IKS

INFORMATYKA KOMPUTERY SYSTEMY

CENA 120 ZŁ

DODATEK „ŻOŁNIERZA WOLNOŚCI" Nr 5/1988 ISSN 0860—2794

Propozycje

Żywioł i koniunktura — czyli komputery sprowadzane przez Prosystem w sklepach Centralnej Składnicy Harcerskiej — strona 15.

Czy przypadkowo skasowane zbiory można odtworzyć? — odpowiedź na stronie 22.

Informatyczny słownik angielsko-polski — Uwaga to może się przydać — strona 29.

Nauka • Technika

Zachodnioniemiecka firma SIGMA-SPORT zaprezentowała na wystawie w Nowym Jorku rower wyposażony w komputer, który pokazuje cykliście aktualną prędkość jazdy, szybkość średnią i maksymalną, jaką rowerzysta może na danej trasie rozwinąć przy swoich możliwościach fizycznych, liczbę przejechanych kilometrów, siłę nacisku na pedały - a także liczbę skurczów serca rowerzysty.

■ ■ ■

W Japonii opracowano elektroniczny system poszukiwania abonentów, umożliwiający błyskawiczne odnalezienie danej osoby i uzyskanie z nią połączenia telefonicznego. Ludzie, którzy z racji obowiązków służbowych wędrują po różnych pomieszczeniach - wkładają karteczkę ze swym kodem do specjalnej przystawki aparatu telefonicznego, który jest akurat w pobliżu. Informacja ta przekazywana jest natychmiast do komputera, który steruje pracą centrali telefonicznej.

Ciąg dalszy na str. 8

Programuję w ciemno...

W numerze

Klasówka przy klawiaturze

Zanim fala namiętności mikrokomputerowych dotarła do Polski wraz ze stosunkowo tanimi modelami Spectrum, Atari i Amstrada, informatycy, nauczyciele i wiele innych osób zaczęło sobie zadawać pytanie. „Kiedy te cudowne maszynki dotrą do polskich szkół?" W to, że wcześniej, czy później muszą dotrzeć, mało kto wątpił. Komputer stał się synonimem nowoczesności, a jakby na to nie spojrzeć, próg nowoczesności, legendarny XXI wiek, przekroczy za 12 lat. Dzisiejsze 6-latki będą ostatnim pokoleniem osiągającym pełnoletność (według dzisiejszych zasad, w „starym" XX wieku). Czyż zatem można pozwolić, by ich starsi koledzy, nie zetkneli się podczas szkolnych lekcji z klawiaturą komputera, by nie poznali jego zasad pracy i nie przyzwyczaili się do rewolucji niesionej przez masowo stosowaną informatykę?

Jak to najczęściej bywa, wówczas gdy można zarobić trochę pieniędzy, pojawiają się energiczne firmy. Podobnie było z dostawą mikrokomputerów do polskich szkół. Pierwsze oferty wpłynęły od firm zagranicznych oraz spółek z kapitałem mieszanym. Zwyciężyła jednak koncepcja rodzimego komputera szkolnego. Dwa lata temu rozstrzygnięty został konkurs na polski mikrokomputer edukacyjny — wygrała konstrukcja opracowana przez zespół z Zakładu Badań Operacyjnych i Systemów Komputerowych Instytutu Automatyki Politechniki Poznańskiej. Mikrokomputer otrzymał nazwę Junior, nazwę-symbol, podkreślającą jego przeznaczenie. Zastosowany jest w nim mikroprocesor 8-bitowy Zilog Z-80A, ten sam co w najpopularniejszych mikrokomputerach 8-bitowych (Spectrum, Amstrad). Nowsze konstrukcje wykorzystują mikroprocesory 16- i 32-bitowe. Nie w tej jednej dziedzinie decydują finanse i dylemat produkować tanie Juniory, czy nie produkować wcale. Wybór niewielki i trudno się dziwić, że zdecydowano się na pierwszy wariant, jednak Junior jest droższy niż dostępne na rynku 8-bitowce sprowadzane z zagranicy.

Dwa lata, jakie upłynęły od rozstrzygnięcia konkursu na komputer edukacyjny, to krótki okres. Przez te dwa lata uruchomiona została produkcja we wrocławskich zakładach Elwro. Jesienią ubiegłego roku ruszyła produkcja seryjna, w grudniu około 2 tysiące jednostek centralnych czekało na odbiorcę. Nie były to kompletne zestawy, takie jakich sobie życzył klient — Ministerstwo Edukacji Narodowej. Dla fabryki były to już gotowe produkty, co gorsze, że w świetle przepisów, za ich składowanie fabryka powinna zapłacić karę (gdyby nie sprzedano ich w 1987 roku). Po licznych monitach, 31 grudnia, za zgodą Ministerstwa Edukacji Narodowej, wyspecjalizowane przedsiębiorstwo zaopatrujące szkoły w pomoce naukowe Cezas, zakupiło kłopotliwą partię Juniorów. Nowy rok rozpoczął klient — Ministerstwo Edukacji od problemów ze skompletowaniem zestawów. Serwis będzie musiał dodatkowo jeździć do szkół, by zamontować brakujące urządzenia peryferyjne — wiadomo, kto poniesie koszty.

W naszych warunkach dwuletni okres uruchamiania produkcji jest małym rezultatem. Te same dwa lata, jeśli zmierzymy nimi rozwój informatyki, zmieniają swoją wymowę. Widać to na specjalistycznych targach sprzętu komputerowego. Pojawiły się nowe konstrukcje komputerów, urządzeń zewnętrznych. ZX Spectrum dwa lata temu uważany był za komputer, dziś traktuje się go jak zabawkę. Magnetofon kasetowy, jako pamięć zewnętrzna został wyparty przez stacje dysków elastycznych. Junior, choć jeszcze praktycznie nie dotarł do szkół, w błyskawicznym tempie starzeje się. Do dnia dzisiejszego nie zostało zakończone testowanie serii próbnych, nadal są problemy ze skompletowaniem pełnych zestawów zamawianych przez Ministerstwo Edukacji.

Szkolna konfiguracja składa się z zestawu sieciowego (ośmiu komputerów wykorzystujących wspólny dysk elastyczny i drukarkę) oraz dwóch dodatkowych, autonomicznych komputerów z dyskami elastycznymi i drukarkami. Podczas pracy komputerów w sieci, jeden z nich (nauczycielski), steruje pracą dysku, drukarki, pozostałe komputery (uczniowskie) pracują jak normalne ZX Spectrum wyposażone w pamięć dyskową i drukarkę. Żaden z użytkowników pracując przy swojej klawiaturze i monitorze, nie odczuwa, że w tym samym czasie z urządzeń zewnętrznych korzystają inni użytkownicy. Przewaga komputerów pracujących w sieci nad pracą pojedynczych komputerów, polega na tym, że można między nimi przeprowadzać transmisję danych, można wykorzystywać mniej urządzeń peryferyjnych, a to obniża koszty. W przypadku Juniorów działających w lokalnej sieci mikrokomputerowej Junet, praca uczniów na komputerze może być obserwowana przez nauczyciela na ekranie jego komputera. Pomaga mu to o tyle w pracy, że nie będzie musiał chodzić po klasie, by patrzeć, jak radzą sobie wychowankowie.

Junior i sieć Junet to pierwszy etap informatyki — sprzęt drugi, to oprogramowanie. Pracą sieci zarządza system operacyjny CP/J, wzorowany na najbardziej rozpowszechnionym dla mikrokomputerów ośmiobitowych dyskowym systemie operacyjnym CP/M. Oprogramowanie przechowywane w pamięci EPROM komputera zawiera edytor, interpreter języka Basic i programy sterujące pracą urządzeń zewnętrznych. Junior jest zgodny ze Spectrum firmy Sinclair Research Ltd. — wszystkie programy napisane dla tego najpopularniejszego mikrokomputera, bez żadnych adaptacji, przeróbek mogą być wykorzystywane w Juniorze. System operacyjny CP/J oraz oprogramowanie systemowe umożliwiają stosowanie najbardziej rozpowszechnionych języków i systemów programowania: Basic, Logo, Pascal, Fortran, C, Forth,

makroasemblery mikroprocesorów Intel 8080, 8085 i Z80.

O ile nad zakupem komputerów czuwają kuratoria (chodzi o to, by nie dopuścić do zakupu modeli nie kompatybilnych ze sobą), to przy zakupie i doborze programów edukacyjnych szkoły są samodzielne. Powstał „Katalog oprogramowania dydaktycznego" — informator dla nauczycieli o tym, jakie programy znajdują się w zbiorach resortu edukacji narodowej. Wybór konkretnego programu zależy od nauczyciela, od jego metod nauczania i doświadczenia.

Informatyka w szkole nie może ograniczać się wyłącznie do sali „komputerowej", z Juniorów powinni korzystać nauczyciele innych przedmiotów — jak z normalnych pomocy naukowych, dlatego potrzebne są dodatkowe, autonomiczne egzemplarze. Z jednej strony mikrokomputer na lekcjach przedmiotów ścisłych może być wykorzystywany jako narzędzie do obliczeń, z drugiej przy dobrych programach edukacyjnych nauka może być atrakcyjną zabawą. Jak twierdzi Zbigniew Rogowski, doradca ministra ds. informatyki brakuje dobrych programów edukacyjnych. Przy ich pisaniu należy połączyć doświadczenia pedagogów i informatyków. Bez tego zawsze będzie czegoś brakować — tak jak w programie Ortografia — zasób słownictwa nie został dobrany pod kątem uczniów szkoły podstawowej, do których był adresowany.

Od 1985 roku prowadzone są kursy przygotowujące nauczycieli do korzystania z mikrokomputerów. Przyjęto zasadę, że informatyki nauczać będą ludzie z przygotowaniem specjalistycznym, by korzystać z komputera na lekcji wystarczy krótsze przeszkolenie. Przedstawiciel producenta zapewniał mnie, że oprogramowanie sieci Junet jest na tyle proste w obsłudze, iż podczas szkoleń organizowanych dla nauczycieli, wystarczała godzina, by samodzielnie mogli prowadzić zajecia z wykorzystaniem sieci, pod warunkiem, że dany nauczyciel wiedział, co można pokazać w konkretnym programie. Oczywiście by poznać sieć należy poświęcić znacznie więcej czasu, trzeba dodatkowo poznać Juniora. Z pewnością nie wszystkich nauczycieli uda się przekonać do korzystania z informatyki — dyrektor Rogowski od swojego francuskiego odpowiednika wie, że we Francji w 1985 roku 30 proc. nauczycieli odcinało się od komputerów, wybierając tradycyjne metody nauczania. Jak będzie u nas, na razie zwolennicy nowoczesności wprowadzają komputery do szkół, kiedy skończą, zobaczymy.

D.O.

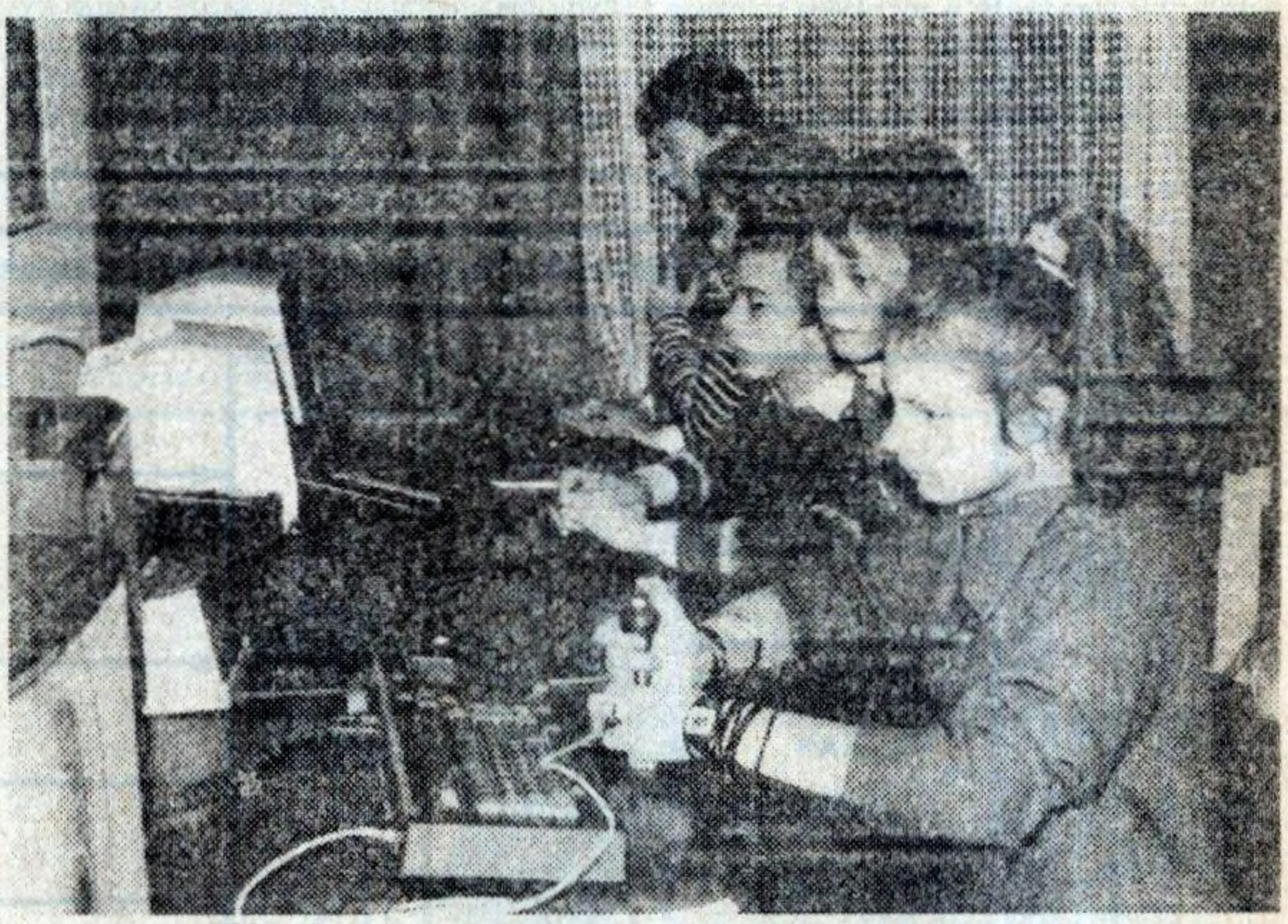
JUNET, czyli sieć lokalna mikrokomputerów ELWRO—800 JUNIOR, zapewnia współpracę kilkudziesięciu mikrokomputerów. Dzięki sieci możliwe jest wspólne wykorzystywanie pamięci dyskowych i drukarek przez wiele mikrokomputerów. Możliwe jest też przesyłanie obrazów, tekstów, programów oraz wybranych obszarów panięci operacyjnej z jednych mikrokomputerów do drugich. Poszczególne komputery pracujące w sieci mogą mieć zróżnicowane uprawnienia do korzystania z niej. Na przykład nauczyciel korzystając z mikrokomputera nauczycielskiego czyli uprzywilejowanego w sieci JUNET, może rozsyłać programy do mikrokomputerów uczniowskich oraz kontrolować prace uczniów, kopiując bez ich wiedzy zawartość ekranu wybranego mikrokomputera na własny ekran. Funkcje te są niedostępne w mikrokomputerach uczniowskich, ale uczniowie poprzez sieć JUNET mogą korzystać z pamięci dyskowych i drukarki mikrokomputera nauczycielskiego. Informacja przesyłana w sieci może być kierowana do jednego, wybranego odbiorcy, do określonych grup odbiorców lub do wszystkich odbiorców. Szybkość transmisji wynosi 64 kb/s. Sieć ma architekturę typu „wspólna szyna", dzięki czemu dołączenie kolejnych mikrokomputerów nie wymaga żadnych zmian we wcześniejszych połączeniach. Do sieci JUNET może być dołączony profesjonalny mikrokomputer ELWRO—800 jako jednostka dużej mocy obliczeniowej.

Drugi rok „Stobit-u"

„Co kupić? Magnetowid, czy komputer" — zastanawiał się jesienią 1986 roku Wiesław Rypina, kierownik Domu Kultury „Stokłosy". Nie wiadomo, jaką podjąłby decyzję, gdyby wśród zapytanych o to osób nie było Jana Ejgerda, zawodowego informatyka mieszkającego w pobliżu. Jak każdy fachowiec pasjonujący się swoją dziedziną, odpowiedział — „komputery, z pewnością będzie z nich większy pożytek. Powstanie klub komputerowy, dzieciaki będą się mogły czegoś pożytecznego nauczyć". W ten sposób powstała sekcja komputerowa „Stobit", nazwa powstała z połączenia części nazwy warszawskiego osiedla Sto-kłosy i jednostki informacji — bit.

Początki, jak to początki, nie były łatwe. Gdyby było odwrotnie, to nie wspominano by o nich z rozrzewnieniem. Zaczęło się od dwóch Unipolbritów, bez stacji dysków. Chętnych było zbyt wielu, osiedle młode, dużo rodzin z dziećmi, do centrum stolicy daleko, a tu pod bokiem taka atrakcja. Organizatorzy przyjęli założenie, że z jednej klawiatury podczas zajęć mogą korzystać dwie osoby. Stąd łatwo było wyliczyć, że przy dwóch komputerach, ośmiu godzinach zajęć tygodniowo można przyjąć 32 osoby. Od razu utworzona została lista rezerwowa, z której zapisane osoby miały pierwszeństwo przy przyjmowaniu. Najłatwiej było zwiększyć liczbę zajęć. W miarę upływu czasu kupiono więcej sprzętu — stację dysków do Unipolbritów, Amstrada CPC 464 i Amstrada CPC 6128 z kolorowymi monitorami.

„Stobit" rozpoczął drugi rok działalności — nadal funkcjonuje lista rezerwowa, co świadczy, że zajęcia prowadzone są w sposób ciekawy. Wśród uczestników zajęć wielu ma własne komputery, do nich należą Rafał Tkaczyński, Paweł Wróbel. Oni szukają czegoś więcej niż zabawy przy klawiaturze; chcą poznać zasady pracy komputera, języki programowania. Czasem trafia do klubu samouk, jak Jakub Połeć, który samodzielnie rozgryzł podstawowe języki programowania, obecnie uczy się asemblera. Przyszedł do klubu, bo „zawsze tu jest fajnie, można z kimś porozmawiać, powymieniać programy, doświadczenia".



Pod koniec zajęć Agnieszka lubi sobie troszkę pograć.

Większość przychodzących do „Stobitu" pierwszy raz zetknęła się z mikrokomputerami. Dla nich, 8-, 12-latków, (w tym wieku jest najwięcej uczestników) spotkania na zajęciach są szansą poznania informatyki. Jak stwierdził jeden z uczestników, „u mnie w szkole nauczyciele prawie nic nie wiedzą o komputerach. Pani bibliotekarka powiedziała, że to takie urządzenie, w którym po naciśnięciu guzika, wypisują się różne informacje". Dzieci śmieją się z takiego określenia komputera, dla nich dawno przestał być czarną skrzynką z guzikiem.

Tekst i zdjęcia: D.O.

Transputer — współczesna alternatywa komputera optycznego?

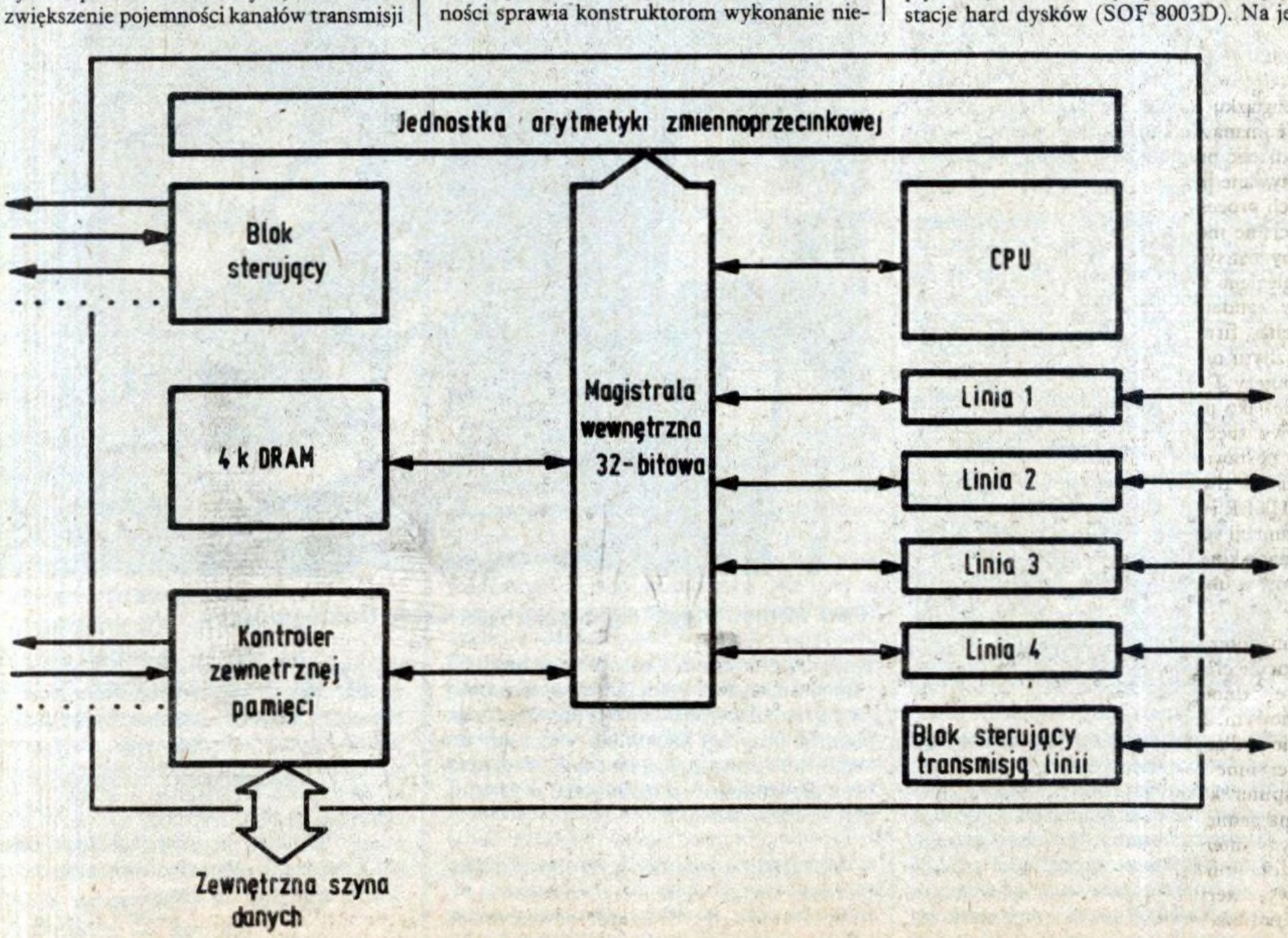
Już od dawna naukowcy, projektanci i konstruktorzy w swoich pracowniach toczą walkę, aby pokonać kolejne bariery techniki i technologii. Postęp badań w holografii [1], rozwój techniki laserowej i optoelektroniki upoważniał do stwierdzeń, że w niedalekiej przyszłości nastąpi pełna adaptacja fal optycznych jako nośnika informacji. Oczekiwano, że w telekomunikacji oraz w teleinformatyce elektronikę uzupełni lub całkowicie zastąpi nowa dziedzina — **fotonika**. Jednakże osiągnięcia współczesnej elektroniki, w szczególności pojawienie się w latach siedemdziesiątych, a następnie burzliwy rozwój mikroprocesorów dokonały rewolucyjnych zmian w wielu dziedzinach życia.

Komputery optyczne

Pomimo że techniczne opracowania pamięci holograficznych nie mają znaczenia przemysłowego, stały się one punktem zwrotnym w badaniach nad wykorzystaniem zjawisk dyfrakcji i interferencji fal świetlnych w opracowaniach komponentów architektonicznych komputerów optycznych (procesorów, pamięci, kanałów transmisji, przetworników WE/WY). Motorem tych zainteresowań były teoretyczne możliwości tj. niezwykła szybkość przesyłu i przetwarzania informacji zakodowanych w postaci fal świetlnych, milionkrotne zwiększenie pojemności kanałów transmisji oraz spodziewana, niewiarygodnie duża gęstość upakowania informacji w odpowiednio skonstruowanych pamięciach. W pracach badawczo-konstrukcyjnych napotkano na całkowicie nowe i skomplikowane problemy natury technologicznej i technicznej. Aktualne ich rozwiązania są niezadowalające. W przeciwieństwie do procesów elektronicznych procesy optyczne są pozbawione cech uniwersalności dla konkretnej realizacji układowej, mogą z dużą szybkością realizować tylko określoną klasę operacji. Zasadniczymi elementami architektury komputerów optycznych są lasery. Przede wszystkim stanowią one źródło światła, a także często decydują o precyzyjności innych komponentów architektury. Pożądane jest, aby wytwarzana wiązka charakteryzowała się bardzo dużą spójnością czasowo-przestrzenną, wysoką stabilnością częstotliwości, dużą mocą i dobrą podatnością na modulację. Z drugiej strony współczesne źródła światła powinny mieć odpowiednio niewielkie gabaryty i wykazywać się dużą żywotnością oraz sprawnością energetyczną. Wymagania powyższe są trudne do spełnienia. Dla przykładu lasery półprzewodnikowe są wielkości łebka od szpilki, niestety dyskwalifikuje je mała moc i niska spójność wytwarzanej wiązki. Natomiast lasery gazowe cechujące się idealnymi parametrami emitowanego światła wykazują dużą podatność na uszkodzenia oraz niską sprawność energetyczną. Wielkie trudności sprawia konstruktorom wykonanie niektórych specjalistycznych elementów i przetworników optycznych o wysokiej precyzyjności w technologii monolitycznej optyki zintegrowanej (scalonej).

Współczesne osiągnięcia i kierunki rozwoju

Aktualne badania i prace konstrukcyjne koncentrują się wokół problemów: hybrydyzacji (połączenia) środowisk — optycznego i mikroelektronicznego, dalszego doskonalenia procesów technologicznych oraz rozwoju optoelektroniki. Równolegle próbuje się poprawić zasadnicze parametry współcześnie istniejących systemów. Trwają poszukiwania nowych rozwiązań sprzętowych i systemowych (struktury RISC, komputery jednoukładowe, wieloprogramowość i wielodostęp oraz zrównoleglenie procesów). Dynamiczny rozkwit przeżywa sfera technologii układów scalonych. W jednym chip'ie można umieścić do 256 mikroprocesorów, zaś najnowszy superchip o powierzchni niespełna 14 cm^2 zawiera 28 milionów mikroelementów. Trzeba było aż 10 lat oczekiwania w sejfach koncernu IBM, aby koncepcja szybkich dysków sztywnych typu Winchester ujrzała światło dzienne. Najnowsze rozwiązania wykorzystują laserowe techniki tworzenia zapisu sygnału cyfrowego i jego odczytu (analogia do fonicznych płyt CD). Firma Sanyo produkuje optyczne stacje hard dysków (SOF 8003D). Na jednej



płycie można pomieścić od 0,5 do 4 GB w zależności od jej średnicy. Firmy 3M oraz Verbatim na targach CeBIT'87 w Hannowerze zaprezentowały dyskietki optyczne umożliwiające wielokrotny zapis i odczyt (pojemność dyskietek od 50 do 400 MB). Odpowiednie stacje są w opracowaniu.

