

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860-1674

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 7 (43)

LIPIEC 1989

CENA 220 ZŁ



**KRADZIONE
TUCZY?**

**DUSZKI
w Action!**

**OBRAZKI
W DATA**

JUMPER

Z DAŁA OD JAPONII

Wprawdzie mamy wakacje, ale przecież wszystko co piękne, szybko się kończy. Dlatego proponuję kilka refleksji na temat szkoły — nie naszej jednak, tylko japońskiej. Otóż również ją ostatnio zaczęto głośno krytykować — za to, iż nie przygotowuje właściwych kadr dla gospodarki narodowej. W tym kontekście zarzuca się tamtejszym władzom oświatowym niedociągnięcia w zakresie komputeryzacji procesu dydaktycznego. Są to oczywiście zarzuty formułowane na zupełnie innym poziomie, niż w Polsce.

W marcu 1988 roku w 24.692 państwowych szkołach podstawowych w Japonii, do których uczęszczają uczniowie klas od pierwszej do szóstej, znajdowały się 9523 komputery. W tym czasie w 14.746 pełnych i niepełnych szkołach średnich, do których uczęszczają uczniowie klas od siódmej do dwunastej, było na stanie 90.619 komputerów.

Widać z powyższego, że podstawowa masa komputerów dydaktycznych skoncentrowana jest na razie w Japonii w szkołach średnich, do których uczęszczają uczniowie w najlepszym „komputerowym” wieku 15—17 lat. Według danych z marca 1988 aż w 93,7 proc. tych szkół znajdowały się komputery do celów dydaktycznych. Na każdą szkołę przypadło statystycznie osiemnaście komputerów.

Efektom krytyki poziomu szkolnej edukacji informatycznej była rewizja planów nauczania i programów wyposażenia szkół w urządzenia elektronicznej techniki obliczeniowej. Zakłada się, że poczynając od roku 1993 nauczanie podstaw informatyki będzie się rozpoczynać już w szkołach podstawowych.

Aby przygotować się do tych nowych zadań, w wybranych szkołach podstawowych Japonii już obecnie uruchomiono eksperymentalne centra nauczające podstaw programowania i korzystania z komputerów. Na przykład szkoła podstawowa w miejscowości Kasiwa (prefektura Tiba) otrzymała 47 komputerów osobistych, a wszyscy nauczyciele znają podstawy

programowania i obsługi komputerów.

Jak twierdzi dyrektor tej szkoły, Iudzō Mera, jego uczniowie — dzieci w wieku od 6 do 11 lat — nie mają lęku i jakichkolwiek zahamowań posługiwania się tą nowoczesną techniką. „Uczniowie posługują się komputerem tak samo jak ołówkiem i kredkami” — mówi z zachwytem. Czyli — zupełnie jak u nas, co jeszcze raz potwierdza, że świat jest mały, a prawidłowości związane z korzystaniem z nowoczesnej techniki przejawiają się tak samo pod każdą szerokością i długością geograficzną.

Polskim szkołom (średnim) daleko jest oczywiście do takiego poziomu nasycenia komputerami jak w Japonii. Niemniej jednak sprzęt do naszych szkół systematycznie trafia i to nie tylko w dużych miastach, ale również i w terenie, gdzie coraz częściej można spotkać porządnie wyposażone pracownie informatyczne. Problemem zaczyna być teraz właściwe tego sprzętu wykorzystanie.

Na razie szkolne nauczanie informatyki odbywa się bardzo chaotycznie, w zależności od osobistych pasji i umiejętności nauczycieli, którym przydzielono „nauczanie informatyki”. Za mało jest natomiast materiałów dydaktycznych, specjalnie przygotowanych do tego celu programów, literatury itp.

Nie sądzę, aby potrafiąco rozwiązać ten problem centralnie — bo w ogóle z centralizmem, nie tylko w nauczaniu informatyki — jakoś nam ostatnio nie wychodzi. Dlatego szansę na poprawę sytuacji należy widzieć w szerokim rozwoju małych firm software'owych. Aby to jednak nastąpiło — warto pomyśleć o zapewnieniu im korzystnych warunków podatkowych. Te „straty” na pewno nam się już niedługo z nawiązką zwrócą!

A piszę o tym już teraz, w wakacje, gdyż to właśnie teraz decyduje się jak będzie wyglądał nadchodzący rok szkolny. Szkoda byłoby go znowu stracić!

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 Płaszczek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, 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

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltanska

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

Cena 150 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 055528 n. 150.000 egz. U-113

ZA MIESIĄC:

- **NARODZINY ROZRYWKI, czyli jak powstały gry komputerowe**
- **ŻABIE ZAWODY dla przed-szkolaków**
- **gry: Terramex, Captain Blood i Steel Thunder**
- **grafika wektorowa**
- **TRZY KANAŁY w muzyce na SPECTRUM**
- **OXFORD PASCAL na Commodore**
- **polskie znaki na ploterze SONY**
- **co jest trudne, co jest proste**





KRADZIONOŚĆ TUCZY

W kwietniu 1726 roku na Jamajce kapitan Green z Bristolu zaokrętował w charakterze bosmana na statku „Elizabeth” niejakiego Williama Fly’ a. Fly wśliznął się na statek z kilku kamratami znanymi z gotowości do każdego nikczemnego czynu. Postanowił bowiem zamordować kapitana i pierwszego oficera aby samemu objąć dowództwo i puścić się na piraterię.

U wybrzeży Gwinei nędznicy pochwycili Greena i wyrzucili go za burtę. Po czym podnieśli na maszt czarną flagę i poczęli łupić statki spotkane na morzu.

Pojmano ich niedaleko Bostonu. Fly został powieszony w łańcuchach u wejścia do przystani w tym mieście. Krótkie było panowanie tego zakamieniałego w zbrodni łotra, któremu brakowało tylko kunsztu i sił, żeby dorównać słynnym infamisom, pładze mórz.

A DZIŚ?

Oczywiście są piraci morscy, ale to nie oni nas interesują. Choć ich metody budzą odrazę i zasługują na podobne traktowanie jak trzysta lat temu.

Są też inni. Wykształceni, inteligentni, posiadający znaczne dochody i nie obawiający się przyszłości — przynajmniej w Polsce.

To piraci komputerowi. Czerpiący zyski ze sprzedaży masowo powielanych programów. Proceder ten jest możliwy, ponieważ w naszym kraju nie ma ochrony prawnej oprogramowania.

Fragment stanowiska Polskiego Towarzystwa Informatycznego w sprawie ochrony prawnej programów komputerowych: ...sytuacja, w której można za darmo otrzymać kopie programów, stwarza społeczne przekonanie, że oprogramowanie nic nie kosztuje lub

kosztuje niewiele. To przekonanie jest błędne i może mieć poważne skutki dla podejmowania decyzji gospodarczych.

Ich owocnej działalności zawdzięczamy powszechne w środowisku użytkowników przekonanie, że komputer jest wszystkim, a program niczym. Można go przecież nabyć za cenę niewspółmiernie niższą od ceny sprzętu.

Oferaty handlarzy giełdowych — 300-400 złotych za jeden program na „Spectrum” czy ogłoszenia prasowe licznych „Studiów komputerowych” proponujących „gry, programy użytkowe, literaturę... i co piąty program bezpłatnie” — potwierdzają tę tezę.

Pewien zachodni biznesmen posiadający udziały w jednej z firm sprzedających na polskim rynku sprzęt komputerowy po odwiedzeniu giełdy tak skomentował to, co zobaczył: „Nie! W tym kraju nie sprzedam ani jednego programu.”

Identyczną zasadę wyznają rodzimi programiści.

...kopiowanie na szeroką skalę oprogramowania zagranicznego bez zgody i woli producentów stanowi łamanie zasad przyjętych w obrocie w ich krajach i jest poważną barierą dla rozwoju prawidłowych stosunków handlowych z tymi producentami.

Zyski ze sprzedaży oprogramowania są w skali światowej znacznie większe od

tych ze sprzedaży sprzętu komputerowego.

Mimo to nie opłaca się pisanie programów dla polskiego użytkownika. Z góry wiadomo, że zysku nie będzie. Program albo zostanie skopiowany i odsprzedany dalej, albo po prostu „puszczony” po znanych.

Stan ten hamuje rozwój rodzimego przemysłu softwarowego. Nie sposób wyobrazić sobie firmy pragnącej wejść ze swoim towarem na zachodni rynek, jeśli w produkcji posługuje się ona kradzionym oprogramowaniem narzędziowym. Nikt z nimi nie będzie handlował. Nie kupuje się u pasera.

Podobna sytuacja może spotkać przedsiębiorstwa posługujące się „pirackim” oprogramowaniem w zupełnie innych celach.

O tym jak skuteczne mogą być tego rodzaju sankcje, niech przekona nas przykład Brazylii.

Amerykańskie korporacje handlowe oskarżyły firmy brazylijskie, eksportujące do USA kawę, o posługiwanie się kradzionym oprogramowaniem. A bez komputerów nie sposób obecnie prowadzić poważnych interesów.

Brazylijczycy zareagowali błyskawicznie na spadek zainteresowania kawą. Nie dosyć, że wykupili prawa użytkowania odpowiednich programów, to dodatkowo wprowadzili u siebie surowe przepisy chroniące producentów oprogramowania.

Bill Gates, szef największej na świecie firmy softwarowej Microsoft powiedział w jednym z wywiadów:

„Komputer kupuje się raz na cztery lata, a nasze programy trzeba kupować stale. Bez nich twój sprzęt staje się kupą złomu”.

ILE KOSZTUJE PROGRAM?

Wystarczy sięgnąć do amerykańskiego miesięcznika „BYTE” aby przekonać się, że nie są to małe kwoty. Oto kilka przykładów z marca br.

- DBase III Plus 433 \$
- Lotus 1-2-3 295 \$
- Ventura Publisher 2.0 470 \$
- Wordstar 2000+ 205 \$
- Turbo Pascal 4.0 58 \$

To są ceny profesjonalnego oprogramowania. Miłośnikom gier wystarczą mniejsze pieniądze.

- Defender of the Crown 37.99 \$
- King of Chicago 37.99 \$
- Sub Battle 27.99 \$
- King Quest IV 35.99 \$
- Leisure Suit Larry II 35.45 \$

Ceny te warto porównać z cenami sprzętu. To daje najlepsze wyobrażenie o rzeczywistej wartości programu. Zwykły 14-calowy monitor kosztuje 60 \$, kolorowy 270 \$, karta VGA do niego 329 \$. Komputer kompatybilny z IBM XT firmy VENDEX TECHNOLOGIES w podstawowej konfiguracji bez twardego dysku — 699 \$.

Z tego zestawienia wynika, że gdyby w Polsce ktoś chciał nabyć autoryzowany program, musiałby mieć bardzo dużo pieniędzy. Tańsze jest zwrócenie się do spółki z o.o. sprzedającej software po cenach, które np. państwowy użytkownik jest w stanie zaakceptować.

W tym miejscu niewielka dygresja.

Kilka lat temu doskonale „szły” w naszym kraju programy zastępujące prace księgowości w przedsiębiorstwach. Dokonywano zakupów sprzętu, organizowano kursy, „wdrażano” ...i po pewnym czasie okazywało się, że programy te są znakomite dla angielskich, amerykańskich czy zachodnioniemieckich przedsiębiorstw. One były napisane nie dla NAS.

Trzeba było przeprosić się z zaręczkami i ponownie przekwalifikować panie Basie i Zosie.

W Stanach Zjednoczonych i Europie Zachodniej firma softwarowa zapewnia jednocześnie 24-godzinny serwis. O każdej porze dnia i nocy użytkownik może skontaktować się telefonicznie z ludźmi, którzy doradzą mu, co ma robić, kiedy program nie chce działać. W przypadku korzystania z pirackich kopii nie jest to możliwe. Często podczas „łamania” program ulega

GRA O JUTRO

uszkodzeniu, co nie pozwala na pełne wykorzystanie jego możliwości.

Piraci oczywiście nie informują o tym swoich klientów.

Zupełnie inna sytuacja panuje na rynku gier komputerowych. Tu nie są potrzebne specjalne programy kopiujące (mimo to warto je mieć!), wystarczy dwukasetowy magnetofon.

Wtajemniczeni twierdzą, że zdolny nastolatek dysponujący prostym zestawem „magnetofon + komputer — jest w stanie zarobić w ciągu miesiąca około 150 tysięcy złotych.

Dawno już pojawiło się u nich przekonanie, że program jest „niczyj”, a zarabia na nim ten, kto potrafi błyskawicznie zorganizować „produkcję” i sprzedaż.

Profesor Władysław M. Turski w ten sposób skomentował to, co dzieje się na giełdach: „Nie ma się czemu dziwić. Ci młodzieńcy nie opanowali w szkole prostej zasady — **nie kradnij**”, dodał też: „Wydaje mi się, że w przyszłości będziemy mieli kłopoty”.

W obronie istniejącego stanu rzeczy często przytaczany jest argument odwołujący się do kosztów, jakie musielibyśmy ponieść z chwilą wprowadzenia w Polsce ochrony prawnej oprogramowania — trzeba byłoby uiścić opłaty licencyjne na rzecz producentów zachodnich.

Jeśli tak się stanie, to z pewnością sytuacja zmusi wielu do racjonalnego gospodarowania funduszami na zakupy. Być może okaże się, że to samo można wyprodukować w kraju taniej!

Uwagi te dotyczą mikrokomputerów. Duże komputery, to zupełnie inne problemy. Po pierwsze w Polsce prawie ich nie ma. Po drugie ich oprogramowanie przygotowywane jest „na zamówienie”, najczęściej w kilku tylko egzemplarzach. W związku z tym łatwo je chronić. Po trzecie zastosowanie oprogramowania dodatkowo eliminuje potencjalnych piratów. Kogo będzie ineresował program obsługi pasażerów nowego portu lotniczego na Okęciu.

Jeśli oczywiście taki port powstanie.

W dziedzinie oprogramowania dużych komputerów dawno już utraciliśmy kontakt ze światową czołówką.

Jeśli w tej chwili myślimy o tym, w jaki sposób zmniejszyć dzielący nas dystans, powinniśmy zacząć zachowywać się jak ludzie cywilizowani.

Polska uczestniczy w pracach Światowej Organizacji Własności Intelektualnej, a mimo to nasze prawo autorskie i wynalazcze jest w niedostatecznym stopniu przystosowane do ochrony programów komputerowych przed pirackim kopiowaniem.

Jak dotychczas nie było w naszym kraju ani jednego procesu o naruszenie praw autorskich w tej dziedzinie!

O ile fakt ten jest uzasadniony aktualnym stanem prawa, to dziwi brak reakcji na handel tłumaczeniami instrukcji i pod-

ręczników wydawanymi w swoistym „drugim obiegu”.

Prawo autorskie jednoznacznie wyowiada się na temat publikacji „bez wiedzy i zgody autora” Tymczasem bez najmniejszych problemów można kupić opisy gier, instrukcje obsługi, podręczniki programowania, itp. Stosnie ceny nie zniechęcają nabywców. Mimo wysiłków krajowych autorów i wydawców brak jest literatury informatycznej, szczególnie tej na poziomie podstawowym, przeznaczonej dla masowego użytkownika.

Stan ten powinien jak najszybciej ulec zmianie.

CO NAS CZEKA?

Z pewnością doczekamy się zmian w prawie autorskim, które uwzględnią interesy programistów. Nie zmieni to od razu starych nawyków, nie znikną z dnia na dzień ludzie handlujący kradzionymi programami. Będzie jednak podstawa prawna do walki z nimi. A co najważniejsze — może wreszcie doczekamy się rodzimego przemysłu softwarowego. Ta produkcja bardzo się opłaca.

Istniejące obecnie firmy, które sprzedają własne programy, nie będą musiały stosować drogich zabezpieczeń hardware'owych. Być może stanie się opłacalna produkcja oprogramowania dostępnego dla przeciętnego odbiorcy.

Mam nadzieję, że przy okazji zmienią się też nasze nieodobre nawyki.

Koszty zwłoki ponosimy wszyscy. Jestem przekonany, że prędzej czy później zachodnie firmy zareagują na bezprawne z ich punktu widzenia praktyki Polaków. To, że dotychczas nie znam przykładu takiej drastycznej reakcji świadczy, że nie jesteśmy dla nich liczącym się partnerem.

Obraz ten byłby niepełny bez informacji o tym, co mimo trudnych warunków próbuje się robić.

Podczas targów „Infosystem 89” spółka „Aplicom” przedstawiła autoryzowaną wersję programu AutoCad wraz z zabezpieczeniem sprzętowym. Podobnie postępuje spółka „Cyfronex” oferująca profesjonalny program składu gazetowego. Inni zabezpieczają się jak itopiań licząc na to, że uda się choć na jakiś czas odstraszyć piratów.

15 grudnia 1988 r. Polskie Towarzystwo Informatyczne przedstawiło swoje stanowisko w sprawie ochrony prawnej programów komputerowych. Trwają też w Sejmie prace nad nowelizacją prawa autorskiego. Mam nadzieję, że jeszcze w tym roku zostaną uchwalone konieczne poprawki. Wtedy może zniknie czarna flaga znad polskich komputerów.

Marek Czarkowski

PROPOZYCJA DEFINICJI PODSTAWOWYCH ELEMENTÓW PRAWA WŁASNOŚCI PROGRAMU

1. Przedmiotem prawa własności programu jest każdy program komputerowy ustalony w postaci:

- a) kodu programowego w wersji do bezpośredniego wykonania przez komputer.
- b) tekstu programu w języku symbolicznym lub innej postaci umożliwiającej automatyczne wygenerowanie kodu programu w wersji do bezpośredniego wykonania przez komputer.

1.1 Przedmiotem prawa własności programu jest również każdy szczegółowy opis programu, tj. taki opis, który umożliwia proste odwołanie do którejkolwiek postaci z wymienionych w p. 1.

2. Przedmiotem prawa własności programu jest osoba, która poniosła nakłady na opracowanie programu. W szczególności, jeżeli program opracowany został w wykonaniu umowy o pracę lub umowy o dzieło, prawo własności programu służy pracodawcy lub zamawiającemu.

2.1 Jeżeli kilka osób poniosło nakłady związane z opracowaniem programu, dla określenia prawa każdej z osób stosuje się odpowiednio przepisy prawa cywilnego dotyczące współwłasności.

2.2 Dornniemywa się, że prawo własności programu służy osobie, której nazwa uwidoczniła jest w szczególnym opisie programu lub która uwidoczniła zostaje w zewnętrznych przejawach działania programu (np. na ekranie, wydruku lub rysunku).

2.3 Bezpośrednim twórcą programu służy prawo ochrony dóbr osobistych zgodnie z zasadami prawa cywilnego. Nie obejmuje ono jednak prawa do decydowania o sposobie wykorzystania programu przez właściciela prawa do tego programu.

3. Prawo własności programu obejmuje prawo do:

- a) wyłącznego rozporządzania programem
- b) pobierania przychodów z tytułu eksploatacji programu

3.1 Prawo własności programu może być w całości lub części zbyte innej osobie

4. Poszczególne uprawnienia w zakresie prawa własności programu mogą być udzielone przez właściciela prawa do programu innej osobie w drodze umowy licencyjnej.

5. Korzystanie z cudzego programu może odbywać się wyłącznie na podstawie umowy zawartej z właścicielem prawa do tego programu lub osobą uprawnioną wg zasad określonych w p. 4.

6. Naruszeniami prawa własności programu są ponadto:

- a) kopiowanie programu,
- b) odtwarzanie tekstu programu w języku symbolicznym lub opisu programu na podstawie kodu programu w wersji do bezpośredniego wykonania.
- c) wykorzystanie istotnych rozwiązań zawartych w programie w opracowaniu innego programu, bez pisemnej zgody właściciela prawa do tego programu lub osoby uprawnionej wg zasad określonych w p. 4.
- 7. Opublikowanie przez właściciela prawa do programu szczegółowego opisu w książce lub czasopiśmie uważane jest za wyrażenie zgody na swobodne wykorzystywanie programu oraz zawartych w nim rozwiązań.
- 8. Usuwanie nazw osoby lub osób, którym przysługuje prawo własności programu oraz stwarzanie w jakikolwiek sposób przekonania, że prawo to przysługuje innej osobie lub osobom są ściągane karnie.
- 9. Właściciel prawa do programu lub osoba upoważniona wg zasad określonych w p. 4 może żądać od każdego, kto narusza zasady określone w p. 5 i 6, zaniechania tych naruszeń oraz wygania korzyści materialnych.

Opracował Jacek Irtlik
Polskie Towarzystwo Informatyczne
Warszawa 15 XII 1988 r.

MICRO

IMPREZA JAKICH WIELE



Tak można podsumować najkrócej wystawę „Infosystem 89” w intencji organizatorów miał to być polski CeBIT. Tymczasem sądząc przynajmniej po ilości i poziomie ofert zagranicznych intencji tej nie zrozumiano na Zachód od Łaby. Poza tak znanymi u nas już firmami jak ABC Data, Soft-Tronik, czy Polsin niewielu odwiedziło nas gości. Tymczasem, chociażby ze względu na dostępność dewiz na przetargach walutowych ci, którzy chcieli nabyć sam sprzęt woleli przeznaczyć na to dewizy. Stąd tak wielki ruch panował chociażby na stoisku wspomnianego „Soft-Tronika”, czy debiutującej na naszym rynku holenderskiej firmy „Grocom”.

Znacznie lepiej od wystawców zagranicznych zaprezentowali się rodzimi. Coraz więcej było kompleksowych systemów, coraz lepiej prezentował się rodzimy software, powstały często w wyniku kooperacji z partnerami zagranicznymi. Przykładem może tu być polska wersja „Auto-CAD-a” proponowana przez „Apikom” i licencjonowana przez brytyjski „Auto-Desk”, czy też „Cyfroset”, w pełni profesjonalny system służący do składu i lamania czasopism i książek. Ten z kolei powstał niejako „na zamówienie” angielskiej firmy „Monotype”, jednego z najpoważniejszych dostawców fotonaświetlań na rynek polski. Wszystko wskazuje zresztą na to, że wkrótce już zarówno „Bajtek”, jak i „Sztandar Młodych” będą korzystać z tego programu.

Dla młodych ludzi odwiedzających wystawę, a także dla wybranych klas kilku szkół poznańskich Fundacja Edukacji Komputerowej przygotowała atrakcję w postaci pracowni komputerowych urządzonych przez zajmujące się wprowadzaniem informatyki do oświaty instytucje z kilku krajów, m.in. z Bułgarii i Wielkiej Brytanii. Tym, którzy wciąż jeszcze mogą tylko marzyć o nauce w takich klasach i którzy nie mieli okazji zobaczyć „Infosystemu” dedykujemy ten fotoreportaż autorstwa Leopolda Dzikowskiego.

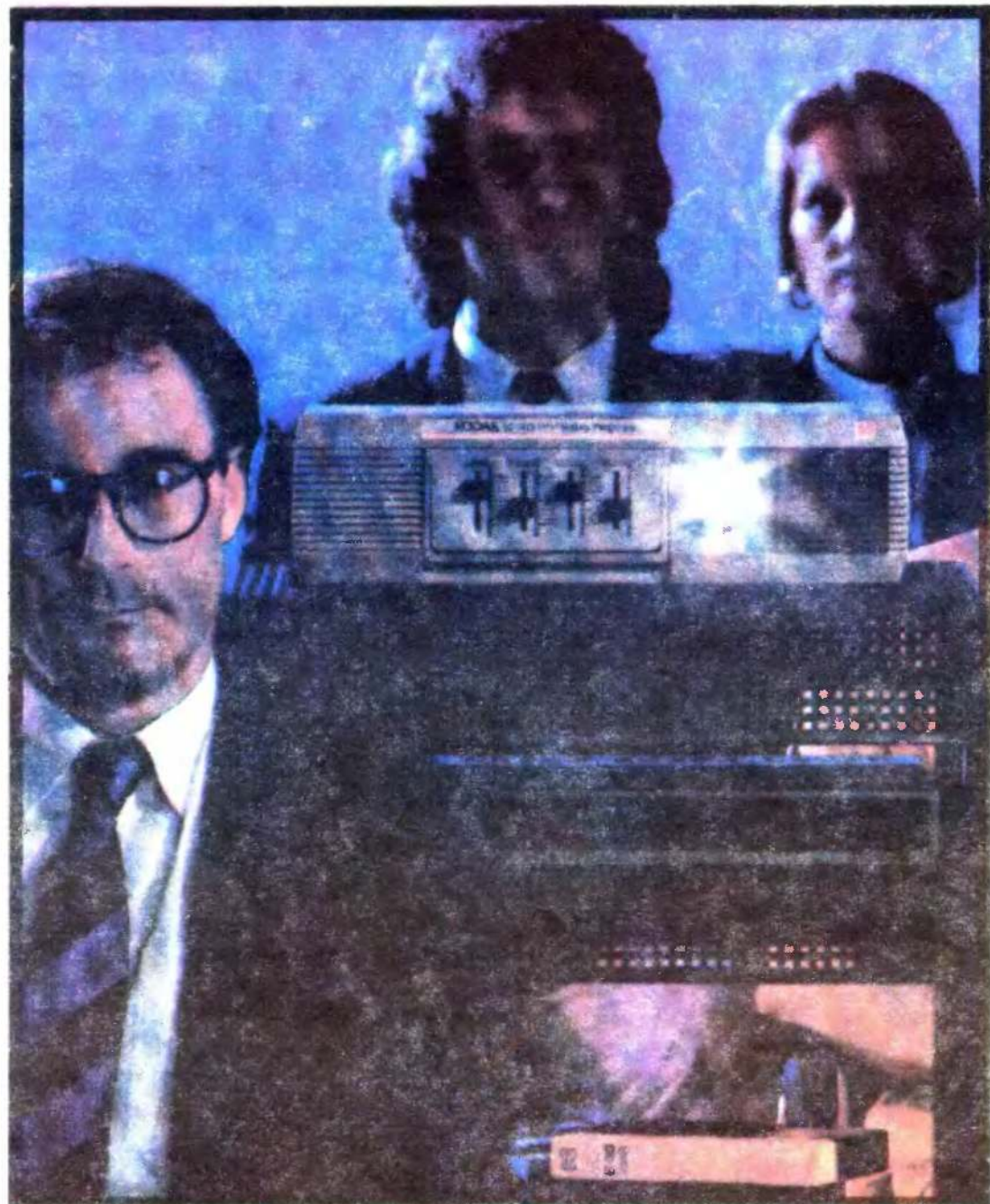
(go)

KARTY TELEFAXU DO IBM PC

Telefaxy jako urządzenia znacznie wygodniejsze od popularnych telexów są coraz częściej używane na świecie. Podstawową ich zaletą jest transmisja informacji w trybie graficznym, co pozwala przesyłać rysunki, fotografie lub tekst w alfabecie arabskim, chińskim albo jeszcze bardziej egzotycznym. Telefaxy są urządzeniami dość drogimi, najczęściej kosztują powyżej 1000 dolarów. Bardzo ciekawą ofertę stanowią więc znacznie tańsze karty rozszerzające do komputerów klasy IBM PC. Na rynku brytyjskim za 300 funtów oferowana jest karta Communicate C-FAX SR. Pozwala ona na pracę z szybkością 9600 bodów. Dostarczone oprogramowanie pozwala na wysyłanie tekstów tylko w formacie ASCII. Brak urządzenia wejściowego do wprowadzania rysunków (scanner) utrudnia pełne wykorzystanie tego bardzo użytecznego urządzenia. Konkurencyjną kartę, ale z lepszym oprogramowaniem dostarcza firma InterQuadram. JT Fax też kosztuje 300 funtów.

Z powodu istotnych ograniczeń rozwiązanie to, aczkolwiek tańsze od kompletnego oddzielnego telefaxu, nie wzbudza powszechnego entuzjazmu i traktowane jest na razie jako przejściowe.

KINO Z KOMPUTERA



Kolorowe monitory wysokiej rozdzielczości zapewniają dziś operatorowi komputera wspaniały obraz. Monitor jest także wystarczającym narzędziem wówczas, gdy chcemy efekty naszej pracy pokazać dwóm- trzem przyjaciołom. Gorzej jednak, a tak często bywa, gdy trzeba zorganizować dla kilkunastu lub kilkudziesięciu osób pokaz lub wykład z pomocą komputera i magnetowidu. Ekran monitora staje się wówczas zdecydowanie zbyt mały, szczegóły niewidoczne.

Lekarstwem na te kłopoty, ma być, według specjalistów firmy „Kodak” wprowadzany właśnie na rynek (premiera na CeBIT 89) projektor LC 500. Od produkowanych już projektorów wideo różni się on zdecydowanie małymi rozmiarami (410x110x245 mm), wagą 5,9 kg) oraz wspomnianą już uniwersalnością. Na tylnej ścianie projektora widnieją wejścia sygnałowe composite-video, RGB-TTL i analogowe-RGBS, zarówno do magnetowidu jak i komputera. Przełącznikiem wybiera się tylko skąd ma być w danej chwili pobierany

wyświetlany właśnie na ekranie obraz.

Niezwykle prosta jest obsługa techniczna projektora. Jego najbardziej zawodną część — żarówkę można wymienić w ciągu kilku sekund nie postępując się przy tym nawet śrubokrętem. Jakość kolorowego obrazu na ekranie (może nim być także zwykła ściana) jest nienaganna.

LC 500 ma stać się, w intencji jego producentów uniwersalną pomocą naukową dla szkół i wyższych uczelni, obowiązkowym wyposażeniem specjalistów od marketingu prezentujących często i w różnych miejscach wyroby swoich firm itp. W ulotce reklamowej Kodaka pisze się także o niewątpliwych zaletach posiadania takiego urządzenia w domu. Rzeczywiście zalet takich jest wiele. Jest jednak wszakże i mała wada, na nasze warunki dyskredytująca to urządzenie jako domową zabawkę — cena. Gdy na przełomie września i października LC 500 pojawi się w sklepach RFN, kosztować ma ok. 8500 marek.

(go)



**DOWÓD
W 90 SEKUND**

Spore zainteresowanie na targach CeBIT 89 wzbudzał prezentowany na stoisku firmy „Kodak” komputerowy system sporządzania dokumentów osobistych składający się z kamery wideo i połączonego z nim komputera wyposażonego w kolorową drukarkę termiczną (prezentowaną już w „Bajkowym Mikromagazynie”).

„Kodak” zaproponował aby, poza własnoręcznym podpisem właściciela wszystkie inne elementy dokumentu (przepusiki, prawa jazdy itp.) powstawały bez montażu. Dotyczy to w szczególności także kolorowego zdjęcia wmontowywanego przez komputer natychmiast po jego zaakceptowaniu przez fotografowanego. Kamera wideo jest znacznie bardziej cierpliwa niż papier fotograficzny — może długo i z uporem „wpatrywać się” w twarz, którą uwiecznia w dokumencie, tak długo dopóki jej operator i fotografowany nie uzgodnią, że właśnie to, a nie inne ujęcie udało się najlepiej.

Wpisana do pamięci komputera „stopklatka” jest następnie drukowana przy pomocy wideodrukarki razem z podawanymi z klawiatury danymi osobowymi, zapisanym za pomocą kodu paskowego numerem dokumentu, a także, jeśli jest to potrzebne, z wprowadzonym

wcześniej znakiem graficznym zastępującym pieczęć.

System „Kodaka” jest podobno bardzo bezpieczny, niebezpieczeństwo podrobienia tak zrobionego dokumentu minimalne. Na dodatek dzięki takiemu systemowi tworzyć można bez większych problemów pełną komputerową kartotekę, w której zapisane będą w formie elektronicznej także wizerunki posiadaczy dokumentów. Przed komputerem nie ukryje się nikt. Nawet broda, wąsy, czy peruka nie mogą go zwieść, w każdej bowiem chwili elektroniczny wizerunek może być poprawiony przez dodanie takich właśnie szczegółów. Nic dziwnego, że na CeBIT-cie największym wzięciem cieszyły się właśnie sporządzane od ręki (ok 90 sek.) na stoisku „Kodaka” takie dowody z małymi „poprawkami” — np. blond piękność z wąsami.

(go)



WPHW Dąbrowa Górnicza

ELEKTROIN

SKLEP Nr 163 41-300 DĄBROWA GÓRNICZA, ul. Sobieskiego 17
tel. 622 371 w godz. 10 - 18

OFERUJE:

- mikrokomputery 8-bitowe (Amstrad, Commodore, Atari)
- mikrokomputery klasy PC XT/AT
- urządzenia peryferyjne (drukarki, plotery, stacje dysków)
- akcesoria i materiały eksploatacyjne (dyskiety, kable, kasety barwiące, pudełka, ...)
- sprzęt video i CTV
- sprzęt elektroakustyczny klasy Hi-Fi
- sprzęt estradowy
- drobny sprzęt elektroniczny i gospodarstwa domowego

SKLEP PROWADZI SPRZEDAŻ POZARYNKOWĄ

PROWADZIMY KOMIS I SKUP W/W ARTYKUŁÓW

W TYM ROKU o 5% TANIEJ!

SERWIS 12-MIESIECZNY PROWADZI:



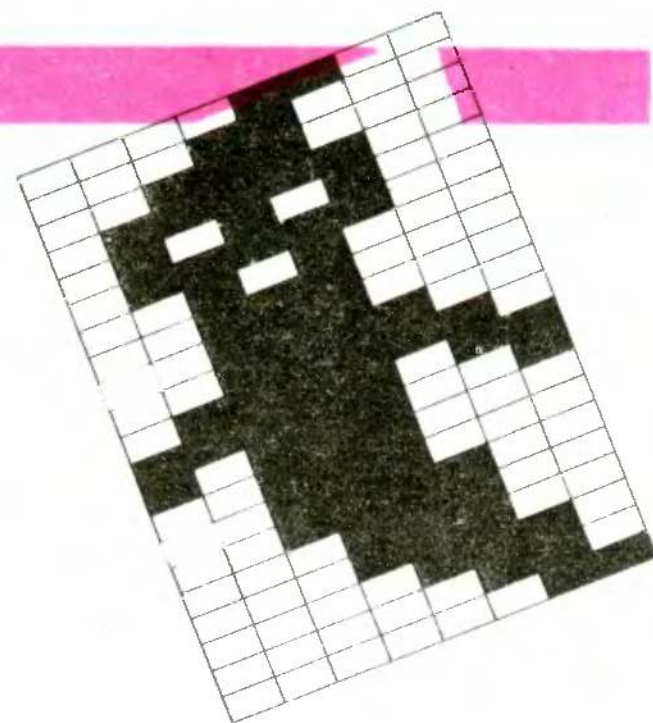
**PRZEDSIĘBIORSTWO
POSTĘPU TECHNICZNEGO
SPÓŁKA z O.O.**

41-303 DĄBROWA GÓRNICZA
Czerwonych Sztandarów 94
tel. 647 148 tlx. 031 28 98

SB 33

DUSZKI W

Action!



