Spectrum + mam kilka pytań:**

- dlaczego używając interfejsu Sinclair nie działa autofire w joysticku,
- w jaki sposób zapisane są „screeny” w nowszych grach, że zajmują ok. 4000 bajtów, gdy w rzeczywistości obraz ma długość 6912 bajtów?

— na rysunkach kilku gier (np.

**Gryzor, Combat School, Platoon) widnieje znane z „Bajtku” nazwisko — M. Stawicki. Czy grafik Maciej Stawicki jest autorem tych rysunków?**

**Piotr Szymański, Warszawa**

Standard Sinclaira wykorzystuje jako sterowanie dziesięć klawiszy liczbowych. 1,2,3,4,5 to tzw. Sinclair Left, zaś 6,7,8,9 i 0 to Sinclair Right. Za pośrednictwem interfejsu podłączonego do szyny krawędziowej poruszanie joystickiem daje ten sam efekt, co wciśnięcie odpowiednich klawiszy. Brak działania autofire może być spowodowany tym, że joystick wysyła sygnały ze zbyt wielką częstotliwością i nie są one interpretowane jako ciąg nacisnięć, lecz jedno. Trudno jest postawić jednoznaczny diagnozę, więc używaj raczej standardu Kempston.

Screen zajmujący 4000 bajtów intryguje niejednego. Plik, w którym zawarty jest rysunek, zawiera go w postaci skompresowanej (bliższe informacje na temat kompresji znajdziesz w „Bajtku” i „Komputerze”). Po wczytaniu do pamięci kilkudziesięciobajtowa procedura zawarta w tym samym pliku dokonuje „rozpakowania” obrazka i posyła go na ekran.

Nie wiemy niestety, co dzieje się z panem Stawickim. Jestem jednak prawie pewien, iż to on jest autorem screenów, na których się podpisuje.

**Wojewódzkie Przedsiębiorstwo Handlu Wewnętrznego Oddział w Tychach**

**VIDEOBIT**

43-100 Tychy, Al.ZMP 77  
tel.276975

poleca między innymi

- sprzęt komputerowy  
Atari ● Commodore ● Amstrad ●  
● IBM PC XT/AT/PS 2
- drukarki STAR, EPSON, AMSTRAD
- Sprzęt audiowizualny
- magnetowidy
- OTV PAL/SECAM
- Videoskopy
- kamery
- anteny satelitarne
- aparaturę badawczo-naukową

Udzielamy gwarancji, prowadzimy naprawy pogwarancyjne. Zapewniamy o atrakcyjnych cenach.

(SB 18)



**JOYSTICK SERVICE**

GUN SHOT  
QUICK SHOT ( I-V )  
VG-125 , CX 40  
200 X i inne

- \* wymiana standardowych styków na mikrołączniki
- \*\* naprawa

Zgłoszenia: Studio komputerowe SEZAM  
D.H. "SEZAM" ulp. — czwartki 16" — 19"  
Rachunki. Zniżki cen dla szkół i klubów.  
Prowadzimy ekspedycję pocztową. Szczegółowe  
Informacje po nadesłaniu koperty zwrotnej.  
Korespondencja : JOYSTICK SERVICE  
02-770 Warszawa 130 skr. poczt. 102

**ATAREX oferuje**

duży wybór programów do komputerów »ATARI« na taśmach kasetowych i dyskietkach

szczegółowych informacji po załączeniu koperty z adresem i znaczkiem udzieli:

ATAREX  
ul. 20 Października 42 ul. 22 Lipca 17  
63000 ŚRODA Wlkp. 62300 WRZEŚNIA