Dość obiecująco przedstawiają się zagadnienia zrównoleglenia procesów. W tej dziedzinie preferowana jest wieloprocesorowość oraz systemy wielokomputerowe. Dzięki temu uzyskuje się pewien poziom współbieżności i wielozadaniowości prowadzonych procesów. Jak wiemy, większość z opracowanych procesów ma strukturę architektoniczną opartą na strukturze zaproponowanej przez Johna von Neumana. Zgodnie z tym licznik rozkazów wymusza operacje według programu zakodowanego w pamięci. Poszczególne operacje są wykonywane w określonej sekwencji kroków. Taka struktura ogranicza szybkość pracy. Dla ominięcia takiej sytuacji stosuje się więcej niż jeden procesor. W typowych rozwiązaniach architektonicznych komputerów procesor centralny steruje i kontroluje poprawność pracy pozostałych komponentów struktury, procesory uzupełniające z zasady wykonują funkcje specjalne (procesor, graphic controller itp.). Technika ta, stosunkowo droga i mało efektywna obciążona jest także niedogodnościami struktury von Neumana. Wieloprogramowość i wielodostęp w niewielkim stopniu poprawiają funkcjonalność struktury. Efekty tych zabiegów są zauważalne (w makroczasie) tylko przy stosunkowo niedużych systemach wykonujących nieskomplikowane algorytmy. Alternatywne rozwiązania dla tych zagadnień stanowią sieci komputerowe i technika decentralizacji „inteligencji" systemu dla procesów słabopowiązanych.

Transputery

W związku z niedogodnościami struktury von Neumana zachodzi pytanie, czy nie można by podzielić program na części, które byłyby wykonywane jednocześnie (w mikroczasie) w różnych procesach. Aby w pełni wykorzystać potencjalne możliwości techniki scalonej potrzebny jest system o dużo większym stopniu równoległego współdziałania niż to jest możliwe w standardowych systemach. W 1985 r. niewielka firma INMOS wprowadziła na rynki zbytu nową jakościowo rodzinę mikro: transputery T212 i T412, które zostały opracowane jako programowalny komponent systemowy specjalnie pod kątem rozwiązania wyżej postawionego zadania. Nazwa ta pochodzi od dwóch słów: TRANSistor oraz comPUTER i jest nawiązaniem do podobieństwa funkcji spełnianych przez transputery w systemach komputerowych z funkcjami tranzystorów w układach elektronicznych.

Transputer posiada procesor centralny, kilka pamięci oraz cztery linie komunikacyjne, będące jednocześnie układem wejściowym i wyjściowym. Do każdej z linii można bezpośrednio podłączyć inny transputer lub poprzez adapter inne urządzenia mikroprocesorowe. Transputer komunikuje się z własną wewnętrzną pamięcią RAM za pomocą równoległej szyny interfejsu z szybkością do 20 Mb/s na każdej linii szyny. Niezależnie od tego 16-bitowy interfejs pamięci lokalnej off-chip T212 posiada stałą konfigurację (z 16-bitową szyną danych). Szybkość przesyłania danych do pamięci lokalnej współpracującej z pamięcią transputera jest 2-, 3-krotnie mniejsza. 32-bitowy T414 posiada o wiele bardziej złożony interfejs. Umożliwia on 13 różnych konfiguracji realizowanych przez zaszycie w kości lub programowo. Szybkość transmisji na liniach komunikacyjnych standardowo wynosi 10—20 Mb/s zależnie od typu zegara.

W kwietniu 1987 r. na rynkach światowych pojawiło się następne dziecko transputerowej rodziny: IMS T—800, o pełnej 32-bitowej architekturze wewnętrznej. Twórca transputera Iann Barron do struktury mikroukładu włączył jednostkę arytmetyki zmiennoprzecinkowej oraz kilka innych, bardzo interesujących ulepszeń. W ten sposób T—800 jest znacznie szybszy od powszechnie stosowanych konkurentów: Intela 80386, Motoroli 68020, Nat-Semi 320—32 czy Fairchild Clippera. Za wadę transputerów uważa się fakt, że INTEL nie jest ich producentem. Wtedy promocja na rynkach byłaby z pewnością skuteczniejsza. Pogląd ten wydaje się zbyt przesadny. Transputerami interesują się firmy z USA i Japonii. Istnieją liczne ich sprzętowe zastosowania, m.in. IBM stosuje z powodzeniem transputery w sterownikach monitorów kolorowych wysokiej rozdzielczości. T—800 ma moc przetwarzania porównywalną z mocą najpopularniejszego dotychczas supermikrokomputera VAX 860. Nie ma w tym nic dziwnego, gdyż od samego początku transputer był projektowany pod kątem współbieżności procesów w dużych systemach komunikacyjnych. Oczywiście nie jest prawdą, że procesory Intel 80386 i Motorola 68020 nie mogą być używane w równoległych procesach, jednakże ich wewnętrzna architektura oraz oprogramowanie są zwrócone na obsługę przede wszystkim własnego mikrośrodowiska. Trudności realizacyjne wynikają z tego, że procesy komunikacyjne między tego typu CPU odbywają się poprzez zewnętrzne szyny. Planowanie bloków funkcjonalnych składających się z 100 takich mikroprocesorów znacznie komplikuje projekt struktury PCB interfejsów dla tych wszystkich szyn, że zadanie wydaje się niewykonalne. Trudnym zadaniem jest także wykonanie oprogramowania bloku, gdyż jednostki CPU mikroprocesorów znacznie lepiej przystosowane są do pełnienia funkcji zarządzających systemem niż współpracy równoległej ze sobą. To co jest więc istotą przystosowania T800 wynika po części z jego architektury wewnętrznej (Rys. 1).

Transputer jest układem opartym na strukturze RISC, zawierającym wszystkie elementy systemu komputerowego i może on wykonywać procesy bez potrzeby dodatkowych urządzeń rozszerzających niezależnie od sygnału zegarowego. Jego konwencjonalna jednostka CPU posiada podwyższone parametry arytmetyczno-logiczne dzięki zastosowaniu w niej 64-bitowej jednostki operacji zmiennoprzecinkowych, umożliwiających tym zwiększenie szybkości ich realizacji do 1,5 MFLOPS[2] (dla T414 wynosiła ona 100 kFLOPS). W dodatku posiada też własną pamięć on-chip DRAM (2 KB), z którą wymiana jest szybsza (80 Mb/s, w stosunku do 20 Mb/s dla off-chip DRAM). Ponadto wyposażony jest także w 4 linie komunikacyjne, przez które może bezpośrednio dokonywać transmisji z innym transputerem lub poprzez adapter z innymi urządzeniami komputerowymi. Możliwa jest bezpośrednia transmisja między dwiema pamięciami dwóch różnych transputerów. Kod transputera jest bardzo zwarty. Większość wykonywanych instrukcji jest jednobajtowa, i wykonywane są w jednym cyklu. W przyszłości planuje się znaczne zwiększenie złożoności struktur wewnętrznych. Do programowania układów transputerowych służy specjalizowany język wysokiego poziomu OCCAM. Dokładniejsze informacje o tym języku oraz innym oprogramowaniu w następnym odcinku z tego cyklu.

---

[1] holografia — polega na odtworzeniu rzeczywistego obrazu poprzez działanie falą odniesienia na zarejestrowany hologram (obraz interferencyjny fali odbitej od przedmiotu oraz fali odniesienia). Teoretyczna gęstość zapisu jest bliska 10^{10}b/cm^2 (uzyskana 10^7), zaś w hologramach przestrzennych 10^{13}b/cm^3.

[2] FLOPS (floating-point operation per secound) — operacje zmiennoprzecinkowe na sekundę.

W.M.S.

— Interesuje nas ktoś, kto zna dobrze ten instrument... rys. Puchalski

W y k r e s

Ludwik PIELA Tomasz MROWIEC

W dzisiejszym odcinku zamieszczamy, zgodnie z obietnicą, kompletny program umożliwiający kreślenie wykresów. Tworzy on, wyświetla i drukuje wykresy słupkowe, liniowe lub punktowe. Dane do kreślenia mogą być wprowadzone z klawiatury (gdy nie ma ich zbyt dużo) lub ze zbioru (zbiorów). Dodatkowo akceptuje on dane z programu przetwarzania arkusza obliczeniowego Visicalc. Napisany w języku BASIC może być używany na wszystkich 8-bitowych komputerach Atari posiadających 48KB pamięci wewnętrznej i stacje dysków lub magnetofon.

Mniej zaawansowani użytkownicy mogą używać go bezpośrednio po wprowadzeniu. Bardziej zaawansowani mogą wprowadzić własne ulepszenia i dostosować go do swoich potrzeb.

W swojej aktualnej postaci program może wyświetlić jednocześnie zawartość trzech zbiorów danych zawierających do 100 wartości każdy. Zbiory muszą zawierać jednakową liczbę danych z przedziału [0,900000].

Po wczytaniu danych program pyta o nazwy wykresu i osi Y. Jeśli nie chcemy nadawać nazw, naciskamy <RETURN>. Jeśli kreślimy dwa lub trzy wykresy jednocześnie, musimy wprowadzić nazwy dla każdego z nich. Dodatkowo musimy zaakceptować (ewentualnie zmienić) wartości minimalną i maksymalną oznaczaną na osi pionowej. Następnie program wybiera skalę dla osi poziomej X i prosi o wybranie sposobu jej opisu (litery, miesiące, liczby, bez oznaczania, ewentualnie własne oznaczenia).

Możemy kreślić następujące rodzaje wykresów: **liniowy**, **punktowy** lub **słupkowy**. W przypadku tego ostatniego musimy być świadomi ograniczeń wynikających z szerokości słupków. Na przykład, wykres słupkowy zawartości trzech zbiorów zmieści się na ekranie , jeśli ma nie więcej niż 12 wartości w każdym zbiorze.

Zbiory danych, przechowujące wartości, które chcemy przedstawić w postaci graficznej, możemy przygotować dwoma sposobami. Pierwszy z nich polega na wykorzystaniu jednego z edytorów tekstów, np.: AtariWriter, PaperClip lub Speed Script. Musimy przy tym przestrzegać następującej zasady: wszystkie liczby muszą być umieszczone w jednej kolumnie. W tym celu po wprowadzeniu liczby naciskamy klawisz <RETURN>. Zatem dane w zbiorze powinny wyglądać następująco:

```
  887
 2745
   91
13019
  562
 6743
```

W przypadku korzystania ze stacji dysków każdy zbiór ma własną nazwę, nadawaną podczas przechowywania. W przypadku magnetofonu wystarczy zapisać zbiory jeden po drugim (będą one oczywiście bez nazw).

Drugi sposób przygotowania danych do wykreślenia polega na wykorzystaniu programu przetwarzania arkusza obliczeniowego VisiCalc. Z jego pomocą możemy przygotować dane w postaci kolumn liczb. Przechowujemy je w pamięci zewnętrznej za pomocą opcji PF (wydruk do zbioru). Najpierw umieszczamy kursor u góry pierwszej kolumny danych arkusza obliczeniowego. Wprowadzamy rozkaz (PF, a następnie nazwę zbioru (dla zbioru na taśmie magnetycznej podajemy C). Po propozycji „lower right" przesuwamy kursor na koniec kolumny danych i naciskamy <RETURN>. Musimy powtórzyć te kroki dla każdej przechowywanej kolumny danych.

Istnieje jeszcze jeden sposób wprowadzania danych do programu WYKRES, przeznaczony głównie do wprowadzania niewielkiej liczby danych dla celów testowania, bądź też poznania możliwości programu. Po propozycji podania nazwy zbioru wprowadzamy „E". Teraz wartości możemy wprowadzać za pomocą klawiatury. Wprowadzanie danych kończymy naciskając CTRL—3 (znak EOF).

Możliwości programu nie będziemy opisywać. Po wprowadzeniu danych są one prezentowane w postaci menu i bardzo łatwo można je sprawdzić samemu.

```
HK 10 REM WYKRESY - cz. 1
OI 20 BRK=1:IF PEEK(53279)=5 THEN
    BRK=0
EA 40 DIM A$(20),F$(15),G$(37),T$
   (40),XL$(300),M$(72),TL$(40),Y
   L$(20),K$(25)
JA 50 DIM CL$(1),Y(100,2),A(100),
   B(100,2),PTS(3)
NE 60 M$="STYLUTMARKWIMAJCZELIPSI
   EWRZPAZLISGRU":M$(37)=M$:CL$=C
   HR$(125)
GF 70 GRAPHICS 0:POKE 82,2:POKE 8
   3,39:CLOSE #1:OPEN #1,4,0,"K:"
   :GOSUB 1930
DO 80 ? CL$:POSITION 17,1:? "WYKR
   ESY":?
VJ 90 ? :? "Program ten kresli wy
   kresy skladajace sie maksymaln
   ie ze 100 punktow na     podst
   awie danych ze zbioru."
NF 95 GOSUB 2000
WR 100 ? :? "Czy chcesz wykaz zbi
   orow dyskowych          (
   T/N)";
FI 110 GOSUB 1280:IF  NOT A THEN
   140
BK 120 ? CL$:TRAP 310:CLOSE #2:OP
   EN #2,6,0,"D:*.*"
CX 130 INPUT #2,A$:? A$;"  ";:IF
   A$(5,9)<>"FREE" THEN 130
KA 140 CLOSE #2:? :? :? "Ile wykr
   esow kreslic jednoczesnie
              (1-3)";
BG 150 GET #1,FILES:FILES=FILES-4
   8:IF FILES<1 OR FILES>3 THEN 1
   50
HA 160 ? FILES:? :A=0:YMAX=0:YMIN
   =900000:G$=" ":G$(36)=" ":G$(2
   )=G$
SO 170 REM LADOWANIE ZBIOROW
MQ 180 A=A+1:? "Podaj nazwe zbior
   u ";A;"  ";:INPUT #16;A$
ZC 190 F$=A$:TRAP 310:CLOSE #2:OP
   EN #2,4,0,F$:B=0:TRAP 250
NO 200 INPUT #2,C:IF C>900000 THE
   N ? "ZBYT DUZA WARTOSC !":GOTO
    100
RV 210 IF C<0 THEN ? "DANE UJEMNE
    !":GOTO 100
DY 220 B=B+1:Y(B,A-1)=C:IF YMAX<C
    THEN YMAX=C
PA 230 IF YMIN>C THEN YMIN=C
UF 240 IF B<100 THEN 200
AS 250 IF B=0 THEN A=A-1:? "BLAD
   LADOWANIA !":GOTO 180
AO 260 PTS(A)=B:G$(A*12-11)=A$:IF
    A<FILES THEN 180
EL 270 IF FILES=1 THEN 320
TF 280 IF FILES=2 AND PTS(1)=PTS(
   2) THEN 320

MU 290 IF FILES=3 AND PTS(1)=PTS(
   2) AND PTS(2)=PTS(3) THEN 320
HB 300 ? " ZBIORY ZAWIERAJA ROZNA
    ILOSC DANYCH !":GOTO 100
MH 310 ? :? "BLAD NR ";PEEK(195):
   GOTO 100

KL 315 REM WYKRESY - cz.2
UX 320 GOSUB 1530:REM SKALOWANIE
WC 330 GOSUB 1800:REM NADANIE NAZ
   W
WD 340 REM MENU
HM 350 TRAP 40000:POKE 702,64:GRA
   PHICS 0:GOSUB 1930:POKE 82,11:
   ? :? "    M E N U":? "  -----
   ----"
AH 360 ? :? "1. HISTOGRAM":? "2.
   WYKRES LINIOWY":? "3. WYKRES P
   UNKTOWY"
PE 370 ? "4. ZMIANA SKALI":? "5.
   ZMIANA NAZW":? "6. NOWE DANE":
   ? "7. KONIEC PROGRAMU"
IO 380 ? :? " CO WYBIERASZ?";:POK
   E 82,2
DV 390 GET #1,A:GH=A-48:IF GH<1 O
   R GH>7 THEN 390
FB 400 ? GH:IF GH=4 THEN GOSUB 15
   30:GOTO 350
NT 410 IF GH=5 THEN GOSUB 1800:GO
   TO 350
LJ 420 IF GH=6 THEN RUN
PJ 430 IF GH=7 THEN ? CL$:END
DK 440 IF GH>1 THEN ? :? :? "CZY
   KRESLIC PIONOWE LINIE (T/N)";:
   GOSUB 1280:VL=A
QZ 450 REM
YA 460 GRAPHICS 8+16:GOSUB 1930:C
   OLOR 1
KD 470 PLOT 56,8:DRAWTO 56,155:DR
   AWTO 319,155
PL 480 FOR A=LOW TO HI STEP (HI-L
   OW)/DIV:C=155-(A-LOW)*SCALE
CB 490 PLOT 50,C:DRAWTO 55,C:T$=S
   TR$(A)
NA 500 X=6-LEN(T$):Y=C-3:GOSUB 74
   0
IC 510 FOR B=59 TO 317 STEP 3:PLO.
   T B,C:NEXT B
CX 520 NEXT A
IH 530 IF TL$<>"" THEN T$=TL$:X=2
   0-INT(LEN(T$)/2):Y=0:GOSUB 740
ID 540 IF YL$<>"" THEN FOR A=1 TO
    LEN(YL$):T$=YL$(A,A):X=0:Y=72
   -LEN(YL$)*4+A*8:GOSUB 740:NEXT
    A
ME 550 IF XLABEL=5 THEN 600
YI 560 A=1:FOR B=1 TO PTS:A$=XL$(
   A,A+LL-1)
```

```
LO 570 FOR C=1 TO LL:T$=A$(C,C):X
   =A(B)/8:Y=149+C*7:GOSUB 740:NE
   XT C
IQ 580 A=A+LL:IF A>LEN(XL$) THEN
   A=1
DV 590 NEXT B
ST 600 IF FILES>1 THEN ON FILES-1
    GOSUB 840,900

HV 610 IF VL>0 AND GH>1 THEN FOR
   A=1 TO PTS:FOR B=154 TO 9 STEP
    -5:PLOT A(A),B:NEXT B:NEXT A
KZ 620 ON GH GOSUB 980,1040,1040
RJ 630 T$="(M)enu":X=8:Y=160:GOSU
   B 740:T$="(D)ruk":X=8:Y=168:GO
   SUB 740
KH 640 GET #1,A:IF A<>77 AND A<>6
   8 THEN 640
OF 650 IF A=77 THEN 350
UI 660 COLOR 0:FOR A=0 TO 55:PLOT
    A,160:DRAWTO A,174:NEXT A:COL
   OR 1:TRAP 630

QY 1515 REM WYKRESY - cz.3
TX 1520 REM SKALOWANIE
CG 1530 ? CL$:? "Ustawienie skali
    osi Y"
MB 1540 ? :? "Podaj dolna wartosc
    skali.":? "0-";YMIN;") 0";
YS 1550 ? "0";:LOW=0:TRAP 1570:I
   NPUT LOW:IF LOW<0 OR LOW>YMIN
   THEN 1540
FH 1560 REM OBLICZENIE SKALI
HZ 1570 HI=YMAX-LOW
NW 1580 POWER=0
QP 1590 IF HI>=100 THEN HI=HI/10:
   POWER=POWER+1:GOTO 1590
HN 1600 IF HI<10 THEN HI=HI*10:PO
   WER=POWER-1:GOTO 1600
TY 1610 HI=INT(HI+0.99)
PE 1620 IF HI/12=INT(HI/12) THEN
   DIV=12:GOTO 1660
JT 1630 IF HI/10=INT(HI/10) THEN
   DIV=10:GOTO 1660
FB 1640 IF HI/8=INT(HI/8) THEN DI
   V=8:GOTO 1660
ZF 1650 HI=HI+1:GOTO 1620
EM 1660 IF POWER>0 THEN FOR A=1 T
   O POWER:HI=HI*10:NEXT A
OY 1670 IF POWER<0 THEN FOR A=POW
   ER TO -1:HI=HI/10:NEXT A
TR 1680 SCALE=144/HI
NV 1690 HI=HI+LOW
OT 1700 ? :? "Gorna wartosc na os
   i Y ";HI;".":? "Zgadzasz sie (
   T/N)?";
KQ 1710 GOSUB 1280:IF A THEN 1760
MY 1720 TRAP 1720:? :? "Podaj gor
   na granice (wieksza niz ";:? YM
   AX;")";:INPUT HI
NF 1730 IF HI<YMAX OR HI>900000 T
   HEN 1720
TT 1740 HI=HI-LOW:GOTO 1580
IZ 1750 REM
JV 1760 ? :? "Przeliczam punkty .
   ..":PTS=PTS(1):FOR A=1 TO PTS:
   FOR B=1 TO FILES
XA 1770 A(A)=INT(245/PTS*(A-0.5)+
   65):A(A)=INT(A(A)/2)*2:B(A,B-1
   )=INT(155.5-(Y(A,B-1)-LOW)*SCA
   LE)
EO 1780 NEXT B:NEXT A:RETURN

SD 1790 REM WYKRESY - cz.4
NQ 1800 ? CL$:? "Nadawanie nazw :
   (";PTS;" punktow )"
RU 1810 ? :? "Podaj nazwe wykresu
   .":INPUT TL$
QE 1820 ? :? "Podaj nazwe osi Y."
   :INPUT YL$
MP 1830 IF FILES=1 THEN 1890
YU 1840 ? :? "Podaj oznaczenia dl
   a zbiorow danych.":? "(Maksyma
   lnie 8 znakow)"
RS 1850 ? :K$=" ":K$(24)=" ":K$(2
   )=K$:G$(37)=" "
OZ 1860 FOR A=1 TO FILES:? "Zbior
   ";A;" (";G$(A*12-11,A*12);")
   ";
VM 1870 INPUT A$:IF A$<>"" THEN K
   $(A*8-7)=A$
QQ 1880 NEXT A:K$(25)=" "
YI 1890 ? :? "Oznaczenie osi X  W
   lasne  , Litery  , Miesiace  ,
   Numery , lub Bez opisu ?";

SA 1900 POKE 702,64:GET #1,XLABEL
   :GOSUB 1980:XLABEL=XLABEL-74:I
   F XLABEL<1 OR XLABEL>5 THEN 19
   00
EX 1910 ? CHR$(XLABEL+74):IF XLAB
   EL<5 THEN ON XLABEL GOSUB 1310
   ,1440,1460,1490
AX 1920 RETURN
XA 1930 POKE 710,12:POKE 712,12:P
   OKE 709,2:POKE 77,0
IY 1940 IF BRK THEN POKE 16,112:P
   OKE 53774,112
BG 1950 RETURN

NF 725 REM WYKRESY - cz.5
FR 730 REM WYDRUK TEKSTU W GR.8
XO 740 X=INT(X):Y=INT(Y):COLS=40:
   START=(PEEK(89)*256+PEEK(88))+
   Y*COLS+X
HY 750 FOR E1=1 TO LEN(T$):E3=ASC
   (T$(E1))
XR 760 IF E3<32 OR (E3>127 AND E3
   <160) THEN E3=E3+64:GOTO 780
C8 770 IF E3>31 AND E3<96 THEN E3
   =E3-32
WM 780 CHARSET=PEEK(756)*256+E3*8
TG 790 FOR E2=7 TO 1 STEP -1
IA 800 POKE START+E2*COLS,PEEK(CH
   ARSET+E2):NEXT E2
YD 810 X=X+1:IF X>COLS THEN START
   =START+COLS*8:X=0
JC 820 START=START+1:NEXT E1:RETU
   RN

SO 1300 REM WYKRESY - cz.6
KM 1310 ? CL$:? "Oznaczenia na os
   i x dla ";PTS;" punktow"
JO 1320 ? :? "ILE ZNAKOW (1-3)?";
ZR 1330 TRAP 1330:GET #1,LL:LL=LL
   -48:IF LL<1 OR LL>3 THEN 1330
DP 1340 ? LL:? :? "Podaj oznaczen
   ie dla punktow.          Nacisn
   ij RETURN Jesli chcesz"
QZ 1350 ? "powielic oznaczenia.":
   A=1
XP 1360 ? :? "Podaj oznaczenie pu
   nktu ";A;".  >";:B=1
IQ 1370 GET #1,C:IF C=155 AND B=1
    AND A>1 THEN ? "POWIEL":RETUR
   N
ZU 1380 IF C<32 OR C>123 THEN 137
   0
KJ 1390 XL$(A*LL-LL+B)=CHR$(C):?
   CHR$(C);
LA 1400 IF B<LL THEN B=B+1:GOTO 1
   370
ZE 1410 IF A<PTS THEN A=A+1:GOTO
   1360
SX 1420 ? :RETURN
OM 1430 REM LITERY
NP 1440 FOR A=65 TO 90:XL$(A-64)=
   CHR$(A):NEXT A:LL=1:RETURN
EV 1450 REM MIESIACE
EB 1460 TRAP 1460:? :? "Miesiac p
   oczatkowy (1-12)";:INPUT A:A=I
   NT(A):IF A<1 OR A>12 THEN 1460
MA 1470 LL=3:B=A*3-2:XL$=M$(B,B+3
   5):RETURN
JC 1480 REM LICZBY
OD 1490 TRAP 1490:? :? "Zaczynamy
    od NR (0-900)";:INPUT A:A=INT
   (A):IF A<0 OR A>900 THEN 1490
NM 1500 XL$=" ":XL$(300)=" ":XL$(
   2)=XL$
OQ 1510 LL=LEN(STR$(A+PTS)):FOR B
   =1 TO PTS+1:XL$(B*LL-LL+1)=STR
   $(A+B-1):NEXT B:RETURN

OY 825 REM WYKRESY - cz.7
QZ 830 REM
GQ 840 T$="Znak    -          -":
   T$(9)=K$(1,8):T$(21)=K$(9,16):
   X=7:Y=182:GOSUB 740
ON 850 IF GH>1 THEN X=108:Y=185:F
   =1:GOSUB 1110:X=204:F=2:GOSUB
   1110:RETURN
SC 860 FOR A=106 TO 110:PLOT A,18
   2:DRAWTO A,188:NEXT A
RB 870 PLOT 202,188:DRAWTO 202,18
   2:DRAWTO 206,182:DRAWTO 206,18
   8:DRAWTO 202,188:PLOT 203,185:
   DRAWTO 205,185
ZU 880 RETURN
TZ 890 REM TRZY ZBIORY

YC 900 T$="Znak
          -":T$(9)=K$(1,8):T$(2
   1)=K$(9,16):T$(33)=K$(17,24)
RS 910 X=0:Y=182:GOSUB 740
ZX 920 IF GH>1 THEN X=52:Y=185:F=
   1:GOSUB 1110:X=148:F=2:GOSUB 1
   110:X=244:F=3:GOSUB 1110:RETUR
   N
EE 930 FOR A=50 TO 54:PLOT A,182:
   DRAWTO A,188:NEXT A
KI 940 PLOT 146,188:DRAWTO 146,18
   2:DRAWTO 150,182:DRAWTO 150,18
   8:DRAWTO 146,188:PLOT 147,185:
   DRAWTO 149,185
NH 950 PLOT 242,188:DRAWTO 242,18
   2:DRAWTO 246,182:DRAWTO 246,18
   8:PLOT 244,182:DRAWTO 244,188
ZR 960 RETURN
FL 970 REM HISTOGRAM
KR 980 FOR A=1 TO PTS
FL 990 IF FILES=1 THEN B=-2:GOSUB
    1180
KL 1000 IF FILES=2 THEN B=-4:GOSU
   B 1180:B=2:GOSUB 1190
NZ 1010 IF FILES=3 THEN B=-8:GOSU
   B 1180:B=-2:GOSUB 1190:B=4:GOS
   UB 1230
CT 1020 NEXT A:RETURN
FT 1030 REM WYKRES PUNKTOWY I LIN
   IOWY
XP 1040 FOR A=0 TO FILES-1:FOR B=
   1 TO PTS
SJ 1050 X=A(B):Y=B(B,A):F=A+1:GOS
   UB 1110
NQ 1060 NEXT B:NEXT A
II 1070 IF GH=3 THEN RETURN
FL 1080 FOR A=0 TO FILES-1:PLOT A
   (1),B(1,A):FOR B=2 TO PTS:DRAW
   TO A(B),B(B,A)
ED 1090 NEXT B:NEXT A:RETURN
WO 1100 REM KRESLENIE PUNKTOW
FB 1110 COLOR 0:PLOT X,Y-2:DRAWTO
    X,Y+2:COLOR 1
CQ 1120 IF F=1 THEN PLOT X,Y:PLOT
    X,Y-1:PLOT X+1,Y-1:DRAWTO X+1
   ,Y+1:DRAWTO X-1,Y+1:DRAWTO X-1
   ,Y-1


BE 1130 IF F=2 THEN PLOT X,Y:PLOT
    X-1,Y-2:DRAWTO X+1,Y-2:PLOT X
   +2,Y-1:DRAWTO X+2,Y+1
IK 1140 IF F=2 THEN PLOT X+1,Y+2:
   DRAWTO X-1,Y+2:PLOT X-2,Y+1:DR
   AWTO X-2,Y-1
MG 1150 IF F=3 THEN PLOT X-2,Y-2:
   DRAWTO X+2,Y+2:PLOT X+2,Y-2:DR
   AWTO X-2,Y+2
AT 1160 RETURN
IT 1170 REM
DE 1180 FOR C=B TO B+4:PLOT A(A)+
   C,B(A,0):DRAWTO A(A)+C,155:NEX
   T C:RETURN
JS 1190 COLOR 0:FOR C=A(A)+B+1 TO
    A(A)+B+3:PLOT C,B(A,1):DRAWTO
    C,154:NEXT C:COLOR 1
GA 1200 PLOT A(A)+B,155:DRAWTO A(
   A)+B,B(A,1):DRAWTO A(A)+B+4,B(
   A,1):DRAWTO A(A)+B+4,155
QK 1210 IF B(A,1)<152 THEN FOR C=
   B(A,1) TO 153 STEP 3:PLOT A(A)
   +B,C:DRAWTO A(A)+B+4,C:NEXT C
AJ 1220 RETURN
OQ 1230 IF B(A,2)>153 THEN 1250
TR 1240 COLOR 0:PLOT A(A)+B+1,B(A
   ,2):DRAWTO A(A)+B+1,154:PLOT A
   (A)+B+3,B(A,2):DRAWTO A(A)+B+3
   ,154:COLOR 1
KO 1250 PLOT A(A)+B,155:DRAWTO A(
   A)+B,B(A,2):DRAWTO A(A)+B+4,B(
   A,2):DRAWTO A(A)+B+4,155
BU 1260 PLOT A(A)+B+2,B(A,2):DRAW
   TO A(A)+B+2,155:RETURN
WA 1280 ? "T";:POKE 702,64:GET #
   1,A:IF A<>78 AND A<>84 AND A<>
   155 THEN 1280
QW 1285 IF A=155 THEN A=84
CU 1290 ? CHR$(A):A=(A-78)/11:RET
   URN

PN 670 REM WYKRESY - cz.8
LD 680 CLOSE #5:OPEN #5,8,0,"P:"
GN 690 TRAP 1960:DM=PEEK(88)+PEEK
   (89)*256:DM=DM+40*191
TC 700 FOR X=DM TO DM+39
```

```
KK 705 LA$=CHR$(K0):LA$(192)=LA$:
   LA$(2)=LA$
KT 710 W=USR(1536,X,ADR(LA$)):? #
   5;CHR$(27);"9";CHR$(27);"A";CH
   R$(0);CHR$(192);LA$:NEXT X
NQ 715 CLOSE #5
RO 720 TRAP 40000:GOTO 630
GK 1960 GRAPHICS 8+16+16:GOSUB 19
   30:POKE 712,64:POKE 752,1
JI 1965 ? CHR$(125);CHR$(29);CHR$
   (29);"       ***DRUKARKA ODLAC
   ZONA***"
NZ 1970 FOR X=1 TO 90:NEXT X:GRAP
   HICS 8+16+32:GOSUB 1930:GOTO 6
   30
HP 1980 IF XLABEL=87 THEN XLABEL=
   75
KT 1990 IF XLABEL=66 THEN XLABEL=
   79
CM 1995 RETURN
BV 2000 RESTORE 2030:FOR T=1 TO 6
   1:READ Q:POKE 1535+T,Q
ML 2010 NEXT T:DIM LA$(193):RETUR
   N
CW 2030 DATA 104,104,141,21,6,104
   ,141,20,6,104,141,27,6,104,141
   ,26,6,160,193,173,255,255,136,
   240,35,141,255,255,238
QQ 2040 DATA 26,6,240,21,173,20,6
   ,56,233,40,141,20,6,144,4,24,7
   6,19,6,206,21,6,76,19,6,238,27
   ,6,76,33,6,96
```