Grafika graczy i pocisków obsługiwana jest przez układy komputera, istnieje więc możliwość wykorzystania jej w każdym języku. Można ją stosować również w Action!

Obsługa duszków wymaga zaprojektowania całościowego systemu, który umożliwi realizację wszystkich możliwych funkcji. Przede wszystkim konieczne jest zdefiniowanie kilku zmiennych globalnych, które uproszczą stosowanie grafiki graczy i pocisków. Są to trzy zmienne proste oraz siedem tablic:

PMMode — określa wybraną rozdzielczość;
HitClr — służy do kasowania rejestrów kolizji;
PMBAAdr — adres początkowy pamięci duszków;
HPos — zawiera poziome pozycje duszków;
VPos — zawiera pionowe pozycje duszków;
MMask — zawiera maski bitowe do odczytu danych pocisków;
PMHPos — ustala poziome pozycje duszków;
PMSize — ustala szerokość duszków;
PMField — określa wielkość pamięci dla każdego duszka;
PMEmpy — określa wielkość wolnej pamięci powyżej PMBAAdr.

Elementy tablic, które mają wymiar osiem, dotyczą poszczególnych duszków. Tablica MMask ma cztery elementy, gdyż dotyczy tylko pocisków. Cztery elementy tablicy PMSize określają szerokość graczy, a piąty szerokość wszystkich pocisków. Tablice trzejelementowe służą do określania parametrów w zależności od wybranej rozdzielczości.

WŁĄCZANIE I WYŁĄCZANIE

Przygotowanie do pracy systemu duszków wymaga zarezerwowania odpowiedniego obszaru w pamięci komputera oraz wyzerowania wszystkich rejestrów, które z nim współpracują. Następnie — w zależności od wybranej rozdzielczości — ustalany jest adres początkowy pamięci duszków (PMBAAdr) i jego starszy bajt (PMBBase). Dla P/MG jest zawsze wyznaczony obszar pamięci leżący poniżej pamięci obrazu. Dla zabezpieczenia tego obszaru zmieniana jest wartość szczytu pamięci (MemHi). Zapamiętanie poprzedniej wartości w zmiennej OldMHi umożliwia późniejsze przywrócenie stanu pierwotnego.

Wszystkie te czynności wykonuje procedura PMGraphics, której parametrem jest żądana rozdzielczość. Wartość dwa odpowiada tu rozdzielczości dwuliniowej, wartość jeden — jednoliniowej, zaś zero powoduje wyłączenie duszków. Każda inna wartość parametru jest zmieniana na zero w początkowej fazie procedury, dzięki czemu unika się błędnych wywołań. Konieczne jest także sprawdzenie, czy grafika P/M była wcześniej uruchomiona. Brak takiej kontroli spowodowałby zajęcie całej pamięci operacyjnej już po kilku zmianach rozdzielczości. Warto zauważyć, że poprzez użycie tablicy PMEempy zmniejsza się zajmowany przez duszki obszar pamięci do niezbędnego minimum. Ponadto adres początkowy pamięci duszków jest porównywany z końcowym

adresem programu, co chroni przed zniszczeniem treści programu przez dane duszków.

KOLOR DUSZKA

Do wyboru koloru duszka służy procedura PMColor, którą wywołuje się podobnie jak biblioteczną procedurę SetColor, lecz zamiast numeru rejestru koloru trzeba podać numer gracza. Kolory pocisków są zawsze takie same jak kolory graczy o identycznych numerach. Działanie tej procedury jest bardzo proste i nie wymaga opisu.

Zarówno w tej procedurze, jak i we wszystkich pozostałych przyjęty został system oznaczeń stosowany w Basicu XE. W systemie tym duszki mają numery od 0 do 7. Duszkami 0–3 są gracze o takich numerach, a duszkami 4–7 są pociski o numerach mniejszych o cztery, czyli również od 0 do 3. Na przykład, parametr wskazujący gracza 1 ma wartość 1, a wskazujący pocisk 1 ma wartość 5.

ADRES DANYCH

Równie prosta jest funkcja PMAdr, która zwraca adres początkowy obszaru pamięci zajmowanego przez dane wskazanego gracza (parametry od 0 do 3). Ponieważ dane wszystkich pocisków zajmują ten sam obszar, to dla parametrów od 4 do 7 zwracana jest ta sama wartość. Sposób obliczenia adresu jest oczywisty dla każdego, kto zna podział pamięci pomiędzy poszczególne duszki.

KASOWANIE PAMIĘCI

Często zachodzi konieczność całkowitego skasowania obszaru danych konkretnego duszka. Stosuje się to w celu usunięcia tego duszka lub całkowitej zmiany jego kształtu. Operacja ta jest realizowana przez procedurę PMClr, której parametrem jest numer duszka. Wykonanie tej czynności dla graczy jest bardzo łatwe — wystarczy użyć procedurę Zero. Pociski natomiast sprawiają znacznie większy kłopot, gdyż każdy z nich zajmuje tylko dwa bity każdego bajtu danych. Zmusza to do skorzystania z pętli FOR oraz tablicy masek MMask. W obu przypadkach adres kasowanego bloku określany jest przy pomocy funkcji PMAdr.

POZYCJA DUSZKA

Ruch dowolnego duszka można zrealizować wywołując procedurę PMMove. Jej parametrami są kolejno: numer duszka, współrzędna pozioma i współrzędna pionowa. Na początku podana współrzędna pionowa jest porównywana z jej aktualną wartością. Jeśli duszek przesuwany jest w poziomie (dy=0), to wystarczy wpisanie nowej współrzędnej x do tablic HPos i PMHPos. W przeciwnym przypadku trzeba dokonać przeniesienia danych duszka o liczbę bajtów wskazaną przez zmienną dy. W tym celu cały blok pamięci jest najpierw przepisywany do pomocniczej tablicy temp, a dopiero potem na właściwe miejsce.

Takie postępowanie jest konieczne dla zapewnienia poprawnego ruchu w dół i w górę. Niestety, wymaga to przepisania pół lub ćwierć kilobajta (zależnie od rozdzielczości) i jest stosunkowo powolne (jak na Action!). Na szczęście Action! umożliwia bardzo proste włączenie do programu procedur w języku maszynowym. Pierwszą pętlę FOR można zastąpić taką procedurą:

```
LDY #0
LDA pptr
STA $A0
LDA pptr+1
STA $A1
LOOP LDA ($A0),Y
AND mask1
STA temp,Y
INY
CPY plien
BNE LOOP
```

listing 1

Zmieniając to na kod liczbowy otrzymamy tablicę, którą należy umieścić w procedurze PMMove we wskazanym miejscu (zamiast pętli).

```
[ $A0 0 $AD pptr $B5 $A0 $AD pptr+1
  $B5 $A1 $B1 $A0 $2D mask1 $99 temp
  $C8 $CC plien $D0 $F2 1
```

listing 2

Podobna procedura posłuży do przepisania zawartości tablicy temp do odpowiedniego obszaru pamięci duszków. Jest ona nieco dłuższa, gdyż trzeba pilnować, aby nie został naruszony blok danych następnego duszka.

```
LDX #0
LDY offy
LDA pptr
STA $A0
LDA pptr+1
STA $A1
LOOP LDA ($A0),Y
AND mask2
ORA temp,X
STA ($A0),Y
INY
CPY plien
BNE SKIP
LDY #0
SKIP INX
CPX plien
BNE LOOP
```

listing 3

Po zmianie na liczby otrzymujemy drugą tablicę. Zastępujemy nią drugą pętlę FOR.

```
[ $A2 0 $AC offy $AD pptr $B5 $A0 $AD
  pptr+1 $B5 $A1 $B1 $A0 $2D mask2
  $1D temp $91 $A0 $C8 $CC plien $D0 2
  $A0 0 $E8 $EC plien $D0 $E6 1
```

listing 4

W tym przykładzie doskonale widoczna jest elastyczność systemu Action!. Użytkownik nie musi znać miejsca, w którym zostaną umieszczone procedury i wykorzystywane przez nie dane.

Wpisane do tablicy zmienne określające adresy zmiennych zostaną przez **Action!** zamienione na odpowiednie wartości w procesie kompilacji programu.

TWORZENIE DUSZKA

Mając już procedury niezbędne do operowania duszkami na ekranie możemy przystąpić do stworzenia ich. Będzie to polegało na umieszczeniu zestawu danych w odpowiednim miejscu pamięci oraz ustaleniu wybranej szerokości i pozycji na ekranie. W tym celu należy wywołać procedurę **PMCreate**, a jako jej parametry podać kolejno: numer duszka, tablicę zawierającą dane duszka, liczbę elementów tej tablicy, szerokość duszka i współrzędne początkowej pozycji. Procedura przepisuje tylko podane dane na właściwe miejsce oraz wpisuje pozostałe parametry do odpowiednich rejestrów i tablic. Jeśli we wskazanym obszarze znajduje się już inny wzór duszka, to zostanie on usunięty tylko w tych miejscach, gdzie pokrywa się z nowym wzorem.

KOLIZJA

Do pełnego szczęścia brakuje już tylko wykrywanie kolizji duszków z innymi duszkami i elementami planszy. Wykorzystamy tu funkcję **PMHit**, która zwraca wartość i w przypadku wystąpienia wskazanej kolizji, a zero, gdy takiej kolizji nie było. Parametrami funkcji są dwa numery. Pierwszy z nich wskazuje duszka, którego kolizję badamy. Drugi parametr jest numerem obiektu, z którym badany duszek mógł się zderzyć. Wartość tego numeru z zakresu od 0 do 3 odpowiada numerowi gracza, zaś wartość od 4 do 7 jest zwiększonym o cztery numerem rejestru koloru, który był wykorzystany do narysowania elementu planszy. Jest to więc numer użyty w procedurze **SetColor** zwiększony o cztery.

Funkcja **PMHit** nie kasuje zawartości rejestrów kolizji, co umożliwi kolejne sprawdzanie kilku zderzeń. Po wykonaniu wszystkich niezbędnych wywołań tej funkcji trzeba skasować te rejestry, aby zapewnić poprawne wykrywanie dalszych kolizji. W tym celu wystarczy przypisać zmiennej **HitClr** dowolną wartość. UWAGA: wszystkie rejestry kolizji są kasowane jednocześnie.

ZMIANA GRAPHICS

Zmiana trybu graficznego obrazu przy aktywnych duszkach może spowodować częściowe nałożenie się wykorzystywanych obszarów pamięci. Aby tego uniknąć, trzeba przed każdą procedurą **Graphics** wyłączyć duszki. Czynność tę także przerzucimy na barki komputera. Wykorzystamy tu sposób działania **Action!**, który powoduje, że każda nowa definicja procedury lub zmiennej powoduje anulowanie poprzedniej o tej samej nazwie. Dotyczy to również procedur bibliotecznych. Napiszemy więc, po prostu, nową procedurę **Graphics**, która zastąpi procedurę biblioteczną.

STOSOWANIE

Na zakończenie wypada wyjaśnić, jak korzystać z opisanego zestawu procedur. Należy umieścić go na początku pisanego programu, a po nim wpisać słowo **MODULE** i resztę własnego programu. Teraz można już swobodnie korzystać ze wszystkich pokazanych procedur.

W celu uzyskania duszków należy w dalszej części programu wykonać kilka prostych czynności:

- ustalić odpowiedni tryb graficzny (**Graphics**);
- wybrać rozdzielczość duszków (**PMGraphics**);
- wyczyścić odpowiedni obszar pamięci (**PMClr**);
- utworzyć wybrane duszki (**PMCreate**).

Następnie można dowolnie przesuwać utworzone duszki przy pomocy **PMMove**, badać ich zderzenia przez **PMHit**, usuwać je przy użyciu **PMClr** oraz ponownie tworzyć wywołując **PMCreate**. Możliwa jest również zmiana rozdzielczości, lecz niszczy to wszystkie istniejące duszki. Grafika graczy i pocisków jest aktywna, aż do zmiany trybu graficznego lub wywołania procedury **PMGraphics(O)**.

Wojciech Zientara

```

; Duszki w Action!
; Wojciech Zientara
; Copyright (c) Bajtek

BYTE PMMode=IOI,HitClr=#DOIE
CARD PMBAdr

BYTE ARRAY
  HPos(8)=[O O O O O O O O],
  VPos(8)=[O O O O O O O O],
  MMask(4)=[#FC #F3 #CF #3F],
  PMHPos(8)=#D000,PMSize(5)=#D008

CARD ARRAY
  PMField=[O 256 128],
  PMAEmpty=[O 768 384]

; Włączanie/wyłączanie duszków
PROC PMGraphics(BYTE mode)

  BYTE DMAct1=#22F, GTIAct1=#26F,
  PMBase=#D407,PMGraf=#DOOD

  CARD MemHi=#2E5,OldMH1,AppMH1=#E

  CARD ARRAY PMMask=[O #F800 #FC00],
  PMMem=[O #800 #400]

  IF mode > 2 THEN
    mode = 0
  FI
  Zero(PMHPos,8)
  Zero(HPos,8)
  Zero(VPos,8)
  Zero(PMSize,5)
  Zero(PMGraf,5)
  IF PMMode#0 THEN
    MemHi=OldMH1
    DMAct1=#22 GTIAct1=0
  FI
  IF mode=0 THEN
    DMAct1=#22 GTIAct1=0
  ELSE
    IF mode=1 THEN
      DMAct1=#3E
    ELSE
      DMAct1=#2E
    FI
    OldMH1=MemHi
    PMBAdr=(MemHi-PMMem(mode)-#80) &
    PMMask(mode)
    PMBase=PMBAdr RSH 8
    IF PMBAdr<AppMH1 THEN
      DMAct1=#22 GTIAct1=0
    RETURN
  FI
  MemHi=PMBAdr+PMAEmpty(mode)
  GTIAct1==#C0%1
FI
PMHods=mode
RETURN

; Ustalenie koloru duszka
PROC PMColor(BYTE n,hue,lum)

  BYTE ARRAY ColPM(4)=#2C0

  n==#3
  ColPM(n)=(hue LSH 4) % (lum & #0F)
RETURN

; Adres danych duszka
CARD FUNC PMAAdr(BYTE n)
  n==#7
  IF n>=4 THEN
    n=0
  ELSE
    n==+1
  FI
RETURN(PMBAdr+PMAEmpty(PMMode)+(n*
  PMField(PMMode)))

; Kasowanie obszaru pamięci duszka
PROC PMClr(BYTE n)
  CARD ctr
  BYTE ARRAY PIAAdr
  n==#7
  PIAAdr=PMAAdr(n)
  IF n<4 THEN
    Zero(PIAAdr,PMField(PMMode))
  ELSE
    n==-4
    FOR ctr=0 TO PMField(PMMode)-1
      DO
        PIAAdr(ctr)==&MMask(n)
      OD
  FI
RETURN

```

```

;Umieszczenie duszka na pozycji x,y
PROC PMMove(BYTE n,x,y);
  CARD i
  BYTE offy,plien,mask1,mask2
  INT dy
  BYTE ARRAY temp(256),pptr
  IF PMMode=0 THEN RETURN FI
  n==#7
  dy=y-VPos(n)
  IF dy=0 THEN
    PMHPos(n)=x HPos(n)=x RETURN
  FI
  pptr=PMAAdr(n)
  plien=PMField(PMMode)
  IF dy>=0 THEN
    offy=dy
  ELSE
    offy=plien+dy
  FI
  IF n<4 THEN
    mask1=255 mask2=0
  ELSE
    mask2=MMask(n&3) mask1=mask2!#FF
  FI
  FOR i=0 to plien-1 ; tu pierwsza
    DO ; procedura
      temp(i)=pptr(i)&mask1 ; w języku
    OD ; maszynowym

  FOR i=0 to plien-1 ; tu druga
    DO ;
      pptr(offy)==&mask2 % temp(i)
      offy==+1 ;
      IF offy>=plien THEN ; procedura
        offy=0 ; w języku
      FI ; maszynowym
    OD ;

    VPos(n)=y PMHPos(n)=x HPos(n)=x
  RETURN
; Uruchomienie duszków
PROC PMCreate(BYTE n BYTE ARRAY pm
  BYTE len,size,x,y)
  BYTE i,mask,tmp,temp,oldsize=[O]
  BYTE ARRAY pptr,msize=[O 1 0 3]
  n==#7
  IF n<4 THEN
    mask=0
  ELSE
    temp=n&3 mask=MMask(temp)
  FI
  pptr=PMAAdr(n)
  FOR i=0 to len-1
    DO
      pptr(i+y)==&mask%pm(i)
    OD
    size==-1
    IF n<4 THEN
      PMSize(n)=size
    ELSE
      tmp=(msize(size) LSH (temp LSH 1))
      oldsize==& mask % tmp
      PMSize(4)=oldsize
    FI
    PMHPos(n)=x HPos(n)=x VPos(n)=y
  RETURN

; Kolizje duszków
BYTE FUNC PMHit(BYTE n,cn)
  BYTE ARRAY pmpf(8)=#D000,pmpm(8)=#D008
  n==#7
  IF n<4 THEN
    n==+4
  ELSE
    n==-4
  FI
  IF cn<4 THEN
    RETURN ((pmpm(n) RSH cn) & 1)
  ELSE
    cn==#3
    RETURN ((pmpf(n) RSH cn) & 1)
  FI
RETURN (0)
; Nowa procedura Graphics
PROC Graphics(BYTE mode)
  Close(6) PMGraphics(O)
  Open(6,"S:",(mode & #FO) ! #IC,mode)
RETURN

```


ATARI BEZ TAJEMNIC

Wśród ogromnej rzeszy użytkowników Atari 800XL/65XE znajduje się spora grupa osób, które chętnie poznałyby system operacyjny komputera. Zdobycie jego tajemnic dostarczy wiele satysfakcji wszystkim tym, którzy nie boją się zmagania z trudnymi problemami. Zdobyte tą drogą informacje przydadzą się nie tylko jako pomoc w programowaniu. Pozwolą one zrozumieć jak funkcjonuje komputer i optymalnie wykorzystać jego zalety.

Po kilku takich słowach zaczęły w niejednej głowie zrodzić się pytania, jak zabrać się do złamania systemu operacyjnego. Zadanie to nie jest takie proste, nawet jeśli dysponujemy odpowiednimi pakietami oprogramowania.

Wymaga to wielu godzin pracy i żmudnych analiz. Znacznie łatwiej jest skorzystać z gotowych pomocy.

Pierwszą w Polsce i jedną z najlepszych w świecie książek, w których znajdziemy bardzo szczegółowy opis systemu operacyjnego, jest czteroczęściowa „Mapa pamięci Atari XL/XE” napisana przez Wojciecha Zientarę. Pierwsza część, „Podstawowe procedury systemu operacyjnego”, to klucz do poznania całego zagadnienia.

Opisano tu, co dzieje się w komputerze po włączeniu zasilania, jak rozpoznaje on dołączone urządzenia zewnętrzne i w jaki sposób czytane są informacje zapisane na magnetofonie, stacji dysków. Kolejny blok procedur, to procedury przerwań. Są one jedną z najmocniejszych stron systemu operacyjnego Atari i dlatego programiści bardzo chętnie posługują się nimi przy pisaniu programów. W dalszej części książki opisano procedury zmienioprzecinkowe, które znajdują zastosowanie we wszelkiego rodzaju obliczeniach.

Dla wygody czytelników autor opracował szereg dodatków ułatwiających korzystanie z książki. Znajdują się wśród nich zestawienie adresów wszystkich procedur systemu operacyjnego, tabela z adresami rejestrów, zmienne systemowe i słowniczek terminów informatycznych.

Książkę poleciłbym szczególnie tym, którzy poważnie myślą o pisaniu programów na Atari XL/XE. Będzie ona dla nich jedną z podstawowych pomocy.

(J.J.)

Wojciech Zientara, „Mapa pamięci Atari XL/XE — Podstawowe procedury systemu operacyjnego”, Systemy Ośrodka Elektronicznej Techniki Obliczeniowej, Warszawa 1988, Wyd. I, Nakład 5000 egz.

ROCKFORD

— DLA KAŻDEGO —

Z pewnością każdy zna śmieszna figurkę Rockforda z gry „Boulder Dash”. Wpadłem na pomysł, aby wykorzystać ją do uatrakcyjnienia programów edukacyjnych dla dzieci.

Programy edukacyjne pisane w GRAPHICS 0 są uzupełniane postacią Rockforda, który jest dla dzieci uosobieniem komputera. Gdy dzieci odpowiadają, on śmieje się za poprawną odpowiedź lub gani za złą.

Pokazany podprogram należy dołączyć do własnego programu. Jego użycie jest bardzo proste. Procedura GOSUB 870 powoduje ukazanie się Rockforda na pozycji określonej zmiennymi X i Y. Potem można wykonać dowolną sekwencję procedur tak, jak jest to pokazane w wierszu 1080. W międzyczasie na ekranie może się pojawiać tekst, dzieci mogą odpowiadać na pytania itp. Usunięcie postaci Rockforda wykonuje procedura GOSUB 890.

Zdeněk Rubý

```
SD 10 REM : ROCKFORD
FM 20 REM : Zdenek Ruby
DA 30 REM Copyright (C) Baitek
BB 40 REM
IR 100 GRAPHICS 0:POKE 710,0:POKE 712,0:P
OKE 752,1
QG 105 POSITION 2,7:?"-----
-----"
GY 110 POSITION 5,10:?"Obraz zgasnie ter
az na minute"
MU 115 POSITION 2,13:?"-----
-----"
QF 120 FOR DEL=1 TO 500:NEXT DEL
SM 125 REM DEFINIOWANIE ZNAKU ROCKFORDA
KF 130 I=PEEK(106)-4:POKE 106,I
NJ 135 GRAPHICS 0:POKE 756,I:I=256*I
OO 140 POKE 752,I:?" ";:POKE 710,1
IA 145 FOR J=0 TO 1023:POKE I+J,PEEK(5734
4+J):NEXT J
JZ 150 RESTORE 160:FOR J=1 TO 200:READ K:
POKE 519+J+I,K:NEXT J
BC 155 GOTO 1000:REM DEMONSTRACJA
DF 160 REM ROCKFORD
PT 161 DATA 12,31,31,51,51,31,3,0
XJ 162 DATA 15,18,39,192,7,12,12,60
BN 163 DATA 48,248,248,204,204,248,192,0
GE 164 DATA 240,72,228,3,224,48,48,60
SM 165 DATA 12,31,31,63,51,31,3,0
VF 166 DATA 12,31,31,63,63,31,3,0
EL 167 DATA 48,248,248,252,204,248,192,0
HM 168 DATA 48,248,248,252,252,248,192,0
NC 169 DATA 15,18,39,192,7,12,60,0
DK 170 DATA 15,18,39,192,7,60,0,0
GH 171 DATA 240,72,228,3,224,48,60,0
KE 172 DATA 240,72,228,3,224,60,0,0
LQ 173 DATA 15,18,39,24,7,12,12,60
EY 174 DATA 15,18,23,12,7,12,12,60
WK 175 DATA 240,72,228,24,224,48,48,60
XV 176 DATA 240,72,232,48,224,48,48,60
WA 177 DATA 240,72,232,48,224,48,60,0
WH 178 DATA 15,18,23,12,7,12,60,0
FN 179 DATA 15,18,23,12,7,12,12,60
FZ 180 DATA 243,74,228,0,224,48,48,60
RF 181 DATA 255,192,228,194,201,228,210,1
93
LC 182 DATA 255,3,11,3,7,75,147,3
VH 183 DATA 194,196,200,194,228,200,192,2
55
JN 184 DATA 131,67,11,71,35,23,3,255
II 185 DATA 255,255,195,227,243,255,255,2
55
SE 195 REM Kostka
MT 200 POSITION X,Y:?" CHR*(25);CHR*(25)
DN 205 POSITION X,Y+1:?" CHR*(25);CHR*(25)
ND 210 REM FOR I=1 TO 5:GOSUB 320:NEXT I
ZP 215 RETURN
NV 225 REM Rozprysk
EY 230 POSITION X,Y:?" CHR*(21);CHR*(22)
AE 235 POSITION X,Y+1:?" CHR*(23);CHR*(24)
ZG 240 RETURN
PR 295 REM PAUZA I
```

```
UH 300 FOR DEL=1 TO 20:NEXT DEL:RETURN
IA 315 REM PAUZA II
XH 320 FOR DEL=1 TO 100:NEXT DEL:RETURN
DQ 395 REM Postoj Rockforda
OP 400 FOR I=0 TO 1:FOR J=0 TO 1
QR 405 POSITION X+I,Y+J:?" CHR*(2*I+J+1)
HN 410 NEXT J:NEXT I:GOSUB 320:RETURN
QR 435 REM Mruganie oczami
SR 440 GOSUB 320
IG 445 POSITION X,Y:?" CHR*(5);CHR*(7)
RV 450 GOSUB 300
KO 455 POSITION X,Y:?" CHR*(6);CHR*(8)
RX 460 GOSUB 300
IK 465 POSITION X,Y:?" CHR*(5);CHR*(7)
RZ 470 GOSUB 300
ZO 475 POSITION X,Y:?" CHR*(1);CHR*(3)
ZQ 480 RETURN
WM 495 REM Przytupywanie
NR 500 FOR I=1 TO 5
IC 505 POSITION X,Y+1:?" CHR*(9)
GP 510 GOSUB 300:GOSUB 300
AX 515 POSITION X,Y+1:?" CHR*(2)
GR 520 GOSUB 300:GOSUB 300
GO 525 NEXT I
JJ 530 GOSUB 320:GOSUB 320
ZW 535 RETURN
LL 545 REM Maszerowanie
OB 550 FOR I=1 TO 5
KB 555 POSITION X,Y+1:?" CHR*(9);CHR*(15)
RY 560 GOSUB 300
XS 565 POSITION X,Y+1:?" CHR*(10);CHR*(16)
SA 570 GOSUB 300
KF 575 POSITION X,Y+1:?" CHR*(9);CHR*(15)
SC 580 GOSUB 300
KC 585 POSITION X,Y+1:?" CHR*(2);CHR*(4)
HF 590 GOSUB 300:GOSUB 300
UB 595 POSITION X,Y+1:?" CHR*(13);CHR*(11)
RN 600 GOSUB 300
VX 605 POSITION X,Y+1:?" CHR*(14);CHR*(12)
RP 610 GOSUB 300
TM 615 POSITION X,Y+1:?" CHR*(13);CHR*(11)
RR 620 GOSUB 300
JR 625 POSITION X,Y+1:?" CHR*(2);CHR*(4)
GU 630 GOSUB 300:GOSUB 300
IO 640 NEXT I:GOSUB 320:RETURN
YV 655 REM Zlozenie rak, tupanie
YZ 660 POSITION X,Y+1:?" CHR*(13);CHR*(15)
HP 665 GOSUB 300:GOSUB 300
BO 670 POSITION X,Y+1:?" CHR*(14);CHR*(16)
HR 675 GOSUB 300:GOSUB 300
OI 680 FOR I=1 TO 5
XH 685 POSITION X+1,Y+1:?" CHR*(17)
HG 690 GOSUB 300:GOSUB 300
WF 695 POSITION X+1,Y+1:?" CHR*(16)
GP 700 GOSUB 300:GOSUB 300
NC 705 NEXT I:RETURN
AJ 715 REM Jedna podniesiona, tupanie
NX 720 FOR I=1 TO 5
JV 725 POSITION X,Y+1:?" CHR*(18);CHR*(4)
GV 730 GOSUB 300:GOSUB 300
LB 735 POSITION X,Y+1:?" CHR*(19)
GX 740 GOSUB 300:GOSUB 300
JE 745 NEXT I:GOSUB 320:RETURN
SF 765 REM Machanie reka
OH 770 FOR I=1 TO 5
AK 775 POSITION X,Y+1:?" CHR*(19);CHR*(20)
HF 780 GOSUB 300:GOSUB 300
LJ 785 POSITION X,Y+1:?" CHR*(19);CHR*(4)
HH 790 GOSUB 300:GOSUB 300
NU 795 NEXT I:RETURN
IB 805 REM Wymazanie
IH 810 POSITION X,Y:?" CHR*(32);CHR*(32)
YT 815 POSITION X,Y+1:?" CHR*(32);CHR*(32)
ZI 820 RETURN
ZT 825 REM Trzask
SL 830 FOR I=15 TO 1 STEP -1
US 835 SOUND 0.5,2,I:SOUND 1,60,4,I
OJ 840 SOUND 2,10,8,I:SOUND 3,10,8,I
QF 845 NEXT I:FOR DEL=1 TO 10:NEXT DEL
QM 850 FOR I=0 TO 3:SOUND I,0.0,0:NEXT I
AD 855 RETURN
KK 865 REM Pokazanie Rockforda
TU 870 POSITION X,Y:GOSUB 200:GOSUB 830:G
OSUB 230:GOSUB 830:GOSUB 400:GOSUB 830
:RETURN
LJ 885 REM Usuniecie Rockforda
ZB 890 POSITION X,Y:GOSUB 230:GOSUB 830:G
OSUB 200:GOSUB 830:GOSUB 320:GOSUB 810
:GOSUB 830:RETURN
CP 995 REM DEMONSTRACJA
QI 1000 A=INT(RND(0)*3):IF A=2 THEN A=-1
DT 1010 X=19+10*A
TB 1020 B=INT(RND(0)*3):IF B=2 THEN B=-1
EF 1030 Y=11+7*B
FR 1040 POSITION X-2,Y+2
YQ 1050 FOR K=1 TO 6:?" CHR*(160);:NEXT K
FC 1060 C=INT(RND(0)*6)+1
PB 1070 GOSUB 320:GOSUB 870:GOSUB 320
TQ 1080 ON C GOSUB 440,500,550,660,720,77
0
RB 1090 GOSUB 320:GOSUB 890:GOSUB 320
IQ 1100 POSITION X-2,Y+2:?" "
RB 1110 GOSUB 320:GOTO 1000
```


GRAFIKA W KODZIE MASZYNOWYM

Prędzej czy później przyjdzie czas na Twój pierwszy program w kodzie maszynowym. Po nim nadejdą następne, coraz to bardziej złożone. Ale pewnego dnia zapragniesz, by Twój program nie tylko pisał i liczył, lecz na przykład grał i rysował.

Problemem muzyki w kodzie maszynowym, wykorzystującej zarówno wewnętrzny mini-procesor dźwiękowy spectrum, jak i prezentowany niedawno trzykanałowy generator będziemy się jeszcze nieraz zajmować. Na razie skupmy się na grafice.

Spotykana w najlepszych grach grafika tworzona jest przez doświadczonych programistów przy użyciu sprzętu szesnastobitowego. Raffaele Cecco, autor takich gier, jak Exolon, Cybernoid I i II oraz Stormlord, używa do pracy trzech komputerów. Na Olivetti PC tworzy szkielec gry, na Atari 1040 ST — grafikę i muzykę, zaś wszystko przelewa na Spectrum (i nie tylko) za pośrednictwem jednostki systemu CP/M firmy Philips.

My nie będziemy naśladować Raffaele, nie z powodu braku sprzętu, czy umiejętności, lecz przede wszystkim z powodu braku dostępu do biblioteki procedur, bez których nie powstanie żadna gra.

Na poziomie jednostkowego użytkownika Spectrum sprawą trudną jest nawet tworzenie grafiki wektorowej, tzn. złożonej z punktów i odcinków. Tematowi temu poświęcimy nieco miejsca na łamach „Bajtki”, lecz zaczniemy od sprawy najprostszej — płaskiej grafiki złożonej z punktów, łuków, odcinków i okręgów.

Istnieją dwie możliwe do wyboru drogi. Pierwsza polega na przechowywaniu w pamięci swego rodzaju mapy z zakodowaną treścią obrazu. Treść ta odkodowywana jest przez odpowiedni program, który identyfikuje część rysunku i odtwarza go po kolei.

Druga droga opiera się na kilku (-nastu) podstawowych procedurach zawartych gdzieś w pamięci, do których odwołuje się wykonywany program za każdym razem, gdy ma coś do narysowania.

Oba sposoby są dobre, choć i tak niezwykle pracochłonne w realizacji. Pozostawiam to więc dociekliwym i upartym Czytelnikom.

Aby jednak pomóc im, by nie błakali się po pamięci, postaram się przedstawić, jak wykorzystać w kodzie maszynowym zwykłe, znane z Basic'a komendy: PLOT, DRAW i CIRCLE.

PLOT jest najprostszą komendą, do której odwołują się wszystkie pozostałe, służąca do rysowania komendy. Jej obsługa znajduje się ROM-ie pod adresem #22E5 (dziesiątka 8933). Nie radzę jednak nikomu, by śledził działanie zaczynającego się od #22E5 programu, gdyż uwikła się w ROM bez szans na wypłatanie. Planujemy wprawdzie w „Bajtku” kompletny opis ROM-u Spectrum, lecz jest to sprawa dość dalekiej przyszłości.

Parametrami wejściowymi komendy PLOT są współrzędne punktu, który ma zostać zapalony. Współrzędną poziomą umieszcza się w rejestrze B, zaś pionową w C. Aby oszczędzić sobie pracy, zaś mikroprocesorowi taktów zegara, dobrze jest wykonać prostą operację $BC = B + 256 * C$ i programik:

```
LD B,128
LD C,88
CALL 8933
RET
```

zastępować:
LD BC,22656
CALL 8933
RET

Raz czy dwa, nie ma to znaczenia, ale przy programach powyżej 2 KB różnica, głównie w zajmowanej pamięci, jest wyraźnie widoczna.

DRAW czyli „ciągnij linię” jest komendą pozwalającą na wyrysowanie odcinka (właściwie wektora) o początku tam, gdzie znajduje się ostatni postawiony punkt (zmiennne systemowe COORDS — 23677 i 23678) do punktu, na który wskazują podane przy DRAW liczby — współrzędne wektora.