BIURO HANDLOWO-USŁUGOWE  
**TECHNIMAX**  
5P.2.00  
81-969 Gdynia 2 S.P. 150

- SPRZĘT KOMPUTEROWY
- DRUKARKI, PLOTTERY
- DYSKIETKI, TAŚMY BARNIACZ
- SPRZĘT AUDIO-WIZUALNY
- TELEWIZORY, MAGNETOWIDY
- MAGNETOFONY, RADIA
- ORAZ WSZELKI INNY SPRZĘT ELEKTROTECHNICZNY
- POLECAMY RÓWNIEŻ PROGRAMY NA KOMPUTERY ATARI XE/XL



10—14 Października 1989  
Hala Widowiskowo-Sportowa  
„Spodek” Katowice

# II MIĘDZYNARODOWE TARGI INFORMACJA 1989

TARGI WYMIANY INFORMACJI, HANDLU, PROMOCJI,  
TARGI, NA KTÓRYCH WINIENES' BYĆ OBECNY,  
SEMINARIA, SZKOLENIA  
DESKTOP PUBLISHING  
INFO-MED

- Wystawa literatury i oprogramowania firm: Microsoft, Ashton Tad, Osborne i innych.
- Wystawa wydawnictw: Data Baker, Chip, Mark-Technik, Komputer i Bajtek.
- Udział w br. wielu firm zagranicznych i krajowych.
- Wyprzedaż z dużą bonifikatą artykułów wystawienniczych za złotówki i walutę.

Wśród zwiedzających nagrody: video, telewizory kolorowe, literatura, wycieczki zagraniczne, w tym do ZSRR na podobne targi w Moskwie w dniach 10—20 listopada br. oraz targi CEBIT w Hanowerze 1990 roku.

**Zapraszamy**

Prosimy wyciąć i przesłać na adres organizatora, kupony wezmą udział w losowaniu nagród.

Prosimy o rezerwację ..... szt. biletów (1 szt. — 1500 zł)

Prosimy o rezerwację hotelu ..... ilość osób

Adres zakładu (osoby) .....

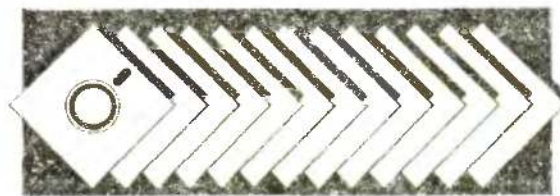
..... miejscowość .....

tel. .... tlx .....

(SB-45)

## KLUBY

### KONKURS O ZŁOTĄ DYSKIETKĘ BAJTKA



#### „MNEMONIK”

— Wrocław —

Klub Komputerowy „MNEMONIK” powstał w 1986 roku przy Zakładach Elektronicznych „ELWRO” we Wrocławiu. Trudno więc się dziwić, że wyposażony jest w całości w Juniorskie, polskie komputery edukacyjne. Członkowie klubu chwają sobie bardzo te komputery. Szczególnie możliwość pracy w sieci. Szkoda tylko — ale to już zupełnie inna sprawa — że o zaletach Juniora nie mogą się przekonać także uczniowie z wszystkich polskich szkół.

Członkami klubu są dzieci pracowników „ELWRO” oraz uczniowie Elektronicznych Zakładów Naukowych — pomaturalnej szkoły przygotowującej średnią kadre techniczną dla „ELWRO”. Jak w każdym klubie są zaawansowani i początkujący. Dla tych ostatnich organizuje się kursy programowania i obsługi komputerów. Bardziej doświadczeni wolą pracować indywidualnie.

Byłby to więc bardzo typowy klub, jakich powstało w ostatnich latach wiele, gdyby nie pasja prezesa klubu, Władysława Salika. Zastosowanie mikrokomputerów w edukacji dzieci specjalnej troski. Pasja, którą potrafił zarazić nie tylko członków klubu, ale także wiele osób z Wrocławia i z całej Polski.

„MNEMONIK” włączony został jako Fundator do Ogólnopolskiej Fundacji Edukacji Komputerowej i korzysta w ramach Fundacji z subkonta dla Dzieci Specjalnej Troski.