Dla lepszego poznania programu, w celu ewentualnego zmodyfikowania go, omówimy krótko funkcje realizowane przez poszczególne segmenty programu.

Linie 80—310 realizują wprowadzanie danych. W ramach tego segmentu linie 120—130 odczytują i wyprowadzają na ekran katalog zbiorów dyskietki, oczywiście tylko wtedy, gdy chcemy sprawdzić jej zawartość. Wprowadzanie danych realizowane jest w liniach 200—310. W trakcie tej czynności następuje kontrolowanie zakresu wartości danych oraz wyznaczanie wartości maksymalnej i minimalnej (potrzebne później podczas skalowania). Segment ten wprowadza dane z podanej liczby zbiorów i przechowuje je w tablicy Y. Sprawdza także, czy liczba danych we wszystkich zbiorach jest jednakowa.

Kolejną czynnością wykonywaną przez program jest skalowanie wartości w celu odpowiedniego rozmieszczenia ich na ekranie. W tym momencie znane są wartości minimalna i maksymalna. Użytkownik otrzymuje propozycję wybrania położenia osi X, od 0 do Ymin, w zależności od potrzeb. Realizują to linie programu 1530—1550. Następnie w liniach 1570—1690 dobierany jest najlepszy współczynnik skalowania. Po wyznaczeniu go użytkownik otrzymuje propozycję górnej granicy na osi Y (linie 1700—1740) i może zgodzić się z tym lub nie. W przypadku braku akceptacji możemy sami wybrać górną granicę, która musi być większa od Ymax, i następuje ponowne dobieranie współczynnika skalowania. W liniach 1760—1780 następuje przeliczenie wartości danych na współrzędne ekranu oraz wybranie symbolu oznaczającego na wykresie punkty z każdego zbioru.

Następnie program przechodzi do przygotowania opisu wykresu (linie 1800—1920). W tym celu pyta użytkownika o nazwę wykresu, oznaczenie osi Y, oznaczenie dla danych z poszczególnych zbiorów oraz sposób opisu osi X. W ostatnim przypadku możemy wybrać jedno z oznaczeń standardowych — miesiące, litery, liczby, bez opisu — lub wprowadzić własne. Mogą one mieć od 1 do 3 znaków. Poszczególne opisy osi X realizowane są przez podprogramy zawarte w liniach 1310—1510.

Po wykonaniu powyższych czynności program prezentuje na ekranie swoje możliwości (linie 350—380), które możemy wybrać wprowadzając cyfrę z przedziału 1—7. W przypadku wybrania opcji 1, 2 lub 3 przechodzimy do wykonania czynnośc przygotowawczych, tj.: wykreślenia os współrzędnych i opisania ich (linie 460—610). Jak widać dosyć często wywoływany jest podprogram rozpoczynający się od linii 740. Jest to segment (procedura) drukujący tekst na ekranie w trybie graficznym 8.

Wykres kreślony jest, w zależności od wybranej opcji, przez różne segmenty programu. I tak, wykres słupkowy kreślony jest przez instrukcje zawarte w liniach 980—1020 i 1180—1260, wykres punktowy: 1040—1070 i 1110—1160, a wykres liniowy: 1040—1090 i 1110—1160.

Po narysowaniu wykresu możemy przejść do menu lub wykonać kopię ekranu na drukarce. Informują o tym napisy w lewym dolnym rogu ekranu. Wydruk zawartości ekranu realizowany jest przez instrukcje zawarte w liniach 680—720, które wykorzystują procedurę w języku maszynowym, umieszczoną na stronie 6 przez instrukcje w liniach 2000—2040.

W programie WYKRES dostarczone są „domyślne" odpowiedzi na część pytań. Jeśli nam one odpowiadają wystarczy nacisnąć klawisz <RETURN>, jeśli nie, to należy podać własną propozycję, już bez naciskania klawisza <RETURN>. Dotyczy to głównie pytań, na które odpowiadamy TAK (T) lub NIE (N). Poprawną realizację takiego rozwiązania zapewnia podprogram umieszczony w liniach 1280—1290, który przekazuje odpowiedź w postaci wartości logicznej (A=1 jeśli TAK, A=0 jeśli NIE).

Działania programu nie można przerwać za pomocą klawisza BREAK. Zapewnia to podprogram w liniach 1930—1950, który musi być wywoływany po każdym użyciu instrukcji GRAPHICS. Istnieje jednak możliwość wyłączenia zakazu przerywania działania programu. Wystarczy uruchomić go z wciśniętym klawiszem <SELECT>. Realizują to instrukcje w linii 20.

Do wprowadzenia programu WYKRES proponujemy wykorzystać Edytor BASIC'a („IKS" nr 4 z 1987 roku).

Tomasz MROWIEC
Ludwik PIELA

CPC

Po wczytaniu i uruchomieniu programu gdy będziesz chciał wydrukować tekst z ekranu napisz CALL &A000.

```
10 REM Kopiuje tekst z ekranu na
       drukarke
20 MEMORY &9FFF:FOR x=&A000 TO
   &A03C
30 READ a$:POKE x,VAL("&"+a$):NEXT
40 DATA cd,81,bb,cd,11,bc,3c,47
50 DATA 3e,0a,cb,27,10,fc,47,0e
60 DATA 19,21,01,01,cd,75,bb,c5
70 DATA cd,60,bb,cd,2c,a0,3e,09
80 DATA cd,5a,bb,10,f3,c1,0d,c8
90 DATA c5,3e,0a,cd,36,a0,3e,0d
100 DATA cd,36,a0,c3,18,a0,cd,2b
110 DATA bd,d8,c3,2c,a0
```

Nauka • Technika

W Japonii opracowano nowego typu suchy elektrolit, który umożliwia zminiaturyzowanie akumulatora. Wytwarzany jest on w formie niezwykle cienkiej kartki, dzięki czemu sam akumulator może mieć grubość nie przekraczająca 0,1 mm. Forma elektrolitu sprawia, że nowego typu akumulator jest odporny zarówno na niskie, jak i wysokie temperatury - w granicach od -60 do +100°C. Może on być ponadto eksploatowany w próżni na statkach kosmicznych. Miniakumulatory będą mogły być powszechnie stosowane zarówno w odbiornikach radiowych i magnetofonach, jak i nowoczesnych kartach kredytowych z wbudowanymi układami scalonymi. Obecnie prowadzone są testy doboru odpowiednich elektrod do płaskiego elektrolitu

Ciąg dalszy na str. 22

PROGRAM 52 COMMODORE

Tłumacz grafiki

Przy współpracy C-64 z drukarką inną niż Commodore lub nie posiadającą interfejsu do współpracy z tym komputerem może pojawić się problem polegający na tym, że drukarka nie będzie drukować poprawnie programów zawierających grafikę C-64.

Poniższy program tłumaczy grafikę C-64 na język angielski lub kod ASCII, które „rozumie" i drukuje poprawnie każda drukarka.

```
1 PRINT 'TLUMACZ'
2 PRINT CHR$(147)CHR$(5):POKE 53281,0:FO
R X=0TO5:PRINT:NEXT
3 PRINT:PRINT TAB(17)'TLUMACZ'
4 FOR X=0TO3000:NEXT
15 CLR:OPEN 15,8,15
16 U$=CHR$(145):RT$=CHR$(18):LS$='
   [26 SPACJI]      '
17 RO$=CHR$(146):WT$=CHR$(5):RD$=CHR$(28
):N$=CHR$(0):POKE 53280,0:GOTO 700
60 INPUT#15,D1,D2$,D3,D4:RETURN
200 IF A=32 THEN A$=' ':RETURN
201 IF A=17THEN A$='[CSRS D]':RETURN
203 IF A=145THEN A$='[CRSR G]':RETURN
204 IF A=147THEN A$='[CLR/HOME]':RETURN
205 IF A=157THEN A$='[CRSR L]':RETURN
206 IF A=19THEN A$='[HOME]':RETURN
207 IF FT$='S'ANDA=13THENPRINT#1:RETURN
208 IF A=13THENA$='[RETURN]':RETURN
209 IFA=14 THENA$='[LOWER CASE]':RETURN
210 IF A=5THENA$='[WHT]':RETURN

211 IF A=18THENA$='[REV ON]':RETURN
212 IF A=20THEN A$='[DEL]':RETURN
213 IF A=28THEN A$='[RED]':RETURN
214 IF A=30THEN A$='[GREEN]':RETURN
215 IF A=31THEN A$='[BLUE]':RETURN
216 IF A=141THEN A$='[SHIFT RETURN]':RET
URN
217 IFA=142THENA$='[UPPER CASE]':RETURN
218 IF A=144THENA$='[BLACK]':RETURN
219 IF A=146THEN A$='[REV OFF]':RETURN
220 IF A=148THEN A$='[INST]':RETURN
221 IF A=156THEN A$=[PUR]':RETURN
222 IF A=158THEN A$='[YEL]':RETURN
223 IF A=159THEN A$='[CYN]':RETURN
224 IF A=255THEN A$='[PI]':RETURN
225 IF A=133THEN A$='[F1]':RETURN
226 IF A=134THEN A$='[F3]':RETURN
227 IF A=135THEN A$='[F5]':RETURN
228 IF A=136THEN A$='[F7]':RETURN
229 IF A=137THEN A$='[F2]':RETURN
230 IF A=138THEN A$='[F4]':RETURN
231 IF A=139 THEN A$='[F6]':RETURN

232 IF A=140 THEN A$='[F8]':RETURN
233 IF A=160 THEN A$=' ':RETURN
234 IF A=151 THEN A$='[GREY 1]':RETURN
235 IF A=152 THEN A$='[GREY 2]':RETURN
236 IF A=153 THEN A$='[GREY 3]':RETURN
237 IF A=129 THEN A$='[ORANGE]':RETURN
238 IF A=150 THEN A$='[LT RED]':RETURN
239 IF A=153 THEN A$='[LT GREEN]':RETURN
240 IF A=154 THEN A$='[LT BLUE]':RETURN
241 IF A=149 THEN A$='[BROWN]':RETURN
250 X$=STR$(A):X$=MID$(X$,2,3):A$='[CHR$
('+X$+')]':RETURN
300 IF A=153 THEN A$='PRINT':RETURN
301 IF A=151 THEN A$='POKE':RETURN
302 IF A=152 THEN A$='PRINT#':RETURN
303 IF A=129 THEN A$='FOR':RETURN
304 IF A=131 THEN A$='DATA':RETURN
305 IF A=137 THEN A$='GOTO':RETURN
306 IF A=139 THEN A$='IF':RETURN
307 IF A=141 THEN A$='GOSUB':RETURN
308 IF A=142 THEN A$='RETURN':RETURN
309 IF A=163 THEN A$='TAB(':RETURN
310 IF A=164 THEN A$='TO':RETURN
311 IF A=175 THEN A$='AND':RETURN
312 IF A=176 THEN A$='OR':RETURN
313 IF A=199 THEN A$='CHR$':RETURN
314 IF A=140 THEN A$='RESTORE':RETURN
315 IF A=170 THEN A$='+':RETURN
316 IF A=171 THEN A$='-':RETURN
317 IF A=172 THEN A$='*':RETURN
318 IF A=173 THEN A$='/':RETURN
319 IF A=174 THEN A$='[']':RETURN
320 IF A=177 THEN A$='>':RETURN
321 IF A=178 THEN A$='=':RETURN
322 IF A=179 THEN A$='<':RETURN
323 IF A=194 THEN A$='PEEK':RETURN
324 IF A=167 THEN A$='THEN':RETURN
325 IF A=133 THEN A$='INPUT':RETURN
326 IF A=132 THEN A$='INPUT#':RETURN
327 IF A=143 THEN A$='REM':RETURN
328 IF A=159 THEN A$='OPEN':RETURN
329 IF A=160 THEN A$='CLOSE':RETURN

330 IF A=169 THEN A$='STEP':RETURN
331 IF A=201 THEN A$='RIGHT$':RETURN
332 IF A=147 THEN A$='LOAD':RETURN
333 IF A=203 THEN A$='GO':RETURN
340 IF A=128 THEN A$='END':RETURN
341 IF A=130 THEN A$='NEXT':RETURN
342 IF A=134 THEN A$='DIM':RETURN
343 IF A=135 THEN A$='READ':RETURN
345 IF A=138 THEN A$='RUN':RETURN
346 IF A=144 THEN A$='STOP':RETURN
347 IF A=145 THEN A$='ON':RETURN
348 IF A=148 THEN A$='SAVE':RETURN
349 IF A=149 THEN A$='VERIFY':RETURN
350 IF A=150 THEN A$='DEF':RETURN
351 IF A=154 THEN A$='CONT':RETURN
352 IF A=155 THEN A$='LIST':RETURN
353 IF A=156 THEN A$='CLR':RETURN
354 IF A=157 THEN A$='CMD':RETURN
355 IF A=197 THEN A$='VAL':RETURN
356 IF A=161 THEN A$='GET':RETURN
357 IF A=165 THEN A$='FN':RETURN

358 IF A=166 THEN A$='SPC(':RETURN
359 IF A=168 THEN A$='NOT':RETURN
360 IF A=180 THEN A$='SGN':RETURN
361 IF A=181 THEN A$='INT':RETURN
362 IF A=182 THEN A$='ABS':RETURN
363 IF A=184 THEN A$='FRE':RETURN
364 IF A=184 THEN A$='FRE':RETURN
365 IF A=185 THEN A$='POS':RETURN
366 IF A=186 THEN A$='SQR':RETURN
367 IF A=187 THEN A$='RND':RETURN
368 IF A=188 THEN A$='LOG':RETURN
369 IF A=189 THEN A$='EXP':RETURN
370 IF A=190 THEN A$='COS':RETURN
371 IF A=191 THEN A$='SIN':RETURN
372 IF A=192 THEN A$='TAN':RETURN
373 IF A=193 THEN A$='ATN':RETURN
374 IF A=195 THEN A$='LEN':RETURN
375 IF A=196 THEN A$='STR$':RETURN
376 IF A=158 THEN A$='SYS':RETURN
377 IF A=198 THEN A$='ASC':RETURN
378 IF A=200 THEN A$='LEFT$':RETURN
379 IF A=202 THEN A$='MID$':RETURN

380 IF A=146 THEN A$='WAIT':RETURN
381 IF A=162 THEN A$='NEW':RETURN
382 IF A=136 THEN A$='LET':RETURN
383 IF A=255 THEN A$='[PI]':RETURN
390 X$=STR$(A):X$=MID$(X$,2,3):A$='CHR$(
'+X$+')':RETURN
700 PRINTCHR$(147):PRINT:PRINT:PRINT:PRI
NT:PRINT
730 NF$='':INPUT'NAZWA PLIKU';NF$
732 IF LEN(NF$)<1 OR LEN(NF$)>16 THEN PR
INT U$U$:GOTO 730
740 FT$='':PRINT'RODZAJ PLIKU:'RT$'P'RO$
'RG LUB 'RT$'S'RO$'EQ';:INPUT FT$
742 IF FT$<>'P' THEN IF FT$<>'S'THEN PRI
NT U$U$:GOTO 740
750 FD$='':PRINT'WYDRUK NA'RT$'E'RO$'KRA
N LUB 'RT$'D'RO$'RUKARKE';:INPUT FD$
752 IF FD$<>'E'THEN IF FD$<>'D'THEN PRIN
T U$U$:GOTO 750
754 IF FD$='E' THEN FD=3
756 IF FD$='D' THEN FD=4
760 SK=0:PRINT'ILE BITOW DO POMINIECIA
'RT$'(RETURN)'RO$' =0';:INPUT SK
800 PRINT CHR$(147):FOR X=0TO4:PRINT
810 PRINT RT$'POSZUKIWANIE: 'NF$' ,'FT$
830 IF FD=3 THEN TY$='EKRAN'
835 IF FD=4 THEN TY$='DRUKARKE'
840 PRINT TAB(3)'WYPROWADZENIE NA: 'TY$
845 PRINT TAB(5)RT$' < F1 = PRZERWA >'
850 CLOSE 8:OPEN 8,8,8,'0:'+NF$+','+FT$+
',R'
852 IF SK>0 THEN FOR X=0TO SK:GET#8,A$:N
EXT:X=0
855 CLOSE 1:OPEN 1,FD
856 PRINT#1,NF$
860 GOSUB 60:IF A>0 THEN PRINT CHR$(147)
D1;D2$,D3;D4
865 IF FT$='P'THEN GET#8,A$,B$
870 IF FT$='P'THEN GET#8,A$,B$
874 IF FT$='S' THEN 905
875 GET#8,C$,D$:F1=ASC(C$+N$):F2=ASC(D$+
N$)
880 IF ST=0 THEN PRINT#1,(F2*256)+F1;
890 GET#8,A$:A=ASC(A$+N$):IFA>0THENC=0
891 GETBK$:IFBK$=CHR$(133)THEN GOTO1000
892 IFA=0THENFL=0:PRINT#1:C=C+1:GOTO870
893 IF ST=0THEN 900
895 IF ST=64 THEN 1000
896 PRINT'STAN : 'ST:GOTO 1000
900 IF FL=0 AND A=34 THEN FL=1:GOTO905
901 IFFL=1ANDA>93THENGOSUB 200:GOTO 910
902 IF A=34 THEN FL=0
903 IF A<33 THEN GOSUB 200:GOTO 910
904 IF A>93 THEN GOSUB 300:GOTO 910
905 PRINT#1,CHR$(A);:A=0:GOTO 890
910 PRINT#1,A$;:GOTO 890
1000 PRINT#1:GOSUB 60
1001 PRINTCHR$(5):CLOSE1:CLOSE8:CLOSE15
1010 IF D1>0 THEN PRINT'BLAD-'
1011 PRINT'STAN DYSKIETKI:'D1;D2$,D3;D4
1015 PRINT:PRINT'CZY CHCESZ SPROBOWAC JE
SZCZE RAZ ?(T/N)'
1020 GETQ$:IF Q$='' THEN 1020
1030 IF Q$='T' THEN CLR:GOTO 15

1040 END
READY.
```

Program może współpracować również z VIC-20, należy tylko opuścić lub odpowiednio zmienić linie 1 i 20 (ustalenie koloru tablicy i ramki).

Z programu mogą korzystać (niestety) tylko posiadacze stacji dysków. Tłumaczyć (listować) i drukować można za jego pomocą pliki programowe (PRG) i sekwencyjne (SEQ), zarówno na ekran, jak i na drukarkę. Możliwy jest także skok do środka pliku przed przystąpieniem do drukowania, co pozwala na zaoszczędzenie papieru.

Działanie programu możemy w dowolnej chwili przerwać klawiszem F1.

Program może być bardzo przydatny dla osób, które:

— posiadają drukarkę inną niż Commodore,

— posiadają drukarkę Commodore, ale chcą lepiej poznać sposób zapisu znaków graficznych,

— mają trudności w zrozumieniu grafiki Commodore i nie posiadają drukarki.

Opracował: Tadeusz CISEK

Wskaźniki i funkcje w języku „C”

H. KRASUSKI Z. DYJAK

Język C staje się coraz częściej stosowanym narzędziem programowania. Jego użytkownicy podkreślają wiele zalet tego języka, takich jak: szczególna przydatność do tworzenia oprogramowania systemowego, możliwość definiowania i przetwarzania nowoczesnych struktur danych oraz prostota i zwartość budowanych programów. **Inną jego zaletą, być może najistotniejszą jest to, że nie jest on związany z żadnym konkretnym sprzętem i napisane w nim programy mogą być uruchamiane na dowolnym komputerze, dla którego opracowano kompilator tego języka.**

Intensywny wzrost liczby mikrokomputerów użytkowanych zarówno przez amatorów, jak i profesjonalistów znacznie przybliża ich użytkownikom możliwość programowania w języku C, tym bardziej że jego kompilatory dostępne są w zestawie oprogramowania prawie każdego systemu mikrokomputerowego, w tym również mikrokomputerów domowych, jak AMSTRAD czy ATARI. Warto zatem zainteresować się bliżej tym językiem.

Język C jest językiem programowania ogólnego zastosowania. Jednym z celów jego zaprojektowania było wypełnienie luki istniejącej pomiędzy takimi językami jak BASIC a językiem asemblera. Język C stworzony przez programistów systemowych stanowi wysokopoziomową alternatywę dla języków asemblerowych. Jest on znacznie szybszy i oferuje programiście więcej możliwości niż BASIC (np. wykonywanie operacji logicznych na pojedynczych bitach, tworzenie programów strukturalnych), a jednocześnie jest mniej „błędotwórczy” i bardziej wydajny niż język asemblera. Programy napisane w języku C, mimo iż działają nieco wolniej niż programy w kodzie maszynowym, są jednak znacznie łatwiejsze w tworzeniu i testowaniu. Takie elementy języka jak: wskaźniki, funkcje rekurencyjne i wygodne w użyciu struktury sterujące czynią nawet potencjalnie skomplikowany program łatwym do zakodowania i przetestowania. Co więcej, programy w nim napisane, w przeciwieństwie do programów w języku asemblera mogą być przeniesione na dowolny inny komputer kosztem naprawdę niewielkich modyfikacji.

Programiści poznający język C, w sytuacji gdy wcześniej posługiwali się takimi językami, jak: PASCAL, FORTRAN, czy BASIC są początkowo zaskoczeni zastosowaniem w nim kilku rozwiązań, które trudno znaleźć w innych językach programowania. Szczególnie dwa z nich zasługują na bliższe przedstawienie — wskaźniki i funkcje, w tym również funkcje biblioteczne.

Funkcje biblioteczne w języku C stanowią zbiór procedur dostarczanych z kompilatorem (biblioteka kompilatora), realizujących operacje wejścia/wyjścia programu, obliczenia numeryczne itp. Można zatem zapytać: cóż jest odmiennego w bibliotece funkcji języka C w odniesieniu do innych języków programowania? Różnica polega na tym, że w innych językach programowania instrukcje realizujące wyżej wymienione operacje stanowią element języka, nieodłącznie z nim związany. Natomiast w języku C funkcje nie są elementem języka. Na przykład programując w języku BASIC można użyć instrukcji PRINT do wyprowadzenia danych na urządzenie zewnętrzne, mając świadomość, że instrukcja ta jest elementem tego języka. Natomiast w języku C nie ma instrukcji przeznaczonej do wyprowadzania danych, mało tego, w ogóle nie ma w nim instrukcji przeznaczonych do obsługi tzw. wejścia/wyjścia programu. Jak zatem takie operacje realizować w tym języku? Otóż do wykonywania tych operacji służą w języku C funkcje zawarte w bibliotece kompilatora. Każdy kompilator języka C zawiera bibliotekę funkcji zdefiniowanych przez twórców języka oraz pewną liczbę funkcji specyficznych dla konkretnej implementacji sprzętowej. Na przykład: kompilator języka C opracowany dla mikrokomputerów ATARI XL/XE (DEEP BLUE C /DBC/) zawiera bibliotekę funkcji realizujących arytmetykę zmiennoprzecinkową. Jest to biblioteka specyficzna dla tego kompilatora, której istnienie spowodowane jest niestandardowym sposobem wykonywania obliczeń zmiennoprzecinkowych na tym sprzęcie. W kompilatorach dedykowanych dla mikrokomputerów, w których zastosowano standardowe liczby zmiennoprzecinkowe taka biblioteka w ogóle nie istnieje. Obok niej kompilator DBC posiada standardowe funkcje języka C takie jak: printf[], strlen[], getchar[], putchar[], tolower[], toupper[] i inne. Funkcje o takich nazwach występują we wszystkich kompilatorach języka C, bez względu na sprzęt, na który je opracowano. Konsekwencją faktu, że funkcje nie są elementami języka C są drobne różnice w sposobie ich realizacji na konkretnym typie mikrokomputera.