Zawarta w ROM-ie procedura DRAW wymaga jako parametrów czterech liczb, zawartych w parach rejestrów BC i DE. Para BC przechowuje współrzędne wektora w liczbach bezwzględnych (bez znaku, a więc ze znakiem +), zaś DE znaki odpowiednich współrzędnych, w kodzie U2 (najstarszy bit zapalony — znak -, zgaszony — znak +) wraz z zapalonym najmłodszym bitem. W sumie więc DE przechowuje wartości funkcji SGN (sigmum) współrzędnych x i y z rejestrów B i C. Postępowanie to trochę marnotrawne, ale co zrobić.

DRAW zaczyna się od adresu #24BA (dziesiątka 9402) i narzuca jeszcze jedno ograniczenie. Mianowicie zmienia zawartość pary H'L, która powinna być odtworzona przed powrotem do Basic'a. W związku z tym przykładowa procedura DRAW wyglądać będzie następująco:

```
LD (23677), 255
LD (23688), 175
EXX
PUSH HL
EXX
LD BC,22656
LD D,255
LD E,1
CALL 9402
EXX
POP HL
EXX
RET
```

Pozostała nam procedura rysująca okrąg. Nic prostszego: jej początek znajduje się pod adresem #2D23 (11555). Przesyłanie parametrów wejściowych jest trochę nietypowe, za to mniej kłopotliwe. Polega to na umieszczeniu na stosie kalkulatora trzech liczb, które procedura CIRCLE odczytuje odpowiednio jako współrzędne środka i promień okręgu. Umieszczenie na stosie kalkulatora danej liczby odbywa się za pośrednictwem procedury STACK-A z ROM-u, zaczynającej się od adresu #2D28 (11560). Przesyła ona na stos kalkulatora liczbę z akumulatora.

Przykładowe wywołanie procedury CIRCLE wyglądać będzie więc następująco (para H'L również musi być zachowana!):

```
EXX
PUSH HL
EXX
LD A,128
CALL 11560
LD A,88
CALL 11560
LD A,80
CALL 11560
CALL 11555
EXX
POP HL
EXX
RET
```

Jak pewnie wiecie, w Basic-u podając przy DRAW trzeci parametr można uzyskać bardzo ciekawe efekty. Jest to realizowane również i z poziomu kodu maszynowego, za pośrednictwem osobnej procedury #2394 (9108). Trzy parametry wejściowe przesyła się na stos kalkulatora tuż przed wywołaniem tej procedury, podobnie jak w przypadku DRAW, należy też pamiętać o zachowaniu pary rejestrów H'L.

Na koniec proponuję wklepanie programiku rysującego zabawną twarz. Zaczepnięty został z miesięcznika „Sinclair User” nr 75, zaś jego autorem jest Damien Scattergood, ekspert od ROM-u Spectrum.

To tylko przygrzywka do ogromnego tematu, jakim jest grafika komputerowa. Istnieje tak wiele sposobów i tak wiele pomysłów, że nie sposób jest je tu nawet wymienić. Na razie życzę powodzenia w zmaganiach z mini-grafiką obiecując, że powrócimy wkrótce do tematu grafiki wektorowej.

Marcin Przasnyski

```
10 REM GEBE NA EKRANIE
20 REM by Damien Scattergood/SU
30 BORDER 0: PAPER 0: INK 5: OVER 1: CLS
40 CLEAR 29999
50 FOR F=0 TO 131: READ A: POKE 30000+F,A
60 NEXT F
70 RANDOMIZE USR 30000: PAUSE 0
100 DATA 217, 229, 217, 62, 100, 205
110 DATA 40, 45, 62, 100, 205, 40, 45
120 DATA 62, 69, 205, 40, 45, 205, 45, 35
130 DATA 62, 68, 205, 40, 45, 62, 116
140 DATA 205, 40, 45, 62, 16, 205
150 DATA 40, 45, 205, 45, 35, 62, 132
160 DATA 205, 40, 45, 62, 116, 205, 40, 45
170 DATA 62, 16, 205, 40, 45, 205, 45, 35
180 DATA 62, 100, 50, 125, 92, 62, 80
190 DATA 50, 126, 92, 1, 0, 32, 17, 1, 1, 205
200 DATA 186, 36, 62, 96, 205, 40, 45
210 DATA 62, 32, 205, 40, 45, 62, 2
220 DATA 205, 40, 45, 62, 64, 50, 125
230 DATA 92, 62, 64, 50, 126, 92, 205
240 DATA 148, 35, 1, 68, 116, 205, 229
250 DATA 34, 1, 69, 117, 205, 229
260 DATA 34, 1, 132, 116, 205, 229
270 DATA 34, 1, 133, 117, 205, 229
280 DATA 34, 217, 225, 217, 201
```

JUMPER

System operacyjny TOS dla ZX Spectrum charakteryzuje się m.in. hierarchiczną strukturą katalogu. Jego wadą jest konieczność pamiętania, w jakim podkatalogu umieściliśmy dane.

Wadę tę likwiduje poniższy program — „JUMPER”. Umożliwia on swobodną „wędrówkę” po podkatalogach, jak również wgranie w każdej chwili dowolnego programu lub danych (nagrywanych przez SAVE * „nazwa” CODE). Wymagane jest, aby program w BASIC-u posiadał rozszerzenie .BAS, a zbiór danych — .COD. W przypadku napotkania innego rozszerzenia wyświetlany jest komunikat o błędzie (linia 5025).

W celu wgrania dowolnego zbioru lub przejścia do podkatalogu, należy klawiszami 6 i 7 przesunąć migający prostokąt na nazwę zbioru (lub katalogu) i nacisnąć „0”. Powrót do poprzedniego katalogu jest możliwy po wciśnięciu „1”.

Program należy zapisać na dysku przez: **CLEAR: SAVE * „START” LINE 10: ATTR * „START”** i będzie on automatycznie wczytywany po naciśnięciu **RESET**.

Większość liczb w programie zastąpiono przez wyrażenie z „VAL” lub „PI”, aby maksymalnie zmniejszyć ilość zajmowanej przez niego pamięci — zajmuje on w ten sposób jedną jednostkę alokacji. Nic oczywiście nie stoi na przeszkodzie, aby rozbudować go według własnych potrzeb.

Jacek Zapala

```
10 BORDER NOT PI: PAPER NOT PI
: INK VAL "7": CLS: LET a$=""
20 CAT *: LET n=VAL "6": PRINT
#NOT PI; "6-DOL 7-GORA
0-OK 1-Powrot do ostatniego
katalogu"
30 PRINT AT n, NOT PI; OVER SGN
PI; FLASH SGN PI; a$: BEEP VAL "
.1", VAL "40"
40 IF INKEY$="7" AND n>VAL "6"
THEN PRINT AT n, NOT PI; FLASH
OT PI; OVER SGN PI; a$: LET n=n-S
GN PI: GO TO VAL "30"
45 IF SCREEN$(2,6)>"0" AND IN
KEY$="1" THEN DRAW *: RUN
50 IF INKEY$="6" THEN IF n<=VA
L "20" THEN LET n=n+SGN PI: GO S
UB 1000: PRINT AT n, SGN PI, 0; FL
ASH NOT PI; OVER SGN PI; a$: GO T
O VAL "30"
60 IF INKEY$="0" THEN GO TO VA
L "5000"
70 GO TO VAL "40"
1000 LET n$="": FOR x=NOT PI TO
VAL "8": LET s$=SCREEN$(n,x): I
F s$<>" " THEN LET n$=n$+s$
1015 NEXT x: LET n$=n$+" "
1020 FOR x=VAL "9" TO VAL "11":
LET s$=SCREEN$(n,x): IF s$<>" "
THEN LET n$=n$+s$
1030 NEXT x
1040 IF n$="" THEN LET n$="": L
ET n=n-SGN PI
1045 PRINT AT n, NOT PI; OVER SGN
PI; FLASH NOT PI; a$
1050 RETURN
5000 GO SUB VAL "1000"
5010 LET r$=n$(LEN n$-VAL "3" TO
LEN n$): IF r$=".BAS" THEN LOAD
*n$
5020 IF r$=".COD" THEN INPUT "Po
daj RAMTOP (ENTER-RAMTOP program
owy) "; LINE q$: IF q$<>" " THEN
LET q=VAL q$-SGN PI: LET l=USR "
a": POKE l, LEN n$: FOR z=SGN PI
TO LEN n$: LET l=l+SGN PI: POKE
l, CODE n$(z TO z): NEXT z: CLEAR
a: LET n$="": LET n=PEEK USR "a
": FOR z=SGN PI TO n: LET n$=n$+
CHR$(PEEK USR "a"+z): NEXT z:
GO TO VAL "5030"
5025 LET e$=n$(LEN n$-VAL "3" TO
LEN n$): IF e$<>".DIR" AND e$<>
".COD" THEN CLS: PRINT "Wczytaj
sobie ten program jak chcesz,
ja nie wiem, co z nim zrobic!
": STOP
5027 IF r$=".DIR" THEN LET d$=n$
(SGN PI TO LEN n$-VAL "4"): GO S
UB *d$: RUN
5030 CLS: LOAD *n$CODE
```


LICZNIK

Myślę, że wiele osób widziało program, który ładuje się z tzw. licznikiem.

Mam na myśli gry EXOLON, KANE itp. Podczas ładowania wartość licznika zmniejsza się aż do zera, gdy program jest już cały w pamięci. Jest to o wiele bardziej eleganckie od zwykłych „pasków”, gdyż licznik jest nie tylko efektywny graficznie, ale także dobry „na nerwy”, ponieważ w każdej chwili wiemy, ile czekania nam jeszcze zostało.

Przedstawiony program LICZNIK umożliwia ładowania programów w ten właśnie sposób.

Główna część kodu maszynowego pochodzi z loadera do programu BOULDER DASH IV, reszta to przystosowanie do ładowania według danych z nagłówka, tak, by użytkownik mógł korzystać z programu bez znajomości assemblera. LICZNIK znajduje się przy końcu pamięci, od adresu 64500 do 69252. Uruchamia się go komendą **RANDOMIZE USR 65100**. Program wczytuje wtedy nagłówek (pierwszy napotkany) pod adres 65082 i ładuje tylko bloki typu PROGRAM i BYTES, nie akceptuje tablic.

LICZNIK, jak już wyżej wspominałem, umiejscawia dany program w pamięci na podstawie danych w nagłówku. Jeżeli podczas ładowania wystąpił błąd, LICZNIK wraca do BASIC-a (konkretnie do miejsca wywołania, może więc również wrócić i kontynuować wykonywanie kodu maszynowego). Nie przerywa działania programu, tylko umieszcza w komórce 65099 wartość 0 (normalnie 255).

Jeżeli nie odpowiadają nam atrybuty licznika (standardowo czarne litery na białym tle), to

dzięki **POKE 64910**, (wartość negatywu atrybutu) możemy je zmienić.

Od adresu 64562 począwszy znajdują się kształty cyfr 6 w negatywie) od zera do dziewięćki kolejno. Kształty można przededefiniować według własnego upodobania, podobnie jak tworzymy znaki graficzne (UDG).

Jeżeli chcemy korzystać z programu LICZNIK z poziomu assemblera, to podajemy parametry takie same, jak dla systemowej procedury ładowania, tj.

```
LD IX, adres początku
LD DE, długość
LD A, FFF
SCF
CALL 64500
```

LICZNIK nie jest programem krótkim, ale myślę, że warto zdobyć się na odrobinę cierpliwości i wpisać go.

Jacek Marczewski

```
10 DATA 64500,24
20 DATA "F5D5ED532CFCCD0FFED1F
137CD3AFC215827D9C33F05AE7723130
B78B120F6C340"
30 DATA "3AFE4E4FFF000115FCFE5
0310003434F444520202020202003020
0640064FEC6D4"
40 DATA "BABABABAC6FEFEFECEEEEE
EEEE2FEFEC6BAF6EEDE02FEFE82F6E6F
ABAC6FEFEF638"
50 DATA "E6D6B682F6FEFES2B6E6F
ABAC6FEFEE6DE86B8BAC6FEFE82FAF6F
6EEEF6FEFEC6A8"
60 DATA "BAC6BABAC6FEFEC6BABAC
2FAC6FEFEC6BABABABAC6FE140815F30
90E00D93E0A0C"
70 DATA "D3FEDBFE1FE620F6014FB
F00CD20FD30FA21150410FE2B7CB520F
9CD1CFD30EBA8"
80 DATA "069CCD1CFD30E43EC6B83
0E02420F106C9CD20FD30D578FED430F
4CD20FDD0D95C"
90 DATA "3A1AFC4F321FFCD926000
680181F082007300FDD7500180FCB11A
DC0791F4F1302"
100 DATA "1807DD7E00ADC0DD23180
806B22E01CD1CFDD03ECBB8CB1506B0D
203FD7CAD6760"
110 DATA "7AB320CA7CFE01C9CD20F
DD0C33BFDD04C83E7FDBFE1FD0A9E6202
8F3792F4FE608"
120 DATA "07F608D3FE37C9D979B72
840211FFC357E285BFE053042211BFC3
42C2C352A1B6C"
```

```
130 DATA "FC7EC632BF16FC2A1DFC1
A771C241A771C241A771C241A771C241
A771C241A77CC"
140 DATA "1C241A771C241A77D9C32
3FD061710FE3E00ED44A7D9C323FD3E4
732FA5A32FB93"
150 DATA "5A32FC5A32FD5A2323060
DC382FD064FED5B16FC2A18FC10F2B9F
D581506051040"
160 DATA "FE7EC3ECFD7AE607CAD1F
D15F2D3FD502D3A00603A00603A0060C
3ECFD7E7E7DCE"
170 DATA "E607CAE4FD2DF2E6FD682
5ED44C3ECFD7E7E7CE607280125ED531
6FC2218FC2160"
180 DATA "15FC221BFC21FE502210F
C3E40321FFCED44ED44ED44A7D9C323F
D2A2CFC060512"
190 DATA "CB3CCB1D10FA1119FC3EF
F01E8033CB7ED4230FA09878787124D4
4292909291B74"
200 DATA "7BFE1520E4C9202843294
A414D534F465427383920202000F3F5C
5D6E5DDE5FD41"
210 DATA "E508D9F5C5D5E5D908DD2
13AFE1111003E0837CD86053A3AFEFE0
02831FE0320FA"
220 DATA "E6DD2A47FEED5B45FE3EF
F37CDF4FB3806AF324BFE18053EFF324
8FED908E1D1BF"
230 DATA "C1F1D908FDE1DDE1E1D1C
1F1FBC9ED5B535C2A695C2BCDE5192A8
95C2BED4B45AA"
240 DATA "FECDE51623ED5B49FE192
```

```
24B5C3A48FEE6C0200A2A47FEAF22425
C32445CDD2AD1"
250 DATA "535CED5B45FE3EFF37CDF
4FB38A918A104"
260 REM
270 REM
280 REM
290 REM
300 CLEAR 64499
310 PRINT "URUCHOMIENIE - RAND.
USR 65100"
320 GO SUB 9990
330 SAVE "LICZNIK"CODE 64500,75
2
340 STOP
9990 REM
9991 REM
9992 REM
9993 READ A,S: FOR F=10 TO S+10
STEP 10
9994 READ L8: LET L=LEN L8: LET
S=0: LET K=2
9995 LET A8=L8(K-1): LET B8=L8(K
)
9996 LET C=(CODE A8-48-(7*(A8)*0
"))+16+CODE B8-48-(7*(B8)*0"))
9997 IF K<L THEN POKE A,C: LET S
=S+C: LET K=K+2: LET A=A+1: GO T
O 9995
9998 IF S-255>INT (S/255)(<)C THE
N PRINT "BLAD W LINII ";F+10: ST
OP
9999 NEXT F: RETURN
```

PERKUSJA

Amstrad, po przejęciu firmy Sinclair, nie tylko utrzymał produkcję ZX Spectrum 128, ale wprowadził na rynek dwa nowe modele ZX Spectrum oznaczone +2 i +3, zgodne ze standardem 128 KB, a nawet wyposażone w tryb emulacji Spectrum 48.

Te trzy komputery różnią się od swoich młodszych braci nie tylko pamięcią, ale i układem do generacji dźwięku. „Gumiak” nigdy nie miał zbyt wielkich możliwości dźwiękowych, ale już w ZX Spectrum 128 wprowadzono trójkanałowy generator AY 3-8910, który wcześniej zastosowany był w ośmiobitowych maszynach Amstrada.

Zwiększone możliwości muzyczne Spectrum wykorzystywane są w wielu grach, powstałych w 1987 roku i później. Istnieje także kilka programów muzycznych (ja miałem do czynienia na razie z trzema — The Music Box 128, Music Studio 128 i Mozart).

Po takim wstępie chciałbym zaprezentować procedurę, która może przydać się tym wszystkim, którzy w

swoich programach chcieliby wykorzystać dźwięki z generatora a zwykła instrukcja PLAY już im nie wystarcza. Po uruchomieniu programu otrzymujemy do dyspozycji dziesięć dźwięków o perkusyjnym brzmieniu. Większa jego część jest napisana w kodzie maszynowym lecz do prawidłowej pracy potrzebny jest choć fragment programu w Basicu — co najmniej od linii 200. Pomysł został zaczerpnięty z programami MICRO-RYTHM i jest to jego najbardziej uproszczona wersja.

Opis programu:

- 5—80 przepisanie do pamięci kodu maszynowego
- 100—130 przepisanie do pamięci kodu dźwięków
- 200—210 odtwarzanie dźwięków
- 220 wyłączenie generatora
- 250 kody dźwiękowe
- 300 graficzne odwzorowanie aktualnego dźwięku

Najważniejszą częścią programu jest kod maszynowy, umieszczony w dwóch niezbyt odległych od siebie miejscach pamięci. Ze względu na takie rozmieszczenie konieczne jest użycie instrukcji skoku. Z tego powodu kod nie jest relokowalny.

Program „Perkusja” może być uruchomiony także na Spectrum 48 z dołączonym generatorem dźwięków i dlatego dźwięk wyłączany jest instrukcjami OUT a nie PLAY.

W linii 50 umieszczamy cyfry od 0 do 9 — dane dla perkusji. Trzeba jednak pamiętać, by na jej końcu znajdowała się liczba 255 i cyfr nie było więcej niż 280. Potem można już uruchomić procedurę instrukcją RUN. Kom-

puter zamilknie na kilka chwil, by wczytać kod maszynowy.

Przy prawidłowym przepisaniu procedury, w lewym górnym rogu ekranu powinno pojawić się słowo ZAPIS. Oznacza to, że komputer przystąpił do przepisywania kodów dźwięków. Czas wykonywania tej operacji zależy od ilości cyfr w linii 250. Następnie, także w lewym górnym rogu pojawi się napis ODTW, oznaczający rozpoczęcie odtwarzania zapisanych dźwięków. Ponadto na środku ekranu pojawią się cyfry od 0 do 9 i czarny pas, który będzie wskazywał numer dźwię-

ku odtwarzanego w danym momencie.

Program jest tylko propozycją wykorzystania zwiększonych możliwości komputera ZX Spectrum 128 i od niedawna, również 48. Jego rozbudowę pozostawiam Czytelnikom. Chciałbym jednocześnie zachęcić Was do eksperymentów z tym komputerem. Często ich efekty są zadziwiające. Posłuchajcie tylko muzyki z takich gier jak TURBO BOAT czy PROFESIONAL BMX SIMULATOR i spróbujcie ułożyć podobne. Życzę powodzenia.

Michał Sobieszuk

```
5 LET S=0
10 FOR I=49439 TO 49637: READ a: LET S=S+A: POKE I,a: NEXT I: FOR I=49692 TO 4
9790: READ a: LET S=S+A: POKE I,a: NEXT I
20 IF S<18331 THEN PRINT "BLAD !!!": STOP
30 DATA 0.0,56.7,15.0,56.7,12.0,56.7,9.0,56.7,5.0,56.7,2.2,211.26,15.3,2.0,14.
3,49.6,13.3,96.12,12.3,143.17,11.1,12.1,12.1,12.1,8.1,12.1,5.1,12.1,1.1,12.1,0
35
40 DATA 1.244,26.15,1.254,0.14,2.8,6.13,2.18,12.11,2.28,17.10,3,232,15,15,3,13
2,0,13,3,32,15,11,2,188,0,9,2,88,15,7,1,244,10,15,1,246,30,13,1,244,10,11,1,246,
30
45
50 DATA 9,1,244,10,7,0,100,0,15,0,110,31,14,0,120,15,13,0,110,31,11,0,100,0,9,
2,111,14,15,2,158,20,14,2,205,26,13,2,252,0,12,3,43,6,11,2,211,20,15,3,2,25,12,3
55
60 DATA 49,31,9,3,96,5,5,3,143,11,2,0,200,31,12,0,210,15,11,0,220,10,9,0,210,5
7,0,200,58,117,194,254,19,200,58,118,194,203,39,203,39,71,135,135,128,95
65
70 DATA 22,0,33,32,193,25,58,117,194,95,22,0,25,14,253,6,255,52,1,237,121,6,19
1,126,237,121,35,6,255,62,0,237,121,6,191,126,237,121,35,6,255,62,6,237,121,6
75
80 DATA 191,126,237,121,35,6,255,62,8,237,121,6,191,126,237,121,58,117,194,198
4,50,117,194,201,0,0,194,198,4,50,126,194,201,12
85
100 STOP: PRINT AT 0,0:"ZAPIS": FOR I=49152 TO 49438: READ s
110 IF S=255 THEN GO TO 200
120 IF S>0 AND S<-9 THEN POKE I,S
130 NEXT I
200 PRINT AT 0,0:"ODTW.": FOR I=0 TO 9: PRINT AT 9,(I*3):I: NEXT I
210 READ aS: FOR I=49152 TO 49438: LET z=PEEK I: POKE 49781,0: POKE 49782,z: PR
INT AT 10,0:AS(1 TO 3*(9-z)+1):"
OMIZE USR 49692: PAUSE 1: NEXT a: NEXT I
220 OUT 65533,7: OUT 49149,255
230 GO TO 100
250 DATA 255
300 DATA "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```


MUZYKA!!!

Generator dźwięku zwany w skrócie AY zadziwił wszystkich. Trzy kanały i to w dodatku stereo zadowolą najbardziej wybredne ucho.

Tak więc muzyka Spectrum porównywalna jest już z muzyką Amigi, nie mówiąc o tym, że bije na głowę Atari, Commodore i inne komputery, które dotąd zdawały się dominować.

Ale jest jedno „ale”. Każda gra na Commodore wykorzystuje jak nie trzy, to dwa kanały swego procesora dźwięków. Podobnie jest z Atari. Natomiast w przypadku Spectrum tylko część gier obsługuje generator. Wprawdzie od momentu wprowadzenia na rynek modelu 128 większość gier posiada rozszerzone możliwości muzyczne, lecz część z nich ujawnia się tylko na Spectrum 128.

Nie jest jednak źle. Powstają wciąż gry i programy muzyczne działające na Spectrum 48 z dołączonym generatorem. Poniżej przedstawiona jest lista najpopularniejszych z nich.

Ostatnio rozpowszechniona jest moda na „wyciąganie” muzyki gier i umieszczanie w jednym programie kilkunastu takich ekstraktów. Moda ta pojawiła się na giełdzie „Bajtka” i stąd też wyszły pierwsze takie programy.

Celuje w tym miły człowiek-hacker podpisujący się Chris. Aby urozmaicić i tak wspaniałe efekty dźwiękowo-muzyczne, w programach swoich umieszcza bardzo długie, przewijające się napisy, kogo to on nie lubi i który z hackerów giełdowych jest beznadziejny. Komentarze te pisane są niby po angielsku, przy użyciu czasu Simple Present i pięciu obraźliwych słów, w różnych mutacjach.

Postępowanie takie nie jest godne naśladowania. Nieprzyjemne są soczyste komentarze i pomijając wartość programów pod względem muzycznym, są one haniebne. Jeśli Chris w istocie chce błyszczeć, może spróbować sposobów mniej rażących otoczenie.

Kto chce słuchać dobrej muzyki, musi więc ją sobie albo sam napisać (polecam „Wham — The Music Box 128” — opis wkrótce w „Bajtku”) lub korzystać z gier, najlepiej z poniższej listy:

Ace II
After Burner
Agent X II
Amaroute
ATF
Bear a Grudge
Bionic Commando
Boulder Dash Construction Kit
Bubble Bobble

Chronos
Cyberoid
Cyberoid II
Daley Thompson's Olympic Challenge
Death Ride
Death Stalker
Defcom
Eagle's Nest
Enduro Racer
Exolon
Foxx Fights Back
G.I. Hero
Gryzor
Hot Shot
Hydrofool
Ikari Warriors
International Karate + Marauder
Mask II
Motos
Nether World
Neverending Story
Outrun
Pacland
Pacmania
Platoon
Profesional BMX Simulator
Raw Recruit
Rex
Robocop
Sigma 7
Skate Crazy
Soccer
Spitting Image
Supertest
Tai-Pan
Target Renegade
Thunder Cats
Tiger Road
Turbo Boat
Venom
Vindicator
Vixen
Where Time Stood Still
Wizball
Xenon

Marcin Przasnyski



WERYFIKATOR

PO RAZ TRZECI

Oto trzecia wersja programu „Weryfikator”, służącego do kontroli wpisywanych listingów. Uważam, iż jest to wersja jak na razie najdoskonalsza. Nie zajmuje obszaru przeznaczanego na grafikę użytkownika, współpracuje z Interfacem 1 i jest krótsza od wersji II. Dodatkowo suma kontrolna wyświetlana jest przed numerem linii, zaś zmieniając kod w linii 130 można uzyskać wydruk sumy w nawiasach. Jest to szczególnie ważne w przypadku używania drukarek pozabawionych możliwości drukowania w negatywie.

Podobnie, jak pozostałe wersje, tak i ta oblicza sumę kodów instrukcji modulo (256) pomniejszoną o 9.

Andrzej M. Grossman

```

[90] 10 REM WERYFIKATOR III
[84] 20 REM A.M.GROSSMAN 1988
[88] 30 REM RANDOMIZE USR 64809-Wydruk, POKE 64810,2 - na ekran, POKE 64810,3 - na drukarkę
[85] 40>REM RANDOMIZE USR 64787-Suma kontrolna dla linii z kursorem
[87] 50 REM RANDOMIZE USR 64806-Wyłączenie sumy kontrolnej
[50] 60 CLEAR 64763: LET S=0
[92] 70 FOR I=64764 TO 64970
[85] 80 READ A: POKE I,A: LET S=S+A
[90] 90 NEXT I
[84] 100 IF S<>25902 THEN PRINT "Błąd"
[55] 110 STOP
[58] 120 REM nawiasy
[60] 130 POKE 64819,91: POKE 64820,0: POKE 64832,93: POKE 64835,0
[90] 140 DATA 245,229,213,197,253,203,071,254,042,073,092,205,169,253,204,089,253,193,209,225
[85] 150 DATA 241,255,201,038,254,046,000,059,054,252,035,016,251,054,252,052,254,237,071,237
[80] 160 DATA 094,201,237,085,201,062,002,205,001,022,042,083,092,229,062,020,215,062,001,215
[80] 170 DATA 253,203,071,190,205,089,253,052,020,215,175,215,062,032,215,225,205,085,024,062
[80] 180 DATA 013,215,175,237,091,075,092,237,082,025,056,217,201,205,191,253,237,066,175,087
[64] 190 DATA 134,237,160,234,096,203,214,089,030,028,245,015,015,015,015,205,115,253,241,230
[41] 200 DATA 015,198,048,254,058,056,002,198,007,253,203,071,126,040,036,038,088,107,095,253
[45] 210 DATA 126,083,047,230,127,119,038,064,022,000,235,041,041,041,237,075,054,092,009,235
[85] 220 DATA 006,008,026,119,036,028,016,250,044,093,201,215,201,229,042,083,092,084,093,193
[81] 230 DATA 126,184,032,004,035,126,043,185,208,197,205,191,253,024,240,035,035,078,035,070
[90] 240 DATA 035,009,003,003,003,003,003,201
    
```


JEZYK MASZYNOWY CZ. 6

W układzie dziesiętnym operacje dzielenia i mnożenia przez dziesięć polegają w wielkim uproszczeniu na przemieszczaniu kropki dziesiętnej odpowiednio w lewo i prawo o jedną pozycję. Tak jak dzielenie i mnożenie przez dziesięć są bazowymi działaniami w dziesiętnym układzie liczenia, tak w układzie dwójkowym odpowiednikami są dzielenie i mnożenie przez dwa.

Wykonujemy je równie prosto przez przemieszczenie bitów w bajcie (lub bajtach jeżeli liczba zapisana jest w więcej niż jednym bajcie) w prawo lub lewo.

Zapiszmy w układzie binarnym liczby 50, 100, 200 dane dziesiętnie. Otrzymamy:

50 — %00110010
100 — %01100100
200 — %11001000

Jak widać przy podwajaniu liczb bity w bajcie przemieszczają się w lewo o jedną pozycję, przy czym na miejsce najmłodszego bitu w bajcie wpisywane jest zero. Instrukcją służącą do takiego mnożenia przez dwa jest ASL (Arithmetic Shift Left) lub inaczej arytmetyczne przesunięcie w lewo. Gdybyśmy wpadli na pomysł pomnożenia przez dwa otrzymanej liczby 200 to wyniknie problem. Jak wiemy liczby 400 nie da się zapisać w jednym bajcie. Nastąpi więc przepełnienie sygnalizowane ustawieniem znacznika przeniesienia na 1 (C=1). Tak więc można powiedzieć, że instrukcja ASL powoduje wpisanie wartości 0 do najmłodszego bitu w bajcie i wpisanie najstarszego bitu z bajtu (odpadającego po wykonaniu mnożenia) do znacznika przeniesienia C.

Dobrze by było gdybyśmy dysponowali instrukcją działającą podobnie do ASL, lecz wpisującą do najmłodszego bitu nie zero, lecz wartość znacznika C. Taka instrukcja umożliwiłaby nam prowadzenie obliczeń na liczbach zapisanych w kilku bajtach. Instrukcją taką w procesorach 65xx jest ROL (Rotate left) — rotuj w lewo. Działa ona tak jak to przedstawiono na rysunku.

Przykładowe użycie tych instrukcji (pamiętajmy, że operacje arytmetyczne wykonywane są wyłącznie w akumulatorze):

LDA \$2750
ASL lub ROL
STA \$2750
BRK

Program ten powoduje przemnożenie przez dwa zawartości komórki pamięci o adresie \$2750 i zapisanie wyniku do tej samej komórki.

Jeżeli naszym celem jest pomnożyć liczbę zapisaną w trzech bajtach (\$2750-\$2752) przez np. 4 to musimy napisać następujący program (LISTING1). Mnożenie przez 4 to dwukrotne mnożenie

```
02729 9D 50 27 STA $2750,X
0272C 88      DEY
0272D D0 E3   BNE $2712
0272F 00     BRK
```

nie przez dwa, dlatego też do rejestru .Y służącego dalej jako licznik wpisana zostaje wartość dwa. Aby nie używać dwóch instrukcji ASL i ROL, ustawiamy znacznik przeniesienia na zero, a w związku z tym pierwsza instrukcja ROL zadziała tak jak ASL, czyli do najmłodszego bitu w bajcie wpisane zostanie 0. Następnie trzy razy powtarzana jest sekwencja załadowania zawartości komórki do akumulatora, poczynając od najmłodszego bajtu, dokonania mnożenia na tym bajcie, zwiększenia zawartości rejestru indeksującego .X o jeden. Następnie licznik .Y jest zmniejszany o jeden i sprawdzony zostaje warunek czy mnożenie zostało wykonane dwa razy. Jak widać, tak jak większość pierwotnych wersji naszych programów tak i ten jest mało uniwersalny. Jedyna rzecz jaką możemy sterować to ilość wykonywanych mnożeń przez dwa — zawartość rejestru .Y. Aby uogólnić ten program oraz aby mieć pretekst do wprowadzenia czterech nowych instrukcji powstał program znajdujący się na LISTINGu 2.

READY.

MONITOR

PC SR AC XR YR SP

; B000 00 00 00 00 FB

```
02710 A0 02   LDY ##02
02712 1B     CLC
02713 A2 00   LDX ##00
02715 B0 1C 27 JSR $271C
02718 88     DEY
02719 D0 F7   BNE $2712
0271B 00    BRK
0271C B0 50 27 LDA $2750,X
0271F 2A    ROL
02720 9D 50 27 STA $2750,X
02723 E8     INX
02724 0B    PHP
02725 EA    NOP
02726 E0 03  CPX ##03
02728 0B    PHP
02729 6B    PLA
0272A B0 60 27 STA $2760
0272D 6B    PLA
0272E 0D 60 27 ORA $2760
02731 4B    PHA
02732 2B    PLP
02733 D0 E7   BNE $271C
02735 60    RTS
02736 00    BRK
```

Jaka różnica — spytaście. Mimo że jest on dłuższy od pierwszego, to nie ulegnie on już żadnym wydłużeniom — jest uniwersalny. Ilość mnożeń regulowana jest zawartością rejestru .Y, a ilość bajtów, które ulegną wymnożeniu wynika z instrukcji w linii \$2726, w której następuje porównanie liczby określającej ilość bajtów z ilością dokonanych już wymnożeń.

Program wykorzystuje podprogram wykonujący pojedyncze mnożenie trzech bajtów przez dwa. W podprogramie tym pojawiają się instrukcje operujące na stosie. Instrukcje PHP i PHA powodują zapisanie zawartości odpowiednio rejestru słowa stanu i akumulatora na stosie. Instrukcje PLP i PLA powodują ściągnięcie ze stosu i załadowanie do rejestru słowa stanu lub akumulatora odpowiedniego bajtu. Stos działa według zasady LIFO (Last In First Out) czyli ostatnia wartość zapisana na stosie będzie pierwszą odczytaną z niego. Dlaczego użyto tych instrukcji w tym podprogramie? Otóż jak już wspomnieliśmy w celu ich prezentacji i pokazania sposobu działania w programie jak również w celu ugruntowania wcześniej nabytej wiedzy dotyczącej operacji logicznych. Oczywiście program ten można napisać prościej. Pozostawiam to jednak Wam. Jak pamiętamy przy mnożeniu za pomocą instrukcji ROL ważną rolę odgrywa znacznik przeniesienia C natomiast przy porównywaniu CPX jego wartość może się zmienić co może mieć wpływ na dalsze obliczenia. Dlatego przed porównaniem zapamiętałem cały rejestr słowa stanu na stosie, przepisałem go do akumulatora (gdyż bezpośrednio rejestru

słowa stanu nie można załadować do akumulatora) a następnie zapamiętałem zawartość akumulatora w komórce pamięci o adresie \$2760. Następnie dokonałem porównania i nową wartość zapamiętałem na stosie, przeniosłem ze stosu do akumulatora i dokonałem „połączenia” obu rejestrów słowa stanu dbając o to by zmiane nie uległy znacznik przeniesienia C i znacznik zera Z. Reszta programu jest taka jak w pierwszym programie zawartym na LISTINGU1.

Po wpisaniu każdego z tych programów należy zadbać o to by odpowiednie wartości znalazły się w komórkach o adresach \$2750-\$2752. Pamiętajmy, że wartości wpisujemy poczynając od najmłodszego bajtu. I tak jeżeli wpisujemy #\$24 do komórki \$2750, #\$66 do \$2751 i #\$13 do \$2752 to po wykonaniu programów komórki te będą zawierać wartości #\$4D, #\$98, #\$90. Sprawdźcie ręcznie czy takiego wyniku należy oczekiwać.

Mamy więc programy umożliwiające mnożenie przez wielokrotność dwóch. Mnożenie przez dziesięć nie powinno być o wiele trudniejsze. Mając daną liczbę L mnożymy ją przez dwa i zapamiętujemy w jakiejś komórce o adresie XXXX. Następnie liczbę L mnożymy dwa razy przez dwa (razem 8L) i dodajemy zawartość komórki XXXX (8L+2L=10L). Można tę sekwencję uprościć zakładając, że liczba L była zapisana w komórce o adresie XXXX, wtedy L przepisujemy do akumulatora .A, mnożymy zawartość przez 4, dodajemy zawartość komórki o adresie XXXX i całość mnożymy przez dwa. Razem otrzymujemy również 10L. Proponuję napisać własne programy mnożące przez dziesięć i porównać je z rozwiązaniami LISTING 3 i LISTING4. Ambient-

LDA \$XXXX

RSE

LDA \$XXXX

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

RSE

READY.

MONITOR

PC SR AC XR YR SP

; B000 00 00 00 00 FE