W kwietniu 1988 roku odbyła się w Krakowie konferencja naukowa poświęcona zastosowaniu komputerów w nauczaniu dzieci specjalnej troski. Jej organizatorami było Kuratorium Oświaty i Wychowania oraz Szkoła Podstawowa Specjalna nr 131 w Krakowie. Wśród uczestników konferencji nie zabrakło oczywiście Władysława Salika i nie pozostał on wyłącznie biernym słuchaczem konferencji.

Pan Władysław Salik jest z zawodu inżynierem elektronikiem a z powołania pedagogiem. Prócz tego gra na gitarze i śpiewa a nawet koncertuje. Postanowił, że wszystkie dochody uzyskane z koncertów przeznaczać będzie

dla dzieci specjalnej troski. Efektem tej decyzji był komputer SOLUM ofiarowany Szkole Podstawowej Specjalnej nr 115 we Wrocławiu. Jednak pieniądze to nie wszystko. Potrzebni są ludzie. Zaangażowani, przekonani do słuszności tego co robią. Zebrało się ich już sporo, ale potrzeba jeszcze bardzo wielu.

Klub Komputerowy „MNEMONIK” przy Zakładach Elektronicznych „ELWRO” we Wrocławiu, ul. Ostrowskiego 30, 53-238 Wrocław.

#### „FUTURE”

— Oleśnica —

Jest ich dwunastu. Połowa posiada komputery, a druga połowa ma zamiar posiadać. Tak się złożyło, że w klubie znajdują się wyłącznie Commodore 64, choć jego członkowie nie mają nic przeciwko innym komputerom i chętnie przyjmą do swego grona każdego amatora informatyki niezależnie od posiadanego, wykorzystywanego, czy może tylko wymarzonego sprzętu.

Podstawowym celem zorganizowania klubu było odnalezienie w okolicach Oleśnicy użytkowników komputerów i umożliwienie im wymiany doświadczeń, a także prezentowanie zastosowań komputera jako normalnego narzędzia pracy w domu i w szkole.

Efektem kilkumiesięcznej pracy jest między innymi kilka własnych, oryginalnych programów komputerowych z różnych dziedzin. Zgromadzona także biblioteka około 400 programów firmowych.

Początkowo spotkania odbywały się w prywatnych mieszkaniach członków. Było to jednak niezwykle uciążliwe. Dlatego też rozpoczęto starania o uzyskanie jakiegokolwiek własnego lokum. Najpierw sprawa wydawała się beznadziejna, lecz młodych zapaleńców nie tak łatwo zniechęcić. Na początku roku zapukali wreszcie do właściwych drzwi i otrzymali lokal w klubie osiedlowym.

Prezes Klubu — Paweł Leśniorowski ma 16 lat. Większość klubowiczów to jego rówieśnicy. Łączy ich ciekawość rzeczy nowych, niecierpliwość i niewiedza o tym, że czegoś nie da się zrobić...

Klub komputerowy „Future”, ul. Kłownowa 10d/1, 56-400 Oleśnica

### „AKCES-SYSTEM”

Gdańsk ul. K. Marksa 169

tel. (058) 41-19-01

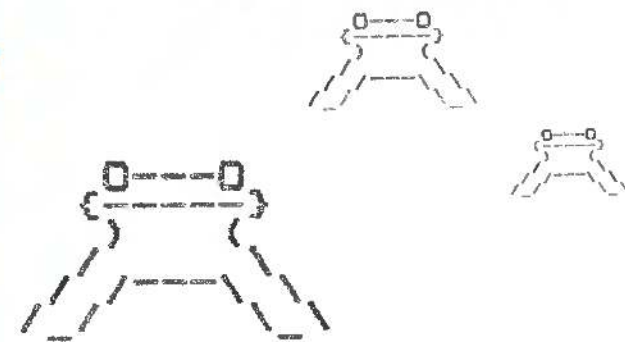
proponuje :