Jednak funkcje biblioteczne nie są tematem niniejszego artykułu. Konstrukcja całego programu w języku C oparta jest o idee funkcji. Każdy program w tym języku musi zawierać przynajmniej jedną funkcję. Oczywiście może również zawierać ich wiele. Ogólnie, funkcjami w języku C nazywane są, definiowane przez programistę, procedury lub podprogramy realizujące określoną operację, na przykład: wyprowadzenie wyników obliczeń na ekran, obliczenia wartości wyrażenia itp. Funkcje mogą być oddzielnie kompilowane i testowane, a następnie w postaci półskompilowanej dołączane do programu podczas fazy linkowania. Ta cecha języka stanowi decydujący czynnik jego „przewagi” nad innymi językami programowania oraz w znacznym stopniu ułatwia i przyspiesza proces tworzenia programu. Jej istotę można zobrazować następującym przykładem: każdy dobrze skonstruowany program przed zakończeniem działania powinien zapytać użytkownika, czy naprawdę chce zakończyć pracę. W przypadku negatywnej odpowiedzi program powinien uruchomić się ponownie. Przy użyciu takich języków programowania, jak BASIC czy PASCAL zakodowanie takiej ewentualności w programie wymaga kilku lub kilkunastu instrukcji. Niestety instrukcji tych musimy ponownie użyć pisząc nowy program, jeśli chcemy aby kończył on się podobnie. Jak problem ten można rozwiązać w języku C? Piszemy funkcję, która będzie realizowała zadanie pytania i analizowanie odpowiedzi użytkownika. Przykładem takiej funkcji jest funkcja o nazwie czydalej[] zamieszczona w programie i (patrz dalsza część artykułu). Funkcję tę kompilujemy i testujemy, a jej kod półskompilowany wprowadzamy do biblioteki własnych funkcji. W każdym nowym programie, gdy będziemy chcieli użyć tej funkcji, to wystarczy ją tylko wywołać, natomiast nie ma konieczności pisania jej ponownie. Oczywiste jest, że jej kod półskompilowany musimy dołączyć podczas linkowania do każdego programu, w którym funkcję tę będziemy wywoływać. To zaś nie stanowi żadnego problemu, gdyż proces linkowania wykonywany jest przez specjalny program dołączony do kompilatora.

Stałe i zmienne programowe, na których funkcja wykonuje operacje nazywane są w języku C argumentami funkcji. Liczba argumentów, które można przekazać do funkcji jest zależna od implementacji sprzętowej kompilatora. Praktycznie jest to liczba dosyć duża, np.: wspomniany wyżej kompilator DBC umożliwia przekazanie do funkcji aż 126 argumentów. Argumenty przekazywane są do funkcji poprzez wartość i ich wartość nie może być zmieniona przez funkcję. Oznacza to, że każda funkcja tworzy własne, prywatne kopie wartości argumentów, na których wykonuje operacje. Kopie te są niszczone w momencie, gdy funkcja wykona przypisane jej zadanie. Wynik obliczeń wykonanych przez funkcję nazywany jest „wartością zwracaną przez funkcję”. Każda funkcja w języku C może zwrócić do miejsca wywołania tylko jedną wartość. W tym miejscu można postawić pytanie: jak zatem przetwarzać tablice za pomocą funkcji? Z powyższego wynika, że przekazanie do funkcji, jako argumentów, elementów tablicy jest praktycznie niemożliwe. I owszem, jest to prawda, gdy chcielibyśmy przekazać do funkcji wartości poszczególnych elementów tablicy. Jednak problem ten można rozwiązać inaczej. Po prostu zamiast przekazywać do funkcji wartości elementów tablicy, można przekazać jako argument funkcji, adres początku tej tablicy. W ten sposób doszliśmy do pojęcia „wskaźnik”.

Wskaźnikiem w języku C jest zmienna, zawierająca adres innej zmiennej. Sposób operowania zmiennymi programu za pomocą ich adresów jest początkowo szokujący dla programisty przyzwyczajonego do używania zmiennych poprzez ich nazwy, jak ma to miejsce w językach wysokopoziomowych. Jednak mechanizm wskaźników, co pokażemy w dalszej części, w niektórych sytuacjach jest znacznie wygodniejszy w stosowaniu i umożliwia tworzenie bardziej zwartego i efektywnego tekstu źródłowego programu.

Język C oferuje programiście również klasyczny sposób przetwarzania tablic, polegający na odwoływaniu się do ich elementów poprzez indeksy. Taką metodę przetwarzania tablic zastosowano w programie 1. Należy w tym miejscu zaznaczyć, że wszystkie programy

zawarte w niniejszym artykule zostały napisane przy wykorzystaniu kompilatora Deep Blue C przeznaczonego dla mikrokomputerów ATARI XL/XE.

Program 1 realizuje funkcję przestawiania elementów tablicy znakowej (łańcucha) w taki sposób, że pierwszy znak staje się ostatnim, drugi przedostatnim, itd. Łańcuch wczytywany jest z klawiatury do tablicy wyraz [120] za pomocą funkcji gets[] z biblioteki AIO kompilatora DBC. Operacja przestawiania znaków realizowana jest w pętli:

```
for(i=0, j=dl—i;i<j; i++,j——)$(
        znak=wyraz[i];
      wyraz[i]=wyraz [j];
       wyraz[j]=znak;
            $);
```

i, j są indeksami „maszerującymi" wzdłuż tablicy wyraz [], i — od jej początku do końca, j — od końca do początku. Zmienna dl zawiera długość tablicy wyraz [] (liczbę znaków). Cały program składa się z dwóch funkcji main[] i czydalej []. Zadaniem funkcji czydalej[] jest tylko eleganckie zakończenie programu. Natomiast zasadnicza część

Program 1

```
/*
 * program Przestawianka
 */
main()$(
char znak,wyraz[120];
int i,j,dl;
 do$(
 printf("\nNapisz dowolny wyraz:\n");
 dl=gets(wyraz);
 if(dl<=0) continue;
 printf("\nNapisales:\n");
   for(i=0;i<=dl-1;i++)
        printf(" %c",wyraz[i]);
/* teraz przestaw znaki we wprowadzonym wyrazie */
   for(i=0,j=dl-1;i<j;i++,j--)$(
     znak=wyraz[i];
     wyraz[i]=wyraz[j];
     wyraz[j]=znak;
    $)   /* koniec petli for  */
/* wyswietl przestawiony wyraz   */
 printf("\nA tak to wyglada po przestawieniu\n");
   for(i=0;i<=dl-1;i++)
      printf(" %c",wyraz[i]);
  $)   /* koniec do  */
 while(czydalej());
$)  /* koniec main  */
/*
 * funkcja czydalej()
 */
czydalej()$(
int stat,odp,T,t,N,n;
char *u,od,zer,i;
i='1';
od='4';
zer='0';
u="K:";
T=84;   /* kod litery T */
t=116;  /* kod litery t */
N=78;   /* kod litery N */
n=110;  /* kod litery n */
stat=open(i,od,zer,u);    /* otwarcie "K:" z iocb 1    */
if(stat!=1)$(
   printf("\n\gBlad otwarcia K: ");
   return;
$)     /* koniec if   */
do$(   /* poczatek do */
printf("\nCzy chcesz kontynuowac?");
odp=cgetc(i);
if(odp!=T && odp!=t && odp!=N && odp!=n)$(
  printf("\n\gOdpowiedz albo T albo N");
  continue;
 $)  /* koniec if */
$)    /* koniec do */
while(odp!=T && odp!=t && odp!=N && odp!=n);
close(i);
return((odp==T || odp==t) ? 1 : 0);
$) /* koniec czydalej() */
```

programu tj.: wczytywanie łańcucha znaków z klawiatury, przestawianie jego elementów oraz wyświetlanie go w postaci wczytanej i po przestawieniu realizowana jest w funkcji main[]. Widać, że pętla ta poza odmiennym sposobem jej zorganizowania (for(...)) nie różni się od pętli realizujących podobną funkcję, tworzonych na przykład w języku BASIC.

Załóżmy, że w tworzonych programach będziemy musieli dosyć często wykonywać operacje przestawiania elementów tablicy znaków. Dobrze byłoby zatem przenieść operację przestawiania do funkcji, której argumentami byłby adres tablicy i liczba jej elementów. Nazwiemy tę funkcję przestaw[]. Wówczas, w miejsce omówionej wyżej pętli możemy umieścić jej wywołanie postaci:

przestaw(tablica, liczbael);

Aby taką funkcję napisać musimy posłużyć się wskaźnikami.

Zmienne wskaźnikowe lub krótko wskaźniki, podobnie jak wszystkie inne zmienne w języku C, muszą być zadeklarowane przed ich użyciem. Każdy wskaźnik posiada typ związany z obiektem (zmienną, tablicą), który wskazuje. Deklaracja: char* U; nakazuje kompilatorowi języka C utworzenie wskaźnika typu znakowego, tzn. takiego, który wskazywał będzie obiekty tego samego typu. Natomiast deklaracja: int* i; nakazuje kompilatorowi utworzenie wskaźnika typu całkowitoliczbowego, tzn., że wskazywane przez niego obiekty są zmiennymi lub elementami tablic tego samego typu. Sam fakt zadeklarowania w programie zmiennych wskaźnikowych nie wiąże jeszcze wskaźnika z tablicą. Aby wskaźnik można było w programie sensownie wykorzystać musi on zostać prawidłowo zainicjowany, tzn. musi mu zostać przypisana wartość będąca adresem jednego z obiektów programu. Używanie wskaźników bez nadania im wartości początkowych jest częstym błędem popełnianym przez początkujących programistów w języku C. Wskaźniki są dla programisty alternatywną metodą przetwarzania dużych obszarów danych. Alternatywną, gdyż jak wynika z przedstawionego programu i obszary takie można przetwarzać stosując metodę indeksowania elementów, z których składają się te obszary. Kompilatory języka C realizując deklarację zmiennej wskaźnikowej nie kontrolują, czy został zadeklarowany obiekt, który ma być wskazywany przez wskaźnik. Stąd tak istotne jest poprawne inicjowanie wartości wskaźników.

Jeżeli umieścimy w programie deklarację:

char tablica [100]

i zadeklarujemy wskaźnik tego samego typu:

char * wsk;

to instrukcja:

wsk = & tablica[0]

inicjuje wskaźnik wsk na zerowy element tablicy, innymi słowy: przypisuje wskaźnikowi wsk adres zerowego (elementy tablic w języku C numerowane są od zera) elementu tablicy. Poprawne w języku C jest również inicjowanie wskaźnika w taki sposób:

wsk = & tablica[i]; gdzie i — zmienna typu int

Operator „&" jest operatorem adresu, powodującym przypisanie zmiennej wskaźnikowej wartości będącej adresem zmiennej stojącej po jego prawej stronie.

W języku C nazwa tablicy jest wskaźnikiem. Kompilator C każde odwołanie do elementu tablicy realizuje posługując się wskaźnikiem do jej początku. Zatem poprawne jest również zainicjowanie wskaźnika instrukcją przypisania:

wsk=tablica;

należy zwrócić uwagę, że w tym przypadku w instrukcji została użyta nazwa tablicy bez nawiasów [..], w których podaje się wartość indeksu elementu tablicy.

Jeżeli w programie zadeklarujemy tablice i zmienną wskaźnikową o takiej samej nazwie, np.:

```
char tablica[100]
 char * tablica;
```

Funkcja 1

```
/*
 *  funkcja  przestaw()
 */
przestaw(wyraz,dlugosc)
char *wyraz;
int dlugosc;
$( /* poczatek funkcji */
char znak;
char *koniec;
dlugosc-=1;
koniec=wyraz+dlugosc;
for(;wyraz<koniec;wyraz++,koniec--)$(
 znak=*wyraz;
 *wyraz=*koniec;
 *koniec=znak;
 $) /* for */
return;
$) /* koniec funkcji przestaw() */
```

to ostatnia deklaracja jest jednocześnie poprawnym zainicjowaniem wskaźnika o nazwie tablica (wskazuje on początkowy element tablicy o tej samej nazwie).

Z faktu, że nazwa tablicy jest traktowana w języku C jak wskaźnik wynika, że deklaracja wskaźnika jest równoważna deklaracji tablicy. NP.: jeżeli zadeklarujemy wskaźnik

char * kom;

to jest to równoważne deklaracji tablicy o nieokreślonej liczbie elementów. Oczywistym jest, że po takiej deklaracji wskaźnik ten je-

program 2

```
/*
 * program  Przestawianka
 */
main()${
char znak,wyraz[120];
int i,j,dl;
 do${
 printf("\nNapisz dowolny wyraz:\n");
 dl=gets(wyraz);
 if(dl<=0) continue;
printf("\nNapisales:\n");
   for(i=0;i<=dl-1;i++)
        printf(" %c",wyraz[i]);
/* teraz przestaw znaki we wprowadzonym wyrazie */
 przestaw(wyraz,dl);
/* wyswietl przestawiony wyraz   */
 printf("\nA tak to wyglada po przestawieniu\n");
   for(i=0;i<=dl-1;i++)
      printf(" %c",wyraz[i]);
  $)   /* koniec do  */
 while(czydalej());
$)  /* koniec main  */
```

szcze niczego nie wskazuje. Instrukcja: kom=„Naciśnij dowolny klawisz" wprowadzi do tego wskaźnika adres tekstu zawartego pomiędzy znakami „...".

Funkcja przestaw[] napisana z wykorzystaniem zmiennych wskaźnikowych ma postać:

Funkcja ta wykorzystuje dwa wskaźniki o nazwach wyraz i koniec. Wskaźnik wyraz wskazuje początek tablicy o nazwie wyraz, a wskaźnik koniec — koniec też i tablicy. Zatem wskaźniki te pełnią podobną rolę jak indeksy i, j w programie

1. Instrukcja:

znak=*wyraz;

przypisuje zmiennej o nazwie znak element tablicy wyraz wskazywany przez wskaźnik wyraz;

natomiast instrukcja:

wyraz= *koniec;

przypisuje elementowi tablicy wskazywanemu przez wskaźnik wyraz element wskazywany przez wskaźnik koniec, czyli przy pierwszym obiegu pętli do pierwszego elementu tablicy zostanie wprowadzony ostatni, w drugim przebiegu pętli do drugiego elementu zostanie wprowadzony przedostatni, itd.

Z analizy działania tej funkcji wynika, że na wskaźnikach można wykonywać operacje arytmetyczne, jest to zgodne z prawdą. Wskaźniki są adresami. Zatem wykonywanie na nich operacji dodawania i odejmowania jest wykonywaniem operacji na adresach. Wyrażenia w pętli for[] wyraz++ i koniec—— zwiększają i zmniejszają o jeden odpowiednio wskaźniki wyraz i koniec. To zwiększanie o jeden nie należy interpretować jako matematyczne dodanie jedynki. Operacja zwiększenia (zmniejszenia) wskaźnika o jeden oznacza, że po jej wykonaniu wskazuje on następny (poprzedni) element tablicy. Mając to na uwadze nie wolno w programie w języku C dodawać lub odejmować od siebie wskaźników różnych typów np.: char i int. Wynika to z faktu, że zmienne typu int i char są różnej długości.

Możemy teraz napisać nową postać programu 1, używając skonstruowanej funkcji przestaw[].

Reasumując, należy stwierdzić, że wskaźniki i funkcje są elementami języka C, które decydują, że język ten posiada wiele walorów w porównaniu z innymi językami wysokopoziomowymi. Funkcje z możliwością ich oddzielnego kompilowania i tworzenia z ich kodów półskompilowanych bibliotek, są szczególnie cennym mechanizmem. Mechanizm ten stanowi niejako możliwość definiowania własnych, specyficznych „instrukcji", które raz zdefiniowane mogą być zawsze używane.

Wskaźniki natomiast, w połączeniu z cechami instrukcji sterujących języka C, umożliwiają tworzenie bardzo zwartego kodu źródłowego programu. Możliwą do uzyskania zwartość ilustruje przedstawiana prawie we wszystkich publikacjach na temat tego języka instrukcja:

while(*s++=*t++);

która kopiuje łańcuch znaków t do łańcucha s. Czy można zaprogramować to krócej w innym języku?

H. KRASUSKI
Z. DYJAK

— Ale nie ma takiego doświadczenia w wagarowaniu, jak my!

Nauka • Technika

W Japonii opracowano nowy typ komputera optycznego (określony mianem neuro-komputera), który jest w stanie nauczyć się jak odróżniać obrazy. Jego budowa oparta jest na układach elektronicznych skonstruowanych w ten sposób, że imitują one strukturę neuronów ludzkiego mózgu. Nowy komputer optyczny, znajdujący się dopiero w początkowym stadium rozwoju, jest w stanie rozpoznać znaki i kształty, a następnie w miarę dokładnie je odwzorować. W początkowym stadium uczenia się komputer optyczny odtwarza znaki czy kształty w sposób niedokładny. Jednakże po przeanalizowaniu różnic pomiędzy znakami 'na wejściu' i 'na wyjściu' komputer koryguje swoją pamięć i ostatecznie odwzorowuje właściwe znaki. W dodatku, gdy znak został całkowicie zapamiętany przez komputer, jest on w stanie wybrać ten sam znak spośród innych, nawet jeśli jest on nieco zniekształcony. Ta funkcja kojarzenia, zdaniem japońskich naukowców, oznacza, że w przyszłości udoskonalone komputery optyczne będą mogły czytać ręcznie pisane znaki.

Ciąg dalszy na str. 22

Awaryjne przerwanie programu ZX

Często podczas działania programu przewidujemy opcję jego zatrzymania (przerwania) przez naciśnięcie określonego klawisza.

W Basic'u uzyskujemy to przez kontrolę klawiatury rozkazem INKEY$. Ta metoda ma jednak swoje wady: przede wszystkim reakcja programu na naciśnięty klawisz istnieje tylko podczas wykonywania instrukcji INKEY$, to znaczy, że aby reakcja na naciśnięty klawisz była szybka, to ta instrukcja musi wykonywać się często. Po drugie: program jest wolniejszy, gdyż każde wykonanie instrukcji INKEY$ pochłania określony odcinek czasu. Stąd jest to metoda w wielu przypadkach nieekonomiczna.

Istnieje jednak inny sposób wykorzystujący fakt, że co 20 ms komputer kontroluje klawiaturę wykorzystując przerwania w trybie pierwszym (IM 1).

Prezentowany program pracuje w trybie drugim (IM 2). Co 20 ms następuje nie tylko kontrola klawiatury, ale również test, czy naciśnięto zadeklarowany klawisz. Jeśli tak, to zostaje wymuszony skok do określonej przez uzytkownika linii programu. Skok do zadanej linii programu uzyskuje przez wstawienie do zmiennej systemowej NEWPPC numeru tej linii.

Procedura nie działa podczas wykonywania rozkazów GO TO, GO SUB, RETURN ponieważ instrukcje te zmieniają wartość NEWPPC. Nie jest to jednak poważne ograniczenie, gdyż zmienna systemowa LAST_K pamięta ostatnio wciśnięty klawisz i zaraz po wykonaniu wyżej wymienionych rozkazów nastąpi przerwanie.

Współpraca z programem:

RANDOMIZE USR 65529 — włącza tryb pierwszy przerwań, wyłącza działanie programu

RANDOMIZE USR 65517 — włącza tryb drugi przerwań, uruchamia program

komórki o adresach 65526 i 65527 zawierają numer linii w Basic'u, do której nastąpi skok podczas wymuszenia przerwania

komórka o adresie 65528 zawiera kod znaku klawisza wymuszającego przerwanie

UWAGA: jeżeli nastąpi wymuszenie przerwania do nieistniejącej linii, wtedy nastąpi przerwanie wykonywania programu i ukaże się komunikat systemu.

Krzysztof POŹNIAK

```
 10 REM **************************
 11 REM
 12 REM     Krzysztof Poźniak
 13 REM
 14 REM   awaryjne przerwanie
 15 REM         programu
 16 REM
 17 REM                    ©1988
 18 REM
 19 REM **************************

 100 CLEAR 65488
 110 LET s=0
 111 FOR k=65489 TO 65535
 112 READ l
 113 LET s=s+l
 114 POKE k,l
 115 NEXT k
 116 IF s<>5279 THEN BEEP 1,10:
PRINT "zle dane!!": STOP
 120 BEEP 1,10
 125 LET nrlinii=9000
 126 POKE 65526,nrlinii-256*INT
(nrlinii/256): POKE 65527,INT (n
rlinii/256)
 130 PRINT : PRINT "nacisnij ten
 klawisz, ktorym    zatrzymasz P
rogram"
 135 PAUSE 0
 140 LET x$=INKEY$
 145 POKE 65528,CODE x$
 150 PRINT "teraz program przerw
iesz klawiszem: ";x$;" i uruchom
isz od linii ";nrlinii;" obsluge
 przerwania"
 151 FOR k=1 TO 100: NEXT k: PRI
NT "wcisnij ENTER": PAUSE 0:
 155 RANDOMIZE USR 65517
 200 FOR k=1 TO 10000: PRINT AT
10,10;"licznik :";k: NEXT k: STO
P
1000 DATA 205,56,0,245,229,33,24
8,255,58,8,92,190,32,11,42,246,2
55,34,66,92,62,1,50,68,92,225,24
1,201,62,59,237,71,237,94,201,24
,219,40,35,115,237,86,201,0,0,0,
24,5279
9000 RANDOMIZE USR 65529: PRINT
: PRINT "zostala uruchomiona obs
luga przerwania"
```

```
00012  ; program wykorzystuje przerwa-
00013  ; nia maskowalne, ustawia tryb 2
00014  ; przerwan.
00015  ; program nie jest relokowalny.
00016  ;
00017  rej.i  equ  59 ;wartosc rejestru
00018  ;              przerwan
00019  ;
00020  ; adres procedury ROM-u
00021  keyb   equ  38h ;kontrola kla-
00022  ;               wiatury
00023  ;
00024  ; adresy zmiennych systemowych
00025  newppc equ  5c42h
00026  nsppc  equ  5c44h
00027  last_k equ  5c08h
00028  ;
00029          org  65489
00030  ;
00031  ; rozszerzenie przerwania
00032  ; program wykonywany w czasie
00033  ; przerwania
00034  przerw call keyb
00035          push af
00036          push hl
00037          ld   hl,adrznk
00038          ld   a,(last_k)
00039          cp   (hl)
00040          jr   nz,koniec
00041          ld   hl,(nrlin)
00042          ld   (newppc),hl
00043          ld   a,1
00044          ld   (nsppc),a
00045  koniec  pop  hl
00046          pop  af
00047          ret
00048  ;
00049  ; inicjalizacja trybu 2
00050  tryb2  ld   a,rej.i
00051          ld   i,a
00052          im   2
00053          ret
00054  adrprw  jr   przerw
00055  ;
00056  ; zmienne procedury
00057  nrlin  dw   9000 ;numer linii
00058  adrznk db   's'  ;kod znaku
00059  ;
00060  ; inicjalizacja trybu 1
00061  tryb1  im   1
00062          ret
00063          org  65535
00064          jr   13
00065          end
```

PROGRAM 53

ZX SPECTRUM

Wesołe zdania

Ten program ma charakter typowo rozrywkowy.

Generuje za pomocą generatora liczb pseudolosowych (RND) zdania złożone z przypadkowo wybranych słów. Dowcip i sens tych zdań zależą wyłącznie od programisty, który w liniach DATA umieszcza zestawy wyrazów.

Chciałbym nadmienić, że pierwszą wersję tego programu napisał mój kolega z Klubu Mikrokomputerowego „HOBBYTE" Krzysztof Herman.

Krzysztof POŹNIAK

```
 10 REM ************************
 11 REM *                      *
 12 REM *  Krzysztof Poźniak   *
 13 REM *                      *
 14 REM *    wesole zdania     *
 15 REM *                      *
 16 REM *              ©1988   *
 17 REM *                      *
 18 REM ************************
 20 CLS
 30 RESTORE 9000
 40 LET n=5: REM ilosc wyrazow
                 w zdaniu
 50 LET k=10: REM ilosc wyrazow
                 w zestawie
 60 LET i=10: REM maksymalna
                 dlug. wyrazu
 70 DIM t$(n,k,i)
 80 DIM d(n,k)
 90 REM wczytywanie danych
100 FOR m=1 TO n
110 FOR l=1 TO k
120 READ w$
130 LET dlug=LEN w$
140 IF dlug>i THEN PRINT "wyraz
'";w$;"' jest za dlugi": STOP
150 LET t$(m,l)=w$
160 LET d(m,l)=dlug
170 NEXT l
180 NEXT m
190 REM generowanie zdan
200 FOR m=1 TO n
210 LET x=1+INT (k*RND)
220 PRINT t$(m,x,1 TO d(m,x));"
 ";
230 NEXT m
240 PRINT
250 GO TO 200
9000 REM zbior wyrazow
9010 REM
9020 REM wyraz pierwszy
9030 DATA "zwariowany","zakochan
y","krzywy","kulawy","wesoly","z
ly","glucha","tchorzliwy","glodn
y","smieszny"
9040 REM wyraz drugi
9050 DATA "tancerz","piosenkarz"
,"zajac","wilkolak","jasiu","paj
ac","naukowiec","szef","bandyta"
,"wariat"
9060 REM wyraz trzeci
9070 DATA "ugryzl","zamalowal","
podgladal","kochal","ugotowal","
zjadl","obrabowal","pocalowal","
wyrzucil","wynalazl"
9080 REM wyraz czwarty
9090 DATA "okropnego","wielkiego
","pieknego","garbatego","stareg
o","zielonego","odwaznego","zgni
lego","madrego","smutnego"
9100 REM wyraz piaty
9110 DATA "amanta","pokrake","do
ktora","muzyka","smoka","pijaka"
,"poete","upiora","ciecia","mura
rza"
```

Ochrona danych podczas wykonywania instrukcji CLEAR

Prezentowany program, a dokładnie linia 70, pozwala na ochronę danych podczas rezerwowania miejsca instrukcją CLEAR. Mechanizm ochrony polega na przypisaniu obszarowi programu obszaru zmiennych na czas działania instrukcji CLEAR w linii 70. Po wykonaniu tej instrukcji odtwarza się stare granice obszarów.

```
  1 REM ************************
  2 REM
  3 REM    Krzysztof Poźniak
  4 REM
  5 REM  Ochrona danych podczas
  6 REM    wykonywania CLEAR
  7 REM
  8 REM                  ©1988
  9 REM ************************

 10 CLEAR 50000
 20 LET a=1
 30 PRINT "zmienna a=";a
 40 PRINT "CLEAR=";PEEK 23730+2
56*PEEK 23731
 50 PRINT #0;"wcisnij dowolny k
lawisz"
 60 PAUSE 0
 70 POKE 23728,PEEK 23627: POKE
 23729,PEEK 23628: LET x=PEEK 23
641+256*PEEK 23642-1: POKE 23627
,x-256*INT (x/256): POKE 23628,I
NT (x/256): CLEAR 40000: POKE 23
627,PEEK 23728: POKE 23628,PEEK
23729
 80 PRINT "zmienna a=";a
 90 PRINT "CLEAR=";PEEK 23730+2
56*PEEK 23731
```

Uwaga: gdyby argument instrukcji CLEAR spowodował błąd systemu, wtedy samemu należy ustawić granicę obszaru zmiennych pobierając tę wartość z komórek pamięci o adresach 23728 i 23729. Komórki te nie są używane przez system.