```
02710 A0 02   LDY ##02
02712 1B     CLC
02713 A2 00   LDX ##00
02715 B0 50 27 LDA $2750,X
02718 2A    ROL
02719 9D 50 27 STA $2750,X
0271C E8     INX
0271D B0 50 27 LDA $2750,X
02720 2A    ROL
02721 9D 50 27 STA $2750,X
02724 E8     INX
02725 B0 50 27 LDA $2750,X
02728 2A    ROL
```

Dominik Falkowski



AMIGA TIPS & TRICKS

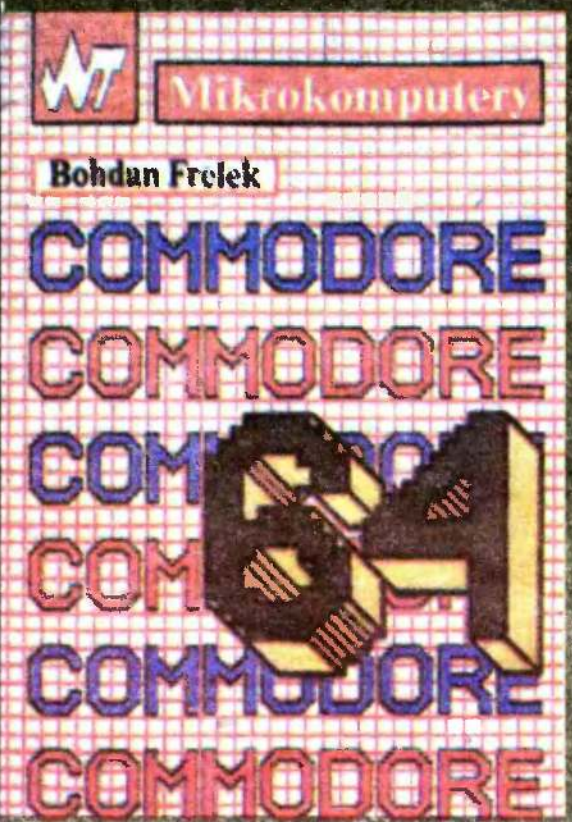
Wśród niezliczonej liczby ciągle pojawiających się książek do komputerów Commodore Amiga wybrałem tę właśnie, gdyż czy to dla C-64, czy C-128 książki opatrzone tym tytułem stawały się zawsze prędko bestsellerami. I tym razem wydawcą jest wszystkim dobrze znana DATA-BECKER. Książka to opasłe tomiszczko zawierające ponad 500 stron podzielone na 17 rozdziałów traktujących o Amiga BASIC, DOS, ikonach, języku maszynowym, operacjach wejścia i wyjścia o budowie samego sprzętu (hardware) i inne. Każdy z rozdziałów porusza obszernie wiele różnych ciekawych tematów np. w rozdziale o Amiga BASIC możemy się dowiedzieć o tworzeniu okienek bez ramek, grafice trójwymiarowej w oparciu o HAM (Hold and Modify), zmianie kolorów, rysowaniu na ekranie monitora w oparciu o inne układy współrzędnych, superszybkiej grafice wektorowej, o drukowaniu itp. Każdy z tematów jest uzupełniony licznymi przykładami programów. Podane są również oddzielne procedury realizujące określone operacje. Moim zdaniem w książce jest za mało rysunków i zdjęć ekranów, które często ułatwiają zrozumienie wykładanego materiału. W tego typu książkach trudno jest nawet wspomnieć o wszystkich zawartych w niej tematach, gdyż każdy z nich (a jest ich mnóstwo) to parę stron w tej książce. Myślę jednak, że Tips & Tricks można polecić wszystkim, którzy podstawy programowania w Amiga BASIC mają już za sobą a poszukują sposobów i sposobików na rozszerzenie możliwości tego komputera — znajdują na pewno odpowiedzi na szereg nurtujących ich problemów. Cena publikacji 59DM.

Amiga Tips & Tricks
DATA BECKER GmbH
Merowingerstr 30
4000 Düsseldorf

COMMODORE 64

Z serii Mikrokomputery Wydawnictwa Naukowo Techniczne ukazała się w ubiegłym roku książka poświęcona w całości popularnemu na całym świecie komputerowi Commodore 64 (C-64). Książek dla Commodore, również w wersji polskojęzycznej, jest na naszym rynku dużo i stale ich przybywa. Jednak potrzebne informacje są często rozproszone po wielu publikacjach i dotarcie do nich jest bardzo utrudnione. Bohdan Frelek (autor książki Commodore 64) w swojej książce zebrał podstawowe informacje dotyczące komputera C-64. Omówił jej budowę, wykorzystanie oraz współpracę z urządzeniami zewnętrznymi. Wspomniał nawet o możliwości podłączenia C-64 do IBM. Opisał programowanie C-64 w języku BASIC (w wersji dla początkujących i bardziej zaawansowanych) i Asemblerze. Przedstawił możliwości uzyskania efektów dźwiękowych i graficznych. Podał również informacje o mikrokomputerach VIC-20, C-16, Plus/4, C-128. Książka ta stanowi kompendium wiedzy o C-64 i byłaby niezwykle cenną pozycją na półkach użytkowników komputerów Commodore gdyby nie kontynuacja zleń tradycji terminologii komputerowej stosowanej w polskich książkach. Spierać się można co do tego czy w informatyce używać wyrazów spolszczonych (np. dżojstik), polskich (manipulator drążkowy stoło-przyczepny) czy ich oryginalnych brzmień (joystick), ale wszyscy zgodni jesteśmy co do tego, że wyrażen tych nie należy mieszać. Więc albo joystick albo manipulator drążkowy. Jeżeli w książce pojawia się słowo cartridge (na określenie modułu) to dlaczego dalej autor używa terminu ładowacz na określenie loadera? Drugą sprawą są niektóre stosowane terminy: Interpreter, syntetyzator, grafika nisko/wysokorozdzielcza i parę innych. Myślę, że już dość dawno wypisano się odnośnie tego, że mówimy interpreter języka programowania czy tryb wysokiej rozdzielczości i szkoda, że młodzi adepci szkoły programowania stykający się z terminologią z książki p. Freleka zaczną od takich niepoprawnych dziwołagów. Jednak w sumie z czystym sumieniem mogę polecić książkę wszystkim tym, którzy z Commodore czy z programowaniem kiedyś się zetknęli. Nie przyswoją oni sobie bowiem zleń terminologii a wyniosą dla siebie dużo nowej wiedzy zawartej w tej książce.

Commodore 64 Bohdan Frelek
WNT 1988 Cena 750 zł



FINAL CARTRIDGE III

Wśród wielu modułów (kart) dołączanych do C-64 rodzina FINAL CARTRIDGE jest chyba najbardziej popularna. Parę lat temu polski rynek czy raczej giełdę podbił FINAL CARTRIDGE II, dziś mam przyjemność przedstawić Czytelnikom jego młodszego brata — FINAL CARTRIDGE III.

Nie jestem pewien, czy polski rynek został zdominowany przez karty FINAL dzięki ich zaletom czy też dzięki stosunkowo łatwemu kopiowaniu i powielaniu w zacisznych pieleszach domowych różnych elektroników amatorów. Na dzień dzisiejszy jednak jest to jeden z najbardziej popularnych modułów dla C-64.

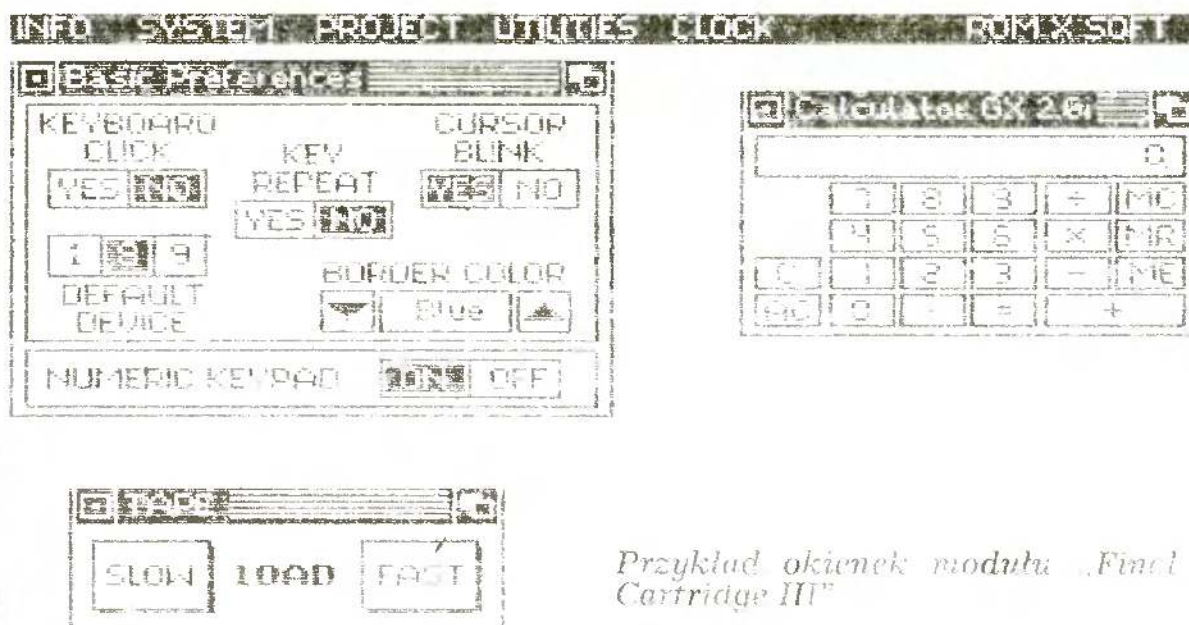
Karta ta zawiera w sobie dwie pamięci EPROM o pojemności 32 KB każda, dzięki czemu zwiększone zostały znacznie możliwości i estetyka programów z zawartych w tym module. FINAL CARTRIDGE III pracuje w oparciu o modną ostatnio technikę „okienek”. Po włączeniu komputera na ekranie ukazuje się menu główne — DESKTOP. W menu głównego mamy dostęp do opcji INFO (mówi o wersji programu i jego autorach), SYSTEM (praca w BASIC, wyłączenie programów Party, FREEZER oraz REDRAW — ponowne wyświetlenie wywołanych uprzednio okienek), PROJECT (podręczny edytor tekstu), UTILITIES (konfiguracja systemu, kalkulator, zarządzanie zbiornymi dyskowymi oraz TURBO dla DATASSETTE) i CLOCK (ustawianie zegara i ewentualnie „alarmu” na wyznaczoną godzinę). Wybor żądanej opcji polega na nalożeniu na strzałki przesuwanej joystickiem lub klawiszami funkcyjnymi i wciśnięciu przycisku. Jak widac autorzy byli chyba mocno zafascynowani systemem GEOS, czego reminiscencje widac nie tylko w menu głównym.

Po wybraniu danej opcji na ekranie ukazuje się okienko zawierające szereg następujących do wyboru. Muszę przyznać, że wiele z nich jest naprawdę pożytecznych i przydatnych w codziennej pracy programisty: „beep” potwierdzający wciśnięcie klawisza, włączony blok numeryczny (tylko jeśli dysponujesz C-128), powtarzanie klawiszy, deklaracja standardowego urządzenia, z którym współpracujesz najczęściej. Jednorazowo można na ekranie wywołać kilka okienek, przy czym wywołane ostatnio nakłada się na wyświetlane już na ekranie. Po przesunięciu strzałki w górny lewy lub prawy róg możemy zamienić okienka (stare nałożą się na nowe); można je także przesuwać w dowolne miejsce ekranu, wyłączać itp. Takie rozwiązanie znacznie ułatwia korzystanie z poszczególnych opcji karty.

Jedną z najczęściej wykorzystywanych możliwości modułu jest program przyspieszający czytanie programów z dyskietki. W porównaniu do wersji II został on znacznie ulepszony — czytanie odbywa się rzeczywiście w bardzo szybkim tempie. Szkoda tylko, że opcja ta nie jest dostępna stale — np. po wczytaniu i uruchomieniu programu DISK DOCTOR procedura ta automatycznie się dezaktywuje.

FINAL CARTRIDGE III zawiera również kilka opcji dla miłośników programowania i grafiki komputerowej. W pierwszym wypadku użytkownik otrzymuje do dyspozycji listę następujących poleceń dodatkowych: APPEND, ARRAY, AUTO, BAR ON/OFF, DEL, DESKTOP, DLOAD, DOS, DSAVE, DUMP, DVERIFY, FIND, HELP, KILL, MEM, MON, OLD, ORDER, PACK, PDIR, PLIST, RENUM, TRACE, TYPE, UNPACK. Z instrukcji tych można korzystać zarówno poprzez wpisanie jak też i wybranie ich ze specjalnego menu wyświetlanego w pierwszej linii ekranu. Sposób wyboru jest identyczny jak to opisano powyżej.

Zreformowany w ten sposób BASIC jest rzeczywiście znacznie bardziej przyjemny aniżeli oryginalna wersja V2.0 C-64. Bardzo przydają się tu przede wszystkim ARRAY (wyświetla na ekranie wszy-



Przykład okienek modułu „Final Cartridge III”

tkie wykorzystywane w programie zmienne), FIND pozwalający na wyszukanie określonego ciągu tekstu, czy MON wywołujący monitor języka maszynowego o sporych możliwościach.

Podobnie jak w wersji II, FINAL CARTRIDGE III wykorzystuje klawisze funkcyjne przypisując im konkretne instrukcje i rozkazy BASIC. Niestety instrukcje BASIC im przypisane są niezmiennie i nie można ich zastąpić innymi; jest to jedna z wad tej karty.

Oprócz wspomnianych powyżej karta ta oferuje ponadto cztery główne opcje: FREEZER, NOTE PAD, PRINT oraz UTILITIES. Pierwsza z nich służy do nagłego zatrzymania wykonywania programu i zapisaniu całej zawartości pamięci RAM na dyskietce lub taśmie; umożliwia to zarówno przenoszenie typowo dyskowych gier na taśmę (oczywiście nie wszystkich!) jak też i omijanie firmowych zabezpieczeń — wystarczy, że program będzie wczytany z dyskietki (czy taśmy) oryginalnej. FREEZER ma kilka opcji dodatkowych pozwalających na zmianę kolorów na ekranie graficznym, wyłączanie kolizji sprite'ów, zmianę portów joysticka, włączenie programowego AUTOFIRE itp. Wśród nich jest także opcja PRINT umożliwiająca uzyskanie wydruku ekranu graficznego czy też tekstowego. Przykładem mogą tu być rysunki do artykułu o programach kopiujących wykonane właśnie za pomocą tej karty.

Już samo bliższe przyjrzenie się opcji PRINT może przyprawić o zawrót głowy. Pomijając już takie „drobiazgi” jak możliwość powiększania rysunku w jednej z płaszczyzn (HORIZONTAL VERTICAL SIZE — maksymalne powiększenie jakie można uzyskać jest dziesięciokrotne) czy wydruk odwrócony o 90 stopni (SIDEWAYS), możliwe jest sterowanie wydrukiem drukarki 24-igłowej, wybór interfejsu (obsługa CENTRONICS jest zapisana w EPROM karty), wybór gęstości wydruku w trybie graficznym itp. Tu na marginesie pewna uwaga: posiadaczom drukarki NL-10 polecam wybór drukarki EPSON, powiększenie rysunku w płaszczyźnie poziomej (HORIZONTAL SIZE=2) i wybranie DOUBLE DENSITY (podwójna gęstość) z menu dla drukarek 8-igłowych. W takim układzie wydruk np. ekranu tekstowego przebiega w trybie graficznym, co daje znacznie lepszą czern i kontury rysunku. Wybór ekranu graficznego lub tekstowego do wydruku odbywa się automatycznie.

Stosunkowo najrzadziej wykorzystuję opcję NOTE PAD będącą małym edytorem tekstu. Przypomina ona trochę NOTEPAD z systemu GEOS, ma jednak znacznie większe możliwości, m.in. pozwala na regulowanie odstępu międzywierszowego i międzyznakowego (tylko na ekranie), pozwala na łączenie zbiorów (dokumentów), wydruk itp. Wydaje mi się jednak, że jest to najsłabszy punkt karty i szkoda, że miejsca zajmowanego przez ten edytor nie przeznaczono na zwiększenie możliwości opcji UTILITIES.

UTILITIES (programy użytkowe) oferują użytkownikowi możliwość wybrania i ustalenia najdogodniejszej konfiguracji systemu. Opcje DISK i TAPE odnoszą się odpowiednio do zarządzania i zbiorami i programami na dyskietkach, jak też wczytywania i zapisywania programów na kasetach w systemie normalnym (SLOW) lub TURBO (FAST). Oprócz wszystkich w zasadzie poleceń dyskowych (choć nie mogę przestać z tego menu np. polecenia „UO-M1”) użytkownik może sortować alfabetycznie programy zapisane w katalogu dyskietki oraz wprowadzać linie oddzielające pomiędzy ich tytułami. Tym niemniej obsługa samego okienka jest nieco bardziej skomplikowana, ponadto FINAL CARTRIDGE III nie dopuszcza możliwości wykonywania operacji na więcej niż jednym zbiorze jednocześnie.

Moduł ten jest wyposażony w dwa przyciski: FREEZER (lewa strona) oraz RESET. Wciśnięcie RESET powoduje wyłączenie karty; jeżeli jednak będziemy przy tym trzymali wciśnięty klawisz RUN/STOP, to przejdziemy automatycznie do BASIC. Z kolei trzymanie wciśniętego klawisza COMMODORE LOGO (C=) spowoduje powrót do DESKTOP. Ta sama zasada obowiązuje w chwili włączania komputera do sieci.

FREEZER może być uaktywniany w dowolnej chwili (poprzez wciśnięcie lewego przycisku), należy jednak pamiętać, że może on nie spełnić swojej funkcji, jeśli „zamrażany” program wykorzystuje przerwanie niemaskowalne NMI. Takie potknięcie zdarzyło się podczas prób wykonania wydruku z planu tytułowej programu GEOS.

FINAL CARTRIDGE III ma jeszcze wiele innych zalet, które z braku miejsca przemilczę. Generalnie jest to na pewno przydatna i znacznie lepsza karta aniżeli jej poprzednik; jest ona przede wszystkim dość uniwersalna, co pozwala mi polecić ją szerokiemu gronu użytkowników, w tym także i początkującym.

Klaudiusz Dybowski

UWAGA. FINAL CARTRIDGE III można kupić na warszawskiej giełdzie i oczywiście nie oryginał, lecz mniej lub bardziej udane kopie. Otrzymałem jednak kilka sygnałów od użytkowników, że nie wszystkie egzemplarze zachowują się prawidłowo. Dlatego też przed zakupem proponowałbym sprawdzenie WSZYSTKICH dostępnych możliwości tej karty, gdyż w przeciwnym wypadku można kupić bubel za ok. 10000 zł.

HEWLETT PACKARD 71B



Model ten należy do nowej generacji ręcznych komputerów personalnych.

Generacja ta charakteryzuje się tym, że w obudowie o wielkości kalkulatora (190mm x 97mm x 25mm) znajduje się prawdziwy komputer dający wyjątkowe możliwości obliczeniowe. Te ręczne „komputerki” posiadają wbudowany język programowania wysokiego rzędu jakim najczęściej jest BASIC, choć spotyka się również PASCAL, FORTH. HP-71B posiada wbudowany BASIC, a dodatkowo można za pomocą specjalnego modułu zainstalować FORTH i Assembler.

Cechy użytkowe HP-71:

Tryb obliczeniowy CALC to potężny, nieprogramowany tryb pracy łatwy do nauki i prosty w użyciu. Umożliwia przeprowadzenie najbardziej skomplikowanych obliczeń. Wyrażenia wprowadzone są w formie algebraicznej od lewej do prawej strony. Dwanastu cyfr w wyniku zapewnia dokładność wszystkich obliczeń. Ponadto tryb CALC współpracuje z trybem programowania BASIC. Polega to na tym, że wartości nadane zmiennym w BASIC mogą być wykorzystane (wywołanie przez nazwę zmiennej) w trybie obliczeniowym i vice versa. Każde wyrażenie wprowadzone w BASIC może być obliczone w trybie CALC. W trybie CALC można posługiwać się wbudowanymi funkcjami numerycznymi, jak również własnymi procedurkami. Można go zdefiniować jako często używaną funkcję i następnie wywoływać ją po zaciśnięciu pojedynczego klawisza. Dodatkowe funkcje traktowane są na równi z wbudowanymi. Oczywiście HP-71 posiada wbudowaną statystykę umożliwiającą przeprowadzenie obliczeń na 15 zmiennych niezależnych; pełny zestaw funkcji trygonometrycznych, logarytmicznych i innych.

Tryb programowania BASIC to 240 funkcji, rozkazów, poleceń i operatorów upraszczających pisanie programów oraz czyniących je przyjemniejszym. Program główny może składać się z szeregu podprogramów, z którymi może wymieniać się parametrami (parametry formalne i aktualne). Z ciekawszych poleceń wymienić trzeba instrukcję PRIVATE ograniczającą osobom niepowołanym dostęp do zbiorów, jak również zabezpieczającą zbiór przed wprowadzeniem poprawek, zmian SECURE (UNSECURE zabezpieczają) odbezpieczają zbiór do głównej pamięci RAM i automatycznie wykonuje polecenia zawarte o tym zbiorze: BEEP wzbogacający programy o przerywniki akustyczne o regulowanej długości i częstotliwości (zakres SH2 — 6200Hz).

Ostatnimi czasy kalkulatory i takie podręczne komputerki coraz częściej wyposażone są przez producentów w systemy operacyjne większe niż te zainstalowane na niejednym prawdziwym komputerze. HP-71 posiada 64 KB system operacyjny umożliwiający przeprowadzenie najbardziej wyrafinowanych i wymyślnych działań. System operacyjny zawsze czuwa! HP-71 posiada pięciopoziomowy stos (rozszerzalny do 16 poziomów) zapamiętujący ostatnio wykonane działania, rozkazy i dający możliwość ich powtórzenia bez powtórzonego opisywania ciągu rozkazów. Znakomicie przyspiesza to przeprowadzenie długich, żmudnych obliczeń dla których jednak z tego czy innego powodu nie warto pisać programu. HP-71 wyposażony jest w cztery pary RAM/ROM. Standardowa pamięć RAM wynosi 17,5 KB, którą można rozszerzyć o 16 KB do łącznych 33,5 KB. Aby umożliwić przyszłe rozbudowanie systemu HP-71 jest zaprojektowany tak aby dać możliwość bezpośredniego adresowania 512 KB pamięci. W celu zaoszczędzenia czasu przy programowaniu, można się posłużyć gotowymi modułami ROM z programami (256 KB). Programy te są opracowywane przez firmę, która dysponuje olbrzymią biblioteką programów do wszystkich wypuszczonych kalkulatorów (komputerów). Można również bezpośrednio u Hewlett-Packard'a zamówić przeniesienie naszych programów na taki gotowy cartridge. Dowolny obszar wbudowanej i zewnętrznej RAM może zestaw wydzielony dla programu i do przechowywania danych, tak aby zbiory były łatwo dostępne i zabezpieczone przed skasowaniem podczas resetu komputera. Można również odłączyć moduły pamięci RAM bez uszkodzenia zbiorów w pozostałej części pamięci RAM.

Często używane słowa kluczowe znajdują się na klawiaturze i mogą być wyświetlone przy naciśnięciu pojedynczego klawisza (tak jak w spectrum). Komputer ten nie posiada jeszcze menu (jak HP-28, HP-27, HP-32) dlatego każdemu klawiszowi (55 klawiszy przypisane jest do trzech funkcji wybieranych klawiszami: żółtym f i niebieskim g). Klawiatura jest w pełni redefiniowana tzn. że każdemu klawiszowi można przydzielić inną funkcję niż ta standardowo do niego przydzielona. Takie możliwości w połączeniu ze specjalnymi nakładkami na klawiaturę umożliwiają wygodne wykonanie programów bez konieczności uczenia się klawiszowania do danego zastosowania.

HP-71 posiada obudowany zegar (o regulowanej prędkości działania 10) w kalendarz. Oprócz ich dużej dokładności działania, również przy wyłączonym komputerze, mogą być one wykorzystane przy programowaniu i uruchamianiu programów jako np. zmienne decyzyjne.

Liczba różnych zbiorów jakie mogą być w pamięci komputera ograniczona jest jedynie dostępną pamięcią RAM. Można się posłużyć 7 różnymi typami zbiorów.

BASIC — zbiory zawierają teksty programów w BASIC-u

BIN — programy w języku maszynowym do wykorzystania jako program lub podprogram

LEX — zbiory przechowujące niezdefiniowane polecenia BASIC

DATA — zbiory przechowujące dane numeryczne i tekstowe

Text — zbiory służące do przekazywania danych do/ z innych komputerów

KEY — zbiory określające aktualną definicję klawiszy na klawiaturze

DATA — zbiory umożliwiające przekazywanie zbiorów do/z kalkulatora serii HP-41 (wkrótce w klanie użytkowników kalkulatorów)

Z ogólnych danych technicznych podaję za producentem parametry tej miniaturki komputera. Waga 340g wraz z czterema bateriami typu AAA. Wyświetlacz 256 różnych znaków wyświetlanych jest w oparciu o matrycę 6 x 8 punktów. Na wyświetlaczu mieszczą się 22 znaki ze scrollingiem w linii do 96 znaków.

Zakres numeryczny rzeczywista precyzja — 9.999999999999999 E99 — 1E-499; 0; 1E-499 9.999999999999999 E 499

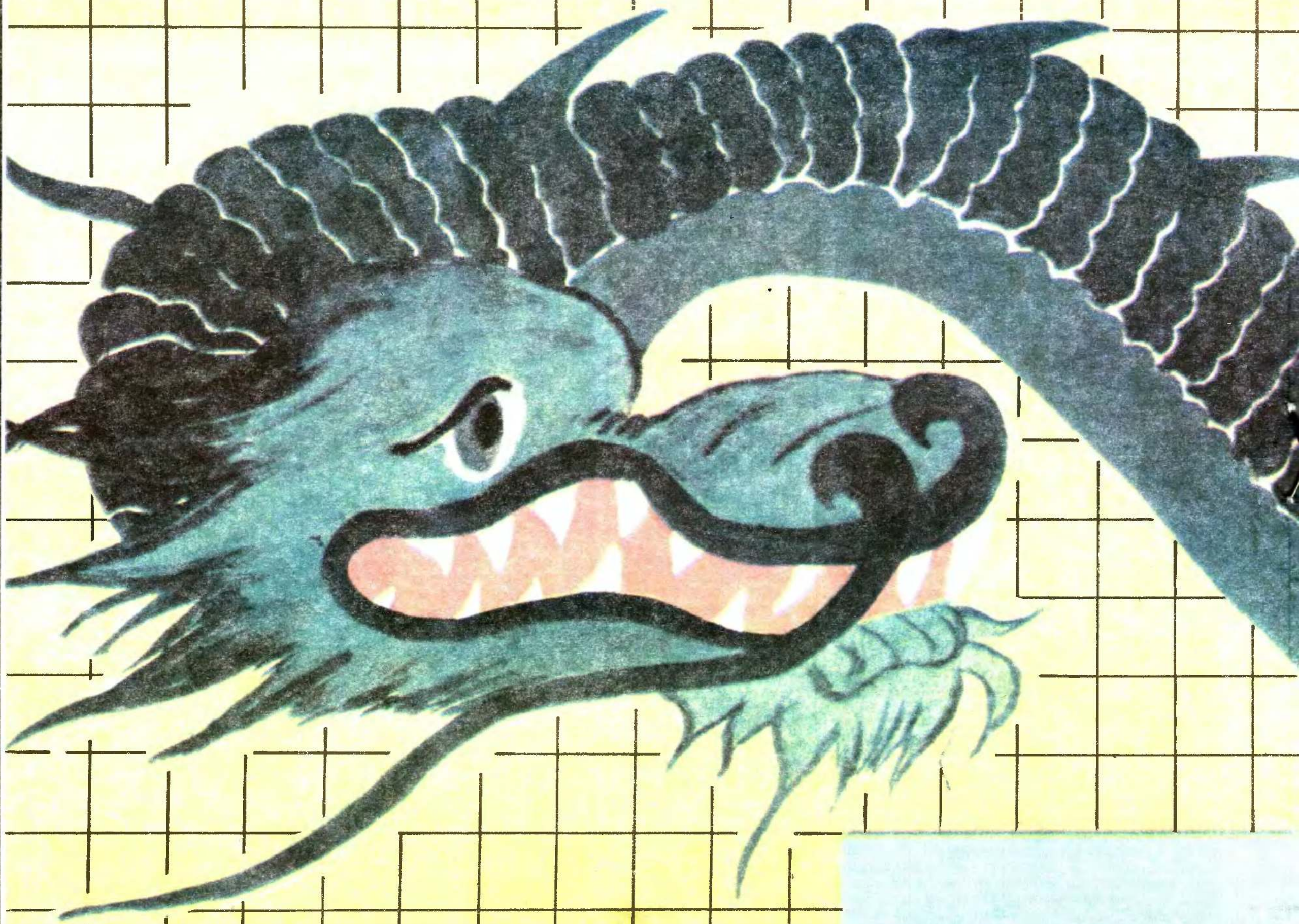
Krótką precyzją: jw. lecz z dokładnością do 4 miejsc po przecinku (lub kropce dziesiętnej do wyboru) liczby całkowite — 99999 — 99999

Wszystkich możliwości tak wyrafinowanego komputera nie da się zawrzeć w tak krótkim artykule. Wspomnę jeszcze o paru cechach wyróżniających HP-71 na tle podobnych komputerów. Czytnik pasków magnetycznych o pojemności 1,3 kB daje możliwość stworzenia biblioteki oprogramowania, przechowywania danych, tekstów i innych informacji na taśmie nośnika bez konieczności każdorazowego „wpakowywania” ich do pamięci. Czytnik napędzany jest ręcznie więc zaoszczędza nam baterii.

Tak jak w komputerach, tak HP-71 posiadał otwartą architekturę. Oznacza to, że można go dostosowywać do różnych często specyficznych zastosowań. Za pomocą interfejsu HP-IL można HP-71 podłączyć do urządzeń peryferyjnych, instrumentów pomiarowych wyposażonych w złącza RS-232C, HP-IB (IEEE 488), GPIO. Za pomocą HP-IL, HP-71 może jednocześnie komunikować się z 30 urządzeniami połączonymi w pętlę. Przy dodatkowym adresowaniu wspomagającym liczba ta rośnie do 930 urządzeń choć osobiście nie wyobrażam sobie takiego zastosowania.

Z urządzeń peryferyjnych wymienić trzeba monitor 80-kolumnowy do HP-71; plotter, drukarkę, strecmer; stację dysków, magnetofon cyfrowy, modem i wiele innych. Reasumując trzeba stwierdzić, że HP-71 jest odpowiednim urządzeniem do zastosowań, do których komputer jest jeszcze do drogi. Dzięki olbrzymim możliwościom i małym gabarytom spełnia funkcje dobrze oprogramowanego komputera z własnym zasilaniem dającego możliwość korzystania z niego wszędzie w podróży, na warsztacie — podłączenia do instrumentów pomiarowych celem pobrania danych do dalszej obróbki itd.

Dominik Falkowski

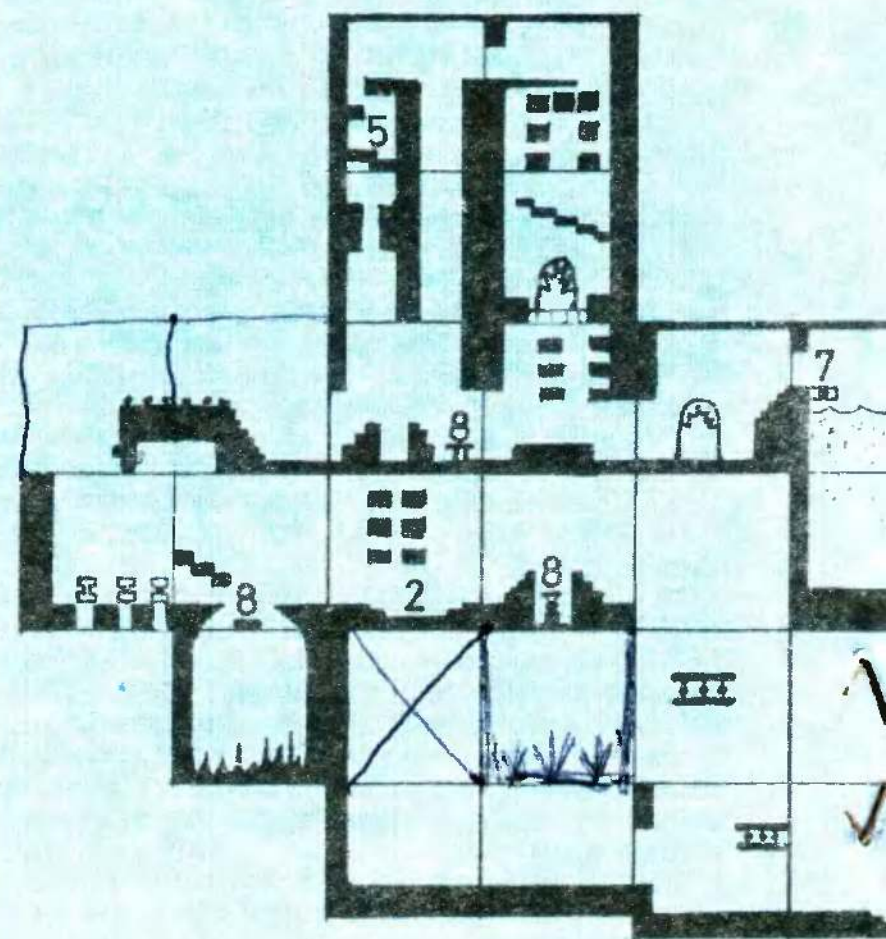


Królestwem, którym jeszcze niedawno rządziłeś, zawładnął podstępny Czarnoksiężnik w postaci olbrzymiej, szkaradnej bestii. Wsączył w serca Twoich byłych poddanych strach i potworną groźbę. Twoje wojska w nierównej walce zostały doszczętnie rozgromione. Po tej porażce musisz sam postarać się unieszkodliwić tyrana. Z pewnych źródeł wiesz, że słabą stroną bestii jest jego, niechroniona skorupa głowy. Wiedział on, że przybędziesz odebrać swoją utraconą koronę. Rozmieszczył po całym zamku demoniczną straż. Swoje magiczne przedmioty dla pewności powierzył sługom w różnych częściach dużej budowli. Musisz je zdobyć i przy ich pomocy odzyskać władzę. Jeśli ci się nie uda Twoja misja, całe państwo pogrzeże się w otchłaniach ciemności i magii potwora. Aby dostać się do niektórych części kamiennego kolosa, będziesz musiał zmienić swoją postać. Czeka cię trudne zadanie.

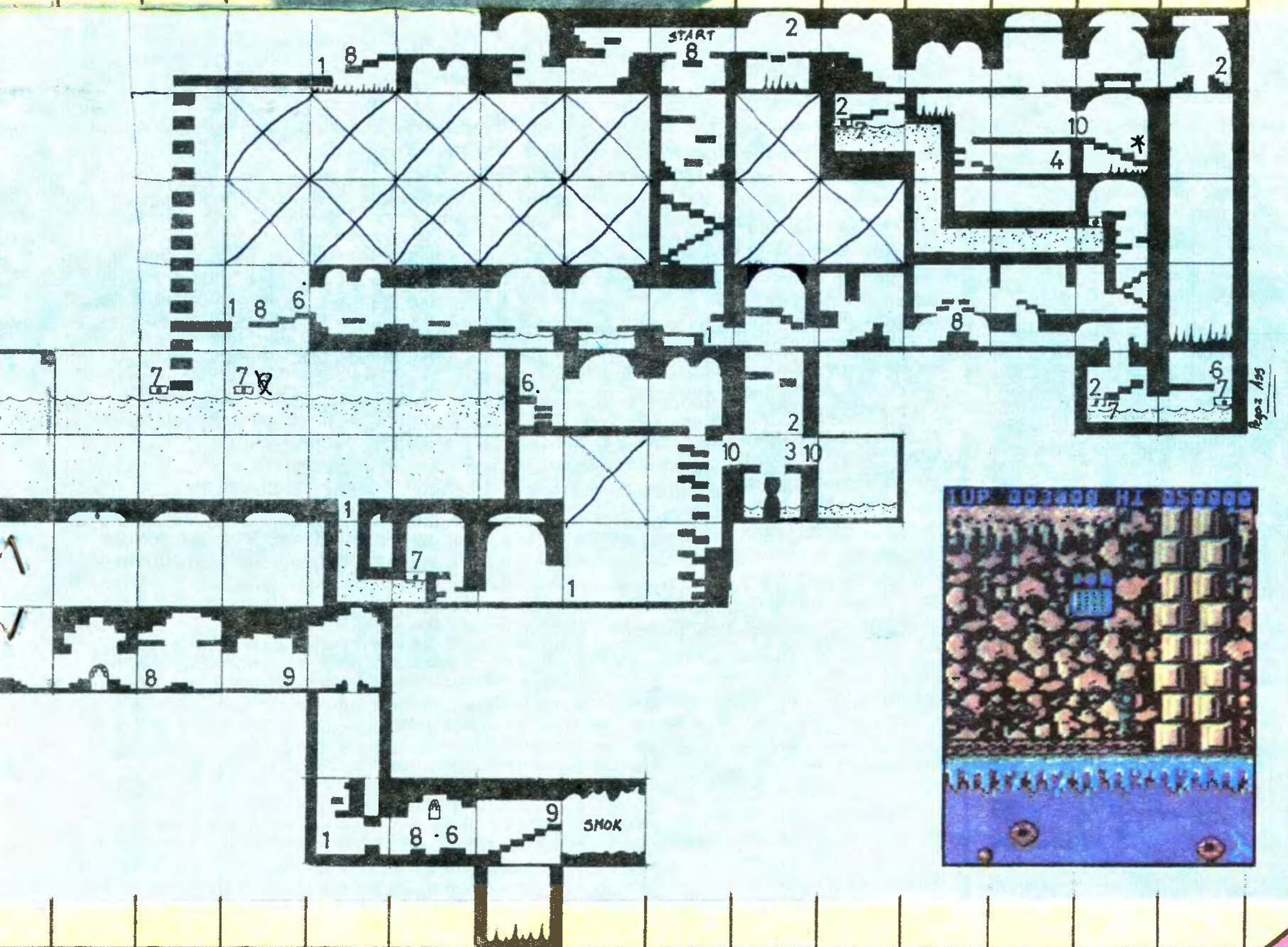
Sebastian Kajdan

Legenda do gry "Draconus"

- 1 - energia
- 2 - ogień w płynie
- 3 - ochrona przeciw demonom
- 4 - magiczny przedmiot
- 5 - oko smoka
- 6 - czar
- 7 - płyta zwyczajna
- 8 - płyta specjalna
- 9 - potwór
- 10 - ściana (znika, gdy masz oko)



DRACOMUS



Bajtek

100

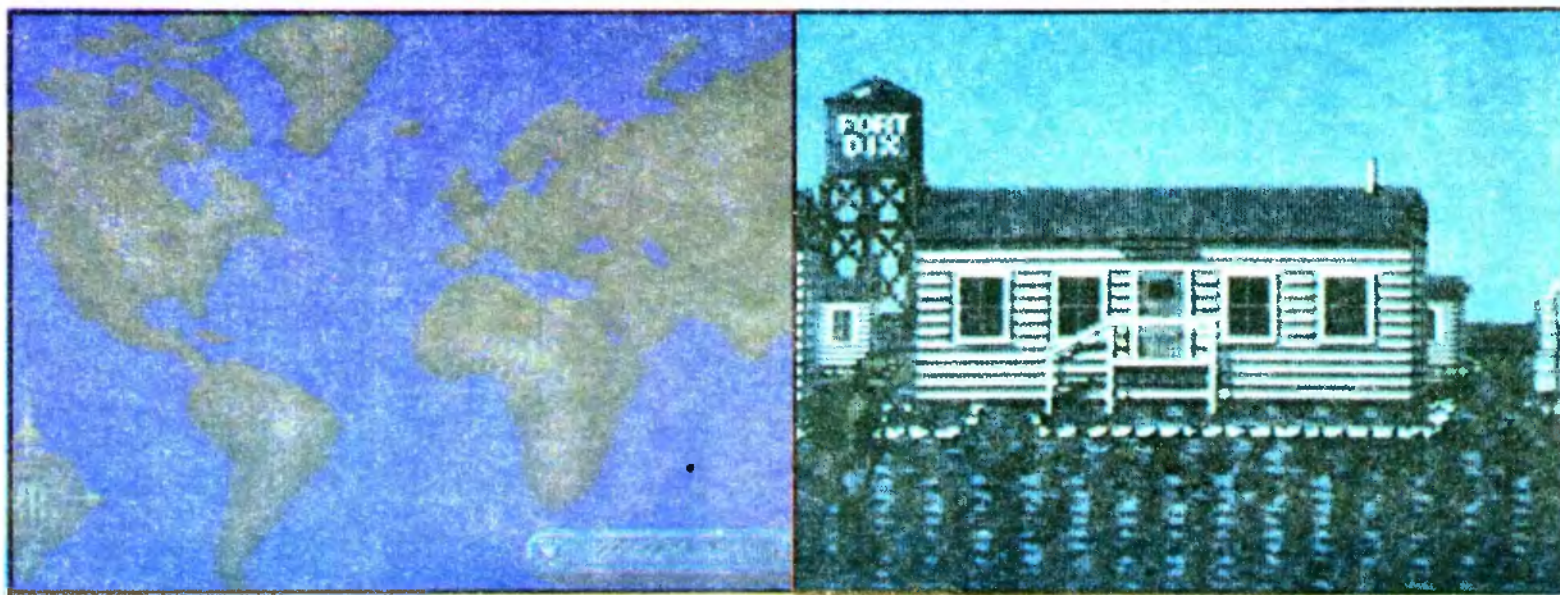
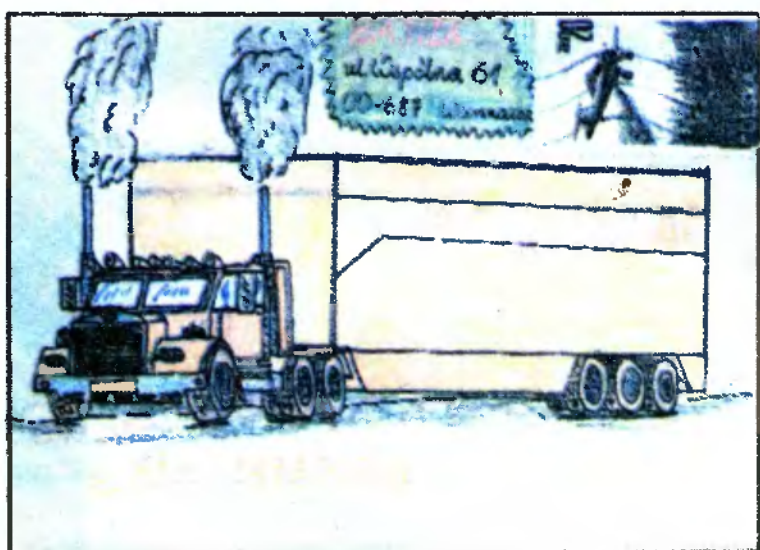
BAJKOWA LISTA PRZEBOJÓW (7/89)

Dialektyka — jedność przeciwieństw jest budulcem tego świata. Również i Listy Przebojów. Gry nowe i stare, wściekle strzelaniny i łagodne tekstówki spotykają się w pierwszej dziesiątce ulubionych gier. Tak jest i w tym miesiącu. Króluje „Operation Wolf”, obecna u nas stosunkowo od niedawna. Dalej również gry nie najstarsze, pewnie zaciekawienie budzi „Police Quest” i „King's Quest” dostępne na niewiele komputerów. A na samym końcu „Green Beret”, historia i klasyka obfitych zręcznościówek.

Na siódme tegoroczne notowanie Listy napłynęło 3590 propozycji na 194 tytuły gier.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 OPERATION WOLF ↑	x	x	x	x
2 STREET FIGHTER ↑	x	x	x	x
3 HOT SHOT ↓		x	x	x
4 POLICE QUEST !	x			
5 GUNSHIP ↑	x	x	x	x
6 ROCKET RANGER !			x	
7 KING'S QUEST ↓	x	x	x	x
8 PLATOON ↓	x	x	x	x
9 DRACONUS ↓			x	x
10 GREEN BERET ↓	x	x	x	x

Kopertę nadesłał Łukasz Wielgosz z Wrzesni



ROCKET RANGER

Jest rok 1940. Niemieckie fabryki pracują na pełnych obrotach, aby dostarczyć wojsku potrzebny sprzęt i amunicję. Messerschmitty tysiącami wędrują na front, wielkie opancerzone czołgi stoją na platformach, czekając na rozkaz. Podkute buty narodowego socjalizmu szykują się do podbicia wielkich obszarów Francji i Rosji, niedługo zjawia się też w Jugosławii i Hiszpanii. Nie ma takiego państwa ani systemu, który zdoła zatrzymać tę szerzącą się w oszołamiącym tempie plagę. Jest natomiast człowiek, mogący całą niemiecką maszynę zmienić w bezwładną bryłę śmiecia.

Pewnej nocy, gdy siedziałś przy biurku rysując model maszyny parowej, pokój zaczął nagle drgać i wibrować. Na stole ukazały się schematy bardzo dziwnego urządzenia, wyglądającego z pozoru jak pojemnik na jedzenie. Jednak, gdy zmontowano go w jednym z laboratoriów, w Twoim ręku znalazł się supernowoczesny plecak raketowy, z turbo odrzutem powierzchniowym oraz scalającą przestrzeni i czasu. Właśnie przy jego pomocy masz szansę walczyć ze znieprawionym faszyzmem oraz z Wehrmachtem, Luftwaffe i Kriegsmarine. Ostateczne zwycięstwo jest w zasięgu graczy o silnym charakterze, którzy spędzą przy komputerze minimum miesiąc.

Gdy zaczynasz, możesz wykonać jedną z czterech czynności:

WAR MAP — mapa świata. Oznaczone są na niej państwa odgrywające jeszcze jakąś rolę w wojnie. Posiadasz pięciu agentów, których możesz dowolnie przemieszczać. Są oni jednak bardzo narażeni na zdemaskowanie, co zawsze kończy się dla nich śmiercią, lub obozem koncentracyjnym. Każdy z wywiadowców może organizować ruch oporu, masz także możliwość zwiększenia szybkości i efektywności ich działania.

FUEL DEPOT — magazyn paliwa. Zaopatrujesz się tutaj w paliwo do swojego plecaka oraz rakiety. Pamiętaj, że nie całe paliwo

zuzyciesz na raz, część zostaw na powrót lub lot do innego kraju.

ROCKET LAB — informacje o ilości zdobytych części od nowej niemieckiej rakiety.

TAKE OFF — start. Przewożony zostajesz do Fort Dix, gdzie startujesz (lub nie) z betonowego pasa. W tym celu na każde dotknięcie nogą ziemi, przyciśnij **FIRE + lewo**. Gdy odezwie się sygnał dźwiękowy informujący o możliwości oderwania się od powierzchni ziemi, wciśnij **FIRE + góra**. Odległość lotu determinuje ilość załadowanego wcześniej paliwa do zbiornika w plecaku. Zależności te znajdują się w załączonej tabelce.

Informacje przekazywane przez wywiadowców mówią zwykle o odkryciu tajnej bazy. Lot do niej jest konieczny, gdyż właśnie tam masz możliwość zdobycia jednej z pięciu części rakiety, którą Niemcy chcą użyć do lotów na Księżyc. Inne wiadomości przekazywane przez agentów dotyczą transportów wojskowych, wykrytych fabryk, itp.

Na samym początku gry, niemieccy komandosi nadlatują sterowcem nad Nowy Jork i pory-

CEL LOTU ROCKET RANGERA

	A	A	A	B	K	K	K	A	B	A	F	N	W	K	L	S	M	P	P	S	H	S	U	Z	V	Z	J
Algieria	0	28	20	43	37	44	25	29	21	16	14	17	15	31	10	19	23	30	50	22	13	24	36	26	40	18	12
Arabia	28	0	41	57	59	65	30	18	14	34	29	27	24	26	20	17	31	12	70	32	33	19	58	23	61	37	22
Atlantyk	20	41	0	32	27	34	37	39	33	18	19	26	22	43	29	31	35	42	44	25	17	36	23	38	30	24	28
Brazylia	43	57	32	0	49	27	32	42	52	50	45	53	47	46	42	54	37	61	25	55	40	44	41	60	26	33	51
Kanada	37	59	27	49	0	28	56	62	51	31	36	38	39	64	44	47	48	55	40	34	35	57	10	41	26	42	43
Kolumbia	44	65	34	27	28	0	52	64	59	47	40	50	48	52	49	57	45	66	17	51	43	61	26	56	12	39	53
Kongo	25	30	37	38	56	52	0	22	23	36	33	34	32	17	24	26	15	31	54	40	29	20	55	33	50	21	27
Afryka Wsch	29	18	39	48	62	64	22	0	15	37	32	33	30	14	23	24	25	19	67	35	21	12	61	28	60	34	26
Egipt	21	14	33	52	51	59	23	15	0	27	24	20	19	25	12	11	18	17	63	28	26	13	50	22	54	19	16
Anglia	16	34	18	50	31	47	36	37	27	0	8	11	13	40	21	22	33	29	53	15	12	30	32	19	41	28	70
Francja	14	29	19	45	36	46	33	32	24	8	0	12	10	37	17	20	30	28	52	16	9	27	35	21	42	25	15
Niemcy	17	27	26	53	38	50	34	33	20	11	12	0	9	35	18	19	32	24	58	10	16	29	30	15	45	31	13
Włochy	15	24	22	47	39	48	32	30	19	13	10	9	0	33	14	12	29	25	57	17	11	26	38	18	44	27	8
Kenia	31	26	43	46	64	52	17	14	25	40	37	35	33	0	28	30	22	27	65	39	36	18	63	34	59	32	29
Libia	10	20	29	42	44	49	24	23	12	21	17	18	14	28	0	13	19	22	56	27	15	16	43	25	46	26	22
Sr. Wschód	19	17	31	54	47	57	26	24	11	22	20	14	12	30	13	0	28	18	64	21	25	23	46	16	52	36	10
Nigeria	23	31	35	37	48	45	15	25	18	33	30	32	29	22	19	28	0	34	49	36	27	17	47	40	43	16	24
Persja	30	12	42	61	55	66	31	19	17	29	28	24	25	27	22	18	34	0	72	26	32	21	54	20	62	41	23
Peru	50	70	44	25	40	17	54	67	63	53	52	58	57	65	56	64	49	72	0	60	51	62	33	66	20	43	61
Skandynawia	22	32	25	55	34	51	40	35	28	15	16	10	17	39	27	21	36	26	60	0	20	31	37	14	27	38	18
Hiszpania	13	33	17	40	35	43	29	21	26	12	9	16	11	36	15	25	27	32	51	20	0	28	34	24	39	23	19
Sudan	24	19	36	44	57	61	20	12	13	30	27	29	26	18	16	23	17	21	62	31	28	0	56	32	55	22	25
USA	36	58	23	41	10	26	55	61	50	32	35	36	38	63	43	46	47	54	33	37	34	56	0	45	24	40	42
Zw. Radziecki	26	23	38	20	41	56	33	28	22	19	21	15	18	34	25	16	40	20	66	14	24	32	45	0	53	44	17
Wenezuela	40	61	30	28	26	12	50	60	54	41	42	45	44	59	46	52	43	62	20	27	39	56	24	53	0	35	49
Zach. Afryka	18	37	24	33	42	39	21	34	29	28	25	31	27	32	26	36	16	41	43	38	23	22	40	44	38	0	30
Jugosławia	12	22	28	51	43	53	27	26	16	70	15	13	8	29	22	10	24	23	61	18	19	25	42	17	49	30	0

POZYCJA ROCKET RANGERA

Gen & Luke '89

wają jednego z najlepszych amerykańskich naukowców — Ottona Barnstorffa, wraz z córką Jane. Jedynym sposobem na ich uratowanie jest odnalezienie sterowca w trakcie powrotu do Niemiec i zajęcie go. Pamiętaj jednak, że trafienie w powłokę wypełnioną wodorem powoduje wielki wybuch, z którego nie uratuje się nikt, oprócz Ciebie.

Gdy uda Ci się dostać do wnętrza sterowca, zadanie Twoje nie będzie jeszcze zakończone. Odpowiednimi słowami powinieneś przekonać

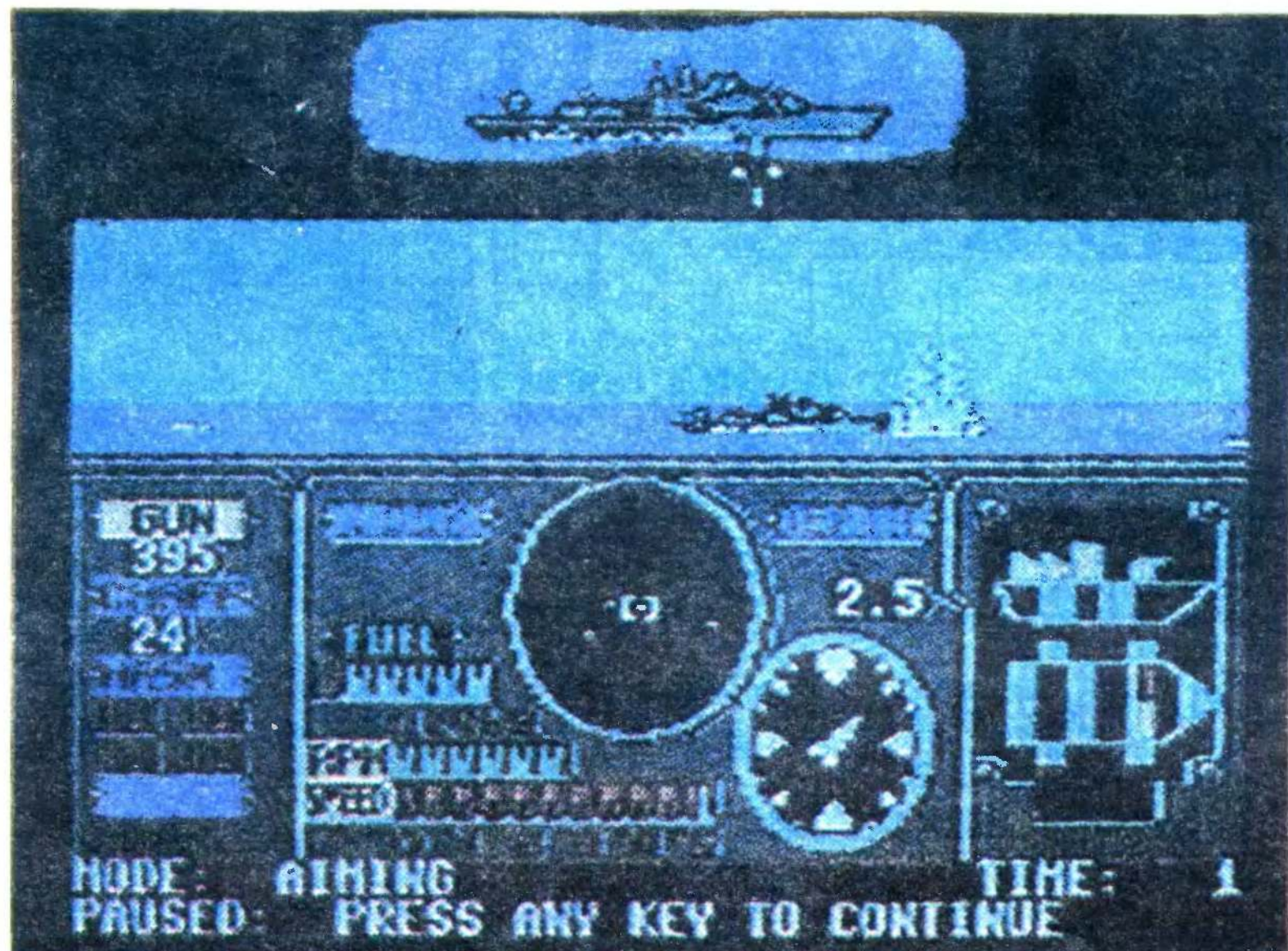
Jane, że jesteś przyjacielem. Jeżeli nie uwierzy, poleci wraz z ojcem prosto do Niemiec, aby już nigdy nie ujrzeć światła Nowego Jorku.

Tak więc los Europy i Świata spoczywa w Twoich rękach. Czy na Białym Domu zawiśnie flaga ze swastyką, czy też staniesz się bohaterem, o którym nie zapomną pokolenia?

Firma: Strike Force

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

Luke



PHM PEGASUS

Pływaliliśmy już różnymi urządzeniami, raz na wodzie, raz pod wodą. Lataliśmy szybko i wolno, wysoko i nisko, ale nigdy jeszcze nie poruszaliśmy się wodorolotem.

PHM Pegasus to nazwa eksperymentalnego wodorolotu, który używany będzie przeciwko piratom. Jedyne dwa istniejące egzemplarze Pegasus ma w posiadaniu armia UK i skrzętnie ich pilnuje. Jak więc ten dostał się w Twoje ręce? Zamieszana jest w to firma EOA, ale lepiej o to nie pytać, gdyż może być to ostatnie zadane pytanie.

Tak więc masz ten wodorolot i nie wiesz, jak się nim posługiwać? Nic prostszego, mogę przybliżyć Ci tę trudną sztukę, lecz mniemam, iż Twoja wdzięczność wyrazi się w miarodajnych jednostkach...

Na początku popatrz dobrze na mapę, rozwieszoną w kabinie. Widzisz tam pewną część morza, kawał lądu i kilka małych punktów. Ten, obwiedziony jasnym okręgiem, to Pegasus, a okrąg pokazuje zasięg jego radaru, a więc i strażników. Inne kropki to jednostki pływające lub latające; obwiedzione okręgami będą Ci przyjazne.

Klawiszami **F1—F4** możesz przełączać nadajnik i wyznaczać kurs dla każdej jednostki. Kursor przesuwasz joystickiem, zaś prędkość regulujesz klawiszami **1—5**.

Czy czas nie płynie zbyt wolno? Jeśli tak, to używając klawiszy **+** i **-** możesz jego upływ odpowiednio zwielokrotnić.

Wciśnij **V** — znajdziesz się przy oknie. Spacją zmieniasz tryb z manewrowania na strzelanie i odwrotnie. W trybie manewrowania poruszanie joystickiem powoduje odpowiednie przyspie-

szanie i zwalnianie oraz skręt w lewo lub prawo. Trybu strzelania używaj, gdy uchwycisz cel w lornetkę.

Zasięg radaru zmieniasz klawiszem **R**. Klawisz **T** powoduje uchwycenie najbliższego celu w lornetkę oraz zmianę celu. Wtedy możesz włączyć tryb strzelania, wybrać rodzaj broni klawiszem **W** (gun — działo, chaff — flary, hasm — rakiety samonaprowadzające średniego zasięgu, gabriel — rakiety samonaprowadzające dalekiego zasięgu), joystickiem przemieścić kursor w lornetce i strzelać.

W czasie pogoni za uciekającymi celami nie zbliż się za bardzo do brzegu. O krytycznej głębokości (płytkości) wody poinformuje wskaźnik **DEPTH** na tablicy przyrządów.

To już wszystko, co potrafi Twój Pegasus. Nie jest to wcale mało, przekonasz się o tym w akcji.

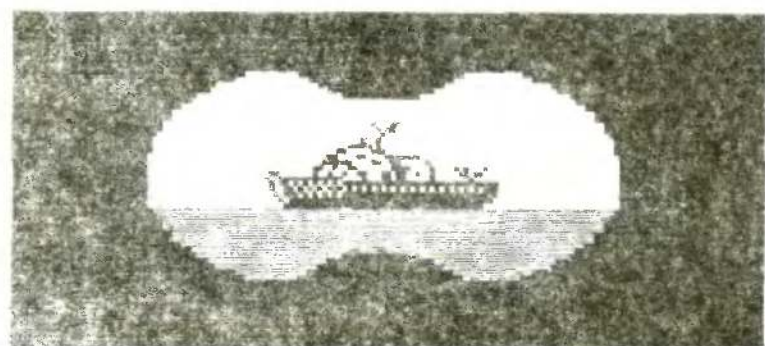
Którą zaś z nich wybierzesz, to zależy od Ciebie. Każda jest inna i różny jest też ich stopień trudności. Tak więc baw się dobrze i nie popsuj wodorolotu, lecz odstaw go na miejsce nietknięty.

Firma: Electronic Arts

Producent: Lucasfilm Games

Komputer: ZX Spectrum 48/+, Commodore 64, Atari ST, IBM PC.

Gen



S.O.S.

Proszę o udzielenie mi informacji o grze UNIVERSAL HERO. Nie wiem, co zrobić w komnacie z komputerem po pojawieniu się napisu „Please identify your self”.

Lukasz Mazurkiewicz ul. Zamieniecka 17 04—158 Warszawa

Posiadam mikrokomputer C—16. Chciałbym rozszerzyć pamięć do 64 Kb. Bardzo proszę o podanie adresu instytucji lub osoby prywatnej mogącej dokonać tej przeróbki. (Okolice Śląska)

Dariusz Bieniek ul. Karola Miarki 21/3 41—500 Chorzów

Próbuję rozszyfrować grę ROBIN HOOD. Chodzę po lesie i w pewnym miejscu Robin zatrzymuje się. Co potem robić? W zamian za pomoc oferuję 150 gier i opisów. Poszukuję także gry BARBARIAN, CONAN, ZORRO II, ZYBEX. Posiadam Atari 800XL.

Daniel Krajewski ul. Wankowicza 9b/10 58—500 Jelenia Góra

Commodorowcy pomóżcie! Nie mogę uruchomić gry BOLDER DASH w wersji na C-16.

Piotr Czesak ul. Klonowa 20/19 00—901 Warszawa

Pilnie potrzebuję na mikrokomputer Spectrum gier KUNG-FU MASTER, POP EYE, GUN FRIGHT. Wymienię je na inne.

Ryszard Opara ul. Zemska 36/17 54—438 Nowy Dwór woj. wrocławski

Gry SPY vs SPY I,II,III, STARQUAKE, THE GOONIES, PANTHER i wiele innych zamienię na BARBARIAN, SPY vs SPY IV, HOBBIT, WINTER GAMES i MISS PACMAN. Posiadam Atari 65XE.

Piotr Pańczyk ul. H. Sawickiej 17/25 43—100 Tychy Jestem posiadaczem Commodore +4. Do czego służy instrukcja SPRAY w programie SPEECH PLUS. Jak grać w MAYHEM, BEACH HEAD, BATTLE i MISSION APOLLO. W zamian służę opisami wielu gier.

Marcin Sokółowski ul. Urzędnicza 9a/44 25—729 Kielce

Proszę o pomoc w zdobyciu gry BARBARIAN w wersji kasetowej na Atari 65XE.

Adam Maik os. Zwycięstwa 11/46 61—646 Poznań

W zamian za opisy do gier PYJAMARAMA, CONQUEST, UNIVERSAL HERO, MONTEZUMA'S REVENGE oferuję mapy gier PHARAOON'S CURSE, STARQUAKE, JET SET WILLY, PITFALL II, BRUCE LEE. Mam Trnexa 2048. Pomóżcie!

Tomasz Szulczyński os. Świerczewskiego 41/8 64/300 Nowy Tomysł

Proszę o informację, jak przejść trzecią i drugą planszę w grze THE GOONIES w wersji na Spectrum.

Paweł Parys ul. Goławicka 4/8 03/550 Warszawa

Posiadam Atari 65XE. Proszę o pomoc w grach SILENT SERVICE! HARD BALL, EASTERN FRONT, DIZZY DICE i FIN ORBIT. W zamian służę pięćdziesięcioma opisami do gier.

Piotr Adamiak, ul. M. Konopnickiej 8h/2, 63-400 Ostrów Wlkp. woj. kaliskie

— KRÓL I KRÓLOWA GIER —



Hanna Puś lat 18, uczennica klasy III (biologiczno-chemicznej) VIII LO im. Adama Mickiewicza w Poznaniu.

Jeszcze nie ma komputera, ale chciałaby mieć Commodore 64.

Lubi wszystkie gry komputerowe. Hobby: Muzyka rozrywkowa.



Marcin Pruszyński lat 17, uczeń klasy II t Technikum Energetycznego w Warszawie. Posiada komputer: Commodore 64.

Ulubiona gra: Tetris. Hobby: muzyka, informatyka, żeglarstwo.

Oto zadanie godne mistrzów klawiatury: mamy zapisany ciąg liczb. Dostając jakąś nową liczbę musimy szybko sprawdzić czy jest ona w tym ciągu. Zadanie takie wynika przy wielu zastosowaniach praktycznych, dodatkowo utrudnione przez to, że nasz ciąg może być modyfikowany w czasie pracy — można do niego dopisać oraz usuwać z niego elementy.

W ubiegłym miesiącu stwierdziliśmy, że wyszukiwanie jest dużo szybsze jeśli nasz ciąg (zapisany w tablicy jednowymiarowej) posortujemy przed rozpoczęciem poszukiwań. Wtedy jednak musimy uważać przy wstawianiu żeby zachować porządek. Wymyśliliśmy dwa sposoby porządzenia sobie — dostawić nowy element na końcu i ponownie wywołać sortowanie, lub elementy większe od wstawianego przesunąć — w prawo o jedną pozycję, tak aby od razu można wstawić na właściwe miejsce.

Pierwsza metoda jest kosztowna (przyjęliśmy, że koszt prymitywnego algorytmu sortowania wynosi N^2), koszt drugiej wynosi ok N operacji. Tak więc wstawiając pojedynczy element nie możemy się wahać ani chwili którą wybrać. Ale jeśli mamy do wstawienia np. 10 czy 20 nowych pozycji od razu, czy nie taniej będzie dopisać wszystkie a następnie raz posortować całość? Intuicyjnie pomysł wydaje się niegłupi, ale dla pewności warto by przeliczyć. Zrobimy to w sposób uproszczony, zakładając, że N jest bardzo dużą liczbą i dlatego zwiększanie jego wartości podczas dodawania kilkunastu elementów można pominąć (zamiast $N+10$ w rachunkach użyjemy N). Jedno posortowanie będzie nas kosztowało N^2 zaś dodanie 10 liczb przez przesunięcie wartości w tablicy będzie kosztowało

10 N czyli dla dużych wartości N znacznie mniej niż jedno jedyne posortowanie. Np. dla $N=1000$ mamy $N=1000000$, a $10 N=10000$, czyli różnica ogromna, i co należy podkreślić zupełnie sprzeczna z naszymi poprzednimi intuicjami. Jednak wymowa cyfr jest jednoznaczna i wniosek ostateczny chyba też — nie wolno nam zbyt po prostu polegać na intuicji, tam gdzie chcemy oceniać efektywność działania algorytmów niezbędne są precyzyjne rachunki. Dla porządku wreszcie dodajmy, że przyjmując do rozważań inny koszt algorytmu sortowanie N liczb (możliwe jest $N \log N$), inny przedział wartości danej N oraz inną ilość jednorazowo dostawianych elementów (np. zbliżoną do wartości N) możemy otrzymać inną ocenę opłacalności.

Tyle zaległych rozrachunków ze wstawianiem. Pomyślmy o drugim problemie — usuwaniu elementów. Tu znowu są przynajmniej dwie możliwości: albo od razu fizycznie usuwamy wartość, tzn. przesuwamy elementy leżące na prawo od usuwanego o jedno miejsce w lewo, także zostaje on zamazany, — albo w miejscu usuwanej wartości wpisujemy jakąś liczbę, o której wiemy, że na pewno nie może wystąpić w naszym ciągu i liczba ta (w niektórych zastosowaniach może to być po prostu zero) będzie oznaczać „to miejsce jest puste”.

WESTCHNIENIE ZA PROSTOTĄ

Pierwsza metoda jest dość prosta koncepcyjnie, ale za to pracochłonna w realizacji — wymaga wielu przesunięć. Druga pozwala zaoszczędzić tych przesunięć ale stwarza nowe problemy: wstawiane „puste elementy” mogą zaburzać porządek tablicy, więc trzeba je odrębnie traktować przy przeszukiwaniu, blokują one miejsce w tablicy, więc trzeba od czasu do czasu wykonać odświeżenie i przesunąć zawartość tablicy w lewo likwidując wszystkie nagromadzone „pustostany” za jednym zamachem. Można też inaczej: dopisując nowy element znaleźć miejsca po usunięciu, a dopiero gdy takiego nie będzie dopisywać na końcu. Można to wszystko jeszcze bardziej rozbudowywać (i oczywiście komplikować), wzdychając tylko do czasu do czasu: „takie to było piękne, proste zadanie...”.

Niestety wszystkie omówione problemy muszą znaleźć rozwiązanie, więc nie uda nam się odzyskać prosto przez ich zignorowanie. Ciągłe przesuwania elementów tablicy też nie mogą nas satysfakcjonować — zamiast odpowiadać czy szukane elementy są w naszym ciągu będziemy w kółko reorganizować zawartość.

Powróćmy jeszcze na chwilę do przykładu, w którym dany ciąg był listą identyfikatorów towarów znajdujących się w magazynie. Zwykle o każdym towarze musimy zapamiętać sporo danych: ilość sztuk, data, dostawy, cena itd. Wtedy nie mamy do czynienia z jednowymiarową tablicą numerów towaru, lecz z tablicą mającą tyle kolumn ile danych mamy w towarze

STRUKTURY DANYCH (3)

Numer	Ilość	Data dostawy	Nr. regału	Nr. półki
5				
11				
34				
43				
223				
241				

Jeden wiersz tablicy stanowi opis jednej pozycji, zaś numery pozycji mieszczą się w pierwszej kolumnie, tzn. nasze poszukiwania wykonujemy w pierwszej kolumnie. Ale przesuwając musimy zawsze przesunąć nie pojedyncze liczby lecz całe wiersze, żeby nie pogubić informacji. Teraz dopiero są powody do wzdychania...

WŁODEK, ZACZYNAJMY OD NOWA

Musimy znaleźć inne rozwiązanie, a kierunek poszukiwań wydaje się być zupełnie jednoznaczny — wszystkie nasze kłopoty wzięły się z szukania nowego i odzyskiwania zwolnionego miejsca w tablicy. Tablica okazała się strasznie sztywną strukturą, za sztywną jak na potrzeby tego zadania. Musimy cofnąć się aż do pierwszego kroku, którym był wybór sposobu reprezentacji danych w pamięci i zapisać dane w inny sposób.

Żeby znowu ominąć szczegóły ustalmy, że wszystkie informacje o jednej części tworzą jeden zapis (często mówimy „rekord” od angielskiego record). Struktury tego rekordu nie będziemy badać, bo może być różna w różnych zadaniach. Ważne jest, że w rekordzie występuje pozycja pozwalająca go jednoznacznie zidentyfikować. Tak więc mając ciąg rekordów możemy wziąć pod uwagę identyfikator każdego z nich i... mamy nasze poprzednie zadanie wyszukiwania — w ciągu liczb (identyfikatorów rekordów) odszukać podaną na wejściu wartość X (i w ten sposób równocześnie odszukujemy rekord o identyfikatorze X), w narysowanej poprzednio tabelce jednym rekordem jest jeden wiersz, a identyfikatorem rekordu jest numer zapisany w pierwszej kolumnie.

Jednak z reprezentacji w postaci tablicy postanowiliśmy zrezygnować, w związku z tym spróbujemy ciąg naszych rekordów zapisać w sposób, który graficznie można przedstawić następująco:

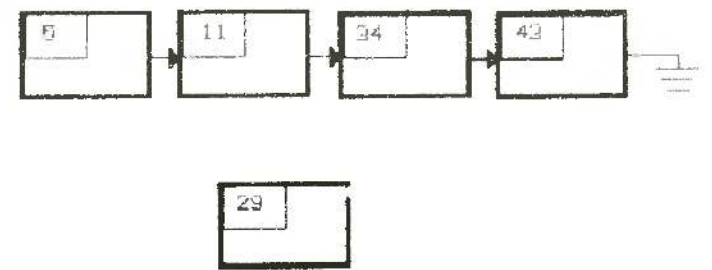


Na razie nie wnikajmy w istotę strzałek łączących rekordy, chcemy tylko by jednoznacznie określały położenie następnego rekordu w pamięci i pozwalały do niego łatwo przejść. Popatrzmy jakie możliwości uzyskamy i jeśli nas zadowolą to zajmiemy się techniczną stroną realizacji.

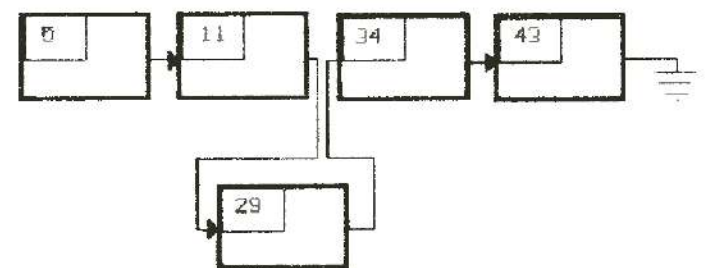
USUWANIE JEDNYM RUCHEM

Spróbujmy zrealizować to co stwarzało nam największe problemy: dopisywanie i usuwanie elementów uporządkowanego ciągu:

Aby dodać nowy rekord musimy zapisać w pamięci jego treść, przy czym możemy to zrobić w dowolnym miejscu.



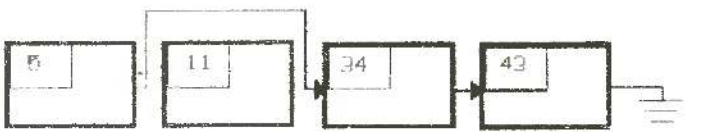
a następnie przeglądamy listę już wpisanych rekordów porównując wartości identyfikatorów z wartością nowo wpisywanego aby ustalić między którymi rekordami trzeba wstawić nowy. Gdy już je znajdziemy po prostu przestawiamy strzałki tak aby nowy rekord wszedł na właściwe miejsce.



Dzięki naszym strzałkom operacja odbyła się pomyślnie bez konieczności przepisywania jakiegokolwiek rekordu z miejsca na miejsce. A co z usuwaniem? Odbędzie się jeszcze prościej wystarczy tylko przesunąć jedną strzałkę. Z ciągu.



chcemy usunąć element o identyfikatorze 11 (czasem zamiast identyfikator mówimy klucz, co jest odpowiednikiem angielskiego terminu key). Trzeba znaleźć właściwy rekord, a następnie ustawić strzałki jak następuje:



A jak szukać danych przy tej metodzie zapisu? Bierzymy pierwszy rekord, sprawdzamy wartość klucza, jeśli jest mówimy to znaleźliśmy, jeśli nie to posługując się wartością strzałki przechodzimy do następnego, sprawdzamy wartość klucza, itd. Aby opisana operacja mogła się zakończyć gdy elementu nie ma musimy mieć możliwość stwierdzenia że doszliśmy do końca. Musimy umieć nadać naszej strzałce pustą wartość, oznaczającą „nie pokazuje na żaden rekord”. Napotkanie takiej wartości będzie oczywiście oznaczać koniec listy. Na naszych rysunkach wartością taką jest



Opisana powyżej (na razie tylko rysunkami) struktura danych to **lista**. Nie jest ona oczywiście moim wynalazkiem, słyszał o niej każdy dobrze wyszkolony informatyk. Listy są bardzo pożyteczne między innymi w sytuacji takiej jak nasza, gdy elementy na których pracujemy pojawiają się i znikają dynamicznie — w czasie pracy.

Jak samo narysowanie schematu listy nie wyczerpuje tematu. Musimy porozmawiać o jej realizacji w praktyce, oraz o problemie gospodarowania wolnym miejscem, ale to wszystko dopiero w następnym numerze.

Andrzej Krul

ILE PŁACIĆ PROGRAMISTOM

Napisales na czyjes zamowienie program liczacy n linii. Zajelo Ci to k godzin czasu, a kazda z nich cenisz a x zl. Rachunek jest prosty: powinienes otrzymac k * x zl. W wielu jednak przypadkach zleceniu dawca chcialby wiedziec, czy podane przez ciebie liczyly sa zgodne z prawda.

Przy tego typu problemach powna pomocą moze byc prosty model matematyczny tego zagadnienia przeffitowany w ksiazce, F. W. Boehma, „Software Engineering Economics”, wylanej przez Prentice Hall w 1981 roku.

Zgodnie z tym modelem, liczba osobomiesiecy potrzebna do napisania programu liczacego n linii wynosi:

$$M = 2.4 * \left[\frac{n}{10.0} \right]^{\alpha}$$

gdzie α jest wspolczynnikiem zależnym od rodzaju programu:
 $\alpha = 1.05$ dla zwyklych programow,
 $\alpha = 1.12$ dla modularnych fragmentow programu,
 $\alpha = 1.20$ dla modyfikacji istniejacego programu.
 Druga wielkoscia charakterujaca opisywany model jest ilosc czasu potrzebna na stworzenie oprogramowania przez zespól ludzi:
 $T = 2.5 * M^{\beta}$
 gdzie β , podobnie jak α , zależy od rodzaju programu:
 $\beta = 0.39, 0.35, 0.32$.

Podzielenie M przez T pozwala odpowiedziec na pytanie ilu programistow nalezy zatrudniec, aby jak najszybciej stworzyc dane oprogramowanie.

Ilustracja przedstawionych wzorow jest ponizszy przyklad. Dla zwyklego programu liczacego 32000 linii otrzymujemy:

$$M = 2.4 * (32)^{1.05} = 91$$

osobomiesiecy.
 $T = 2.5 * (91)^{0.39} = 14$ miesiecy.
 Oznacza to, ze postawione zadanie moze byc wykonane przez zespól 6-7 programistow pracujacych przez 14 miesiecy.

Gdy porownamy wyniki wynikajace z tego modelu z naszymi doswiadczeniami, to czesto okaze sie, ze w swojej pracy uzyskujemy 500-600% „normy”. Niestety, przekonanie o naszej wyzszości nad profesjonalistami, moze byc zluadne. Podane tutaj czasy dotycza calkowicie procesu tworzenia oprogramowania, tzn. nie z wymysleniem algorytmu, napisaniem kodu, jego uruchomieniem i testowaniem. Programy pisane przez amatorow sa czesto niewystarczajaco przetestowane, a przy ich pisaniu nierazko korzystamy z gotowych recept postepowania. Poza tym naprawde trudno jest napisac w warunkach domowych gotowy profesjonalny produkt, który nie padnie przy pierwszej lub drugiej demonstracji, a jednoczesnie bedzie reprezentowal odpowiedni poziom jak sclowy.

Jonasz Mayer

PROGRAMOWAĆ MOŻE KAŻDY

JĘZYK



DLA NAJMŁODSZYCH czyli kolejny wieczór z czarnoksiężnikiem

W poprzednim odcinku poznaliśmy miłego czarnoksiężnika, który znalazł przypadkowo na strychu komputer i podręcznik z czarodziejskimi zaklęciami w języku „C”. Podczas pierwszego wieczoru nauczył się wprowadzać do komputera i wyświetlać na ekranie monitora informacje tekstowe.

Następnego wieczora Czarnoksiężnik ponownie usiadł przy swoim komputerze. Tym razem postanowił nauczyć się wprowadzać i wyświetlać liczby.

W języku „C” liczby dzielimy na dwa rodzaje:

- całkowite — to te, przy pomocy których liczymy samochody, ludzi lub krzesła (także te pożyczone)
- rzeczywiste — to te, przy pomocy których mierzymy odległości między dwoma punktami lub wyrażamy wagę jakiegoś przedmiotu.

Każda liczba zajmuje pewien obszar pamięci komputera. Rozmiar obszaru zajętego przez daną liczbę jest inny dla każdego rodzaju liczb i jest związany z wielkością (zakresem) liczb możliwych do zapamiętania. Jeśli zakres ten jest mały w stosunku do naszych potrzeb, wtedy możemy go rozszerzyć zmieniając rodzaj „całkowita” na „długa” lub „rzeczywista” na „precyzyjna”. Są jeszcze inne możliwości, na początek jednak te informacje nam wystarczą.

Nasz znajomy czarnoksiężnik oglądał niedawno film o piratach. Bohaterowie tego filmu pływali po morzach i często mówili, że przepłynęli ileś mil morskich. My jednak przyzwyczajeni jesteśmy do kilometrów. Czarnoksiężnik postanowił napisać program, który będzie przeliczał mile na kilometry.

Należało w tym celu odwołać się do treści książki. Było w niej napisane:

„Każda liczba, która będzie wczytana do komputera lub obliczona przez program musi być zapamiętana w miejscu pamięci zwanym zmienną. Nazwy wszystkich występujących w programie oraz ich rodzaj muszą być wyszczególnione na początku programu.”

Podobnie jest ze zmiennymi przechowywanymi teksty. Czarnoksiężnik uczył się tego podczas pierwszego wieczoru. Teraz rozpoczął od obmyślenia planu działania programu. Taki plan nazywa się algorytmem.

Jak zapewne wiecie, mila morska ma 1852 metry czyli 1.852 kilometry.

Należy więc:

- 1° wczytać ilość mil
- 2° pomnożyć tę ilość przez skalę 1.852 (w językach programowania używamy kropki dziesiętnej a nie przecinka)
- 3° napisać wynik

Ponieważ chcemy, żeby program działał wielokrotnie ale nie bez końca, musimy wykonać go w pętli. Jednocześnie ustalimy, że pętla zakończy swoje działanie, gdy zerową ilość mił zechcemy zmienić na kilometry.

Czarnoksiężnik usiadł wygodnie w fotelu, wziął do rąk tajemną księgę i zaczął czytać:

„Jednym ze sposobów zapętlenia programu jest wykonanie instrukcji w pętli „powtarzaj”. Ma ona postać:

powtarzaj (start; warunek; zmiana) instr.

W pętli tej najpierw wykona się instrukcja inicjująca „start”, następnie, jeśli „warunek” jest prawdziwy (tzn. „warunek” jest różny od zera) to wykonają się instrukcje „instr” a po niej „zmiana” i ponownie program przechodzi do sprawdzenia „warunku”.

Jeśli chcemy w pętli wykonać kilka instrukcji, wtedy „instr” jest blokiem kilku instrukcji ujętym w nawiasy klamrowe {}. Każda instrukcja musi kończyć się średnikiem;

Trzy elementy w nawiasach () nazwiemy instrukcjami sterującymi zaś „instr” nazwiemy treścią pętli. Każdy z tych elementów może być pusty tzn. może go nie być. Musi pozostać jednak średnik oddzielający poszczególne elementy. Jeśli brak „start” to nie wykona się instrukcja inicjująca. Jeśli brak „warunku” to przyjmujemy, że jest prawdziwy. Jeśli brak jest „zmiany” to znaczy, że pętla ponownie wykona się bez zmiany parametrów sterujących. Najczęściej tak bywa, gdy zmiany tej dokonujemy w treści pętli (czasami tak jest wygodniej). Brak treści pętli „instr” zdarza się wtedy, gdy instrukcje sterujące wykonają wszystkie czynności przewidziane do wykonania w tej pętli. W szczególności instrukcja:

powtarzaj (; ;);
 nie nie wykonuje i jednocześnie pracuje bez końca.

Instrukcje sterujące „start” i „zmiana” mogą składać się z kilku

części. Oddzielamy je od siebie przecinkiem.

Jeśli któryś z elementów instrukcji „powtarzaj” jest zbyt długi i nie mieści się w jednym wierszu to można go napisać w kilku wierszach dzieląc według własnego uznania. W szczególności instrukcja ta może być napisana w formie:

powtarzaj (start;
 warunek;
 zmiana)
 instr.;

Czarnoksiężnik odłożył księgę, po czym usiadł do komputera.

— Najpierw uzupełnię zbiór czarodziejskich zaklęć — pomyślał. Do utworzonego podczas pierwszego wieczoru zbioru <predef.h> dopisał kolejne zaklęcia wybrane z księgi:

- # define rzeczywista float
- # define calkowita int
- # define precyzyjna double
- # define dluga long
- # define powtarzaj for

Teraz zamyslił się głęboko.

— Gdy wczytywałem lub pisałem teksty to w odpowiednim miejscu funkcji „czytaj” lub „pisz” pisałem %s. Co powinienem napisać w przypadku liczb?

Otworzył księgę i czytał: „Funkcje „czytaj” i „pisz” mogą oprócz tekstów obsługiwać także liczby. W przypadku, gdy chcemy wczytać lub napisać liczbę, należy zaznaczyć to w odpowiednim miejscu tekstu zaś po cudzysłowie napisać nazwę zmiennej, gdzie jest lub będzie przechowywana dana wartość. Zaznaczenia dokonujemy w zależności od rodzaju liczby:

%d — wartość jest całkowitą liczbą dziesiętną

%f — wartość jest rzeczywistą liczbą dziesiętną

%c — wartość jest pojedynczym znakiem pisarskim

%s — wartość jest ciągiem znaków pisarskich (tekstem)

Gdy wartość jest liczbą o rozszerzonym zakresie, to odpowiednio przed literami „d” i „f” należy napisać literę „l”.

Oczywiście, w księdze było więcej informacji o formacie danych, dziś jednak poprzestaniemy na tym.

Teraz nasz znajomy zaczął obmyślać szczegóły programu:

— Jeśli chcę w pętli przeliczać mile na kilometry, to instrukcja „start” powinna wczytać ilość mil, „warunek” powinien sprawdzić, czy ta ilość jest różna od zera, „zmiana” powinna porównie wczytać ilość mil a „instr” powinna przeliczać mile na kilometry i wypisać wynik.

— No to do roboty — pomyślał, odłożył księgę i niby pianista, oparł palce na klawiaturze. Na ekranie powstawał tekst programu:

```
<include stdio.h>
<include predef.h>
majster
{
    calkowita mile;
    rzeczywista kilometry;
    powtarzaj (czytaj("%d"), mile);
    mile != 0;
    czytaj("%d", mile);
    {
        kilometry = mile * 1.852;
        pisz("%d" mile = %f kilometry);
    }
}
```

Nowością w tym programie jest zapis

mile != 0;
 który czytamy: „Mile różne od zera”. Symbol 0 oznacza zero (przekreślone dla odróżnienia od litery „O”). Symbol „!=” jest operatorem logicznym, które będą umiowane w następnych odcinkach.

Mieczysław Płacheta

METODY NUMERYCZNE DLA KALKULATORÓW PROGRAMOWANYCH AOS I RPN

I. RÓWNANIA ALGEBRAICZNE. CZĘŚĆ 1.

Wstęp.

Iteracja należy do grupy tych działań w obrębie metod numerycznych, które najskuteczniej przekonują nas o korzyściach płynących z zastosowania kalkulatorów programowanych do wykonywania obliczeń wymagających wielokrotnego stosowania określonej operacji matematycznej. O jej randze świadczy fakt, że jest ona jednym z najważniejszych narzędzi tak praktycznego jak i teoretycznego badania problemów liniowych i nieliniowych. O tych zagadnieniach traktować będzie seria kolejnych artykułów.

Teoria.

Metody iteracyjne wyznaczania zera Σ funkcji $f(x)$ polegają na konstrukcji ciągu kolejnych przybliżeń x_0, x_1, \dots zbieżnego do Σ .
Postać m -- punktowego wzoru iteracyjnego jest następująca:

$$x_{i+1} = g_i(x_i, x_{i-1}, \dots, x_{i-m+1}) \quad i \geq m-1 \quad /1/$$

Jeżeli funkcje g_i -- generujące ciąg /1/ nie zależą od i , wówczas metoda określona wzorem /1/ jest metodą stacjonarną. Jeśli postać funkcji g_i zmienia się od jednej iteracji do drugiej, to metoda jest niestacjonarna.

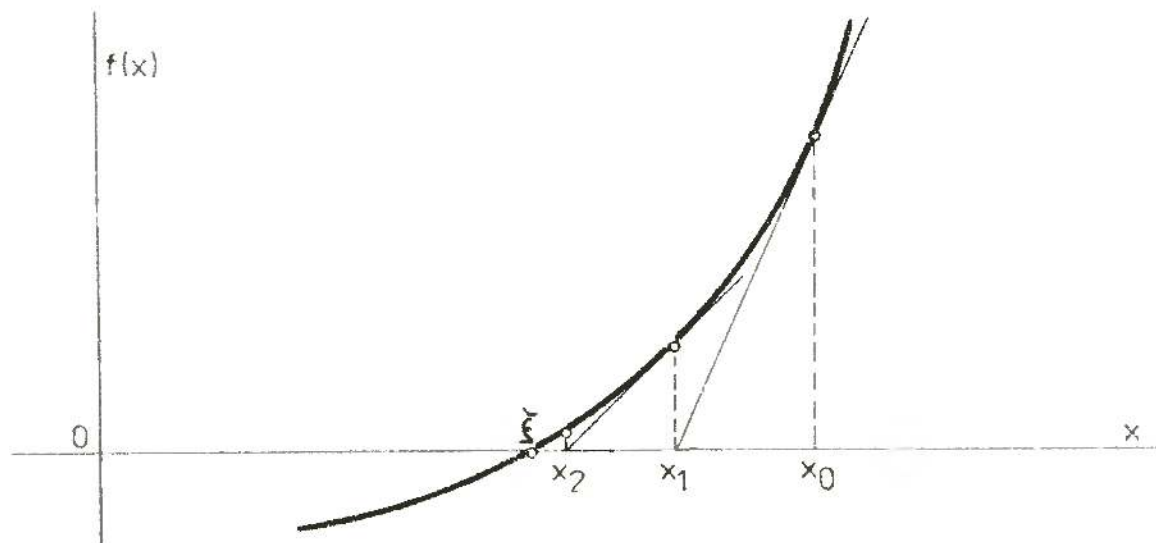
Istnieje kilka metod pozwalających na skuteczne rozwiązanie tego problemu. Do najbardziej znanych należy:

1. Metoda Newtona-Raphsona (stycznych).

Metoda Newtona-Raphsona jest stacjonarną metodą jednopunktową. Aby rozpocząć obliczenia, należy wyznaczyć przybliżenie początkowe x_0 zera Σ . Metoda Newtona-Raphsona będzie zbieżna, tzn. $x_i \rightarrow \Sigma$, jeśli początkowe przybliżenie x_0 będzie dostatecznie bliskie Σ . Posiada ona bardzo prostą interpretację geometryczną. Mianowicie przybliżenie x_{i+1} jest punktem przecięcia osi Ox ze styczną do wykresu funkcji $f(x)$ w punkcie x_i .

Kolejne przybliżenia x_i zera Σ funkcji $f(x)$ oblicza się z wzoru:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad i \geq 0 \quad /2/$$

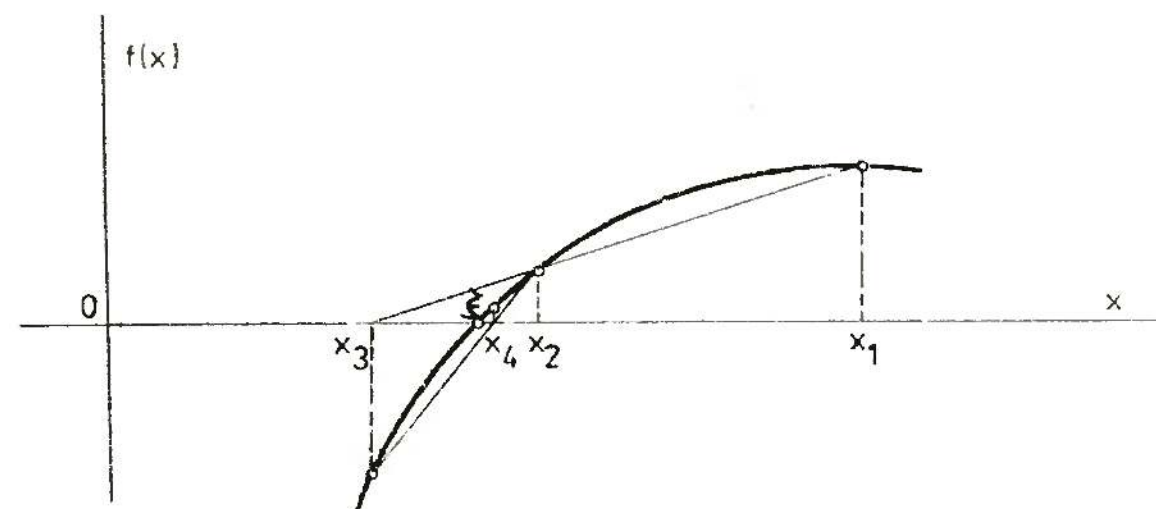


2. Metoda siecznych.

Niech x_0 i x_1 będą dwoma kolejnymi, początkowymi przybliżeniami zera Σ funkcji $f(x)$. Wówczas następne przybliżenie można obliczyć z wzoru:

$$x_{i+2} = x_{i+1} - \frac{x_{i+1} - x_i}{f(x_{i+1}) - f(x_i)} f(x_{i+1}) \quad i \geq 0 \quad /3/$$

Proces ten będzie zbieżny, jeśli x_0 i x_1 będą dostatecznie bliskie zera Σ . Kolejne przybliżenie x_{i+2} jest punktem, w którym sieczna funkcji $f(x)$ poprowadzona w punktach x_i i x_{i+1} , przecina oś Ox . Jest to metoda stacjonarna dwupunktowa.



Interpretacja dla kalkulatorów z Algebraicznym Systemem Operacyjnym AOS™.

Oddajmy pierwszeństwo metodzie Newtona-Raphsona, która jest częściej stosowana w praktyce obliczeniowej. Program umożliwiający obliczenie pierwiastków równania został opracowany na model TI-66 Program mable i przedstawia się następująco:

Krok	Instrukcje
000	LBL A STO 01 0 x↔t CLR R/S
008	LBL B STO 2 R/S
013	LBL D RCL 1 - E : 2nd E = STO 1 2nd Dsz 2 D RCL 1 R/S
030	LBL E

tab. 1

Jego realizacja przebiega w kilku etapach, które charakteryzuje tabela 2.

Kolejność działań	Komentarz
GTO E LRN $f(x)$	Skok bezwarunkowy do etykiety E, wprowadzenie modułu LRN, seria instrukcji klawiszowych wprowadzających funkcję $f(x)$
INV SBR	Instrukcja zamykająca podprogram E
LBL 2nd E $f'(x)$	Skok bezwarunkowy do podprogramu E, seria instrukcji klawiszowych wprowadzających pochodną funkcji $f(x)$
INV SBR LRN	Instrukcja zamykająca podprogram E wyjście z modułu LRN
x_0 A	Wprowadzenie pierwszego przybliżenia pierwiastka równania
i B	Wprowadzenie liczby aproksymacji
D	Wykonanie programu

tab. 2

Przykład:

Obliczyć pierwiastki rzeczywiste funkcji $f(x) = 3x^2 - 3x - 2$, korzystając z metody Newtona-Raphsona.