- rewelacyjny system TURBO 2000 przyspieszający współpracę z dowolnym magnetofonem
- sprzęt mikrokomputerowy ATARI XL/XE
- interfejsy magnetofonu, centronics, TOP DRIVE, HAPPY WARP
- sprzęt mikrokomputerowy ATARI ST
- stacje dysków 5.25"
- doskonale, krajowej produkcji monitory wielosystemowe do ST (niskiej, średniej, wysokiej rozdzielczości) w cenie 299.000,-zł
- sprzęt mikrokomputerowy Commodore

ZADZWON, NAPISZ, PRZYJDŹ I ZOBACZ



# ŻABIE ZAWODY



## Cześć Maluchy!

Proponuję Wam dzisiaj napisanie gry komputerowej — „Żabie zawody”. W zabawie może brać udział dowolna liczba osób. Wszyscy są trenerami żab, które kolejno skaczą po stole. Zwycięza nie żaba najszybsza, lecz ta, która zatrzyma się najbliżej krawędzi stołu. Stół ma długość jednego metra, a żaba może skoczyć od 0 do 50 centymetrów. Problem w tym, że nie wiadomo jak daleko skoczy w danym momencie. Trener może jedynie decydować, czy zaliczyć już uzyskaną odległość, czy ryzykować kolejny skok. Jeśli żaba spadnie ze stołu dostaje za karę w tej kolejce odległość jednego metra.

Pierwszy fragment programu (linie 100—150) to przygotowanie danych początkowych. Komputer pyta o liczbę zawodników biorących udział w zabawie (linie 110 i 120). Po uzyskaniu odpowiedzi deklaruje tablicę liczbową punkty o wymiarze równym liczbie graczy (linia 130). W tej tablicy zapisywane będą punkty karne wszystkich żab.

Zmienna stol oznacza długość stołu. Przypisuje się jej wartość 38 czyli długość ekranu pomniejszoną o dwa (linia 140). Nie zmienia się ona w trakcie działania programu, ale w przypadku uruchamiania programu na innym komputerze może zająć potrzeba zadeklarowania innej wartości zmiennej stol.

Zasadniczy program gry umieszczony jest w liniach 200—530. Pętla od k=1 do 5 (linie 210—530) to pięć powtórzeń kolejki. Znajdująca się wewnątrz pętla od l=1 do lz — liczba zawodników (linie 220—520) to kolejka wyścigów poszczególnych zawodników.

Każdy wyścig rozpoczyna się od ustalenia wartości zmiennej zostało na 100 (linia 230). Zmienna zostało określa chwilową odległość żaby od krawędzi stołu. Na początku gry wynosi ona oczywiście 100 cm. Następnie losowana jest długość skoku żaby — zmienna skok (linia 240) i obliczana jest nowa odległość od krawędzi (linia 250).

Teraz komputer przystępuje do rysowania planszy. Pierwsza czynność to ustawienie kursora w lewym, górnym rogu ekranu (linia 260). Następnie drukowane są napisy informujące, która żaba skacze tym razem (linie 270 i 280). Komputer sprawdza, czy przypadkiem odległość od krawędzi stołu nie jest mniejsza od zera (linia 290). Oznaczałoby to, że żaba skoczyła zbyt daleko i spadła ze stołu. Jeśli tak, komputer skacze do linii 2000, czyli do podprogramu upadek. Jeśli jednak zmienna zostało ma wartość dodatnią, drukowana jest dalsza część planszy (linie 300—470). W liniach 380—410 drukowana jest żaba na stole. W zależności od zmiennej zostało żaba drukowana jest w różnych pozycjach. W linii 460 znajduje się skok do podprogramu rysowania stołu — stol. Podprogram ten będzie wykorzystywany również podczas wykonywania podprogramu upadek. Bezpośrednio po wydrukowaniu stołu drukowane są cztery spacje. W tym miejscu bowiem, po wykonaniu podprogramu upadek będzie znajdowała się żaba, która musi zniknąć.