Krzysztof POŹNIAK

Dyrektor Hans Hofer

Żywioł i koniunktura

Rozmowa z dyrektorem firmy PROSYSTEM COMPUTER TECHNIC — HANSEM HOFEREM

● Jaki jest pana przepis na udany interes w informatycznej branży?

— Nie ma chyba uniwersalnej zasady. Prosystem jest firmą handlową, stąd dla nas najważniejsza jest jakość sprzedawanego towaru i sprawność obsługi klienta. Dodać należałoby jeszcze skuteczną reklamę, dobry serwis, no i oczywiście znajomość rynku.

● I języka...

— Nie ma w tym nic dziwnego. W Polsce bywam 25 razy w roku. Oczywiście język nie jest jak widać najsilniejszą moją stroną. Chociaż bardzo pomaga w pracy.

● Ile już lat pańska firma poznaje Polskę?

— Dziesięć.

● Będzie zatem jubileusz.

— W maju.

● Czy udany?

— Nasze kontakty handlowe są z roku na rok coraz większe. Sprowadzaliśmy i sprowadzamy do Polski odzież z najwyższym znakiem jakości. O trzech lat zajmujemy się również sprzętem komputerowym. Swoim zasięgiem obejmujemy też NRD, Czechosłowację i Związek Radziecki.

● Który z tych krajów jest najpoważniejszym klientem?

— Bez wątpienia Związek Radziecki. To bardzo wymagający, ale i bardzo chłonny rynek. W Polsce mamy znacznie większą konkurencję, powiedziałbym nawet, że informatykę opanował żywioł.

● I dobra koniunktura.

— W tej chwili największe interesy robi się sprzedając sprzęt, gdy tymczasem na całym świecie niezwykle drogie jest oprogramowanie. W Polsce najdroższe nawet programy kopiuje się bezkarnie.

● Dlatego w waszej ofercie znajdują się jedynie komputery i urządzenia peryferyjne?

— Sprzedawanie oprogramowania jest dziś jeszcze nieopłacalne. Nie pomogą zabezpieczenia przed kopiowaniem, jeśli nie ma odpowiednich przepisów chroniących prawa autorskie twórców programów. Teraz można sprzedać kilka egzemplarzy, a już za parę dni pojawią się pierwsze kopie, którymi będą handlować już inni — a my jesteśmy bezsilni. Dlatego wolimy sprzęt.

● W sklepach Centralnej Składnicy Harcerskiej prowadzącej sprzedaż waszych towarów często pojawiają się nowe modele komputerów, ale nie wszystkie znikają?

— Początkowo sądziliśmy, że minikomputery MSX mogą być przeznaczone do szkół. Dyskusja o tym, jaki sprzęt ma trafić do uczniów trwała długo. Teraz nasza oferta, to głównie urządzenia profesjonalne.

● Za złotówki?

— Ilość sprzętu sprzedawanego za złotówki jest ograniczona. Natomiast wszystko możemy dostarczyć za dewizy. W tej chwili w sklepach składnicy można kupić na przykład mikrokomputery BONDWELL BW 38-2, BONDWELL BW-8s, SVI-838 X'PRESS 16, a także stacje dysków, dyski twarde, monitory, drukarki, dyskietki.

● To zestaw standardowy.

— To prawda. Sprzęt o podobnym standardzie sprzedaje prawie każda z wielu działających spółek. Ale sądzę, że nasza pełna oferta jest znacznie atrakcyjniejsza.

● Co proponujecie?

— Przede wszystkim aplikacje profesjonalne, na przykład systemy komputerowego wspomagania projektowania inżynierskiego, tworzone na 32-bitowych komputerach. W skład oprogramowania wchodzą pakiety grafiki trójwymiarowej, programy służące do projektowania obwodów drukowanych i przygotowania klisz do produkcji płytek drukowych, a także systemy sterowania obrabiarkami numerycznymi.

● Dziedzina ich zastosowań jest bardzo wąska.

— Ale właśnie tam coraz częściej wykorzystuje się sprzęt komputerowy. Mamy też coś dla dziennikarzy, a raczej dla wydawców — systemy komputerowej edycji składu i drukowania tekstu i grafiki. Dziennikarzom może się też przydać telefax i telefon z możliwością transmisji obrazu, a urządzenie to uzyskało już atest i jest dopuszczone do zastosowania w krajowej sieci telefonicznej.

● Zainteresowanie sprzętem jest jednak proporcjonalne do możliwości jego zastosowania. Co zatem sprzedaje się najlepiej?

— Wszystko, co utrzymane jest w standardzie IBM. Sądzę, że już niedługo wzrośnie zainteresowanie sieciami i szybkimi, 32-bitowymi komputerami wyposażonymi w zegar o częstotliwości 24 MHz. Moi klienci coraz częściej bardzo konkretnie formułują swoje wymagania, są dobrymi partnerami.

● I tylko takich właśnie panu życzę. Dziękuję za rozmowę.

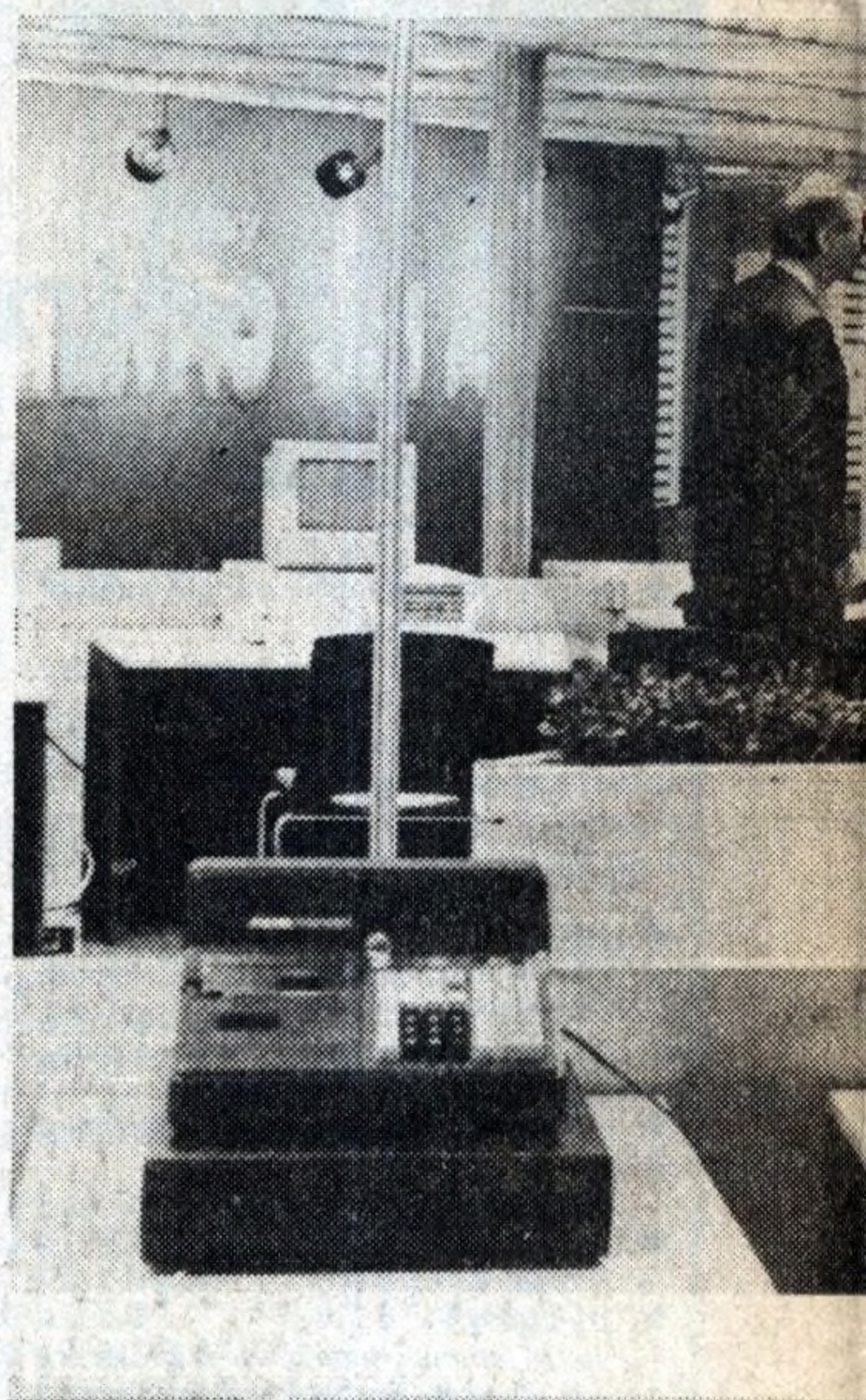
Wiesław CETERA

Telefax

(Foto: S. Bibulski)

INFOSYS

Jeśliby mierzyć skale informatyzacji naszego kraju częstotliwością i rozmachem imprez handlowych to bez wątpienia znaleźlibyśmy się w europejskiej czołówce, a już z pewnością na pierwszym miejscu wśród krajów RWPG. Po „Komputerze 88", kameralnych wystawach w warszawskim „Forum" handlową galę organizował Poznań. Od 25 do 29 kwietnia na terenie Międzynarodowych Targów Poznańskich trwały Międzynarodowe Targii Elektroniki, Telekomunikacji i Techniki Komputerowej „Infosystem '88". Jest to impreza, która w ubiegłym roku gościła we wrocławskiej Hali Ludowej. Zorganizowano ją z inicjatywy „Przeglądu Technicznego" w 120--lecie tej redakcji, co bez wątpienia dobrze podkreślało charakter gazety — organu programowo związanego z postępem. „Infosystem '87" rozbudził wiele namiętności, a patronat, jaki nad nim roztoczył Urząd Postępu Technicznego i Wdrożeń wróżył mu różową raczej przyszłość.



Przez rok impreza dojrzała i znacznie rozszerzyła swą tematykę. Hala Ludowa zrobiła się zbyt ciasna, a i wyspecjalizowana instytucja, jaką są Międzynarodowe Targi Poznańskie potrafi informatyczną fiestę znacznie lepiej przygotować — co już na wstępie podkreśla słuszność tezy, iż żadna działalność nie znosi amatorstwa, o czym warto pamiętać w kontekście różnorodnej działalności informatycznej.

Na „Infosystem '88" przywieziono sprzęt, urządzenia i systemy komputerowe przeznaczone dla wielu dziedzin. Na stoiskach znalazły się także urządzenia teletransmisyjne, radiokomunikacyjne i radiolokacyjne, urządzenia technologiczne dla przemysłu elektronicznego, podzespoły i oprogramowanie. Oferowano też usługi w zakresie przetwarzania danych, a także kompletacji i montażu sprzętu, literaturę fachową.

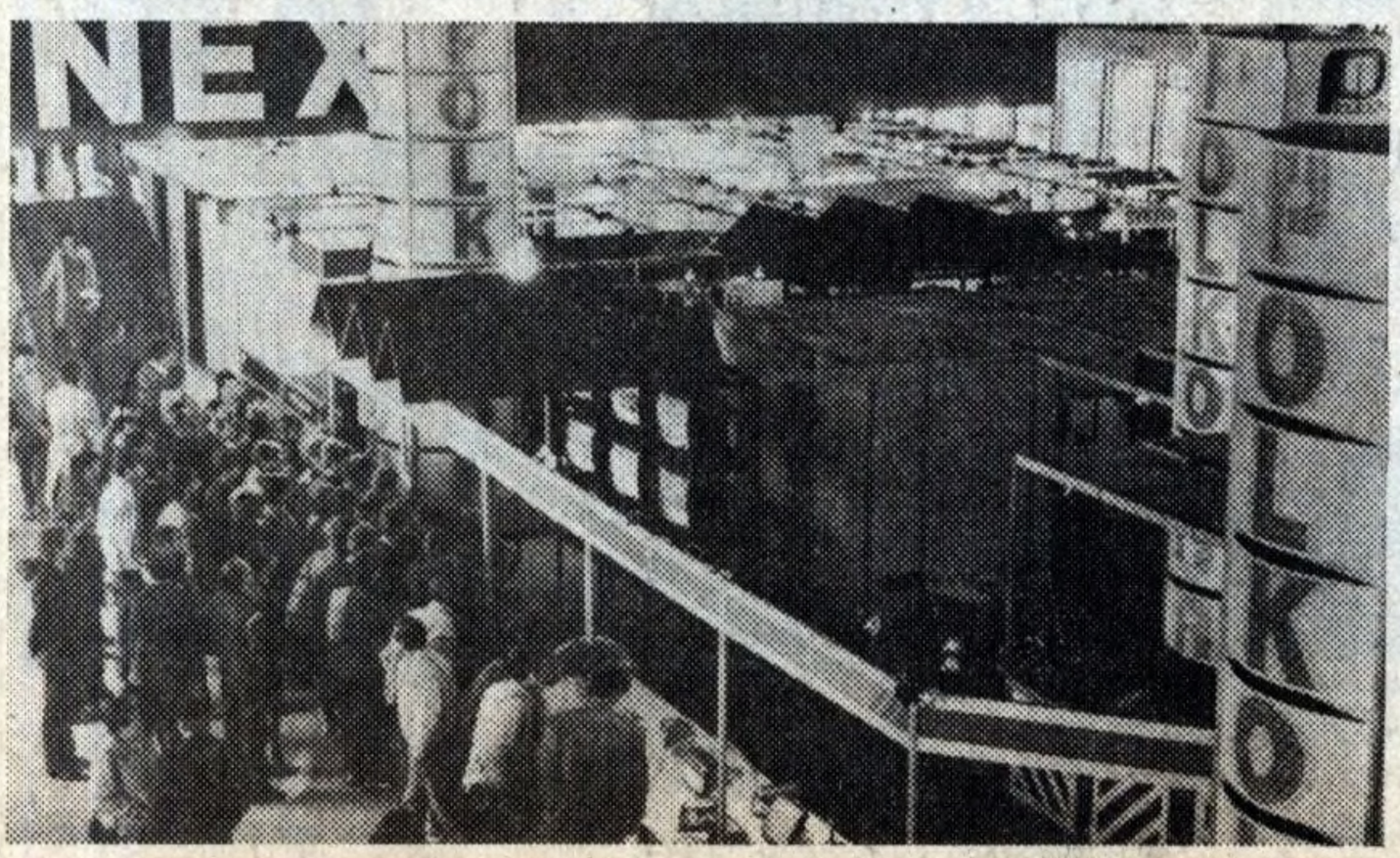
Z zapowiedzi wynikało, że będzie to impreza atrakcyjna, na najwyższym światowym poziomie. Ponad dwieście anonsowanych firm zwiastować miało lepsze jutro dla rozwoju informatyki. Tymczasem nie było ICL — firmy znanej i zadomowionej w Polsce, z udziału w imprezie zrezygnowała firma Olivetti.

Nie sądzę jednak, aby te fakty odbiły się na poziomie naszego rynku informatycznego, nic bowiem nie jest w stanie powstrzymać wzbierającej ciągle fali informatyzacji, która jest przecież synonimem postępu dla jednych i... dobrych interesów dla drugich.

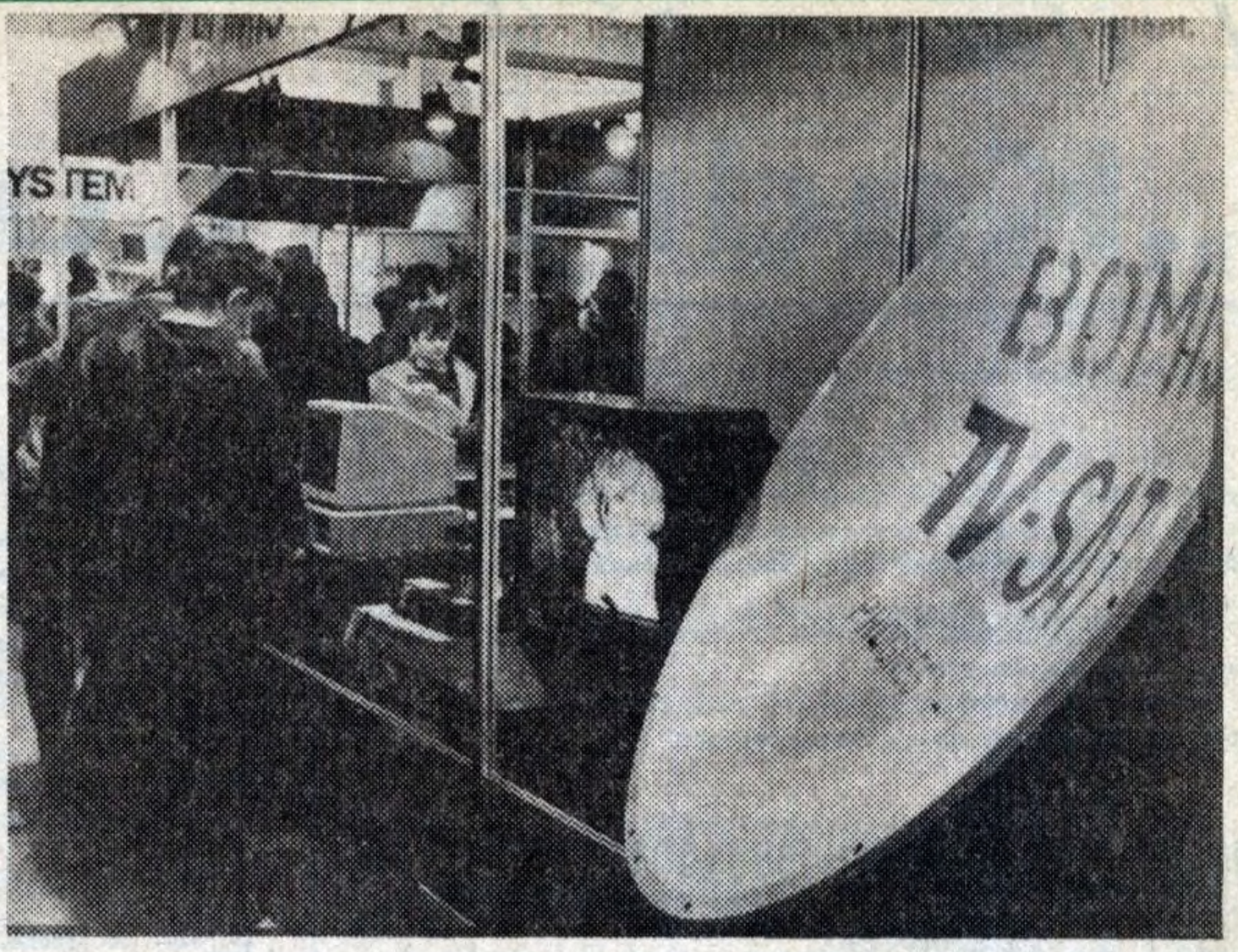
Sprzęt czyli pieniądze

Samopoczucie informatyków miał poprawić rządowy program informatyki, obejmujący siedem potężnych tematów i 17 zamówień rządowych. W 1987 roku na ten cel przeznaczono 50 mld złotych, co jest sumą raczej niewielką, chociażby w porównaniu z wydatkami na alkohol. „Statystycznej" flaszki nie można przeliczyć na komputer, choć kursy obu tych rzeczy trzymają się w Pewexie trzeźwo. Rozsądek stracić można dopiero wtedy, kiedy uspołecznieni i prywatni pośrednicy przeliczają ceny światowe na rodzimą walutę. Polski przemysł elektroniczny jest niedoinwestowany, jego udział w globalnej produkcji wynosi ok. 2,2 procenta (wśród krajów socjalistycznych wyprzedzamy jedynie Albanię i Rumunię). Nasza szansa na przodownictwo w tej dziedzinie (jeszcze nie tak dawno w RWPG wyprzedzał nas jedynie Związek Radziecki) przepadła, a możliwości odrobienia zaległości, realnie oceniając, nie są największe. Tymczasem potrzeby rosną: do końca tego stulecia w oświacie będzie potrzeba 700 tys. mikrokomputerów, w szkolnictwie wyższym 30 tys., w przemyśle i budownictwie ok. 190 tys., zaś w transporcie i komunikacji ok. 20 tys. Tego, niemal miliona urządzeń nie wyprodukuje z pewnością „ELWRO", a i tak nie jest to produkcja stwarzająca konkurencję, ani jakościową, ani cenową Tak, tak ceny komputerów na Zachodzie spadają średnio o kilkanaście procent rocznie (oczywiście nie chodzi o nowości) — tymczasem u nas jest odwrotnie.

Komputery są towarem poszukiwanym, stąd też dobre interesy robią dziś ci, którzy pośredniczą w ich sprzedaży. Listę takich pośredników można ciągnąć w nieskończoność. W tej branży zarabia się pieniądze szybko. Na szczęście jest to zdrowy rynek, który sprawnie eliminuje firmy niesolidne, udzielające na przykład fikcyjnych gwarancji na sprzęt, którego nikt nie potrafi naprawić.



STEM '88



Rzetelni dostawcy są znani. Wśród nich niestety najwyższe ceny mają firmy będące jednostkami gospodarki uspołecznionej, a wyjątek stanowi tu Centralna Składnica Harcerska, która sprzedaje sprzęt sprowadzany przez firmę Prosystem — na ogół niezłej jakości (zgodny z IBM, produkowany na Tajwanie) i nawet tani.

Co i za ile sprzedaje się gdzie indziej? Wystarczy wziąć jakąkolwiek gazetę — znaleźć kolumnę ogłoszeń i sprawdzić telefonicznie. Jeśli ktoś myśli poważnie o kupnie komputera, to cierpliwość może być nagrodzona. Komputery są. Trzeba mieć tylko pieniądze. Tyle sprzęt,

a teraz oprogramowanie czyli dżungla

Komputer wart kilka milionów jest dopiero początkiem drogi informatyki. Ta kupa wysokogatunkowego złomu zda się na niewiele, jeśli brak jest odpowiednich programów, a one swą wartością przewyższają często kilkakrotnie wartość komputera. Takie relacje są na całym świecie, na którym należymy niestety do nielicznej grupy państw nieprzestrzegających praw autorskich twórców programów komputerowych. Mamy bez wątpienia zdolnych i bardzo zdolnych programistów. Nasi „twórcy" (choć cudzysłów w tym miejscu jest raczej niepotrzebny) potrafią ominąć każde zabezpieczenie programu, potrafią dokonać niezbędnych w nim przeróbek, a dokumentację szybko przetłumaczyć. W efekcie pojawiają się takie anonse: „do zakupionego u nas sprzętu dołączamy bogate oprogramowanie" i tu pojawia się obszerna lista programów, które przedstawiają często o wiele większą wartość niż składany w pośpiechu tajwański komputer.

Poważni kupcy spoza naszych granic niechętnie umieszczają w swojej ofercie oprogramowanie. Sprzedaż kilku egzemplarzy po prostu się nie opłaca, bowiem już wkrótce pojawią się pierwsze kopie za połowę ceny i taniej.

Oczywiście mamy w dziedzinie tworzenia oprogramowania niewątpliwe sukcesy. Bez wątpienia na czoło wysuwa się tu Studio Kajkowskich, które nawet konie mówiące ludzkim głosem znają, zapraszając z ekranów telewizorów do salonu firmy w Sopocie.

Proponowane ceny są bezkonkurencyjne (!) podobno, jak i jakość proponowanego oprogramowania. W tym przypadku jednak największe bodaj znaczenie ma dobre imię firmy, która jest już w kraju znaną, a ludzie, którzy ją stworzyli mogą z pewnością zapisać plus po stronie sukcesu.

Jakakolwiek forma wymieniania wystawców „Infosystemu 88" jest daleko niepełna — bo przecież w targach udział wzięło ponad 150 firm.

Pozostańmy zatem przy oferach, w każdej z nich znajdowały się żelazne pozycje: kadry, magazyny, księgowość i finanse. Już pobieżny przegląd skłaniał do refleksji. Wiele z nich napisanych było głównie z myślą o programiście, a nie o użytkowniku. Złośliwi żartowali, że były to propozycje zgodne z zasadą trzech Z: Zrobić, Zainkasować, Zapomnieć. Racja, bo czasami i wstyd na to patrzeć. Niestety pewnie się zdarzy, że ktoś to kupi (ceny powyżej 400 tys. za sztukę).

Wśród ciekawych propozycji programowych odnotować warto edytor tekstowy TAG przygotowany przez Spółdzielnię Pracy Informatyków — „Infoserwis". TAG jest graficznym, pełnoekranowym edytorem tekstów o prostej obsłudze, jasnej komunikacji z użytkownikiem. Kosztuje 180 000. Zainteresowanie wzbudził też podręczny słownik polsko-angielski i angielsko-polski oferowany przez Computech z Krakowa. Producenci żądali za niego 300 000 mając świadomość, że tak naprawdę są to dwa autonomiczne programy umożliwiające dopiero wprowadzenie haseł słownika. Ale i tu cierpliwość może być nagrodzona. Inna firma DORAN, która w tegorocznym Infosystemie udziału nie bierze, proponuje podobny słownik, wypełniony już 15 tysiącami haseł i z dobrym oprogramowaniem, umożliwiającym podobno współpracę z edytorami tekstów za 200 000. I jak tu nie wierzyć w prawdziwą konkurencję, która jestem przekonany sprawi, że rynek informatyczny będzie jednym z najzdrowszych. Czego sobie i Czytelnikom życzę.

W. CETERA

Lilavati (5)

„Kadryl Lucasa”

Danuta KWASIŻUR
Mieczysław SKONIECZNY

Kolejny, piąty już, odcinek opisujący rozrywki matematyczne zgromadzone przez S. Jeleńskiego w „Lilavati” poświęcony będzie prostej grze planszowej nazwanej przez autora „Kadrylem Lucasa”. Nazwa gry pochodzi od nazwiska matematyka francuskiego Edwarda Lucasa.

Gra jest przeznaczona dla 1 osoby, ale można także szukać rozwiązania w zespole kilku graczy.

Prezentowany program przedstawia zasady gry na ekranie informacyjnym, jak pokazano na poniższym rysunku.

```
NA KWADRACIE 49-POLOWYM ROZSTAWIONE SA
KRAZKI CZERWONE I CZARNE.
POLE SRODKOWE JEST WOLNE. KORZYSTAJAC Z
TEGO POLA NALEZY PRZEPROWADZIC CZERWONE
KRAZKI NA MIEJSCE CZARNYCH, A CZARNE NA
MIEJSCE CZERWONYCH.
MOZNA PRZESUWAC KRAZEK ALBO NA SASIADU-
JACE POLE WOLNE ALBO TEZ PRZESKAKIWAC
JEDNYM KRAZKIEM PRZEZ DRUGI SASIEDNI,
GDY POZA NIM ZNAJDZIE SIE POLE PUSTE.

NACISNIJ DOWOLNY KLAWISZ
```

Następnie po naciśnięciu dowolnego klawisza ukazuje się plansza gry oraz informacje dodatkowe:

a) w górnym wierszu:
— dotychczasowy rekord (ilość punktów odpowiada ilości posunięć);
— aktualny wynik (ilość posunięć w bieżącej grze);

b) w wyszczególnionym oknie:
— kolumna i rząd pionka, który chcemy przesuwać;
— kolumna i rząd pola, na które chcemy przesunąć pionka.