```
GTO E LRN
( RCL 1 x^2 x 3 - RCL 1 x 3 - 2 )
INV SBR
LBL 2nd E
( RCL 1 x 6 - 3 )
INV SBR LRN
```

Ponieważ nasza funkcja jest wielomianem stopnia drugiego zakładamy, że posiada 2 miejsca zerowe. Po wprowadzeniu do kalkulatora wartości pierwszych aproksymacji pierwiastków równania, kolejne przybliżenia ich rzeczywistych wartości podają tabele poniżej:

0 A		1 A	
1	B	- 0.666666667	1.666666667
2	B	- 0.457607433	1.457607433
3	B	- 0.457607433	1.457607433
4	B	- 0.457427108	1.457607108
5	B	- 0.457427108	1.457607108
.			
.			
10	B	- 0.457427108	1.457607108

Janusz Mitura

Literatura:

- Björck A., Dahlquist G. — Metody numeryczne. PWN, Warszawa 1987.
- Kozarski M., Szurmak Z. — Minikalkulatory w obliczeniach naukowych i technicznych. WNT, Warszawa 1980.
- Praca zbiorowa — Poradnik Inżyniera. Matematyka. T.2. WNT, Warszawa 1987.
- Texas Instruments Inc. — Sourcebook for programmable calculators. 1978.
- Texas Instruments Inc. — TI-66 Programmable. Manual. 1983.



ABC Data

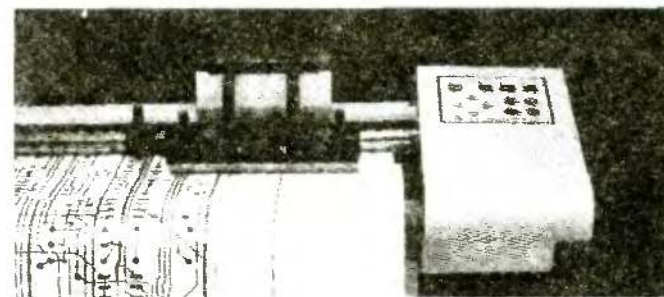


proponuje

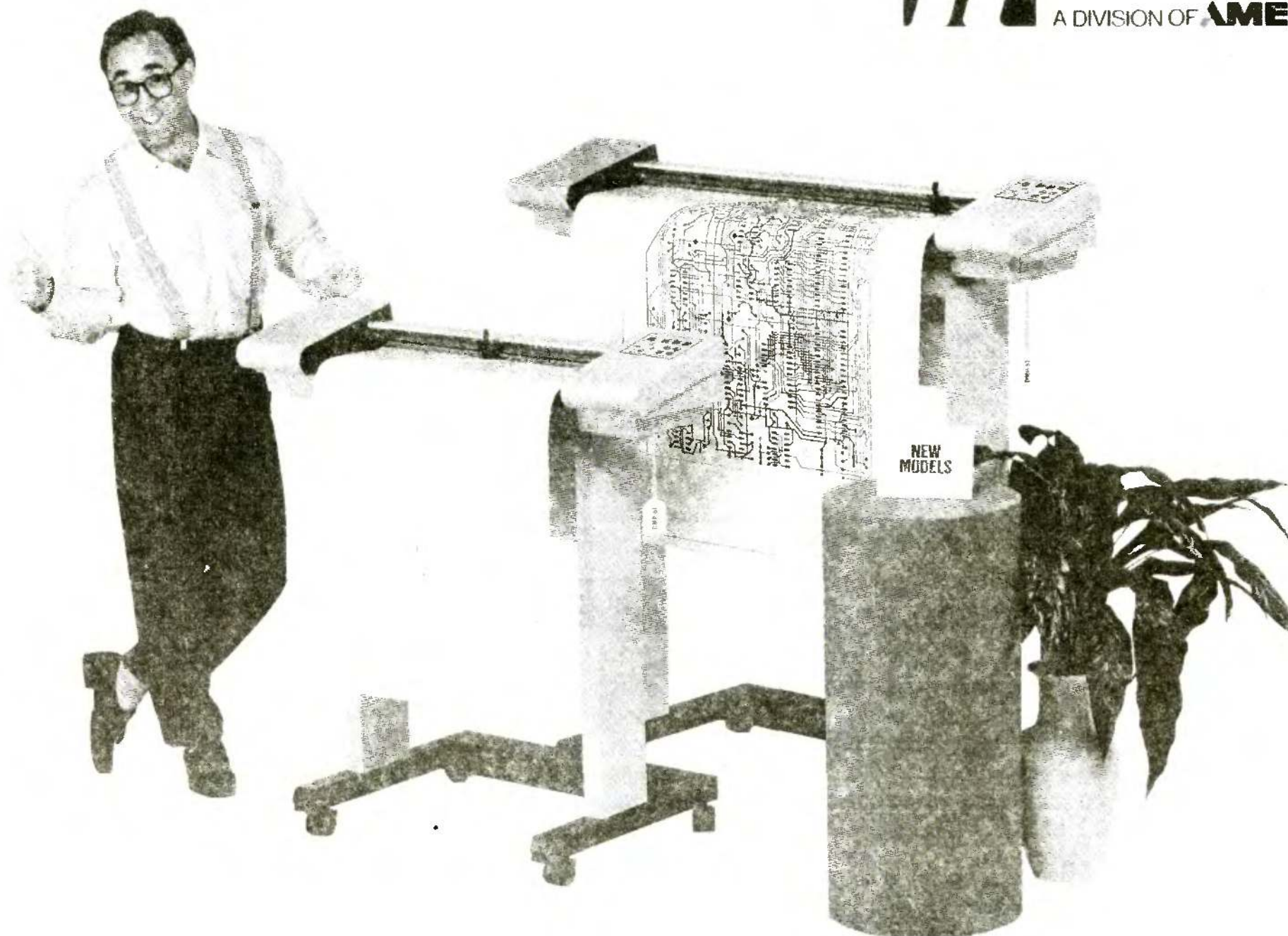
- * plotery tablicowe
- * wielkoformatowe plotery bębnowe
- * digitizery najnowszej generacji
firmy Houston Instrument

Ditmar-Koel-Str. 13
2000 Hamburg 11
tel. (040) 31 40 03
tel. (040) 319 5874
tlx. 21 66 002
fax. (040) 31 91 783

- * **atrakcyjne ceny**
- * **szybkie dostawy**
- * **gwarancja jakości**



**HOUSTON
INSTRUMENT**
A DIVISION OF **AMETEK**



PROGRAMOWAĆ MOŻE KAŻDY

WYSZUKIWANIE, PORZĄDKOWANIE I SORTOWANIE

W ciągu ostatnich kilku tygodni parę osób pytało mnie o efektywne algorytmy sortowania. Problem może być błahy jeśli chodzi o uporządkowanie np. w kolejności rosnącej co najwyżej kilkudziesięciu danych numerycznych.

Kłopoty pojawiają się gdy mamy do czynienia ze zbiorami danych (ang. files) o wielkości prawie takiej jak pamięć operacyjna naszej maszyny. Tu już nasze „prywatne” algorytmy nie wystarczają i trzeba sięgnąć po literaturę fachową, w której też nie zawsze potrafimy znaleźć algorytm najefektywniejszy czyli najszybszy. Do momentu, w którym się nie zainteresowałem tym problemem uważałem, że najszybszymi algorytmami są metoda Shell'a i Bąbelkowa (ang. Bubblesorting) wraz z mutacjami. Jakież było moje zdumienie gdy po dotarciu do odpowiednich źródeł, oprogramowaniu, uruchomieniu i przetestowaniu kilku nowych algorytmów okazało się, że metody te są wolne jak żółw w porównaniu z metodami tzw. HeapSorting czy QuickSorting. Zaczniemy jednak od początku.

Najprostszą postacią struktury charakterystyczną dla danych są tak zwane zbiory danych. Zbiór danych składa się z szeregu zapisów (ang. records), które z kolei mogą składać się z pewnej liczby pól (ang. fields). Przykładami zbiorów danych w życiu codziennym może być kartoteka zawierająca w sposób usystematyzowany nazwiska osób, ich adresy i inne dane lub też słownik, w którym każdemu hasłu odpowiada jakiś zapis wyjaśniający.

Jednym z najprostszych procesów, jakie dają się określić w odniesieniu do zbiorów danych jest proces wyszukania danych.

Problem wyszukiwania powstaje wtedy gdy trzeba zidentyfikować zapis za pomocą pewnego klucza (pewnej części identyfikowanego zapisu). Przyjmuje się, że różne zapisy mają różne klucze i wszystkie odstępstwa od tej zasady muszą być wyraźnie sygnalizowane. Wyszukiwanie jest więc procesem lokalizacji pewnego zapisu w obrębie danego zbioru na podstawie klucza tego zapisu np. w zbiorze nazwisk i adresów mamy znaleźć pełny zapis odpowiadający jakiemuś nazwisku. Wyszukiwanie określa się jako pomyślne, gdy zapis poszukiwany został odnaleziony, w przeciwnym wypadku określa się je jako niepomyślne. Aby dać jakieś wyobrażenie o wyszukiwaniu przedstawię dwie metody:

— wyszukiwanie proste — Przy zastosowaniu tej metody przyjmuje się, że zapisy umieszczone są w pamięci w sposób zwarty, tworząc wektor lub tablicę, przy czym porządek w jakim występują klucze zapisów nie jest z góry znany. W takiej sytuacji można się posłużyć jedynie metodą wyszukiwania prostego: zapisy pobiera się ze zbioru pojedynczo w pewnej określonej kolejności; w każdym pobranym zapisie sprawdza się, czy jego klucz zgadza się z kluczem zapisu poszukiwanego; jeśli nie ma zgodności proces prowadzi się dalej aż do wyczerpania całego zbioru zapisów. Zaletami tej metody jest oprócz jej niezwyklej prostoty możliwość rozbudowywania zbioru danych przez dopisywanie. Można też w sposób bezkonfliktowy skreślać określone zapisy jeśli wyszukiwanie jest zorganizowane w ten sposób, że zapisy puste są pomijane. Wadą tej metody jest jej powolność. Liczba cykli wyszukiwania (ilość porównań zapisu z kluczem) wynosi 1, jeśli od razu trafia się na zapis poszukiwany, oraz wynosi N (liczba zapisów w zbiorze danych), gdy poszukiwany zapis znajdował się na końcu zbioru. Tak więc średnia liczba cykli wyszukiwania dla pomyślnego wyniku wyszukiwania jest wartością podanych wyżej liczb.

Średnia liczba cykli w wyszukiwaniu prostym = $(1+N)/2$. Gdy N jest małe lub gdy powolność algorytmu może być tolerowana wskazane jest użycie tego algorytmu ze względu na jego logiczną prostotę.

— wyszukiwanie binarne — jest to ważny przykład wyszukiwania dychotomicznego. Rozumie się przez nie proces, w którym każdy cykl wyszukiwania łączy się z podziałem zbioru lub tej jego części, która może zawierać poszukiwany zapis, na dwie części, po czym jedna z tych części może być odrzucona. Przy zastosowaniu tej metody zapisy można przechowywać w pamięci w postaci zwartej tablicy, która powinna być uporządkowana według kluczy zapisów. Zakłada się więc, że na zbiorze kluczy jest określona pewna relacja porządkująca, pozwalająca bezpośrednio określić dla każdego dwóch różnych kluczy, który z nich jest wcześniejszy w danym zbiorze. Każdy cykl wyszukiwania binarnego pociąga za sobą zmniejszenie części zbioru, w której może znajdować się poszukiwany zapis o połowę w stosunku do części poprzedniej. Objasnia to pochodzenie słowa „binarna”. Średnią liczbę cykli wymaganych dla pomyślnego wyniku wyszukiwania można w przybliżeniu otrzymać w następujący sposób: ponieważ każdy cykl zmniejsza zbiór o połowę to po wykonaniu n cykli pozostaje w zbiorze $N \cdot 2^{(-n)}$ zapisów. Wyszukiwanie kończymy gdy pozostaje nam tylko jeden zapis ten poszukiwany. Osiąga się to po n cyklach przy czym n oznacza najmniejszą liczbę całkowitą, która spełnia nierówność $N \cdot 2^{(-n)} = 1$. Stąd otrzymujemy, że: $\log N \leq \text{liczba cykli wyszukiwania binarnego} \leq 1 + \log N$. \log jest logarytmem o podstawie równej 2. Ponieważ przy szybkim wzroście N logarytm rośnie powoli widać z tego, że dla dużych N wyszukiwanie binarne jest bardziej ekonomiczne niż wyszukiwanie proste (oczywiście jeśli pomija się koszt utrzymania zbiorów w stanie uporządkowanym).

Kolejnym procesem odnoszącym się do zbiorów danych jest porządkowanie wewnętrzne, czyli sortowanie. Porządkowanie wewnętrzne zbioru jest takim przemieszczeniem zapisów, w wyniku którego klucze następują po sobie w pewnej ustalonej kolejności. Jednym z zastosowań porządkowania jest przygotowanie zbioru do procesu wyszukiwania binarnego. Takie przekształcenie zbioru często nazywa się sortowaniem, a nie porządkowaniem. Sortowanie wymaga, na ogół przemieszczenia zapisów, przy czym każdy zapis może kilkakrotnie zmienić swoje miejsce, zanim nie zostanie osiągnięte wymagane uporządkowanie zbioru. Dla realizacji procesu sortowania są potrzebne tylko klucze sortowania, które pozwalają określić sposób postępowania w tym procesie. W przypadkach, w których klucz stanowi stosunkowo niewielką część dużego zapisu, można zaoszczędzić sporo czasu, gdy podczas sortowania traktuje się klucze nie jako elementy pełnych zapisów lecz jako składniki nazwijmy to skorowidza, zawierającego odniesienia do pełnych zapisów.

Jak już wspominałem na wstępie, istnieje wiele użytecznych metod sortowania. Najlepsze spośród tych metod są na ogół jednocześnie bardziej ekonomiczne choć często osiąga się to kosztem wielkości potrzebnej pamięci i złożoności logicznej algorytmu. Dlatego też określenie miary jakości przy systematycznym porównywaniu metod napotyka na trudności. Wybór „najlepszej” metody sortowania zbioru zależy od dostępnej informacji odnoszącej się do stopnia uporządkowania zbioru, co stanowi dodatkową komplikację. Oczywiście efektywność algorytmów zależy od złożoności zbioru danych, od tego czy sortowaniu podlegają liczby czy też teksty (problem reprezentacji i interpretacji tychże w komputerach) itp.

Poniżej chciałem przedstawić 7 algorytmów sortowania. Moim zdaniem są to najlepsze algorytmy o prostej budowie, a więc umożliwiające

korzystanie z nich na mikrokomputerach. Algorytmy przedstawiam jako procedury (o nazwach algorytmu sortującego) napisane w Pascal-u. Poniżej wszystkich algorytmów przedstawiam test szybkości tychże. Test został przeprowadzony na komputerze Commodore 128 dla różnej wielkości zbiorów:

100, 200, 300, 400, 500, 1000, 2500, 5000, 10000, 30000 elementów (wielkość zbioru przechowywana jest w zmiennej NumItems).

Sortowaniu podlegały liczby xx z przedziału (0,9999) generowane w sposób losowy i przechowywane w wektorze Sort-Array-Type. Wektor ten definiujemy w programie głównym w następujący sposób:

Type Sort-Array-Type = Array [0..NumItems] of integer; Wynik czasowy dla danego zbioru jest średnią arytmetyczną czasów uzyskiwanych przy 3 próbach sortowania (chyba, że czas ten przewyższał 2 godziny, wtedy bowiem program sterujący przechodził do testowania następnego algorytmu). Wynik testu podawany jest w formacie gg/mm/ss czyli godziny/minuty/sekundy.