Po narysowaniu planszy komputer oczekuje na decyzję gracza (linie 480—500). Jeśli gracz naciśnie klawisz k, wówczas zawartość elementu tablicy punkty o numerze 1 — numer żaby, zostanie zwiększona o wartość zmiennej zostało (linia 510) i pętla FOR l=1 to lz ... NEXT l zostanie zamknięta.

Po zakończeniu pętli zewnętrznej komputer przechodzi do wydrukowania tablicy wyników (linie 600—650) i kończy program (linia 660).

Podprogram stol nie wymaga szczegółowego opisu. Warto natomiast przyglądać się podprogramowi upadek (linie 2000—2180). Najpierw odpowiedni element tablicy punkty zwiększany jest o 100 (linia 2010). Następnie drukowana jest plansza (linie 2020—2150). W liniach 2100—2120 drukowane są spacje w celu skasowania żaby na stole. Po narysowaniu stołu drukowana jest żaba na podłodze (linia 2150). Wykonywanie podprogramu zostaje zatrzymane do chwili naciśnięcia dowolnego klawisza (linie 2160 i 2170).

Jak najmniej upadków życzę Twoim żabom

Romek

```

100 REM ***** liczba zawodników *****
110 PRINT "Podaj liczbę zawodników"
120 INPUT lz
130 DIM punkty(lz)
140 LET stol=38
150 CLS
200 REM ***** gra *****
210 FOR k=1 TO 5
220   FOR l=1 TO lz
230     LET zostało=100
240     LET skok=INT(RND(1)*50)
250     LET zostało=zostalo-skok
260     LOCATE 1,1
270     PRINT "żaba nr"; l
280     PRINT
290     IF zostało<0 THEN GOTO 2000
300     PRINT "Skok  -"; skok; "cm"
310     PRINT "Zostało -"; zostało; "cm"
320     PRINT
330     PRINT "k - koniec"
340     PRINT
350     PRINT punkty(l); "punktów karnych"
360     PRINT
370     PRINT
380     FOR m=1 TO (100-zostalo)*stol/l
390       PRINT " ";
400     NEXT m
410     PRINT "&";
420     FOR m=1 TO zostalo*stol/100
430       PRINT " ";
440     NEXT m
450     PRINT
460     GOSUB 1000
470     PRINT " "
480     LET k%=INKEY$
490     IF k%="" THEN GOTO 480
500     IF k%<>"k" THEN GOTO 240
510     LET punkty(l)=punkty(l)+zostalo
520     NEXT l
530   NEXT k
600 REM ***** wyniki *****
610 PRINT
620 PRINT "   Tablica wyników"
630 FOR i=1 TO lz
640   PRINT "Zawodnik"; i; "-"; punkty(i);
650   PRINT "punktów karnych"
660 END

```

```

1000 REM ***** stół *****
1010 FOR i=1 TO stol
1020   PRINT "-";
1030 NEXT i
1040 FOR i=1 TO 4
1050   PRINT
1060   PRINT " II";
1070   FOR j=1 TO stol-8
1080     PRINT " ";
1090   NEXT j
1100   PRINT "II";
1110 NEXT i
1120 RETURN
2000 REM ***** upadek *****
2010 LET punkty(l)=punkty(l)+100
2020 PRINT "żaba spadła ze stołu!"
2030 PRINT "zalicza się 1 m"
2040 PRINT
2050 PRINT "dowolny klawisz"
2060 PRINT
2070 PRINT punkty(l); "punktów karnych"
2080 PRINT
2090 PRINT
2100   FOR m=1 TO stol
2110     PRINT " ";
2120   NEXT m
2130 PRINT
2140 GOSUB 1000
2150 PRINT " &"
2160 LET k%=INKEY$
2170 IF k%="" THEN GOTO 2160
2180 GOTO 520