Przed wykonaniem każdego ruchu należy wprowadzić te parametry wybierając numerycznie: kolumnę i rząd (wg oznaczeń planszy). Taką sytuację przedstawia poniższy rysunek:

(oczywiście pionki mają dwa kolory: czarny i czerwony).

Przed wykonaniem każdego ruchu programu bada, czy jest on dopuszczalny (zgodny z zasadami gry). W wypadku wprowadzenia błędnych parametrów mikrokomputer wyświetla jeden z komunikatów:

```
POLE ZAJETE;
PRZESUNIĘCIE NIEMOŻLIWE;
```

i powtórnie wraca w stan oczekiwania na parametry.

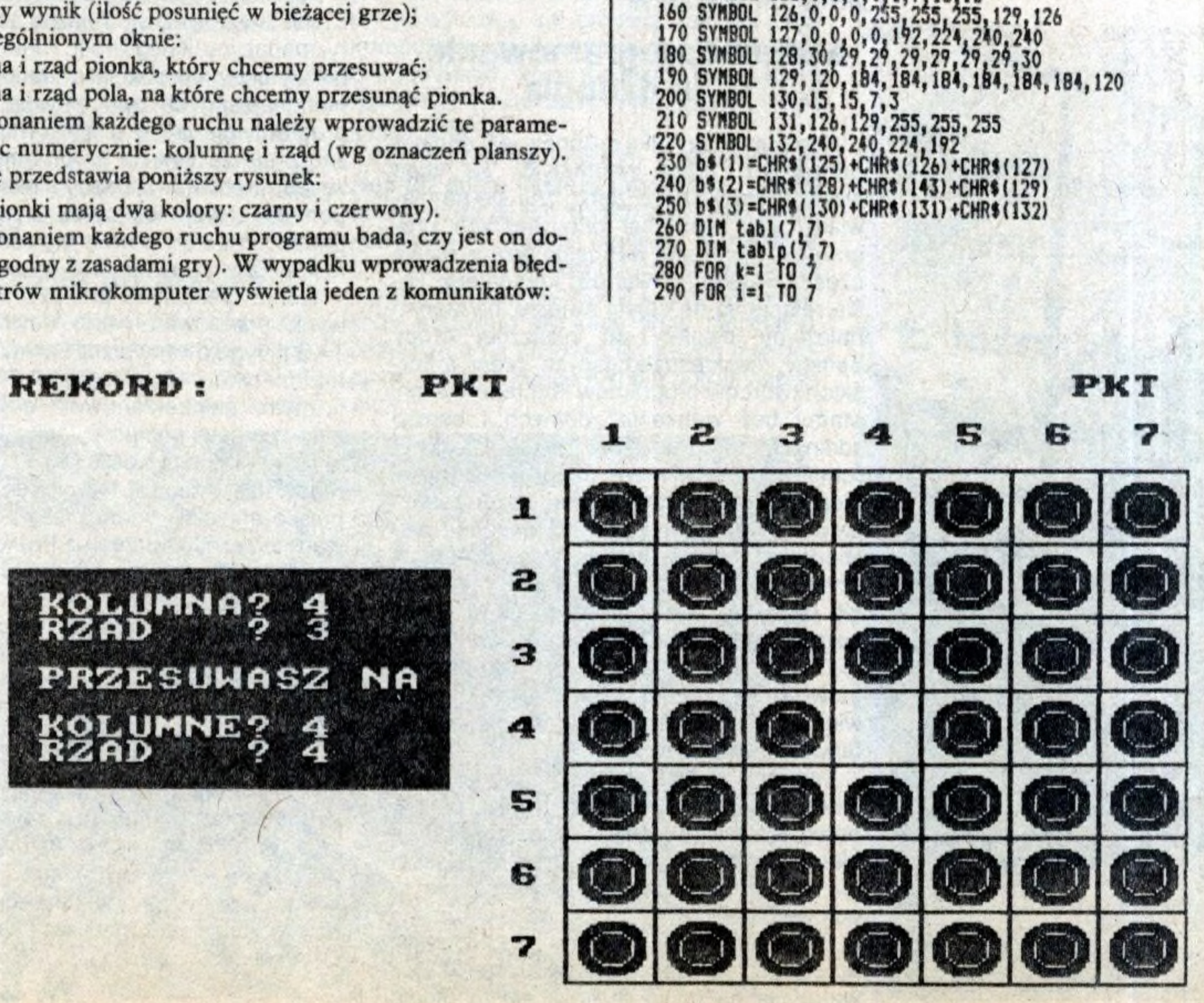
W programie opracowanym w języku BASIC mikrokomputera AMSTRAD CPC 6128 wyszczególniono 2 tablice:

TABL(7,7) — zawiera wartości 0, 1 lub 2 dotyczące aktualnego rozstawienia pionków (0 — pole puste, 1 — pionek czarny, 2 — pionek czerwony);

TABLP(7,7) — zawiera rozstawienie końcowe pionków i służy do badania, czy gra zakończyła się sukcesem.

Poszczególne fragmenty programu realizują następujące czynności:

```
10 MODE 1:INK 0,26:INK 1,0:INK 2,6:INK 3,14:PAPER 3:BORDER 14:CLS:PEN
1
20 PRINT "NA KWADRACIE 49-POLOWYM ROZSTAWIONE SA  KRAZKI CZERWONE I CZ
ARNE.                POLE SRODKOWE JEST WOLNE. KORZYSTAJAC Z TE
GO POLA NALEZY PRZEPROWADZIC CZERWONE KRAZKI NA MIEJSCE CZARNYCH, A CZ
ARNE NA MIEJSCE CZERWONYCH."
30 PRINT "MOZNA PRZESUWAC KRAZEK ALBO NA SASIADU- JACE POLE WOLNE ALBO
 TEZ PRZESKAKIWAC  JEDNYM KRAZKIEM PRZEZ DRUGI SASIEDNI,  GD
Y POZA NIM ZNAJDZIE SIE POLE PUSTE."
40 PEN 2:LOCATE 1,20:PRINT "NACISNIJ DOWOLNY KLAWISZ"
50 IF INKEY$="" THEN GOTO 50
60 PEN 1:MODE 1:CLS:WINDOW £1,20,40,5,25:PAPER £1,0:CLS £1
70 rek=0:g1=0
80 PAPER 3:PEN 2:LOCATE 1,1:PRINT "REKORD:       PKT " 
90 GRAPHICS PEN 1:PLOT 302,336:DRAWR 336,0:DRAWR 0,-336:DRAWR -336,0:D
RAWR 0,336
100 FOR k=0 TO 288 STEP 48
110 PLOT 350+k,334:DRAWR 0,-334
120 PLOT 302,288-k:DRAWR 334,0
130 NEXT k
140 SYMBOL AFTER 125
150 SYMBOL 125,0,0,0,0,3,7,15,15
160 SYMBOL 126,0,0,0,255,255,255,129,126
170 SYMBOL 127,0,0,0,0,192,224,240,240
180 SYMBOL 128,30,29,29,29,29,29,29,30
190 SYMBOL 129,120,184,184,184,184,184,184,120
200 SYMBOL 130,15,15,7,3
210 SYMBOL 131,126,129,255,255,255
220 SYMBOL 132,240,240,224,192
230 b$(1)=CHR$(125)+CHR$(126)+CHR$(127)
240 b$(2)=CHR$(128)+CHR$(143)+CHR$(129)
250 b$(3)=CHR$(130)+CHR$(131)+CHR$(132)
260 DIM tabl(7,7)
270 DIM tablp(7,7)
280 FOR k=1 TO 7
290 FOR i=1 TO 7
```



```
300 READ tablp(k,i)
310 NEXT i
320 NEXT k
330 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,0,1,1,1,2,2,2
,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
340 RESTORE 390:FOR k=1 TO 7
350 FOR i=1 TO 7
360 READ tabl(k,i)
370 NEXT i
380 NEXT k
390 DATA 2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,0,2,2,2,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
400 pkt=0:FOR k=1 TO 7
410 FOR i=1 TO 7
420 b=tabl(k,i):GOSUB 910
430 NEXT i
440 NEXT k
450 PAPER 3:PEN 1:FOR j=1 TO 7
460 LOCATE 18+j*3,3:PRINT USING "£";j;
470 LOCATE 18,3+j*3:PRINT USING "£";j;
480 NEXT j
490 PAPER 3:PEN 1:LOCATE 31,1:PRINT "      PKT "
500 WINDOW £2,1,15,9,17:PAPER £2,1:CLS £2:PEN £2,0
510 PKT=0
520 CLS £2:PEN 0:PAPER 1:LOCATE 1,10:INPUT " KOLUMNA";K1
530 IF k1>7 OR k1<1 THEN GOTO 520
540 INPUT " RZAD   ";RZ1
550 IF rz1>7 OR rz1<1 THEN GOTO 540
560 PRINT:PRINT " PRZESUWASZ NA"
570 LOCATE 1,15:INPUT " KOLUMNE";K2
580 IF k2>7 OR k2<1 THEN GOTO 570
590 INPUT " RZAD   ";RZ2
600 IF rz2>7 OR rz2<1 THEN GOTO 590
610 IF TABL(RZ2,K2)=0 THEN GOTO 630
620 CLS £2:SOUND 1,358:LOCATE 1,13:PRINT " POLE ZAJETE":FOR T=1 TO 800
:NEXT T:GOTO 520
630 IF K1=K2 AND RZ1=RZ2-1 THEN GOTO 720
640 IF K1=K2 AND RZ1=RZ2-2 THEN GOTO 720
650 IF K1=K2 AND RZ1=RZ2+1 THEN GOTO 720
660 IF K1=K2 AND RZ1=RZ2+2 THEN GOTO 720
670 IF K1=K2-1 AND RZ1=RZ2 THEN GOTO 720
680 IF K1=K2-2 AND RZ1=RZ2 THEN GOTO 720
690 IF K1=K2+1 AND RZ1=RZ2 THEN GOTO 720
700 IF K1=K2+2 AND RZ1=RZ2 THEN GOTO 720
710 CLS £2:SOUND 1,358:LOCATE 1,12:PRINT " PRZESUNIECIE":PRINT " NIEMO
ZLIWE":FOR T=1 TO 800:NEXT T:GOTO 520
720 K=RZ1:I=K1:B=0:GOSUB 910:
730 K=RZ2:I=K2:B=TABL(RZ1,K1):GOSUB 910:
740 TABL(RZ2,K2)=TABL(RZ1,K1):TABL(RZ1,K1)=0
750 PKT=PKT+1
760 PAPER 3:PEN 1:LOCATE 31,1:PRINT USING "£££££";PKT
770 FOR i=1 TO 7
780 FOR j=1 TO 7
790 IF tabl(i,j)<>tablp(i,j) THEN GOTO 520
800 NEXT j
810 NEXT i
820 CLS £2:PRINT £2:PRINT £2," BRAWO !":PRINT £2:PRINT £2,"UDALO SIE !
":PRINT £2
830 FOR k=5 TO 14:SOUND 1,258:SOUND 1,320:BORDER k:FOR t=1 TO 10:NEXT
t:NEXT k
840 IF g1=1 THEN  g1=1:PEN 0:PAPER 3:LOCATE 9,1:PRINT USING "£££££";re
k:GOTO 210
850 IF PKT<REK THEN PRINT £2,"USTALILES NOWY REKORD":REK=PKT:PEN 0:PAP
ER 3:LOCATE 9,1:PRINT USING "£££££";rek
860 :INPUT £2,"GRASZ DALEJ ? (T/N)";K$
870 IF K$="T" OR K$="t" THEN GOTO 340
880 IF K$="N" OR K$="n" THEN END
890 GOTO 210
900 IF INKEY$="" THEN GOTO 900
910 PAPER 0:PEN b:FOR a=0 TO 2:LOCATE 17+i*3,2+a+k*3:PRINT b$(a+1);:NE
XT a
920 GRAPHICS PEN 1:PLOT 302+i*48,382-k*48:DRAWR 0,-46:DRAWR -48,0:RETU
RN
```

10—50 — wyświetlenie ekranu informacyjnego;
60—510 — rysowanie planszy, deklaracja tablic i nadanie wartości początkowych zmiennych;
520—600 — wprowadzanie przez grającego parametrów do przesunięcia pionka;
610—710 — badanie, czy przesunięcie jest dopuszczalne (zgodne z zasadami gry);
720—760 — realizacja przesunięcia pionka na planszy, wykorzystująca podprogram 910—920 — rysowania pionka na wybranym polu.

Wytrwałym Czytelnikom, którzy wprowadzą ten program do pamięci mikrokomputera życzymy ustalenia rekordu trudnego do pobicia. Autor „Lilavati" podaje, że minimalna ilość posunięć, po których gra kończy się sukcesem wynosi 120 pkt.

Po zakończeniu jednej gry (komunikat: BRAWO! UDAŁO SIĘ! i dodatkowy efekt dźwiękowy) program pyta czy „GRASZ DALEJ?". Można wtedy przystąpić do nowej gry naciskając klawisz T.

Prosty program graficzny dla mikrokomputera AMSTRAD CPC 6128

Mikrokomputery serii AMSTRAD 6128 posiadają stosunkowo bogate oprogramowanie narzędziowe. Jedną z zalet tego komputera są spore możliwości graficzne, które pozwalają tworzyć ciekawe i nieraz skomplikowane obrazy. Pełne możliwości graficzne CPC 6128 udostępniają użytkownikowi także pakiety programów narzędziowych jak np. SPRITE, czy AUTOCAD. Dla początkującego użytkownika są one często skomplikowane, trudne w użyciu i mało komunikatywne (konwersacja prowadzona jest w języku angielskim). Wszystko to sprawia, że informatyk-amator nie wykorzystuje tych pakietów, a chcąc tworzyć ekrany graficzne sięga do możliwości języka BASIC. Tym samym traci sporo czasu, wykorzystując instrukcje SYMBOL, DRAW, PLOT osiąga wreszcie zamierzony cel. Nabywa przy tym przekonania, że aby osiągnąć ciekawe efekty graficzne trzeba się sporo namęczyć.

Dlatego też początkującym użytkownikom proponujemy proste narzędzie, które pozwoli im szybko uzyskiwać na ekranie różne kompozycje. Przy pewnej wprawie w obsłudze tego programu można pokusić się o tworzenie nawet bardziej skomplikowanych obrazów.

```
10 MODE 1:INK 0,0:INK 1,26:PAPER 0:CLS:PEN 1:INPUT"tryb (0,1,2)";tryb
15 INPUT "ramka (0-26)";ram
17 BORDER ram
20 p1=0
30 MODE tryb
40 DIM TRYBY(3,2)
50 FOR I=1 TO 3:READ TRYBY(I,1),TRYBY(I,2):NEXT i
60 DATA 20,16,40,4,80,2
70 WINDOW £1,1,TRYBY(TRYB+1,1),1,1
80 N1=0
90 DIM kol(tryby(tryb+1,2))
100 FOR v=1 TO tryby(tryb+1,2):kol(v)=-1:NEXT v
120 CLS £1:INPUT"CZYTAC ZBIOR(t/n)";k$
130 IF k$="n" OR k$="N" THEN GOTO 400
140 IF k$="T" OR k$="t" THEN GOTO 160
150 GOTO 120
160 CLS £1:INPUT £1,"TLO";TLO
170 KOL(1)=TLO:N1=N1+1
180 FOR I=2 TO TRYBY(TRYB+1,2)
190 CLS £1:PRINT £1,"KOLOR";I-1;:INPUT £1,K
200 IF K=-1 THEN GOTO 240
210 kol(i)=k:N1=N1+1
220 NEXT I
230 GOTO 270
240 FOR J=I TO TRYBY(TRYB+1,2)
250 KOL(J)=-1
260 NEXT J
```

```
270 INPUT £1,"PODAJ NAZWE";ZB$
280 CLS£1:LOAD zb$
290 PEN 1:CLS £1:PRINT £1,"P-POMOC"
295 REM 'wybor opcji'
300 A$=INKEY$
310 IF A$="" THEN GOTO 300
320 IF A$="R" OR A$="r" THEN GOTO 440
330 IF A$="K" OR A$="k" THEN GOTO 430
340 IF A$="M" OR A$="m" THEN GOTO 870
350 IF A$="F" OR A$="f" THEN GOTO 850
360 IF A$="z" OR A$="Z" THEN GOTO 820
370 IF A$="P" OR A$="p" THEN GOTO 1120
380 IF A$="d" OR A$="D" THEN GOTO 1370
390 GOTO 300
400 CLS £1:INPUT £1,"TLO";TLO
410 INK 0,TLO:PAPER 0:CLS
420 KOL(1)=TLO
430 GOSUB 1040
440 GOSUB 480
450 PLOT x,y:GOSUB 660
460 GOTO 290
470 END
475 REM 'sterowanie kursorem'
480 |SCREENCOPY,2,1
490 x=0:y=0:p1=1:PLOT x,y
500 A$=INKEY$
510 IF A$="" THEN GOTO 500
520 IF INKEY(6)=0 THEN RETURN
530 IF ASC(a$)=240 THEN GOTO 580
540 IF ASC(a$)=241 THEN GOTO 600
550 IF ASC(a$)=242 THEN GOTO 620
560 IF ASC(a$)=243 THEN GOTO 640
570 GOTO 500
580 IF y=398 THEN GOTO 500
590 |SCREENCOPY,1,2:y=y+2:PLOT x,y:GOTO 500
600 IF y=0 THEN GOTO 500
610 |SCREENCOPY,1,2:y=y-2:PLOT x,y:GOTO 500
620 IF x=0 THEN GOTO 500
630 |SCREENCOPY,1,2:x=x-2:PLOT x,y:GOTO 500
640 IF x=638 THEN GOTO 500
650 |SCREENCOPY,1,2:x=x+2:PLOT x,y:GOTO 500
655 REM 'podprogram rysowania plotami'
660 A$=INKEY$
670 IF A$="" THEN GOTO 660
680 IF INKEY(6)=0 THEN RETURN
690 IF ASC(a$)=240 THEN GOTO 740
700 IF ASC(a$)=241 THEN GOTO 760
710 IF ASC(a$)=242 THEN GOTO 780
720 IF ASC(a$)=243 THEN GOTO 800
730 GOTO 660
740 IF y=398 THEN GOTO 660
750 y=y+2:PLOT x,y:GOTO 660
760 IF y=0 THEN GOTO 660
770 y=y-2:PLOT x,y:GOTO 660
780 IF x=0 THEN GOTO 660
790 x=x-2:PLOT x,y:GOTO 660

800 IF x=638 THEN GOTO 660
810 x=x+2:PLOT x,y:GOTO 660
815 REM 'zapis zbioru na dysku'
820 CLS £1:INPUT £1,"PODAJ NAZWE";ZB$
830 CLS£1:SAVE ZB$,B,&C000,&4000
840 END
845 REM 'wypelnianie kolorem'
850 GOSUB 1040
860 GOSUB 480:MOVE X+2,Y:FILL N:y=0:x=0:GOTO 290
865 REM 'mazanie'
870 GRAPHICS PEN 1:GOSUB 480
880 A$=INKEY$
890 IF A$="" THEN GOTO 880
900 IF INKEY(6)=0 THEN GRAPHICS PEN 0:PLOT X,Y:GRAPHICS PEN 1:GOTO 290
910 IF ASC(a$)=240 THEN GOTO 960
920 IF ASC(a$)=241 THEN GOTO 980
930 IF ASC(a$)=242 THEN GOTO 1000
940 IF ASC(a$)=243 THEN GOTO 1020
950 GOTO 880
960 IF y=398 THEN GOTO 880
970 GRAPHICS PEN 0:PLOT X,Y:y=y+2:GRAPHICS PEN 1:PLOT x,y:GOTO 880
980 IF y=0 THEN GOTO 880
990 GRAPHICS PEN 0:PLOT X,Y:y=y-2:GRAPHICS PEN 1:PLOT x,y:GOTO 880
```

Ponieważ podczas pracy, program będzie przechowywał w pamięci kilka obrazów ekranu, dlatego też niezbędne jest wprowadzenie przed jego uruchomieniem programu BANKMAN.

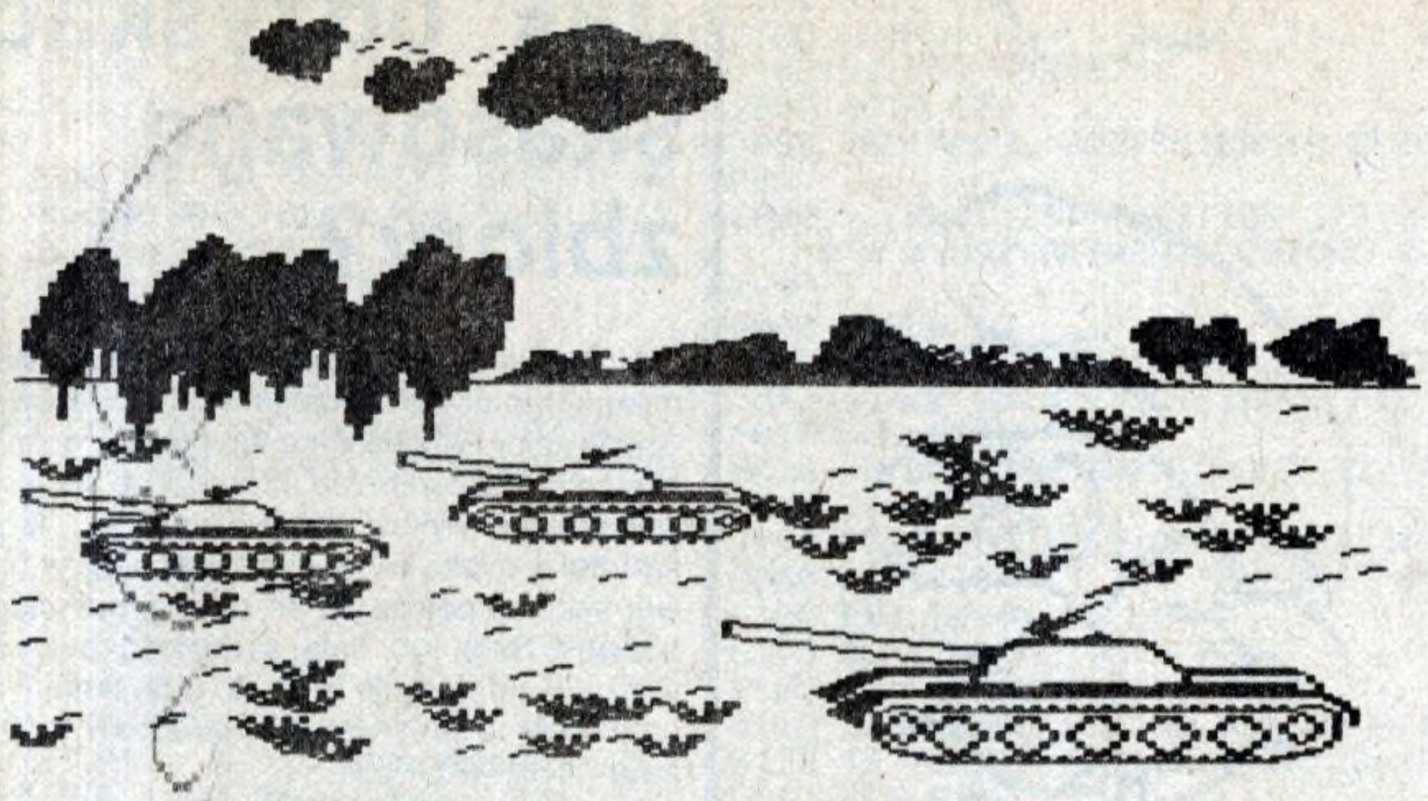
Po inicjacji pracy program żąda odpowiedzi na pytanie „Tryb?" — Należy wprowadzić liczbę z przedziału 0—2. W wybranym trybie będzie można maksymalnie wykorzystać:
0—16 kolorów;
1—4 kolory;
2—2 kolory.

Następne pytanie brzmi: „Czytać zbiór (t/n)?". Jeżeli jest to kolejny etap pracy mający na celu dokończenie obrazu zapisanego wcześniej na dysku, to należy podać nazwę tego obrazu, jaką nadał użytkownik przy poprzednim seansie.

W dalszej kolejności podajemy kolory:
— tła;
— ramki;
— poszczególnych atramentów.