```
Procedure BubbleSort (Var A : Sort-Array-Type;
Var I,J,N,X3 : Integer);
Begin
```

```
  For I := 2 to NumItems do
    For J := NumItems DownTo I do
      If A[I-1] > A[J] then
        Begin
          X3 := A[I-1];
          A[I-1] := A[J];
          A[J] := X3;
        End;
  End;
```

```
Procedure ShakerSort (Var A : Sort-Array-Type;
Var I,K,L,R,X3 : Integer);
Begin
```

```
  L := 1; R := NumItems; K := NumItems;
  Repeat
    For J := R DownTo L do
      If A[J-1] > A[J] then
        Begin
          X3 := A[J-1];
          A[J-1] := A[J];
          A[J] := X3;
          K := J;
        End;
    L := K+1;
    For J := L To R do
      If A[J-1] > A[J] then
        Begin
          X3 := A[J-1];
          A[J-1] := A[J];
          A[J] := X3;
          K := J;
        End;
    R := R-1;
  Until L > R;
End;
```

```
Procedure InsertionSort (Var A : Sort-Array-Type;
Var I,J,N : Integer);
Begin
```

```
  For I := 2 To NumItems do
    Begin
      X := A[I]; A[I] := 0; J := I-1;
      While X < A[J] do
        Begin
          A[J+1] := A[J];
          J := J-1;
        End;
      A[J+1] := X;
    End;
End;
```

```
Procedure BucketSort (Var A : Sort-Array-Type;
Var I,J,K,N,BL,2 : Integer);
Begin
```

```
  For I := 1 to NumItems do
    Begin
      X := A[I]; J := 1; K := 0;
      While J <= R do
        Begin
          If X <= A[J] then
            R := R-1;
          Else J := R+1;
        End;
      For J := 1 to R do
        Begin
          A[J+1] := A[J];
          A[J] := X;
        End;
      End;
    End;
```

```
Procedure SelectTopSort (Var A : Sort-Array-Type;
Var I,J,K,X : Integer);
Begin
```




Dla dwóch ostatnich algorytmów dla N=10000 otrzymałem odpowiednio 359 i 148 a dla N=30000 odpowiednio 1340 i 618.

Dla tak dużych zbiorów pozostałe algorytmy nie dawały rezultatu nawet po 6 godzinach.

Przykładowy odczyt dla N=5000 i algorytmu SelectionSort: 1 godzina, 55 minut, 10 sekund.

Przykładowy odczyt dla N=100 i algorytmu BubbleSort: 6 sekund.

Dominik Falkowski

```
Begin
  K := 1; N := A[1];
  For J := 1+1 To NumItems do
    If A[J] < N Then
      Begin
        K := J; N := A[J];
      End;
  A[N] := A[K];
  A[K] := N;
End;
```

```
Procedure HeapSort (Var A : Sort-Array-Type;
  Var L,R,K,X3 : Integer);
```

```
Procedure Shift;
Label 13;
Var I,J : Integer;
Begin
  I := L; J := 2*I; N := A[I];
  While J <= R do
    Begin
      If J < R Then
        If A[J] < A[J+1] Then
          J := J+1;
      If N > A[J] Then
        GoTo 13;
      A[I] := A[J];
      I := J;
      J := 2*I;
    End;
  13: A[I] := N;
End;
```

```
13: A[I] := N;
End;
```

```
Begin
  L := (NumItems Div 2) + 1; R := NumItems;
  While L > 1 do
    Begin
      L := L-1; Shift;
    End;
  While R > 1 do
    Begin
      X3 := A[L];
      A[L] := A[R];
      A[R] := X3;
      R := R-1;
    End;
End;
```

```
Procedure QuickSort (Var A : Sort-Array-Type);
Procedure Sort (L,R : Integer);
  Var I,J,K,X3 : Integer;
```

```
Begin
  I := L; J := R;
  X3 := (A[L]+A[R]) Div 2;
  Repeat
    While A[I] < X3 do
      I := I+1;
    While X3 < A[J] do
      J := J-1;
    If I <= J Then
      Begin
        X3 := A[I];
        A[I] := A[J];
        A[J] := X3;
        I := I+1;
        J := J-1;
      End;
  Until I > J;
  If L < J Then
    Sort (L,J);
  If I < R Then
    Sort (I,R);
End;
```

```
Begin
  Sort (1,NumItems);
End;
```

Algorytm	N	1000	2000	3000	4000	5000	10000	25000	50000
BubbleSort	1	6	1	241	5411371234110121	---	---	---	---
ShakerSort	1	5	1	221	48112512181	9881	---	---	---
InsertionSort	1	2	1	81	171	311	501	317119531132161	---
BinaryInsertionSort	2	1	81	181	321	561	369119131115371	---	---
SelectionSort	1	3	1	111	251	4511101	439128531155101	---	---
HeapSort	1	1	1	21	41	61	81	171	501
QuickSort	10.51	11	21	31	41	81	251	561	---

INTERSOFT

00-443 Warszawa ul. Górnośląska 9/11 tel 21-56-08 tlx 81-72-45

Programy Edukacyjne

	Cena	ZX SPECTRUM	AMSTRAD CPC	ATARI XL/XE
1. DIODA-1 Animowany film ilustrujący zjawiska na łęczy p.n.	19.000		X	
2. ELEKTRONIKA Ilustrowany test z elektroniki, uczyć podstawowych pojęć i symboli elektronicznych.	10.000		X	
3. GRAWITACJA Symulacja wzajemnego oddziaływania dwóch ciał w ich polu grawitacyjnym.	8.000		X	
4. KINEMATYKA 2 Pakiet trzech programów na temat pola grawitacyjnego.	8.000		X	
5. SOCZEWKI Powstawanie obrazów rzeczywistych i pozornych w soczewkach wklęsłych i wypukłych.	5.000		X	
6. FUNKCJA LINIOWA Własności wykresu funkcji liniowej oraz wpływ współczynników na wykres.	6.000		X	
7. PRZEKROJE Graficzne przedstawienie przekrojów brył geometrycznych. Projektowanie własnych brył.	8.000	X	X	
8. STOLICE EUROPY Gra dydaktyczna ucząca położenia stolic państw Europy.	11.000		X	
9. GRAMATYKA 1 Ćwiczenia z logicznego rozbioru zdań.	7.000		X	
10. JĘZYK ANGIELSKI Tłumaczenie i wymowa zdań w oparciu o podręcznik ŁSżubitko.	8.000	X	X	
11. KRÓLOWIE POLSKI Mapka kojarzenia faktów historycznych z okresami panowania poszczególnych królów Polski.	8.000		X	

JAK KUPIWAĆ PROGRAMY? - Indywidualnie: za gotówkę lub za zaliczeniem pocztowym - instytucje: za gotówkę lub czekiem. Ceny programów nie zawierają ceny nośników. Na życzenie zamówienia realizujemy na nośnikach klientowi. Pełen katalog wraz z krótkimi opisami wysyłamy gratis.

INTERSOFT

SB-19

ATARES

Spółka z o.o.
Chorzów, ul. Jesionowa 3
tel. 465-719

oferuje:
użytkownikom IBM usługi programistyczne na zamówienie zgodnie z indywidualnymi wymogami klienta.
Użytkownikom Atari najnowsze rozwiązania sprzętowo-programistyczne jak:
kasetowy system transmisji „BLIZZARD TURBO” 6000 bodów (magnetofon po przeróbce czyta 10x szybciej, praktycznie bez błędów), system digitalizacji dźwięku „CRYSTAL SOUND”, cartridge do obsługi systemu „BLIZZARD TURBO”, TOP DRIVE 1050 (format 180 KB, transmisja 70 KB), MAXI TURBO DRIVE (kopiuje wszystkie zabezpieczenia, zabezpiecza wszystko).
ATARI 256 DUAL SYSTEM (umożliwia pracę równoczesną z kilkoma programami-systemami), rozszerzenia pamięci ATARI, naprawę sprzętu ATARI najnowsze oprogramowania ATARI, COMMODORE, SPECTRUM.
Firma prowadzi skup — sprzedaż komputerów domowych, sprzętu RTV, video itp. K-106

JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Biuro Reklamy „Sztandaru Młodych”, Warszawa 1, skr. pocztowa 363, ul. Wspólna 61, tel. 28-02-56

Cena reklamy czarno-białej wynosi 750 zł za cm². Do ceny podstawowej doliczane jest 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej. Ogłoszenie drobne kosztuje 450 zł za 1 słowo. Wyżej wymienione kwoty można również wpłacić bezpośrednio na konto MAW: Państwowy Bank Kredytowy w Warszawie, III Oddział Warszawa nr 370015-5757.

W miejscu na korespondencję prosimy zaznaczyć, że wpłata dotyczy ogłoszenia w Bajtku. Ogłoszenia drobne oddawane są do druku po otrzymaniu dowodu wpłaty z NBP.
Ceny te obowiązują od stycznia 1989 roku.

Wymienię programy dla SPECTRUM, ATARI, COMMODORE.
Janusz Wałaszek, skrytka pocztowa 1, 33-106 Tarnów 8.
(SB 25)

SERWIS KOMPUTERÓW TEST

40-164 Katowice ul. Modrze-wiowa 24/33
poleca naprawy:
ATARI 600, 800, 65, 130, XL, XE
COMODORE 16, 116, +4, 64, 128
DISC DRIVE 1541, 1570, 1571, 1050
MAGNETOFONY COMODORE
DRUKARKI
godz. 9-11, 16-18.
(SB 30)

AGENCJA WYSYŁKOWA ELEKTRONICZNA A.S. SERVICE 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)

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
- stacja dysków 5,25"
- doskonałe, krajowej produkcji monitory wielosystemowe do ST (niskiej, średniej, wysokiej rozdzielczości) w cenie 299.000,- zł
- sprzęt mikrokomputerowy COMMODORE

ZADZWOŃ, NAPISZ, PRZYJEDŹ I ZOBACZ

(SB-49)

SPECTRUM, TIMEX programy. Adam Gruzlewicz, ul. Działkowa 15, 96-300 Żyrardów.
(G-51)

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



Impreza, na której winienes być obecny

INFORMACJA '89



10 - 14 październik '89

Hala Widowiskowo-Sportowa "Spodek"-Katowice

- Informatyka w zarządzaniu - INFO '89
- Informatyka w medycynie - INFOMED
- Międzynarodowe sesje z udziałem Stowarzyszenia Dziennikarzy Nauki i Techniki
- Seminaria o tematyce: informatyka audio-video telewizja satelitarna

Jeśli chcesz być obecny wytnij poniższy kupon i wyślij

INF '89

Jestem zainteresowany:.....stoisko pokazowe
stoisko informacyjne

Nazwisko.....

Stanowisko.....

Firma.....

Dziedzina zainteresowań.....

Adres.....

.....

telefon..... telex.....

Adresat:

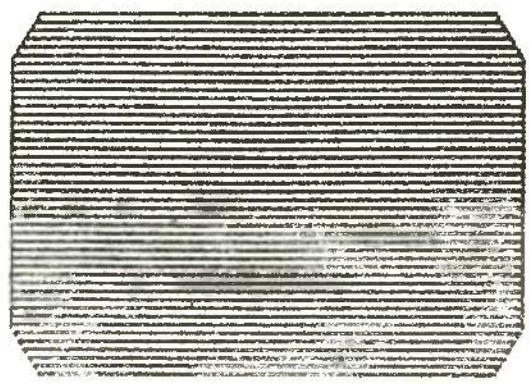
Janusz Gołuch

"PRO - INFO"

40-001 Katowice 1 skr. poczt. 1347

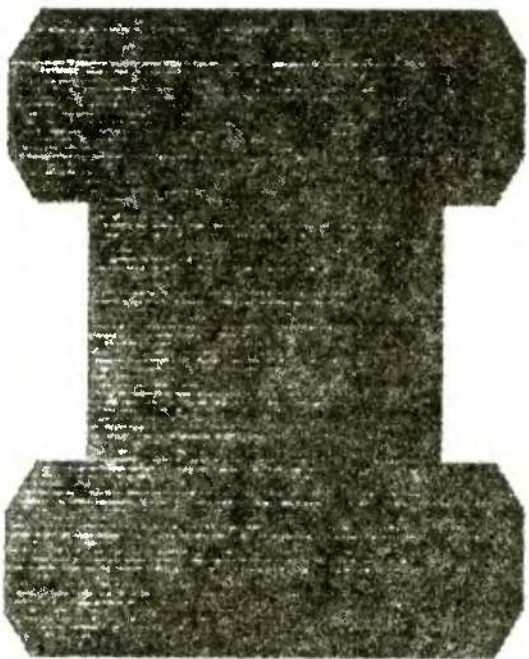
tel. (832) 53-42-88

tlx. 312401 info pl



Let your firm not be absent at

INFORMACJA '89



10 - 14 october 1989

Show Hall "Spodek", Katowice, Poland

All the firms electronics or informatics and willing to promote and present their products in Poland are welcome to participate in our Fair.

Clearly then, the Show Hall "Spodek" in October 89 is the place to be at if new business is your goal. For further information please complete and return the coupon or telephone (832) 5342-88 or telex 312401 info pl.

Now we'd like to hear from you.....

I am interested in:exhibition stand
.....information stand

Name.....

Position.....

Company.....

Business Category.....

Address.....

.....

Telephone.....Telex.....

to: Janusz Goluch
"PRO-INFO" Co. Ltd.
40-001 Katowice 1 P.O.Box 1347
tel. (832)5342-88 tlx. 312401 info pl.

CENY NA DZIEŃ 22.07.89

	Gielda Bajtka tys. zł	Sklep Bajtka tys. zł	Komis tys. zł	Pewex \$	RFN DM	Wik. Brytania #	CSH i inne tys. zł
SINCLAIR							
ZX 81	100	—	—	—	—	—	—
ZX Spectrum 48	380	350	400	115	80	40	—
ZX Spectrum +	420	380	440	—	90	55	—
Timex 2048	430	—	470	146	—	—	300
ZX Spectrum 128+	—	—	—	—	—	85	—
ZX Spectrum 128+2	—	—	—	—	140	100	—
ZX Spectrum 128+3	710	—	—	—	280	150	—
drukarka Seikosha GP 50s	—	250	—	—	—	—	—
Interface Kempston	15	10-15	—	—	10	9	11

COMMODORE

Commodore 64	700	720	740	200	290	120	—
VC 20	—	200	—	—	—	—	—
C 16	200	300	—	—	80	—	—
C 116	250	280	—	—	70	40	—
C Plus 4	350	400	450	—	150	—	—
C 128	1000	1000	—	299	399	170	—
C 128 D	2000	—	—	—	820	250	—
Amiga 500	2500	2900	—	450	899	280	—
Magnetofon 1531	130	130	180	48	30	49	—
Stacja dyskóv 1541	580	700	—	—	355	—	—
Stacja dyskóv 1571	820	800	—	299	460	160	—
Drukarka MPS 803	550	600	—	—	260	100	—

ATARI

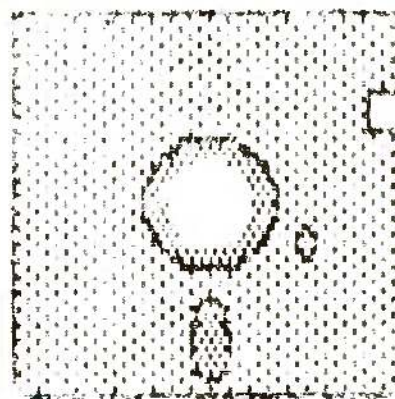
Atari 800 XL	440	500	550	—	—	—	—
Atari 65 XE	450	510	570	114	100	80	—
Atari 130 XE	550	700	—	199	220	110	—
Atari 520 ST	1800	2000	—	—	—	260	—
Atari 1040 ST	—	—	—	—	1140	499	—
Magnetofon XC 12	130	140	—	36	40	42	—
Stacja dyskóv 1050	700	700	770	185	300	100	—
Stacja dyskóv 520 STM	—	500	—	—	—	120	—
Drukarka 1029	500	650	—	—	—	—	—

AMSTRAD

Amstrad 464 mono.	750	700	—	—	350	150	—
Amstrad 664 mono.	—	900	—	—	—	—	—
Amstrad 6128 mono.	1200	—	—	—	670	220	—
Amstrad PCW 8256	—	—	—	—	—	300	—
Amstrad PCW 8512	—	—	—	—	—	390	—
Amstrad PCW 9512	—	—	—	—	—	450	—
Stacja dyskóv do 464	—	—	—	—	380	100	—

SHARP

Sharp MZ 700	—	400	—	—	—	—	—
Sharp MZ 800	—	450	—	—	—	—	—
Dyskietki 5.25 cala	2.5	2-2.5	3	1	0.7	0.5	4-9
Dyskietki 3.5 cala	8	7.5-9	8-10	—	5	1-2	9-10
Dyskietki 3 cale	12	—	—	—	6	2	12
Joystick	25	16-30	20-40	5	10	5	18
Monitor Neptun	70	80	—	—	—	—	70



INDYWIDUALNY
BANK
DANYCH

Rafał i Jakub Palejczyk nawiążą kontakt z posiadaczami mikrokomputera Timex 2048 w celu wymiany doświadczeń i oprogramowania. Adres: 66-400 Gorzów. ul. Kołłątaja 56 m 8.

Agata Kujath, lat 10. Posiada Atari 800 XL, magnetofon XC 12, monitor Neptun 156. Oprogramowanie: około 140 gier oraz programy edukacyjne. Proponuje wymianę gier. Adres: 71-302 Szczecin, ul. Wrońskiego 8/1.

Dariusz Łukasz, lat 17. Posiada Atari 65 XE, stację dyskóv 1050 (TURBO), magnetofon XC 12 (BLIZZARD), około 800 gier i programów użytkowych. Pragnie nawiązać korespondencję z posiadaczami Atari w celu wymiany oprogramowania, doświadczeń i literatury. Adres: 43-346 Bielsko-Biała, ul. Biedronki 18.

Sławek Przybylski, lat 16. Posiada Timex 2048 oraz około 250 programów. Zainteresowania: informatyka. Proponuje wymianę programów i doświadczeń. Specjalizacja programy kopiujące. Adres: 71-401 Szczecin, ul. Wyzwolenia 75a/62.

Cezary Konieczny, lat 26. Posiada Schneidera CPC 6128, stację dyskóv 5 1/4 firmy Vortex (F 1X). Oprogramowanie kilkakset różnych programów, m.in. własne programy projektowania i instalacji własnych znaków na drukarki zgodne z Epsonem (9 igłowe) w tym draft i N.L.Q.

Pragnie wymienić oprogramowanie oraz zachodnie czasopisma poświęcone Amstradowi. Adres: 61-418 Poznań, ul. Żerkowska 37.

Konrad Kwiatkowski, lat 13. Posiada ZX Spectrum +. Proponuje wymianę doświadczeń, oprogramowania oraz opisów gier. Adres: 09-402 Płock, ul. Szopena 61 m 33

Tomasz Gul, lat 13. Posiada mikrokomputer Atari 65 XE, stację dyskóv LDW 2000. Oprogramowanie: ponad 500 gier oraz około 100 programów użytkowych. Proponuje wymianę literatury oraz oprogramowania. Adres: 36-100 Kolbuszowa Dolna 11.

Marek Zydorczak, posiada Spectrum 48KB, monitor Neptun 156B, magnetofon, interfejs Kempston oraz Sinclair 1 i 2, Cursor i joystick. Posiada około 800 programów w tym programy użytkowe. Interesuje się piłką nożną, koszykówką oraz informatyką. Proponuje wymianę oprogramowania i doświadczeń z użytkownikami Spectrum. Adres: 41-706 Ruda Śląska, ul. Kosynierów 4c/7.

Mirosław Winczewski, lat 51. Posiada ZX Spectrum + oraz około 500 programów i gier. Proponuje wymianę oprogramowania. Adres: 44-100 Gliwice, ul. Bolesława Śmiałego 4 m 1.

Stanisław Bejko, prosí o kontakt posiadaczy mikrokomputera ZX 81, poszukuje literatury oraz oprogramowania. Adres: 67-100 Nowa Sól. ul. Przyszłości 3c/38.

Paweł Borek, lat 17. Posiada mikrokomputer Commodore C-64, stację dyskóv 1541, monitor KOLOR BHC. Nawiąże kontakt w celu wymiany programów oraz literatury, posiada dużą ilość literatury w języku angielskim. Adres: 55-200 Oława, ul. Chrobrego 76/10.

ATARI ZX SPECTRUM

INSTRUKCJE • OPISY LITERATURA

Szkoły i Kluby-zniżka
Katalogi-gratis
Co piąty program-gratis
Wysyłka na cały kraj
Wypożyczalnia programów
D.H Sezam Ilp godz. 16-19
00-849 Warszawa UPT 66, skr.p. 14

D-81/82

Na listy czytelników odpowiada
Dominik Falkowski

Jestem uczniem II klasy L.O. w Kwidzynie. Interesuję się historią, głównie II wojny światowej i dlatego moją pasją są gry strategiczne dotyczące tego okresu. Poznałem na razie dwie, tzn. „Arnhem” oraz „Desert Rats” na ZX Spectrum. Chciałbym prosić o informację na temat innych gier tego typu. Jaka jest możliwość dotarcia do większej ilości takich gier i oryginalnych instrukcji do nich (...)?

Adam Nosko, Kwidzyn

Powstało bardzo wiele gier strategicznych, gdyż temat wojny i strategii pasjonuje w równym stopniu, jak programowanie. Oto mini-lista najpopularniejszych gier tego typu, opracowanych na ZX Spectrum: D Day (ładowanie w Arnhem), Battle of Britain (bitwa o Anglię), Falklands (konflikt falklandzki), Theatre Europe (wizja III wojny światowej), Operation Hornumt (bliżej nieznaną operacją), Their Finest Hours (ostatnie chwile Niemców w Europie).

Jeśli chodzi o ich zdobywanie, to jedynym źródłem jest — niestety — giełda i kolekcje. Brak jest bowiem licencjonowanych kopii gier zachodnich, zaś polskie gry w zasadzie nie powstają. Oczywiście nabijanie kiesy handlarzom tylko dlatego, że „oni mają, a ja nie” leży tu na granicy prawa i bezprawia, ale jak na razie nic nie udało się zrobić. Podobnie ma się sprawa z instrukcjami do gier; w przypadku zakupu legalnego (oczywiście nie w Polsce) do gry dołączana jest instrukcja. A my, jak na razie, musimy ratować się również giełdą, no i oczywiście „Bajtkiem”.

Od kilku miesięcy posiadam mikrokomputer TIMEX 204B. Jestem z niego zadowolony i nie zamienilibym się na żaden inny. Poznałem już trochę jego możliwości, lecz na niektóre pytania wciąż nie mogę znaleźć odpowiedzi (...).

Mam sporo programów nagranych w systemie TURBO-COPY autorstwa Tadeusza Wilczka. Interesuje mnie, w jaki sposób można je wprowadzić do pamięci, ponieważ podczas wczytywania z magnetofonu ani jeden program nie jest przez mikrokomputer przyjmowany. (...) Czy mogę bez obaw podłączyć interface Kempston na dwa joysticki do TIMEX-a? (...) Czy można uzyskać dźwięk z komputera przez głośnik telewizora? (...) Czy można programowo wyłączyć głośnik w TIMEX-ie, by w nocy nie przeszkadza domownikom w spaniu? (...) Czy wyjdzie choć jeden numer specjalny „Tyłko o Spectrum”? (...)

Zbigniew Fronckiewicz, Poznań

COPY-COPY komendą LOAD AT 65000, po czym uruchomić komendą USR 65017. Od tej chwili wszystkie odczyty i zapisy na (z COPY COPY odbywały się będą w systemie TURBO. Prędkość transmisji ustala się komendą PAUSE n, gdzie n jest liczbą od 0 do 0/0 — najwolniej, 9 — najszybciej).

W ten sposób możesz wczytać swoje programy na COPY COPY i po zmianie prędkości zapisu nagrać je na taśmę w celu późniejszego uruchomienia. W przypadku korzystania z dobrego magnetofonu, można wczytywać na czystą pamięć programy nawet 3 razy przyspieszone. System TUR-

Drogi Bajtku!

BO, COPY służy do archiwizacji programów; dzięki niemu na jednej taśmie przechowywać można nawet 100 programów!

Interface Kempston powinien działać bez zarzutu z TIMEX-em, choć mogą zdarzyć się przypadki złej współpracy. Tym niemniej możliwość zepsucia TIMEX-a interfacem Kempston jest prawie żadna, choć w Twoim przypadku może się ujawnić ta jedna na 100 szansa.

Ani Spectrum, ani TIMEX nie podają sygnału Audio na wyjście antenowe. Tak więc uzyskanie głosu przez telewizor tak, jak na przykład w Commodore lub Atari jest niemożliwe. Istnieje przystawka o nazwie „Currah microSpeech”, której jedną z funkcji jest uzyskanie dźwięku przez głośnik telewizora. Jeśli chcesz wykonać to sam spróbuj połączyć wyjście MIC z gniazdem „głośnikowym” w telewizorze — czasem daje to rezultat. Zaś najprostszym z możliwych wzmocnienie głosu uzyskasz podłączając MIC do gniazda mikrofonowego magnetofonu, odłączając kabel EAR-gniazdo słuchawkowe. Po włączeniu magnetofonu na nagrywanie (bez kasy!) z jego głośnika popłyną dźwięki. Co do wewnętrznego głośnika TIMEX-a, to jego odłączenie programowe jest niemożliwe, zaś sprzętowe ogranicza się do użycia lutownicy (raz na zawsze) lub wbudowaniu przetwornika w obudowę mikrokomputera.

Ulegając namowom Czytelników na temat numeru specjalnego o Spectrum, zmieniliśmy zdanie i numer taki ukaże się. Nie nastąpi to jednak szybko, bowiem w przygotowaniu są na razie następujące numery specjalne: Atari III, Gry II, Commodore II i Atari IV. „Bajtko” tylko o Spectrum spodziewajcie się więc pod koniec bieżącego roku.

Chciałbym wiedzieć, w jaki sposób tworzy się efekty specjalne na ekranie, jak tworzyć procedury przerwań, gdzie one się znajdują w pamięci i w ogóle wszystko o przerwaniach. Proszę o wydrukowanie krótkiego programu rysującego koła w trybie graficznym 7. Bardzo proszę o doradzenie mi jakiejś książki z zakresu mikroprocesora 6502.

(nazwisko i adres do wiadomości redakcji)

Odpowiedzi na pytania zawarte w liście zapełniłyby prawie cały numer „Bajtko”. Niektóre z tych tematów były już poruszane w naszym piśmie. Polecam Ci lekturę książki warszawskiego SOETO: „Assembler 6502” oraz „Poradnik programisty Atari”

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

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

K-133

ATASERW

42-100 Tychy
ul. Lenczewicza 46/3
Tel. 27-69-66

oferuje świetne rozwiązania sprzętowe do ATARI XL/XE:

1. TURBO DOS-wspaniały DOS na kartidżu
2. TOP DRIVE 1050-samodzielny montaż /recenzja INFORMIK III/883.
3. INTERFEJS CENTRONIKS
4. ROZSZERZENIA PAMIĘCI
5. BASIC XE-kartridż
6. PROGRAMATOR EPROM
7. PROGRAMY UŻYTKOWE

12 miesięcy gwarancji. Informacje i zamówienia telefonicznie (wtorek 8-12, środa, czwartek 16-18) i listownie po otrzymaniu koperty zwrotnej.

K-128

Naprawa komputerów
Perzyński Warszawa 24-93-91
D-74

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 (jelonki)
tel. 37-76-65.

(K-118)

TURBO 2000
System transmisji danych dla ATARI
— 7 razy szybciej
— 7 krotna oszczędność taśmy
— niezawodność pracy
— pełna automatyka działania
— bogata biblioteka programów
To nowy standard w Polsce — kup nie zwlekaj!
„STUDENT — SERVICE” 42-200
Częstochowa
ul. Zawadzkiego 29, tel. 573-75

K-141

ATARI, programy na dyskietkach, napisz-otrzymasz bogatą ofertę. ATARICA, skr 46 82-300 Elbląg

G-82

Programy na Atari Tanio! Robert Hysa
ul. Miła 10/19 41-250 Czeladź

G-73

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

D-84

ATARI

- ponad 1500 programów
 - na kasetach i dyskietkach
 - co piąty program gratis
 - gwarancja jakości
 - rachunki
 - katalogi bezpłatnie
- ART-SOFTWARE
66-542 Zwierzyn P-1

(D-46)

OTV RADZIECKIE
JUNOST ELEKTRONIKA SILELIS
oraz STACJONARE:
NAPRAWA ● KINESKOPY ●
DEKODERY PAL ● FONIA CCIR
WEJŚCIA MONITOROWE
INTER-SERWIS,
ul. Rutkowskiego 10/12
00-020 Warszawa, tel. 27-47-72
D-76

ZX Spectrum

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

D-73

Innowacyjno-Wdrożeniowe Towarzystwo Techniczne



poleca profesjonalne oprogramowanie:

- system „Karta Drogowa”
- kompleksowe rozliczenie bazy transportowej,
- system „Kadry” i „Płace”.

Ponadto oferujemy kompletację i dostawę sprzętu niezbędnego do funkcjonowania w/w systemów. Komputery wyłącznie renomowanych firm zachodnioeuropejskich. Niskie ceny. Natychmiastowa dostawa. Zapewniamy dostawy materiałów eksploatacyjnych (papier, taśmy itp.)

Nasz adres:
Tarnobrzeg 39-400
ul. M. Kopernika 28
tel. 22-11-01 tlx. 062 404
skr. poczt. — 141

K-84

SPRZĘŻENIE ZWROTNE

(ta druga jest dopiero przygotowywana do druku). Znajdziesz tam odpowiedzi na większość nurtujących Cię pytań.

1. Czy jest możliwa komunikacja Atari z IBM za pomocą dyskietki?
2. Jak można używać CP/M na Atari? Jakie przeróbki są konieczne, aby było to możliwe?

Jerzy Ratajczak
Warszawa

1. Podobnie tak. Istnieje program kopiujący pliki tekstowe zapisane w formacie Atari na format IBM i odwrotnie. Wymaga on jednak bardzo specyficznego sformatowania dyskietki przez IBM i — jak dotychczas — w redakcji nie udało nam się tego dokonać.

2. Jedynym znanym mi sposobem użycia CP/M na Atari jest wykorzystanie specjalnej przystawki ATR-8000. Była ona produkowana w USA, lecz nie wiem, czy jeszcze można ją nabyć, gdyż przestała się pojawiać w ofertach publikowanych w zachodnich czasopiśmie. CP/M jest systemem przeznaczonym dla procesora Z80. Nie można więc wykonać w komputerze przeróbek umożliwiających korzystanie z niego (chyba, że przerobić Atari na Amstrada CPC-6128 albo Commodore 128).

W wielu programach zaobserwowalem płynne przesuwanie się tekstu w poziomie. Jak można uzyskać taki efekt na Atari Basicu lub Turbo Basicu?

Grzegorz Kępiński
Warszawa

Zadawalający efekt płynnego przesuwania tekstu można uzyskać wyłącznie przy pomocy języka maszynowego. Zwykle wykorzystuje się przy tym przerwania VBLKI lub TIMER2. Użycie w tym celu dowolnego interpretera Basicu nie da zadawalającego wyniku, gdyż napis będzie przesuwany bardzo wolno, a ponadto program nie może wtedy wykonywać innych czynności. Oczywiście procedurę przerwania można dołączyć do programu w Basicu.

1. Jak zrealizować instrukcję GRAPHICS z poziomu języka maszynowego?
2. W drugim bajcie programu typu „boot tape” podana jest liczba rekordów

do odczytania. Z ilu bajtów składa się rekord? Czy trzeba dopełnić ostatni rekord zerami, gdy jest on niepełny?

3. W „Bajtku” 2/89 podano, że funkcja USSR zwraca zawartość komórek 203 i 204, natomiast w „Atari Basic” wymienione są komórki 212 i 213. Co jest prawdą?

Karol Egeman
Częstochowa

1. Instrukcja GRAPHICS w pełni odpowiada instrukcji OPEN 6,A,B,„S:”, gdzie A i B są parametrami wybranego trybu graficznego. W języku maszynowym trzeba więc użyć procedury otwierającej kanał IOCB, czyli zastępującej odpowiednią instrukcją OPEN.

2. Standardowy rekord zapisu na taśmie składa się zawsze ze 128 bajtów. Jeśli ostatni rekord jest niepełny, to zostaje automatycznie uzupełniony zerami. Ponadto na końcu jest transmitowany specjalny rekord wskazujący zakończenie pliku. Sposób transmisji może zostać zmieniony przez ingerencję w system operacyjny komputera.

3. W „Bajtku” nastąpiła pomyłka. Wynik funkcji USSR jest pobierany z komórek 212 i 213.

1. Podczas pisania programów w Action!, po naciśnięciu RESET komputer przechodzi do DOS-u. Powoduje to utratę programu. Czy jest na to jakaś rada?

2. Od jakiego adresu zaczyna się edytor Action!?

3. Co oznaczają nawiasy trójkątne w katalogu plików dyskowych?

4. Jak można odczytać wartość licznika PC?

Roland Zimek
Opole

1/2. Oryginalny system Action! jest umieszczony na cartridge'u, który zajmuje obszar od adresu 40960 (sA000) do 49151 (sBFFF), i jest odporny na RESET. Zauważona wada wynika z użytkowania pirackiej kopii systemu.

3. Nawiasy trójkątne oznaczają pliki, które są niedostępne dla DOS 2.0, czyli przekraczają pojedynczą gęstość zapisu. Oznaczenie to zostało wprowadzone po opracowaniu DOS 2.5, który zastąpił DOS 2.0.

4. Stan licznika programu PC nie można odczytać.

REKLAMUJ SIĘ W BAJTKU!

**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)

**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, Ashtona 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)

miejsceowość

tel. tlx

(SB-45)

Cześć Maluchy!

Przyszł do Was z wizytą kolega. Nie należy — jak Wy — do Komputerowych Przedszkolaków, a więc komputer jest dla niego urządzeniem egzotycznym i nieco tajemniczym. Po wypróbowaniu kilku gier kolega odkłada drążek sterowy i pyta: „Czy mógłbyś mi pokazać, na czym polega to programowanie?”

W tym momencie stajesz przed poważnym problemem. Jak w ciągu kilkunastu minut przedstawić zasadę programowania? Warto by było napisać także jakiś prosty program. Zwykle taki pokaz kończy się na stworzeniu czegoś podobnego do tego:

```
10 PRINT „Jurek-Ogorek, kielbasa i sznurek”
20 PRINT
30 PRINT
40 GOTO 10
```

(Oczywiście jeśli nasz kolega ma na imię Jerzy.)

Trzeba przyznać — faktycznie — można nazwać to programem i to napisanym jedynie przy pomocy dwóch instrukcji. Kolega stwierdza, że programowanie jest zajęciem fascynującym, ale prawdę mówiąc minę ma nieco zawiedzioną. A może da się wymyślić coś ciekawszego i równie prostego? Właśnie coś takiego chciałem Wam zaproponować. Będą to wyścigi samochodowe napisane przy pomocy tylko jednej instrukcji (PRINT). Rozpoczniemy od zaprogramowania samochodu wyścigowego.

```
110 PRINT " "
120 PRINT " 0-/\-0"
130 PRINT " / 1 \"
140 PRINT " //+\\ "
150 PRINT " !!o!! "
160 PRINT " ()-< U >-() "
170 PRINT " \./ "
RUN
```

```
0-/\-0
 / 1 \"
 //+\\
 !!o!!
 ()-< U >-()
 \./
```

Ready

No dobrze, mamy samochód, ale on stoi w miejscu. Poruszmy go również przy pomocy instrukcji PRINT. Uzupełnimy nasz program o następujące linie:

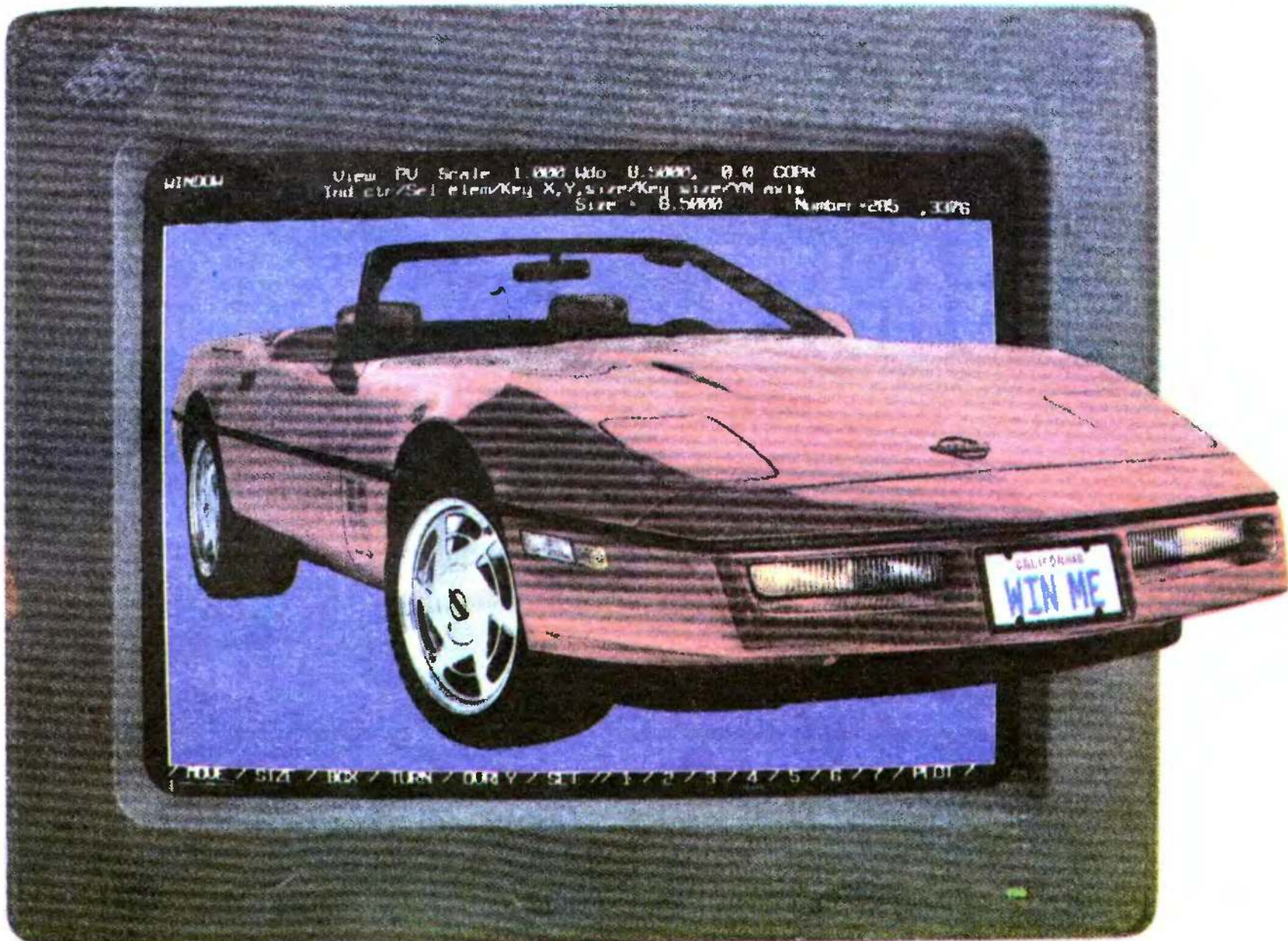
```
101 PRINT: PRINT: PRINT: PRINT: PRINT
102 PRINT: PRINT: PRINT: PRINT: PRINT
103 PRINT: PRINT: PRINT: PRINT: PRINT
104 PRINT: PRINT: PRINT: PRINT: PRINT
105 PRINT: PRINT: PRINT: PRINT: PRINT
171 PRINT: PRINT: PRINT: PRINT: PRINT
172 PRINT: PRINT: PRINT: PRINT: PRINT
173 PRINT: PRINT: PRINT: PRINT: PRINT
174 PRINT: PRINT: PRINT: PRINT: PRINT
175 PRINT: PRINT: PRINT: PRINT: PRINT
```

Ruszył naprawdę. Najpierw wszystkie napisy uciekły do góry, a następnie zza dolnej krawędzi ekranu wyskoczył nasz samochód, przejechał przez cały ekran i zniknął za jego górną krawędzią. Wykorzystaliśmy tutaj zjawisko tak zwanego „przewijania” ekranu, czyli przesuwania wszystkiego, co znajduje się na ekranie do góry w momencie, gdy kolejna linia nie mieści się na ekranie. W przypadku komputera ZX Spectrum i jego licznej rodziny sprawa troszeczkę się komplikuje. Otóż Spectrum po wypełnieniu całego ekranu zatrzymuje program i czeka na naciśnięcie dowolnego klawisza. Można tego uniknąć wpisując po uruchomieniu komputera:

POKE 23692,255: REM *** Tylko Spectrum ***
Oczywiście należy to zrobić tak, aby nie widział tego nasz kolega, bo mu się znowu wszystko pomiesza w głowie.

Ale co to za wyścigi skoro mamy tylko jeden samochód? Czym prędzej musimy nadrobić to karygodne zaniedbanie. Oto program rysujący i uruchamiający kolejne samochody.

```
210 PRINT " x"
220 PRINT " #1-/\-1#"
230 PRINT " b 2 o"
240 PRINT " //-\\"
```



WYŚCIGI SAMOCHODOWE

```
250 PRINT " :x:!"
260 PRINT " ##-1 U 1-##"
270 PRINT " \./ "
271 PRINT: PRINT: PRINT: PRINT: PRINT
272 PRINT: PRINT: PRINT: PRINT: PRINT
310 PRINT " ."
320 PRINT " 0-/\-0"
330 PRINT " ( 4 )"
340 PRINT " </-\\"
350 PRINT " (!#!)"
360 PRINT " []-1 U 1-[]"
370 PRINT " |---|"
371 PRINT: PRINT: PRINT: PRINT: PRINT
410 PRINT " "
420 PRINT " 0-/\-0"
430 PRINT " 1 3 1"
440 PRINT " //+\\ "
450 PRINT " !!o!! "
460 PRINT " ()-1 U 1-()"
470 PRINT " \./ "
471 PRINT: PRINT: PRINT: PRINT: PRINT
472 PRINT: PRINT: PRINT: PRINT: PRINT
473 PRINT: PRINT: PRINT: PRINT: PRINT
474 PRINT: PRINT: PRINT: PRINT: PRINT
475 PRINT: PRINT: PRINT: PRINT: PRINT
```

Przy okazji usuńmy linie 173, 174 i 175. Wpisując numer linii i naciskając >ENTER/RETURN<, zmniejszą to dystans pomiędzy pierwszym samochodem a następnymi.

Pisząc cały ten program skorzystaliśmy dotychczas tylko z jednej instrukcji — instrukcji PRINT. Musicie przyznać, że efekt jest interesujący. Niestety program wykonuje się bardzo szybko i trzeba go uruchamiać od nowa. Są na to dwie rady. Po pierwsze można dopisywać do programu kolejne samochody, a po drugie można wprowadzić nową instrukcję. Łatwiejsze będzie z pewnością to drugie rozwiązanie, tym bardziej, że ta „nowa instrukcja” nie jest wcale dla nas (a nawet dla Waszego kolegi) nową. Chodzi tu oczywiście o rozkaz RUN. Przyzwyczajaliśmy się korzystając z niego wyłącznie do uruchamiania programu, a tymczasem może on być z powodzeniem wykorzystywany w treści programu.

Dopiszmy więc ostatnią linię.

```
500 RUN
```

Teraz nasz program będzie biegał w kółko, aż mu nie przerwiemy lub nie wyłączymy komputera. Szerokiej drogi i gumowych drzew życzy Wam

Romek

```
101 PRINT: PRINT: PRINT: PRINT: PRINT
102 PRINT: PRINT: PRINT: PRINT: PRINT
103 PRINT: PRINT: PRINT: PRINT: PRINT
104 PRINT: PRINT: PRINT: PRINT: PRINT
105 PRINT: PRINT: PRINT: PRINT: PRINT
110 PRINT " "
120 PRINT " 0-/\-0"
130 PRINT " / 1 \"
140 PRINT " //+\\ "
150 PRINT " !!o!! "
160 PRINT " ()-< U >-() "
170 PRINT " \./ "
171 PRINT: PRINT: PRINT: PRINT: PRINT
172 PRINT: PRINT: PRINT: PRINT: PRINT
210 PRINT " x"
220 PRINT " #1-/\-1#"
230 PRINT " b 2 o"
240 PRINT " //-\\"
250 PRINT " !#!!"
260 PRINT " ##-1 U 1-##"
270 PRINT " \./ "
271 PRINT: PRINT: PRINT: PRINT: PRINT
272 PRINT: PRINT: PRINT: PRINT: PRINT
310 PRINT " ."
320 PRINT " 0-/\-0"
330 PRINT " ( 4 )"
340 PRINT " </-\\"
350 PRINT " (!#!)"
360 PRINT " []-1 U 1-[]"
370 PRINT " |---|"
371 PRINT: PRINT: PRINT: PRINT: PRINT
410 PRINT " "
420 PRINT " 0-/\-0"
430 PRINT " 1 3 1"
440 PRINT " //+\\ "
450 PRINT " !!o!! "
460 PRINT " ()-1 U 1-()"
470 PRINT " \./ "
471 PRINT: PRINT: PRINT: PRINT: PRINT
472 PRINT: PRINT: PRINT: PRINT: PRINT
473 PRINT: PRINT: PRINT: PRINT: PRINT
474 PRINT: PRINT: PRINT: PRINT: PRINT
475 PRINT: PRINT: PRINT: PRINT: PRINT
500 RUN
```


KIJOWSKIE GIGANTY



Pierwszy publiczny pokaz kijowskiego kolosa — An-225 „Mrija” w pełnej krasie.

An-225 Mrija potrafi przetransportować w powietrzu aż... 80 ciężarówek marki ZiŁ.

Świat dowiedział się po raz pierwszy o jego istnieniu w 1977 roku. Niezwykły gość Paryskiego Salonu Lotniczego — szef kijowskiego biura konstrukcji lotniczych Oleg K. Antonow wspominał wówczas dziennikarzom, iż jego zespół inżynierów pracuje nad programem rozwoju ogromnego samolotu transportowego nowej generacji.

Na tym polu technologicznej rywalizacji lotniczej do tradycyjnej sportowej dewizy — szybciej, wyżej, dalej, dorzucić należałoby dwa dodatkowe człony — z coraz większym ładunkiem na pokładzie i bardziej ekonomicznie. Na plany budowy gigantycznych odrzutowych samolotów transportowych stać od dawna tylko dwa największe mocarstwa światowe. W 1968 roku wystartowało do swego dziewiczego monstrem koncernu Lockheed — czterosilnikowy, dwupokładowy C-5 Galaxy. Już wymiary maszyny mówiły same za siebie — długość kadłuba — 82 metry, rozpiętość skrzydeł — 74 metry. Podczas prób okazało się, że Galaxy wychodzi w powietrze przy maksymalnym ciężarze startowym 349 ton. Przez ponad dziesięć lat pokazywany był wszędzie z obowiązkowymi przymiotnikami zaczynającymi się od naj...

Sporą sensacją stała się więc mała zapowiedź w znanym na całym lotniczym świecie roczniku „Jane's All The World's Aircraft 1977—78”, głosząca, iż Antonow przygotowuje samolot dorównujący Galaxy.

W maju 1985 roku wielki czterosilnikowiec opatrzony znakami rejestracyjnymi CCCP-82002 pojawił się nad Paryżem. Na powietrzne pokazy i wystawę na Le Bourget przyprowadził go szef pilotów-oblatywaczy Antonowa-Władimir Terski, prawy fotel drugiego pilota zajmował Jurij Pobol.

Nie trzeba chyba dodawać, że maszyna z Kijowa stała się od razu paryskim szlagierem. Prymat Amerykanów został ostatecznie złamany. An-124 Ruslan był bowiem największym i najsilniejszym samolotem transportowym naszego globu. Pobił Galaxy nie tylko rozmiarami — rozpiętość skrzydeł ponad 80 metrów (długość kadłuba — 76 metrów), ale przede wszystkim masą, którą mógł swobodnie przenosić. Jego maksymalny ciężar startowy — około 405 ton był do prawdy imponujący. Na tym nie koniec. Całkowicie zautomatyzowany, komputerowy system sterowania, z pominięciem elementów mechanicznych łączących wolant pilota z ruchomymi elementami usterzenia, znany na Zachodzie jako *fly by wire* (pisaliśmy już o tego rodzaju systemach w jednym z poprzednich numerów „Bajtka”), był szokującym zagranicznych obserwatorów całkowitym novum.

O tym, iż nie był to czczy zachwyty świadczy rekord Ruslana ustanowiony latem 1985 roku. Ruslan wywinął bowiem ładunek ponad 171 ton na wysokość ponad 11 tys. metrów.

Ale życie pokazało, że biuro konstrukcyjne Antonowa stać na coś o wiele większego, że budowa Ruslana była próbą generalną przed wyzwaniem znacznie poważniejszym. Aby zrozumieć sens najnowszego dokonania kijowskich projektantów musimy się zapuścić na podwórko kosmiczne.

Przygotowując swój program lotów promów kosmicznych Amerykanie rozwiązać musieli wcale nietrywny problem ich sprawdzenia w locie szybowym i transportu powietrznego wahadłowców. Na zlecenie NASA (Agencji Aeronautyki i Przestrzeni Kosmicznej) przebudowano w zakładach Boeinga w Seattle seryjnego czterosilnikowego Jumbo Jeta dostosowując go do przenoszenia orbiterów na grzbiecie kadłuba na specjalnym rusztowaniu. Bezpieczny lot takiego kombinowanego zespołu zapewniły dodatkowe stateczniki pionowe. Pierwszy samodzielny krok w powietrzu eksperymentalnego promu „Enterprise” odbył się właśnie w ten sposób. Zmodyfikowany B-747 wyniósł

go na odpowiednią wysokość. Wahadłowiec zaś, wyczepiwszy się, o własnych siłach, prowadzony przez swoją załogę, dotarł do ziemi. Ten sam specjalny Jumbo Jet przenosi po każdej misji orbitalnej amerykańskie pojazdy z pustynnego lądowiska na stanowisko startowe na Przylądku Canaveral. Maszyna jest także wygodnym środkiem transportu nowych zmontowanych już promów na miejsce finalnego ich przeznaczenia.

Identyczne zagadnienie stanęło przed radzieckimi konstruktorami. Obecnie, po pierwszej bezzałogowej wyprawie wahadłowca Buran, poznaliśmy sporo szczegółów przygotowań do jego dziewiczego lotu pozaziemskiego, który, jak pamiętamy, miał miejsce 15 listopada 1988 roku.

Radziecki samolot kosmiczny montowany był wiosną 1988 roku w ogromnej hali kosmodromu Bajkonur. Wielkie moduły konstrukcyjne wieziono drogą powietrzną na grzbiecie czterosilnikowego odrzutowca dostosowanego specjalnie do tego rodzaju operacji. Znany samolot konstrukcji Miasszczewa przekształcono w latającego tragarza o oznaczeniu 201 M. Zapewne ta właśnie maszyna umożliwiła wstępne skoki w powietrzu i praktyczne przetestowanie w locie systemu automatycznego sterowania Burana.

Rozwój wydarzeń dowiódł jednak, że doraźnie zaadaptowany odrzutowiec starszego typu nie jest jednak w stanie sprostać wymaganiom błyskawicznie rozwijającego się w ZSRR programu eksploracji, jak mawiają Rosjanie „bliskiego kosmosu” przy pomocy pojazdów wielokrotnego użytku, jakimi są wahadłowce. 201 M był jedynie prowizorycznym wyjściem z sytuacji. Szybki transport promów wymagał nowoczesnego samolotu o nieprzeciętnych parametrach. Projekt takiej właśnie maszyny powierzono grupie Antonowa. Zamówienie musiało zostać złożone, kiedy prace nad promem były już w pełnym toku, skoro prototyp kijowskiego supergiganta zaprezentowano na początku grudnia 1988 roku.

Setki osób zgromadzonych na płycie przed hangarem w Kijowie obserwowało transportowego mamuta

wynurzającego się z cienia. Tego jeszcze nie było w Związku Radzieckim. Pierleństwo spowodowała, że wszystko odbyło się tak, jak przy prezentacji najnowszych konstrukcji Boeinga, czy zachodnioeuropejskiej spółki Airbus.

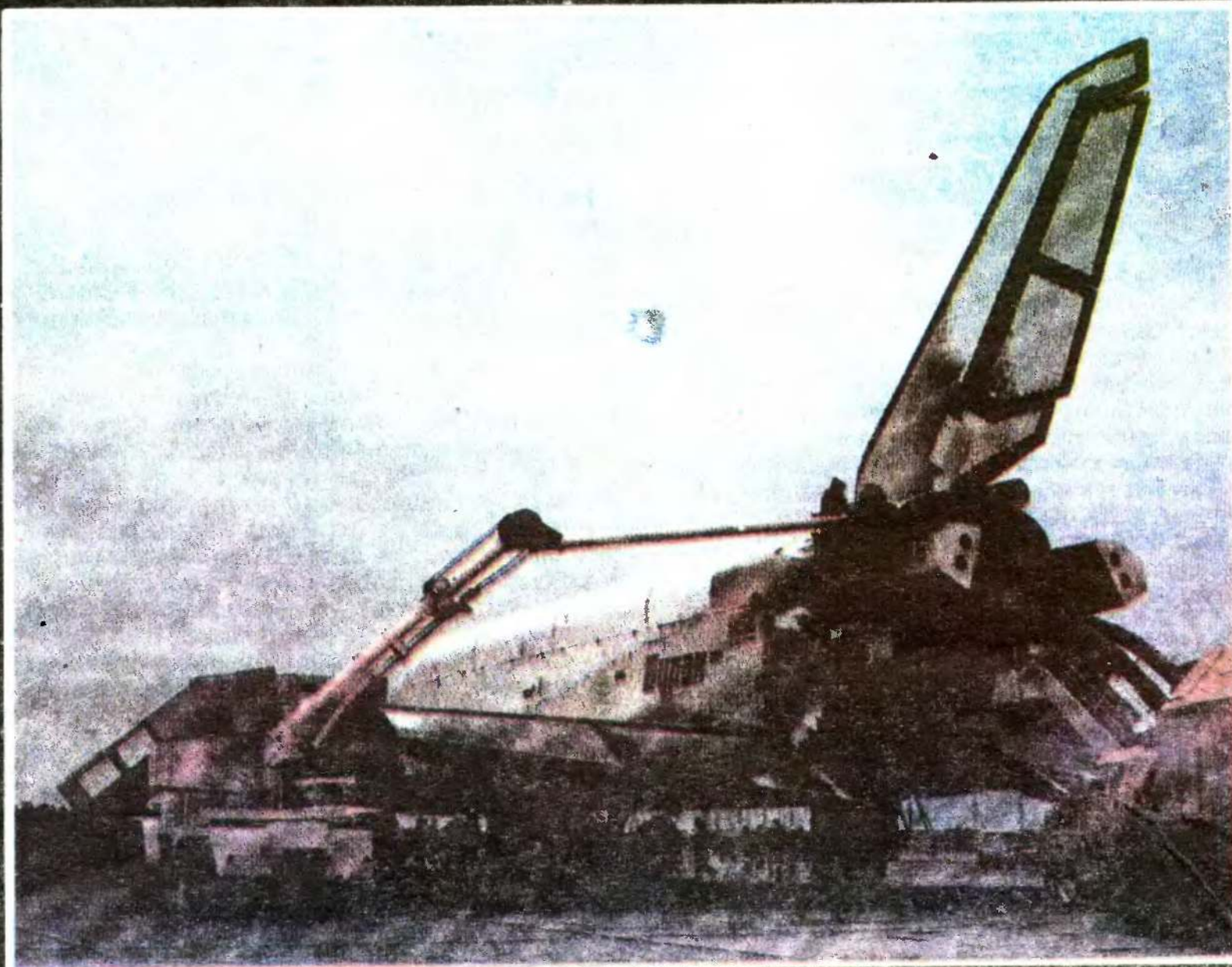
Najnowszy An-225 Mrija (Marzenie) jest kolesem, jakiego dotąd nie było na świecie. Jego maksymalny ciężar startowy szacowany jest na ponad 600 ton! Napędza go sześć zabudowanych na podskrzydłowych wysięgnikach wielkich silników lotariowa D-18T, każdy o ciągu ponad 23 tys. kg. W ogromnej ładowni unieść może swobodnie ciężary o masie przekraczającej 200 ton. Aby uzmysłowić swym Czytelnikom możliwości samolotu moskiewska „Prawda” napisała, iż An-225 potrafi przetransportować w powietrzu aż... 80 ciężarówek marki ZiŁ.

Ale Mrija nie powstał przecież po to, aby wozic samochody. O jego przeznaczeniu świadczą tak zwane „mocne punkty” na grzbiecie kadłuba. Tam właśnie zakotwiczony będzie Buran i jego następcy podczas wahadlowych przelotów między wytwórniami a kosmodromem. Usterzenie maszyny podobne do tego zastosowanego w 201 M sugeruje, iż jego przeznaczeniem będzie dźwiganie na zewnątrz kadłuba przedmiotów nie mieszczących się w brzuchu. Zapewne także elementów startowych rakiety Energia, stanowiącej podstawę napędową radzieckich promów.

22 grudnia An-225 wystartował po raz pierwszy. Potrzebował do rozbiegu 3500 m betonu.

Wszystko to zapowiada poważne przyspieszenie marszu kosmonautyki Kraju Rad. Przyspieszenie, które nie byłoby możliwe bez latających gigantów z Kijowa. To właśnie one spowodują, jak się należy spodziewać, w najbliższej przyszłości redukcję kosztów ponoszonych dotąd na dostarczanie drogą lądową nietypowych frachtów nie tylko na kosmodromy. A goście tegorocznego Międzynarodowego Salonu Lotniczego w Paryżu powinni oczekiwać, za sprawą biura konstrukcyjnego Antonowa, kolejnej sensacji z Kijowa.

Wojciech Łuczak



Prom kosmiczny „Buran” podróżować będzie na grzbiecie... supertransportowca „Mrija”.