```

### ATARI

Wszystko poza napisami piszemy wielkimi literami.

```

10 POKE 82,0: REM *** MARGINES 0 ***
50 DIM K$(5)
150 PRINT CHR$(125)
260   POSITION 0,0
480   IF PEEK(764)<>255 THEN OPEN #5, 4, 0, "K:": GET #5, KOD: CLOSE #5: K$=CHR$(KOD)
2160   IF PEEK(764)<>255 THEN OPEN #5, 4, 0, "K:": GET #5, KOD: CLOSE #5: K$=CHR$(KOD)

```

### SPECTRUM

```

140 LET stol=30
240   LET skok=INT(RND*50)
260   PRINT AT 0,0;
660 STOP

```

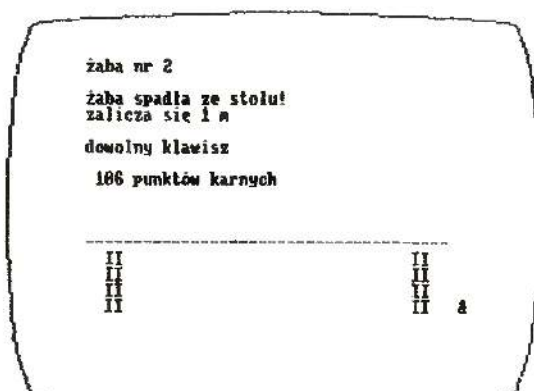
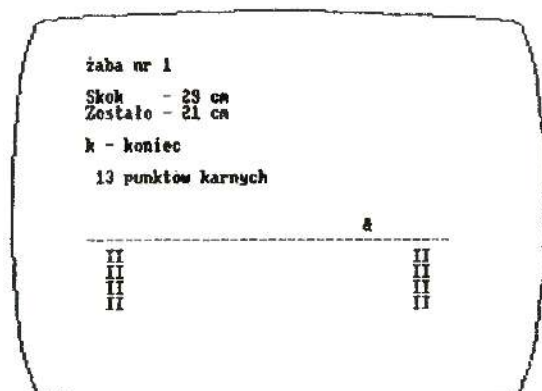
### COMMODORE

Wszystko poza napisami piszemy wielkimi lub wszystko małymi literami.

```

150 PRINT CHR$(147)
260   PRINT CHR$(19)
480   GET K$
2160   GET K$

```







**Gry komputerowe stanowią dla większości zapaleńców mikrokomputerów największy magnes. Ich urok, tajemniczość i niespożyta chęć pokonania komputera przyciąga coraz większe rzesze młodych ludzi. I choć mikrokomputerowy szal skończył się dość dawno temu, to jednak niezmiennie gry komputerowe są nadal na świeczniku.**

W 1985 roku przełamano aż kilka barier zdawałoby się nie do pokonania. Jedną z pierwszych była grafika — porównując choćby URIDIUM i ZAXXON możemy się mocno przekonać jak wielki skok jakościowy został tu dokonany.

W chwili obecnej coraz trudniej jest o grę dobrej klasy. Olbrzymia ilość produkcji powiela już dość oklepane wzory, zarówno pod względem tematyki jak i rozwiązania dźwiękowo-graficznego. Nie twierdzę wcale, że konkretne efekty jest łatwo uzyskać — wydaje się, że znakomity kontroler obrazu VIC dość długo będzie jeszcze inwencją dla programistów.

Coraz częściej spotykam się jednak z pytaniami użytkowników dotyczącymi programowania gier. Jak powstaje gra komputerowa? Ile czasu trzeba nad nią posiedzieć? Jakże programy są potrzebne?