Wprowadzamy liczby całkowite z przedziału 0—26 będące kodami pełnej palety barw tego mikrokomputera.
Po podaniu ww. parametrów wstępnych, możemy korzystać z zestawu następujących opcji programu:

R — rysowanie obrazu,
K — zmiana koloru atramentu,
M — mazanie nakreślonych linii,
F — wypełnianie obiektów kolorami,
Z — zapis obrazu na dysk,
P — pomoc (wyświetlenie pełnego zestawu opcji),
D — drukowanie zawartości ekranu na drukarce.



```
1000 IF x=0 THEN GOTO 880
1010 GRAPHICS PEN 0:PLOT X,Y:x=x-2:GRAPHICS PEN 1:PLOT x,y:GOTO 880
1020 IF x=638 THEN GOTO 880
1030 GRAPHICS PEN 0:PLOT X,Y:x=x+2:GRAPHICS PEN 1:PLOT x,y:GOTO 880
1035 REM 'wybor koloru'
1040 CLS £1:INPUT £1,"KOLOR";K
1050 FOR I=2 TO tryby(tryb+1,2)
1060 IF KOL(I)=K THEN GRAPHICS PEN I-1:N=I-1:CLS £1:GOTO 1110
1070 NEXT I
1080 N1=N1+1
1090 IF N1>TRYBY(TRYB+1,2)-1 THEN PRINT £1,"MASZ JUZ ";:PRINT £1,USING "££";tryb
y(tryb+1,2);:PRINT £1," KOL. !":FOR T=1 TO 1500:NEXT
 T:GOTO 1040
1100 kol(n1+1)=k:INK N1,K:N=N1:GRAPHICS PEN N:CLS £1
1110 RETURN
1115 REM 'pomoc'
1120 CLS £1:|SCREENCOPY,2,1
1130 MODE 1:PEN 1:PRINT "K-ZMIANA KOLORU":PRINT "R-RYSOWANIE":PRINT "M-MAZANIE":
PRINT "F-WYPELNIANIE KOLOREM":PRINT "Z-ZAPIS NA DYSK
U"
1280 LOCATE 1,20:PRINT "NACISNIJ DOWOLNY KLAWISZ"
1290 IF INKEY$="" THEN GOTO 1290
1300 MODE TRYB
1310 FOR K=1 TO TRYBY(TRYB+1,2)
1320 IF KOL(K)=-1 THEN GOTO 1350
1330 INK K-1,KOL(K)
1340 NEXT K
1350 WINDOW £1,1,TRYBY(TRYB+1,1),1,1:|SCREENCOPY,1,2
1360 GOTO 290
1365 REM 'drukowanie ekranu na drukarce'
1370 CLS £1
1380 MEMORY &9FFF
1390 FOR i=&A000 TO &A0B0
1400 READ v:POKE i,v:t=t+v
1410 NEXT i
1420 IF t<>20961 THEN PRINT "BLAD"
1430 DATA 205,166,160,62,27,205,157,160
1440 DATA 62,49,205,157,160,205,186,187
1450 DATA 205,231,187,50,180,160,17,0
1460 DATA 0,33,143,1,34,178,160,62
1470 DATA 7,50,177,160,62,10,205,157
1480 DATA 160,62,13,205,157,160,62,27
1490 DATA 205,157,160,62,76,205,157,160
1500 DATA 62,127,205,157,160,62,2,205
1510 DATA 157,160,14,0,58,177,160,71
1520 DATA 229,197,213,205,240,187,209,193
1530 DATA 33,180,160,190,225,55,32,1
1540 DATA 167,203,17,43,43,16,233,58
1550 DATA 177,160,254,7,40,7,175,203
1560 DATA 17,203,17,203,17,121,205,157
1570 DATA 160,19,229,33,127,2,55,237
1580 DATA 82,225,56,5,42,178,160,24
1590 DATA 193,35,124,181,40,32,43,17
1600 DATA 0,0,34,178,160,62,7,189
1610 DATA 32,146,124,180,32,142,62,4
1620 DATA 50,177,160,24,135,205,46,189
1630 DATA 56,251,205,43,189,201,62,27
1640 DATA 205,157,160,62,64,205,157,160,201
1650 CALL &A000
1660 END
```

OPCJA R

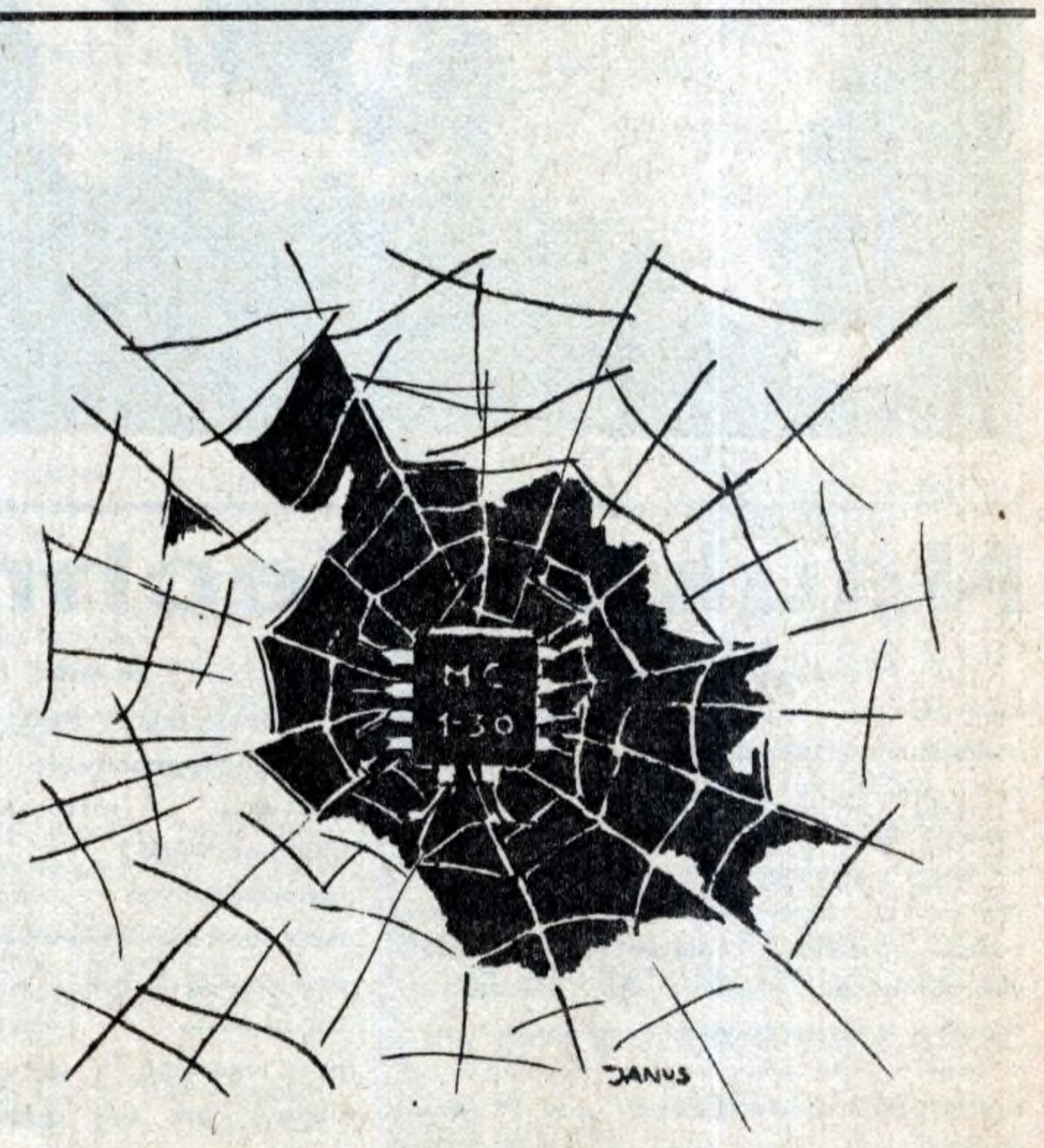
Po jej wybraniu należy na pytanie „KOLOR" wprowadzić kod liczbowy koloru, którym będziemy rysować. W sytuacji, kiedy mamy już maksymalną liczbę kolorów w zadeklarowanym trybie pojawi się stosowny komunikat. Następnie na ekranie pojawi się w lewym dolnym rogu punkt w wybranym kolorze. Przesuwanie punktu do miejsca, w którym chcemy rozpocząć rysowanie realizujemy za pomocą klawiszy funkcyjnych (→, ←, ↑, ↓). Po dotarciu punktu na żądane miejsce wciskamy klawisz ENTER. Od tego momentu klawisze funkcyjne powodują przesuwanie punktu, który kreśli linię. Zakończenie rysowania wybranym kolorem lub aktualnego fragmentu obrazu — klawisz ENTER.

OPCJA K

Po wybraniu tej opcji pojawia się pytanie „KOLOR". Po podaniu kodu liczbowego koloru program bada, czy użytkownik nie przekroczył już limitu barw dla aktualnego trybu pracy. Jeśli tak, to pojawia się komunikat: SĄ JUŻ N KOLORY (gdzie N = 2,4 lub 16) i pada ponowne pytanie „KOLOR". Jeżeli użytkownik wprowadzi dopuszczalny kod liczbowy barwy, to program powraca do kreślonego obrazu ze zaktualizowanym kolorem pióra graficznego.

OPCJA M

Przebieg tej opcji jest analogiczny do opcji R w części dotyczącej poruszaniem piksela. Po naprowadzeniu piksela na żądane miejsce i naciśnięciu ENTER klawisze (↑, ↓, →, ←) sterują kierunkiem wymazywania.



OPCJA F

Po naprowadzeniu punktu na wybrane miejsce (jak w opcjach R, M) i wciśnięciu ENTER następuje wypełnianie obiektu, w którym znalazł się punkt kolorem jego brzegu. (Wywoływana jest instrukcja FILL).

OPCJA Z

Po wybraniu tej opcji program pyta: „PODAJ NAZWĘ". Po zaakceptowaniu podanej nazwy klawiszem ENTER następuje zapisanie aktualnego obrazu na dysku w postaci tzw. „screcn'a"

Uwaga: Nazwa obrazu powinna być 8-znakowa, rozpoczynać się od litery i zawierać litery lub cyfry.

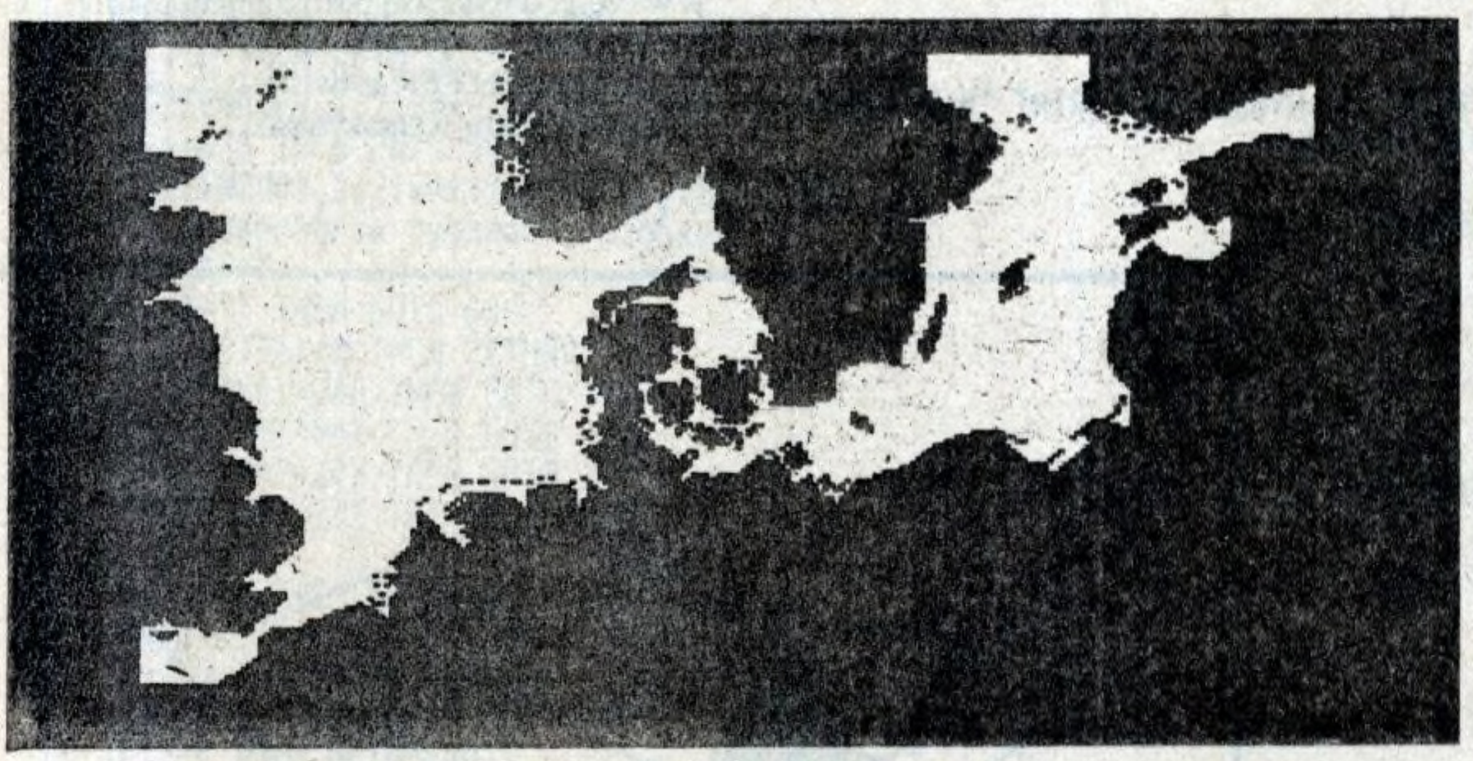
OPCJA P

Po naciśnięciu klawisza P niknie komponowany obraz i na ekranie pojawia się pełen zestaw dopuszczalnych opcji.

OPCJA D

Zalecana jest tylko wtedy, gdy zestaw mikrokomputerowy, na którym pracujemy ma podłączoną drukarkę. Po jej wybraniu jest tworzona kopia ekranu na papierze oczywiście w wersji czarno-białej.

Poniższe rysunki powstały za pomocą tego programu i opcji D. Obrazują one różne stopnie trudności, a jednocześnie spore możliwości tego programu graficznego.



Nauka • Technika

Nowy superkomputer skonstruowano w laboratorium SANDIA w amerykańskim stanie Nowy Meksyk. Jest on zdolny do rozwiązywania skomplikowanych zadań naukowych tysiąc razy szybciej od zwykłych komputerów. Posiada 1024 procesory rozwiązujące jednocześnie różne części większego zadania obliczeniowego. Każdy z tych procesorów odgrywa rolę oddzielnego komputera.

W czasie prób polecono superkomputerowi przygotowanie dokładnego opisu napięć powstających w rdzeniu metalowym znajdującym się pod obciążeniem. Maszyna dała odpowiedź po tygodniu obliczeń - gdy zwykły komputer potrzebowałby na to około 20 lat.

■ ■ ■

Brytyjskie Ministerstwo Komunikacji zaaprobowało plan stworzenia systemu informacji drogowej przekazywanej za pomocą umieszczonych przy jezdniach emiterów podczerwieni do komputerów znajdujących się w samochodach.

Informacje mają dotyczyć przejezdności dróg i nasilenia ruchu: na ich podstawie komputer poinformuje kierowcę o optymalnej trasie przejazdu i najdogodniejszej prędkości.

■ ■ ■

Jak odzyskać skasowane zbiory?

Może się zdarzyć, że w czasie pracy na mikrokomputerze skasujemy przypadkowo ważny zbiór na dyskietce lub jeszcze gorzej zostaną zniszczone lub uszkodzone sektory zawierające wykaz zbiorów na dysku, tzw. kartoteka dysku (ang. directory). Czy w takiej sytuacji należy uznać za stracone skasowany zbiór lub całą zawartość dysku? Otóż nie, informacje te są do odzyskania, a umożliwia to prezentowany program o nazwie DYSK—KOPIER.

Na wstępie trochę teorii. Na każdej dyskietce DOS rezerwuje pewien obszar przeznaczony na zapisanie wspomnianej wyżej kartoteki dysku. Są to sektory o numerach 361—368. Jeżeli kasujemy zbiór, to DOS nie „wymazuje" zawartości sektorów, które kasowany zbiór zajmował, lecz usuwa jego nazwę z kartoteki i aktualizuje zawartość sektora o numerze 360 (tzw. VTOC). A zatem mimo skasowania, zawartość skasowanego zbioru na dyskietce fizycznie nadal istnieje, aż do momentu, gdy zostanie zniszczona przez zapisanie nowych informacji. Problem polega na zidentyfikowaniu sektorów zawierających informacje należące do skasowanego zbioru. Jest to również możliwe, gdy poznamy bliżej strukturę pojedynczego sektora. Jak wiadomo, każdy sektor składa się ze 128 bajtów ponumerowanych od 0 do 127. Pierwsze 125 bajtów (0—124) zawierają informacje użytkownika, natomiast bajty 125, 126 i 127 zarezerwowane są dla DOS—a. Nas interesuje bajt 125, ponieważ jego sześć najstarszych bitów zawiera liczbę (numer) odpowiadającą kolejności wpisu nazwy zbioru, do którego należy dany sektor, w kartotece dyskietki. Po tym wstępie wszystko jest już jasne. Jeżeli mamy na dyskietce dziesięć zbiorów i skasowaliśmy na przykład zbiór piąty, to aby odzyskać jego zawartość należy wybrać z dyskietki wszystkie sektory, które w bajcie 125 mają wpisaną liczbę 4 (numeracja wpisów w kartotece zaczyna się od zera). Po wybraniu należy je zapisać na nową dyskietkę i stracony zbiór jest już odzyskany.

```
1 REM ************************
2 REM PROGRAM KOPIUJACY
3 REM ************************
9 REM MODUL STERUJACY
10 DIM S$(50),TOZ$(50),T$(200),P
US$(128),SE$(23040),SE(180),P$(1
28),INP$(3)
11 PUS$(1)=CHR$(0):PUS$(128)=PUS
$:PUS$(2)=PUS$
12 SE$(1)=CHR$(0):SE$(23040)=SE$
:SE$(2)=SE$
15 POKE 599,0
20 DO=720:VE=0:POKE 1913,80:ND=1
:ZL=1:PISZ=15000:ZROD=18000
25 OPEN #1,4,0,"K:":OPEN #2,4,0,
"E:"
26 GOSUB 29000
30 GOSUB 30000
35 GOSUB 58
40 GOSUB 500
50 GOTO 100
52 X=USR(ADR(S$),ND,360,ADR(P$),
82)
```

```
53 SE=ADR(P$):GS=PEEK(SE+1)+256*
PEEK(SE+2):IF GS=1010 THEN RETUR
N
54 GOSUB 61:POSITION 2,20:? "* *
 * Dysk nie w formacie DOS 2.5 *
 * "
55 FOR G=1 TO 500:NEXT G:GOTO 40
58 X=USR(ADR(TOZ$),ADR(T$),LEN(T
$),0,AD+12):RETURN
60 POKE 752,0:RETURN
61 POKE 752,1:RETURN
100 TRAP 40000:GET #1,A
101 IF A=155 THEN END
102 IF A<65 OR (A)69 AND A<76) O
R (A)77 AND A<>86 AND A<>88) THE
N 100
108 A=A-65
110 GOTO 500+A*500
500 GOSUB 61:? CHR$(125)
510 POSITION 2,1:? "* * * D Y S
K - K O P I E R * * * "
520 POSITION 3,3:? " M E N U
 [RETURN]- WYJSCIE Z MENU"
525 FOR LI=3 TO 11:POSITION LI,4
:? CHR$(13);:NEXT LI

530 POSITION 3,6:? "[";CHR$(66);
"] KOPIUJ SEKTORY [";CHR$(67);"]
 KOPIUJ DYSK"
540 POSITION 3,8:? "[";CHR$(68);
"] DRIVE ZRODL.   [";CHR$(69);"]
 DRIVE KOPI."
550 POSITION 3,10:? "[";CHR$(76)
;"] ODZYSK ZBIORU  [";CHR$(77);"
] TYP DOS-A "
560 POSITION 3,12:? "[";CHR$(86)
;"] VERIFY         [";CHR$(88);"
] FORMATUJ "
700 POKE 559,34:RETURN
999 REM MODUL KOPIOWANIA SEKTORO
W
1000 ? CHR$(125):POSITION 3,2:?
"* * * KOPIOWANIE SEKTOROW * * *
"
1005 GOSUB 60:CZ=1
1010 TRAP 1010:POSITION 2,5:? "N
umer sektora poczatkowego: ";:IN
PUT #2,SS
1020 TRAP 40000:TRAP 1020:POSITI
ON 2,7:? "Numer sektora koncowe
o:     ";:INPUT #2,ZS
1030 IF ZS<SS THEN 1010
1040 IF ZS>720 THEN GOSUB 52
1050 Y=DO:DO=ZS
1055 GOSUB 61:POSITION 8,5:? CHR
$(156);CHR$(156);CHR$(156)
1060 GOTO 1520
1070 DO=Y:CZ=0
1400 GOTO 40
1498 REM MODUL KOPIOWANIA CALEGO
 DYSKU
1500 ? CHR$(125):POSITION 3,2:?
"    * * * KOPIOWANIE DYSKU * *
*"
1510 SS=1
1515 IF DO=1040 THEN GOSUB 52
1520 GOSUB 61
1530 T=1
1540 POSITION 13,5:? "SEKTOR: 0"
1550 POSITION 13,7:? "BUFOR:  0"
1560 GOSUB ZROD
1600 FOR SEK=SS TO DO
1605 AR=ADR(SE$)+(T-1)*128
1610 X=USR(ADR(S$),ND,SEK,AR,82)

1620 IF SE$((T-1)*128+1,T*128)=P
US$ THEN 1633
1630 SE(T)=SEK
1632 T=T+1
1633 POSITION 21,5:? SEK
1636 POSITION 21,7:? T-1
1640 IF T=181 THEN GOSUB PISZ:GO
SUB ZROD
1700 NEXT SEK
1750 IF T<>0 THEN GOSUB PISZ
1850 IF CZ=1 THEN 1070
1900 GOTO 40
2000 ZL=ZL+(ZL=1)-(ZL=2):T$(31,3
1)=STR$(ZL):GOSUB 58
2100 GOTO 100
2500 ND=ND+(ND=1)-(ND=2):T$(71,7
1)=STR$(ND):GOSUB 58
2600 GOTO 100
5998 REM MODUL ODZYSKIWANIA ZBIO
ROW
6000 ? CHR$(125):POSITION 3,2:?
"   * * * ODZYSKIWANIE ZBIORU *
* *"
6005 GOSUB 60
6010 TRAP 6010:POSITION 3,4:? "P
odaj numer zbioru: ";:INPUT FIL
6015 GOSUB 61
6020 IF DO=1040 THEN GOSUB 52
6030 T=1:GES=0
6040 POSITION 13,5:? "SEKTOR: 4"
6050 POSITION 13,7:? "BUFOR:  0"
6100 FOR SEK=4 TO DO
6103 POSITION 21,5:? SEK
6105 AR=ADR(SE$)+(T-1)*128
6110 X=USR(ADR(S$),ND,SEK,AR,82)
6120 IF SE$((T-1)*128+1,T*128)=P
US$ THEN 6200
6122 IF SE$(T*128,T*128)=CHR$(0)
 THEN 6200
6123 SE=ADR(SE$(T*128-2,T*128-2)
)
6124 FN=INT(PEEK(SE)/4)
6125 IF FN<>FIL THEN 6200
6127 POKE SE,PEEK(SE)-FIL*4
6130 SE(T)=SEK
6135 POSITION 21,7:? T
6140 T=T+1:GES=GES+1:IF T=181 TH
EN GOSUB PISZ:GOSUB ZROD
6200 NEXT SEK
6250 GOSUB PISZ
6260 P$(1)=CHR$(0):P$(128)=P$:P$
(2)=P$
6270 P$(1,1)="B":P$(3,3)=CHR$(IN
T(GES/256)):P$(2,2)=CHR$(GES-(IN
T(GES/256)*256))
6280 P$(5,5)=CHR$(INT(SE(1)/256)
):P$(4,4)=CHR$(SE(1)-(INT(SE(1)/
256)*256))
6290 P$(6,16)="ZBIORODZYSK"
6300 X=USR(ADR(S$),ZL,361,ADR(P$
),87)
6340 GOTO 40
6500 DO=DO+320*(DO=720)-320*(DO=
1040)
6510 IF DO=1040 THEN T$(182,192)
=" 2.0 -[2.5]":GOTO 6530
6520 T$(182,192)="[2.0]- 2.5 "
6530 GOSUB 58
6600 GOTO 100
11000 IF VE=1 THEN POKE 1913,80:
T$(142,152)=" TAK -[NIE]":GOSUB
58:VE=0:GOTO 11100
11010 T$(142,152)="[TAK]- NIE ":
GOSUB 58:POKE 1913,87:VE=1
11100 GOTO 100
12000 REM MODUL FORMATUJE DYSKIE
TKE
12010 POSITION 2,14
12020 IF DO=720 THEN ? "Formatow
anie w drive #";ZL;" dla DOS-a 2
.0 (720 sektorow)":LS=0
12030 IF DO=1040 THEN ? "Formato
wanie w drive #";ZL;" dla DOS-a
2.5 (1040 sektorow)":LS=128
12040 POSITION 2,16:? "Napisz T
aby formatowac"
12050 GET #1,A:IF A<>84 THEN POS
ITION 2,17:? "Formatowanie przer
wane":GOTO 100
12070 IF ZL=1 THEN INP$="D1:"
12080 IF ZL=2 THEN INP$="D2:"
12090 TRAP 40000:XIO 254,#5,LS,0
,INP$

12100 POSITION 2,17:? "Formatowa
nie zakonczone":FOR I=1 TO 300:N
EXT I
12150 ? CHR$(125):GOTO 40
14999 REM MODUL ZAPISU NA DYSK
15000 POSITION 3,20:? "Wloz dysk
 [kopie] do drive'u #";ZL
15010 POSITION 15,22:? "Nacisnij
 [RETURN]"
15020 GET #1,A
15030 IF A<>155 THEN 15020
15035 POSITION 3,20:? CHR$(156);
CHR$(156);CHR$(156)
15040 POSITION 13,10:? "SEKTOR:
"
15050 POSITION 13,12:? "BUFOR:
   "
15060 TRAP 40000:TRAP 40
15100 FOR W=1 TO T-1
15110 SK=SE(W)
15113 POSITION 21,10:? SK
15116 POSITION 21,12:? W
15120 AR=ADR(SE$)+(W-1)*128
15130 X=USR(ADR(S$),ZL,SK,AR,87)
15140 NEXT W
15150 T=1
15155 POSITION 13,7:? "BUFOR:  0
   "
15200 RETURN
18000 POSITION 3,20:? CHR$(253);
CHR$(253);"Wloz dysk [zrodlowy]
do drive'u #";ND
18010 POSITION 15,22:? "Nacisnij
 [RETURN]"
18020 GET #1,A
18030 IF A<>155 THEN 18020
18035 POSITION 3,20:? CHR$(156);
CHR$(156);CHR$(156)
18040 RETURN
28000 REM MODUL DEFINIUJACY STAL
E
28001 REM TEKSTOWE I PODPROGRAMY
28002 REM W KODZIE MASZYNOWYM
29000 POKE 709,15:POKE 710,48
29010 RESTORE 29030
29013 AD=155*256

29015 AD=155*256
29020 FOR I=0 TO 9:READ A:POKE A
D+I,A:NEXT I
29030 DATA 0,0,66,12,155,2,2,2,2
,1
29040 DL=PEEK(560)+PEEK(561)*256
:POKE DL,1:POKE DL+1,0:POKE DL+2
,155
29050 POKE AD+10,PEEK(560)+3:POK
E AD+11,PEEK(561)
29060 RETURN
30000 T$(1,40)="DYSK ZRODLOWY
    - DRIVE #1          "
30001 T$(41,80)="KOPIA
    - DRIVE #1          "
30002 T$(81,120)="
                        "
30003 T$(121,160)="FUNKCJA VERIF
Y      TAK -[NIE]       "
30004 T$(161,200)="TYP DOS-A
       [2.0]- 2.5       "
30020 RESTORE 30050
30030 FOR I=1 TO 50:READ A:S$(I)
=CHR$(A):NEXT I:GOTO 30100
30050 DATA 104,104,104,141,1,3,1
04,141,11,3,104,141,10,3,104,141
,5,3,104,141,4,3,104,104,141,2,3
30051 DATA 32,83,228,173,3,3,133
,212,169,0,133,213,96
30100 RESTORE 30150
30110 FOR I=1 TO 50:READ A:TOZ$(
I)=CHR$(A):NEXT I:GOTO 30200
30150 DATA 104,104,133,204,104,1
33,203,104,104,133,205,104,104,1
33,207,104,133,208,104,101,207,1
33,207
30151 DATA 160,0,177,203,56,233,
32,145,207,200,196,205,208,244,9
6
30152 DATA 0,0,0,0,0,0,0,0,0,0,0
,0
30200 POKE 741,255:POKE 742,154
30300 RETURN
```

Oprócz odzyskiwania zbiorów program DYSK—KOPIER realizuje funkcje kopiowania dyskietek. Może on pracować zarówno

pod kontrolą DOS 2.0, jak i DOS 2.5, w zestawie mikrokomputera z jednym lub dwoma napędami dysków.

Wszystkie opcje programu wyglądają następująco:

B — kopiowanie określonej liczby sektorów. Użytkownik podaje numer sektora pierwszego i ostatniego, które mają być skopiowane.
C — kopiowanie całego dysku.
D — przełącznik dla zestawu mikrokomputera z dwiema stacjami dysków — określa, w którym napędzie znajduje się kopiowany dysk.
E — podobnie jak D — określa, w którym napędzie ma być tworzona kopia dysku.
L — odzyskiwanie zbiorów.
M — przełącznik — określa, pod którym DOS-em (2.0 — 2.5) ma pracować program.
V — przełącznik — określa, czy zapis kopii ma być kontrolowany dodatkowo przez odczyt (funkcja VERIFY), czy nie.
X — formatowanie dyskietki.

Podczas kopiowania (odzyskiwania) program wczytuje do bufora w pamięci kolejne sektory. Pojemność bufora ogranicza liczbę wczytywanych sektorów do 180. Po zapełnieniu bufora program zapisuje wczytane sektory na tworzoną kopię i czyta kolejne 180 sektorów itd. itd. Puste sektory nie są zapisywane (kopiowane).