Zacznijmy od tego, że nawet najmniejsza firma produkująca oprogramowanie zatrudnia zwykle zawodowych programistów, znających doskonale nie tylko asembler Commodore, ale również kilka innych asemblerów działających na bardziej profesjonalnych maszynach. Po drugie nie zawsze gra trzeba robić od nowa — czasami tworzy się po prostu jej wersję na dany typ mikrokomputera.

Projektowanie nowej gry zaczyna się zwykle od ułożenia jej scenariusza lub scenariuszy. Jeśli gra „przyjmie się” na rynku, to do-robienie drugiej części nie będzie stanowiło

## Przeciętny czas opracowywania dobrej jakościowo i tematycznie gry wynosi od kilku miesięcy do roku.

już problemu, można więc sprzedać dwa razy tę samą pracę. Nie muszą chyba dodawać, że jest to optyczne.

W dużych firmach istnieje ściśle podział pracy. Programista mający już jakieś doświadczenie ma do dyspozycji nie tylko profesjonalny komputer (np. VAX), lecz również kilku mniej doświadczonych programistów do pomocy. Zleca on im przygotowanie np. zestawu sprite'ów, wykonanie poszczególnych procedur umożliwiających dajmy na to zapis wyników czy wczytywanie poszczególnych poziomów gry. W takich wypadkach określone są zawsze pewne zasady i warunki, jakich ma się programujący trzymać — np. procedura zapisu musi znajdować się w ściśle określonym obszarze pamięci. Od ścisłego i sprawnego układania poszczególnych procedur często zależy dalsza kariera „młodszych” programistów.

Rzadko kiedy gra komputerowa jest układana na konkretnym typie mikrokomputera. Duży, profesjonalny sprzęt daje pod tym względem znacznie większe możliwości — w końcu bez większych problemów można na dużych komputerach wymodelować domowy mikrokomputer ze wszystkimi charakterystycznymi dlań parametrami. Drugim powodem jest szybkość działania „dużego” sprzętu — dzięki temu zmniejsza się znacznie czasochłonność zgrzywania ze sobą (i testowania przy okazji) poszczególnych fragmentów gry.

Oczywiście programiści mają także swoje specjalizacje. Weźmy np. przygotowywanie oprawy muzycznej dowolnej gry. W olbrzymiej większości wypadków muzykę tę tworzy muzyk na profesjonalnym syntezersze sprzężonym ewentualnie z kilkoma innymi instrumentami muzycznymi. Po uzyskaniu pożądanego efektów dźwiękowych są one zapisywane na taśmie, po czym poprzez odpowiednie przystawki tworzy się z tego konkretny program czy raczej dane. Ich wpisanie do pamięci jest już fraszką — w końcu jest to sygnał elektryczny o określonych wartościach.

Zadaniem programisty kordynującego jest teraz połączenie wszystkich opracowanych już procedur i danych w jedną całość. Na przykład podczas tworzenia gry THE EIDOLON kod źródłowy obejmował około 20 000 linii i przekraczał pojemność 500 KB! Z tego należało wybrać najlepsze fragmenty i zespolić je w jedną całość co zajęło dokładnie 24 KB.

Najciekawsze zadanie ma teraz jeden z pomocników. Grę przenosi się na mikrokomputer, po czym przez dłuższy okres czasu (nawet parę tygodni) pomocnik ten wyłącznie „macha” joystickiem dochodząc niemal do perfekcji w danej grze. Ma to na celu wyłapanie wszystkich ewentualnych błędów i słabych punktów, co z kolei pozwala na ich usunięcie przed ukazaniem się gry na rynku.

W końcowej fazie produkcji rozpoczyna się kampania reklamowa. Trzeba przyznać, że graficy od reklam przechodzą nieraz samych siebie. Już jeden rzut oka na reklamę powoduje, że osobnik zarażony wirusem gier zaczyna myśleć skąd zdobyć środki na jej zakup. Mniej więcej w tym samym czasie firma rozsyła bezpłatnie pierwsze wersje do liczących się czasopism komputerowych, wprowadza programy demonstracyjne do takich sieci jak QUANTUMLINK czy COMPUTERSERVE, gdzie żadni rozrywki zapaleńcy gier mogą z bliska przyjrzeć się nowej produkcji.

Jeśli gra chwyci, jeśli recenzje są dobre firma szykuje się do wydania drugiej części. Jeśli porównalbyś terminy ukazania się obu z nich, to okazałoby się, że druga część pojawia się na rynku w kilka miesięcy po pierwszej. Jest to zupełnie normalne, gdyż zwykle w takich wypadkach producent ma w zanadrzu odrzucone podczas pierwszej selekcji fragmenty kodu, które teraz złożą się na drugą część tej gry. Scenariusz jest już zwykle gotowy, wystarczy dorobić kilkanaście procedur i wejść (w podobny sposób jak za pierwszym razem) na rynek.

Najważniejszymi w pracy zawodowych programistów są programy narzędziowe. Mylą się Ci wszyscy, którzy sądzą, że programy te można ot tak po prostu kupić. W olbrzymiej większości wypadków tworzą je sami programiści zgodnie z własnymi potrzebami; programy te są niejednokrotnie bardziej chronione aniżeli cała produkcja danej firmy. I nic w tym dziwnego — opracowanie dobrego programu narzędziowego pochłania czas programisty i pieniądze producenta. Od jakości programu narzędziowego będą dalej zależały efekty w grze i szybkość jej opracowywania, a co za tym idzie — wpływy firmy. Zwykle pierwszym i najważniejszym takim programem jest makroassembler wykraczający daleko poza możliwości programów dostępnych na rynku, nawet profesjonalnym.

Przeciętny czas opracowywania dobrej jakościowo i tematycznie gry wynosi od kilku miesięcy do roku. Na przykład LUCASFILM GAMES jako pierwsza zastosowała w tworzeniu grafiki fraktale, co znacznie zmniejszyło i samą objętość programu, i znacznie polepszyło efekty graficzne. Gra THE EIDOLON tej firmy była jedną z bardziej skomplikowanych; 8 z 24 KB zajmowała grafika wysokiej rozdzielczości (generacja fraktali), 8 KB generacja różnych jaskiń i potworków. Pozostałe 8 KB było przydzielone na potrzeby kontrolera dźwięku (SID) i inne procedury.

Jak z tego widać, napisanie nawet prostej z pozoru gry wymaga sporo czasu i wysiłku. Warunkiem umożliwiającym dokonanie tego jest przede wszystkim doskonała znajomość samego komputera oraz biegłe programowanie w języku wewnętrznym. W przeciwnym wypadku gra będzie wyraźnie „odstawać” jakościowo, co raczej nie zapewni jej większego powodzenia.

Zwiększenie ilości reklam danej gry w czasopiśmie czy sieciach jest nieomylnym znakiem, że program idzie już słabo czy to ze względu na ukazanie się produkcji konkurencji, czy na działalność piratów. Organizuje się wtedy oferty specjalne, obniża ceny, sprzedaje się całe komplety gier. W pewnym momencie reklamy znikają i dana gra pojawia się już wyłącznie w cennikach firm sprzedających oprogramowanie lub też w ogłoszeniach o wyprzedazy. Jest to już jej śmierć ostateczna.

Oczywiście nie wszystkie gry podziela ten sam los. Gry typu ZAXXON czy STEALTH o, powiedzmy sobie, średnio-udanej tematyce giną w niepamięci. Przetrwają natomiast gry gigantyczne jak FLIGHT SIMULATOR II, czy niektóre gry strategiczne, w których bezsensowna strzelanina czy pogon za złodziejem Barabaszem z pędem kłuczy i antalkiem piwa ustępuje modelowaniu sytuacji bądź rozrywce umysłowej na naptawę wysokim poziomie.

ICEMAN

# NARODZINY ROZRYWKI