Aby odzyskać zbiór lub kilka zbiorów należy po uruchomieniu programu włożyć do stacji dysków dyskietkę, z której zbiory będziemy odzyskiwać. Następnie wciskamy klawisz L. Program zażąda podania numeru zbioru, który chcemy odzyskać. Po podaniu numeru program rozpocznie wyszukiwanie sektorów posiadających wpisany w bajcie 125 podany numer. Po odnalezieniu wszystkich sektorów (lub zapełnieniu bufora) program zatrzyma się i wyświetli komunikat „Włóż dysk—kopię do drive'u # n (1 lub 2). Należy wówczas włożyć do stacji dysków czystą sformatowaną dyskietkę i wcisnąć RETURN. Na dyskietkę tę zostanie zapisany odzyskany zbiór. Jego nazwą będzie ZBIORODYSK. Zbiór ten należy skopiować (na przykład opcją 0 DOS-a) na inną dyskietkę.

Opracował H. KRASUSKI

Elektroliza

Program przedstawia zjawisko elektrolizy. Pokazuje on między innymi ruch jonów w woltametrze przed i po podłączeniu napięcia do elektrod. Jeśli chcesz to zobaczyć, uruchom ten program, a jeśli chcesz pokazać to swoim kolegom, nagraj ten program na taśmę magnetofonową i zademonstruj go na lekcji fizyki. Twój nauczyciel z pewnością się ucieszy.

```
0 REM ELEKTROLIZA;J.GOLLA,P.GRZE
GANEK, P.WAGNER
5 GRAPHICS 2+16:POSITION 4,3:? #
6;"ELEKTROLIZA"
10 FOR T=1 TO 3000:NEXT T:DIM A$
(1),B$(1),V$(40)
15 GRAPHICS 0:POKE 710,12:POKE 7
12,8:POKE 709,2:POKE 755,0:POSIT
ION 1,6
20 ? "SPACJA-POWROT DO LISTY ROZ
KAZOW":?
30 ? "KLAWISZ SELECT-PRZEPLYW PR
ADU NA"
31 ? "SCHEMACIE":?
35 ? "KLAWISZ START-SCHEMAT ZEST
AWU":?
40 ? "KLAWISZ HELP-RUCH JONOW":?
45 IF PEEK(53279)=6 THEN GOTO 39
0
50 IF PEEK(53279)=5 THEN GOTO 54
0
60 IF PEEK(732)=17 THEN POKE 732
,0:GOTO 660
65 GOTO 45
390 POSITION 0,20:? "CZY SCHEMAT
 MA BYC OPISANY(T-N)";
392 INPUT B$:IF B$<>"T" AND B$<>
"N" THEN GOTO 390
395 A=7:IF B$="T" THEN A=8
400 GRAPHICS A+16:COLOR 1
405 PLOT 10,30:DRAWTO 10,80:DRAW
TO 80,80:DRAWTO 80,30
410 FOR T=11 TO 79:A=T:IF B$<>"T
" THEN A=2
425 COLOR A:PLOT T,35:DRAWTO T,7
9:NEXT T
430 COLOR 1:FOR T=16 TO 22:PLOT
T,39:DRAWTO T,74:NEXT T
435 COLOR 3:FOR T=68 TO 75:PLOT
T,39:DRAWTO T,74:NEXT T
440 COLOR 1:PLOT 19,39:DRAWTO 19
,5:DRAWTO 45,5:DRAWTO 55,0
442 PLOT 55,5:DRAWTO 130,5:DRAWT
O 130,10:PLOT 130,30
445 FOR T=0 TO 7 STEP 0.5:DRAWTO
 SIN(T)*10+130,COS(T)*10+20:NEXT
 T
450 PLOT 122,26:DRAWTO 130,12:DR
AWTO 138,26:PLOT 124,22:DRAWTO 1
35,22
455 PLOT 130,30:DRAWTO 130,70:DR
AWTO 115,70:PLOT 110,70:DRAWTO 9
0,70
460 DRAWTO 90,25:DRAWTO 72,25:DR
AWTO 72,40
462 PLOT 110,55:DRAWTO 110,85:PL
OT 115,80:DRAWTO 115,60
465 SETCOLOR 1,9,13:SETCOLOR 0,1
2,5:SETCOLOR 2,7,1:SETCOLOR 4,2,
0

467 IF B$="T" THEN GOTO 475
470 SETCOLOR 0,12,1:SETCOLOR 2,7
,15:SETCOLOR 4,2,4:SETCOLOR 1,9,
8:GOTO 530
475 TRAP 475:RESTORE 520:FOR L=1
 TO 13:READ PX,PY,V$
480 DL=PEEK(560)+PEEK(561)*256:D
1=PEEK(DL+4)+PEEK(DL+5)*256
490 FOR U=1 TO LEN(V$):D2=57344+
((ASC(V$(U,U))-32)*8)
500 D3=D1+PY*40+PX+U-1:FOR Z=0 T
O 7:POKE D3+Z*40,PEEK(D2+Z)
515 NEXT Z:NEXT U:NEXT L:B$="
":TRAP 40000
520 DATA 102,5,1-ELEKTROLIT,102,
20,2-AMPEROMIERZ
522 DATA 102,35,3-WYLACZNIK,102,
50,4-ANODA,102,65,5-KATODA,102,8
0,6-BATERIA
525 DATA 13,90,+ -,5,50,1,14,10,
2,6,8,3,14,45,6,2,65,5,9,65,4
530 IF PEEK(764)=33 THEN POKE 76
4,255:GOTO 15
535 GOTO 530
540 GRAPHICS 7+16:SETCOLOR 4,10,
0:RESTORE 650
545 FOR T=1 TO 4:READ A,B,C,D
555 FOR F=A TO B STEP C:E=E+1:CO
LOR E-1:PLOT D,F:IF E=4 THEN E=1
560 NEXT F:NEXT T:COLOR 1:FOR T=
1 TO 4:READ A,B,C,D
575 FOR F=A TO B STEP C:E=E+1:CO
LOR E-1:PLOT F,D:IF E=4 THEN E=1
585 NEXT F:NEXT T:FOR T=1 TO 4:R
EAD A,B,C,D,G
590 FOR F=A TO B STEP C:E=E+0.5:
COLOR E-1:PLOT F,D:DRAWTO F,G
595 IF E=4 THEN E=1
600 NEXT F:NEXT T:FOR T=1 TO 12:
A=RND(0)*40+15:B=RND(0)*30+45
605 FOR F=8 TO 1 STEP -1:E=E+1:C
OLOR E-1:PLOT A+F,B:IF E=4 THEN
E=1
615 NEXT F:A=RND(0)*40+15:B=RND(
0)*30+45
620 FOR F=1 TO 8:E=E+1:COLOR E-1
:PLOT A+F,B:IF E=4 THEN E=1
623 NEXT F:NEXT T:COLOR 3
625 PLOT 5,30:DRAWTO 5,80:DRAWTO
 80,80:DRAWTO 80,30:PLOT 70,40:D
RAWTO 75,40
630 PLOT 21,40:DRAWTO 27,40:PLOT
 24,43:DRAWTO 24,37:PLOT 5,33:DR
AWTO 80,33
635 FOR U=1 TO 15:SETCOLOR 0,10,
U+10:SETCOLOR 1,2,U+5:SETCOLOR 2
,13,9
640 IF PEEK(764)=33 THEN POKE 76
4,255:GOTO 15
645 NEXT U:GOTO 635
650 DATA 45,5,-1,15,5,70,1,150,7
0,20,-1,90,20,45,1,65,15,150
651 DATA 1,5,150,120,-1,70,110,9
0,-1,70,90,65,-1,20
655 DATA 17,13,-1,45,75,67,63,-1
,45,75,113,110,-1,60,80,120,117,
-1,50,90
660 ? CHR$(125):POSITION 3,10:?
"KLAWISZ (Z)  ZAMYKA OBWOD PRADU
"
665 FOR T=1 TO 400:NEXT T
670 POSITION 3,10:? "   PROSZE P
OCZEKAC 15 SEKUND   ":FOR T=1 TO
 400:NEXT T
675 X=PEEK(54279):Y=PEEK(53277):
Z=PEEK(559):O=0:POKE 764,255
680 GRAPHICS 5+16
685 AD=PEEK(106)-16:POKE 106,AD-
16
690 POKE 53277,3:POKE 54279,AD:P
OKE 559,62
695 S=AD*256+1024:FOR I=S TO S+1
024:POKE I,0:NEXT I
700 COLOR 3:FOR T=4 TO 78:PLOT T
,20:DRAWTO T,46:NEXT T:COLOR 1:P
LOT 3,15
705 DRAWTO 3,47:DRAWTO 79,47:DRA
WTO 79,15:COLOR 2
706 FOR T=6 TO 8:PLOT T,15:DRAWT
O T,45:NEXT T:COLOR 1
710 FOR T=73 TO 75:PLOT T,15:DRA
WTO T,45:NEXT T
711 PLOT 8,15:DRAWTO 8,8:DRAWTO
20,8:PLOT 20,5:DRAWTO 20,11:PLOT
 22,3
715 DRAWTO 22,13:PLOT 22,8:DRAWT
O 40,8:PLOT 51,8:DRAWTO 73,8:DRA
WTO 73,15
720 SETCOLOR 1,3,15:SETCOLOR 0,4
,0:SETCOLOR 2,4,8:SETCOLOR 4,4,4
725 GOSUB 900:POKE 704,1:POKE 70
5,1:POKE 706,15
726 POKE 707,15:POKE 53256,1:POK
E 53257,1:POKE 53258,1:POKE 5325
9,17
735 Q=RND(0)*15:H=H-Q:IF H<-80 T
HEN H=30
740 IF PEEK(764)=23 THEN POKE 76
4,255:POKE 53248,60:POKE 53249,6
0:GOTO 742
741 GOTO 745
742 POKE 53250,60:POKE 53251,60:
GOTO 785
745 POKE 53248,100-H:Q=RND(0)*10
:M=M-Q:IF M<-40 THEN M=70
750 POKE 53249,100+M:Q=RND(0)*18
```

```
:N=N-Q:IF N<-80 THEN N=30
755 POKE 53250,100-N
760 Q=RND(0)*24:J=J-Q:IF J<-40 T
HEN J=70:IF PEEK(764)=33 THEN GO
TO 800
765 POKE 53251,100+J:GOTO 735
770 DATA 0,0,0,0,24,24,60,60,102
,102,102,231,129,129,129,129,231
771 DATA 102,102,102,60,60,24,24
,0,0,0,0,0
775 DATA 0,0,0,0,24,24,60,60,126
,126,126,255,129,129,129,129,255
776 DATA 126,126,126,60,60,24,24
,0,0,0,0,0
780 COLOR 3:PLOT 40,1:DRAWTO 50,
1
785 FOR T=1 TO 50:NEXT T:COLOR 3
:O=O+1:PLOT 40,O:DRAWTO 50,O:COL
OR 0
786 PLOT 40,O-1:DRAWTO 50,O-1:IF
 O>=8 THEN COLOR 1:PLOT 12,13:GO
TO 790
788 GOTO 795
790 DRAWTO 17,13:PLOT 60,13:DRAW
TO 66,13:PLOT 63,10:DRAWTO 63,16
:GOTO 800
795 GOTO 785
800 IF PEEK(764)=33 THEN POKE 10
6,PEEK(106)+16:POKE 54279,X:GOTO
 802
801 GOTO 805
802 POKE 53277,Y:POKE 559,Z:POKE
 764,255:GOTO 810
805 GOTO 815
810 FOR T=53248 TO 53255:POKE T,
0:NEXT T:GOTO 15
815 TRAP 825:T=8:T1=73:A1=0:B1=0
:A=0:B=0
820 L=RND(0)*80+80:W=L:A=A+1:RET.
URN
825 TRAP 835
830 V=RND(0)*80+80:R=V:B=B+1:RET
URN
835 L=L+1:W=W-1:V=V+1:R=R-1
840 POKE 53248,W:POKE 53251,L:PO
KE 53250,V:POKE 53249,R
845 IF L>180 THEN L=180
850 IF W<60 THEN W=60
855 IF V>180 THEN V=180
860 IF R<60 THEN R=60
865 IF W=60 AND L=180 THEN GOSUB
 820
870 IF V=180 AND R=60 THEN GOSUB
 830
875 IF A=3 THEN A=0:A1=A1+1:COLO
R 1:PLOT T+A1,15:DRAWTO T+A1,45
880 IF B=3 THEN B=0:B1=B1-1:COLO
R 2:PLOT T1+B1,15:DRAWTO T1+B1,4
5
885 IF B1=-7 THEN A1=0:B1=0
890 IF PEEK(764)=33 THEN GOTO 80
0
895 TRAP 40000:GOTO 835
900 S=AD*256+1024+170:RESTORE 77
0
905 FOR I=S TO S+28:READ B:POKE
I,B:NEXT I
910 S=AD*256+1280+140:RESTORE 77
0
915 FOR I=S TO S+28:READ B:POKE
I,B:NEXT I:S=AD*256+1656
916 RESTORE 775:FOR I=S TO S+28:
READ B:POKE I,B
920 NEXT I:S=AD*256+1792+190:RES
TORE 775:FOR I=S TO S+28:READ B
921 POKE I,B:NEXT I:RETURN
```

Uwaga: Zamiast znaku } (nawiasu wężykowego) należy w programie napisać niedostępny dla drukarki znak \ (pochyloną strzałkę). (Otrzymujemy go naciskając najpierw klawisz ESC, a następnie równocześnie klawisze CONTROL oraz CLEAR.)

Jan GOLLA, Piotr GRZEGANEK,
Paweł WAGNER
Zespół Szkół Chemiczno-Mechanicznych
w Gliwicach

* SUMA KONTROLNA / ETYKIETA *

FY	0	WA	51	NS	101	KL	15
WY	201	JP	301	MP	311	FV	35
DM	401	KC	451	FG	501	NM	60
UW	651	UH	3901	KD	3921	AV	395
ZC	4001	PZ	4051	FA	4101	MR	425
MV	4301	BG	4351	LL	4401	CX	442
FK	4451	PE	4501	NB	4551	OK	460
HQ	4621	KU	4651	GC	4671	UV	470
XC	4751	UX	4801	MW	4901	ZB	500
QD	5151	NF	5201	FT	5221	NM	525
KU	5301	PA	5351	KF	5401	RG	545
PS	5551	EC	5601	PS	5751	VA	585
MP	5901	PY	5951	CJ	6001	PE	605
KH	6151	DH	6201	XY	6231	NY	625
FS	6301	BN	6351	KX	6401	QA	645
TA	6501	AL	6511	DX	6551	WT	660
FT	6651	ZE	6701	RH	6751	IA	680
KB	6851	CE	6901	QL	6951	PL	700
ZZ	7051	AH	7061	IU	7101	CI	711
HY	7151	RD	7201	ND	7251	OT	726
ER	7351	DH	7401	SF	7411	AC	742
ZI	7451	QM	7501	RU	7551	PP	760
BI	7651	ET	7701	AA	7711	CU	775
LL	7761	CP	7801	XA	7851	ZF	786
VL	7881	GS	7901	UT	7951	TW	800
QQ	8011	CF	8021	RN	8051	QQ	810
HX	8151	XF	8201	QX	8251	FT	830
SB	8351	WV	8401	RV	8451	NN	850
CH	8551	IP	8601	EW	8651	LR	870
AY	8751	CO	8801	TD	8851	DV	890
YT	8951	GM	9001	QR	9051	GM	910
US	9151	JN	9161	HB	9201	ZJ	921

Tajemnicze błędy

Na początek proponuję przeprowadzić dwa eksperymenty:

Program 1:

```
10 FOR k=0 TO 10 STEP.1
20 PRINT k
30 NEXT k
```

Program 2:

```
10 PRINT 1—1+1e—30
20 PRINT 1+1e—30—1
```

W pierwszym przypadku dla Amstrada, Atari i Commodore zamiast dokładnych wartości zaczną wyświetlać się liczby z ogonami dziewiątek, a w Spectrum pętla zatrzyma się na wartości 9.9.

Winę za zaistniałe w obu przypadkach anomalie ponosi dokładność obliczeniowa komputera, a ściślej: skończona ilość cyfr po przecinku. Np. 0.1 w zapisie dwójkowym jest nieskończone (tak jak 1/3 w zapisie dziesiętnym). Wobec tego w komputerze jest trzymane przybliżenie 0.1, które podczas wielokrotnego dodawania w pętli potęguje błąd. Jeżeli zamiast 0.1 wstawimy 0.125 (jest ono zapisywane dokładnie), to błędu nie będzie.

Wniosek: Używać należy kroku dającego się zapisać dokładnie

(najlepiej 1):

```
10 FOR k=0 TO 100
20 PRINT k/10
30 NEXT k
```

W drugim przykładzie w linii 20 do 1 dodaje się bardzo mała liczba (1e—30), która ze względu na skończoną ilość liczb po przecinku zostaje obcięta. 1e—30 wymaga postawienia 1 na 30-tym miejscu po przecinku, a komputery nie dopuszczają więcej liczb po przecinku niż 14. Stąd **1+1e—30=1**. Ponieważ dodawanie jest przez komputer wykonywane kolejno, stąd od otrzymanego wyniku zostaje odjęte 1 i powstaje błędny wynik wynoszący 0. Natomiast w linii 10 najpierw zostają od siebie odjęte jedynki dając w efekcie 0, a do 0 zostaje dodana liczba 1e—30. Stąd wynik jest prawidłowy.

Krzysztof POŹNIAK

ZX Spectrum

Semigrafik

```
   1 BORDER 0: CLS : DIM z$(64):
 PRINT PAPER 0;z$: DIM a$(15): D
IM b$(15): DIM c$(15): DIM d$(15
): FOR n=1 TO 12: READ a: LET a$
(n)=CHR$ a: READ b: LET b$(n)=CH
R$ b: READ c: LET c$(n)=CHR$ c:
READ d: LET d$(n)=CHR$ d: NEXT n
: RESTORE
   2 LET a=1+INT (RND*15):
LET b=2+INT (RND*5): LET c=13-b:
 LET d=23-b: LET e=10+b:
 LET f=INT (RND*8): LET g=16+f:
 LET h=31-f: LET i=15-f
   3 PRINT
  AT b,f;a$(a);  AT b,g;a$(a);
  AT b,i;b$(a);  AT b,h;b$(a);
  AT e,f;a$(a);  AT e,g;a$(a);
  AT e,i;b$(a);  AT e,h;b$(a);
  AT c,f;c$(a);  AT c,g;c$(a);
  AT c,i;d$(a);  AT c,h;d$(a);
  AT d,f;c$(a);  AT d,g;c$(a);
  AT d,i;d$(a);  AT d,h;d$(a):
  GO TO 2
   4 SAVE "SEMIGRAFIK" LINE 1
   5 DATA
131,131,140,140,140,140,131,131,
137,134,134,137,134,137,137,134,
138,133,138,133,133,138,133,138,
142,141,139,135,139,135,142,141,
143,143,143,143,141,142,135,139,
135,139,141,142,134,137,137,134
  10 FOR n=USR "a" TO USR "b": ST
EP 2: POKE n,170: POKE n+1,85: N
EXT n: LET a$=CHR$ 144+CHR$ 144+
CHR$ 144
  20 BORDER 0: PAPER 0: INK 7: C
LS : LET i=0: LET p=0
  30 DIM t(8): FOR n=1 TO 8: REA
D t(n): NEXT n
  40 DATA 0,2.039,3.86,4.98,7.02
,8.04,10.08,12
  50 FOR m=i TO 28 STEP 3
  60 FOR n=0 TO 18 STEP 3
  65 IF INKEY$<>"" THEN PAUSE 0
  70 PRINT INK i: PAPER p;AT n,m
;a$;AT n+1,m;a$;AT n+2,m;a$: LET
 p=p+1: IF p=7 THEN LET p=0
  80 NEXT n
  90 PRINT INK i: PAPER p;AT 21,
m;a$: PRINT #0; INK i; PAPER p;A
T 0,m;a$;AT 1,m;a$: LET p=p+1: I
F p=7 THEN LET p=0
 100 LET i=i+1: BEEP .01,14+t(i)
 110 IF i=8 THEN LET i=0
 120 NEXT m: GO TO 50
 200 CLEAR : SAVE "kolor 8*10" L
INE 10: BEEP .2,2: VERIFY "": RE
STORE : GO TO 10
1000 REM
     Program demonstruje kolory,
powstałe ze zmieszania 8 kolorów
palety komputera Spectrum
```

PROGRAM XZ SPECTRUM

Bakarat

BAKARAT to bardzo popularna w XIX w. we Francji gra hazardowa w karty. Jak zwykle w takich przypadkach reguły są proste. Osiem talii kart, bankier, stawki i nieograniczona liczba graczy. Każdy z nich stawia pewną sumę, lecz rozgrywa tylko ten, który postawił najwięcej. On i bankier otrzymują po 2 karty. Wygrywa ten, którego karty mają najwięcej punktów.

Poszczególne karty punktuje się następująco: as — 1, figury i 10 — 0, blotki — ich wartość. Jeśli suma punktów przekracza 9, to oblicza się jej modulo 9. Gracz rozgrywający wygrywa automatycznie mając 8 lub 9 punktów, (chyba że bankier ma tyle samo), czeka mając 6 lub 7 punktów, musi dobrać kartę mając mniej niż 5 punktów, decyduje o dobraniu karty mając 5 punktów.

Bankier zawsze dobiera kartę, jeśli ma początkowo mniej niż 5 punktów. Jeśli rozgrywający wygra, to bank wypłaca wszystkim graczom stawki. Jeśli wygra bankier, to wszyscy tracą postawione pieniądze na jego korzyść.

Zasady gry są nieco uproszczone, co nie przeszkadza przecież w ich modelowaniu. Zresztą powstało wiele odmian bakarata, chociażby CHEMIN DE FER.

Krzysztof MAMCARZ

```
  10 GO SUB 1000
  20 GO SUB 2000
  25 GO SUB 2500
  30 GO SUB 3000
  40 GO SUB 4000
  50 GO SUB 5000
  60 GO TO 25
 100 STOP
1000 REM [ CZARY ]
1010 LET bank=0
1020 DIM k(8,4,13)
1030 LET adres=USR "a"
1040 RESTORE : FOR a=0 TO 31
1050 READ bajtek: POKE adres+a,b
ajtek
1060 NEXT a
1500 RETURN
1700 DATA 0,108,254,254,124,56,1
6,0,16,56,124,254,254,84,56,0,16
,56,124,254,124,56,16,0,56,56,21
4,254,214,16,56,0
2000 REM [ POCZATEK ]
2010 INPUT "Ilu graczy ?",gracz
2020 IF (gracz<>INT gracz) OR (g
racz<1) THEN GO TO 2010
2030 DIM i$(gracz,15): DIM f(gra
cz): DIM s(gracz)
2040 FOR a=1 TO gracz
2050 INPUT "imie gracza ";(a);":
": LINE i$(a)
2060 PRINT a,i$(a)
2070 LET f(a)=10000
2080 NEXT a
2090 RETURN
2500 REM [ START ]
2510 PAUSE 100: LET c=0: LET max
=0: CLS
2520 FOR a=1 TO gracz
2530 IF f(a)<1 THEN LET c=c+1: N
EXT a
2535 IF c=gracz THEN PRINT INVER
SE 1;"WSZYSCY PRZEGRALI": GO SUB
 6000
2540 PRINT "gracz ";a;" :";f(a)
2550 INPUT (i$(a))"ile stawiasz
 ? ";s(a)
2560 IF (s(a)<>INT s(a)) OR (s(a
)<1) THEN GO TO 2130
2570 IF s(a)>f(a) THEN GO TO 211
0
2580 IF s(a)>max THEN LET max=s(
a): LET opoz=a
2590 NEXT a: RETURN
3000 REM [ LOSOWANIE ]
3010 CLS : PRINT INVERSE 1;"BANK
IER"
3020 LET sumab=0: LET sumag=0
3030 FOR i=1 TO 2
3040 GO SUB 7000: LET sumab=suma
b+pkt: GO SUB 7500
3050 NEXT i
3060 PRINT : PRINT INVERSE 1;i$(
opoz)
3070 FOR i=1 TO 2
3080 GO SUB 7000: LET sumag=suma
g+pkt: GO SUB 7500
3090 NEXT i: PRINT
3100 LET sumab=sumab-9*(sumab>9)
3110 LET sumag=sumag-9*(sumag>9)
3120 RETURN
4000 REM [ DECYZJE ]
4010 IF sumab>7 OR sumag>7 THEN
GO TO 4120
4020 IF sumag<5 THEN LET w$="T":
 GO TO 4070
4030 IF sumag>5 THEN GO TO 4100
4040 INPUT (i$(opoz))"Czy dobie
rasz karte ? (T/N)";w$
4060 IF w$="N" OR w$="n" THEN GO
 TO 4100
4070 IF w$="T" OR w$="t" THEN PR
INT : PRINT i$(opoz);" dobiera "
;: GO SUB 7000: LET sumag=sumag+
pkt-9*(sumag>9): GO SUB 7500: GO
 TO 4100
4080 GO TO 4040
4100 IF sumab>4 THEN GO TO 4120
4110 PRINT "bankier","dobiera ";
: GO SUB 7000: LET sumab=sumab+p
kt-9*(sumab>9): GO SUB 7500
4120 PRINT : IF sumab=sumag THEN
 PRINT INVERSE 1;"R E M I S": RE
TURN

4130 IF sumab>sumag THEN PRINT I
NVERSE 1;"BANK WYGRAŁ": GO TO 41
50
4140 PRINT INVERSE 1;i$(opoz);"
WYGRAŁ"
4150 GO SUB 5000
4160 PAUSE 0: CLS
4170 PRINT "STAN GOTOWKI "
4180 PRINT "              "
4190 PRINT : PRINT "BANK ",bank
4200 FOR a=1 TO gracz
4210 PRINT i$(a),f(a)
4220 NEXT a
4230 RETURN
5000 REM [ KONIEC ]
5010 FOR a=1 TO gracz
5020 IF sumab>sumag THEN LET s(a
)=-s(a)
5030 LET f(a)=f(a)+s(a)
5040 LET bank=bank-s(a)
5050 NEXT a
5060 RETURN
6000 REM [ POWTORKA ]
6010 INPUT "gramy dalej ? (T/N)
";w$
6020 IF w$="T" OR w$="t" THEN RE
TURN
6040 STOP
7000 REM [ LOSOWANIE KARTY ]
7010 LET kolor=INT (RND*4)+1
7020 LET karta=INT (RND*13)+1
7030 FOR a=1 TO 8
7040 IF k(a,kolor,karta)=0 THEN
LET k(a,kolor,karta)=1: GO TO 71
00
7050 NEXT a: GO TO 7010
7100 IF karta>9 THEN LET pkt=0
7110 IF karta<10 THEN LET pkt=ka
rta
7120 RETURN
7500 REM [ WYDRUK KARTY ]
7510 IF karta<11 AND karta>1 THE
N LET e$=STR$ karta: GO TO 7560
7520 IF karta=1 THEN LET e$="A"
7530 IF karta=11 THEN LET e$="W"
7540 IF karta=12 THEN LET e$="D"
7550 IF karta=13 THEN LET e$="K"
7560 PRINT e$; INK (0+2*(kolor=1
 OR kolor=3));CHR$ (kolor+143)
7570 RETURN
```

Przeliczanie

```
Program przelicza metryczne
miary długości na mile, jardy,
stopy i cale z dokładnością do
1/64 cala

 1 mile = 1760 yds = 5280 ft
 1 yd   = 3 ft     = 36 in
 1 ft   = 12 in

 1 mile = 1608.64 m
 1 km   = 1094 yd 3 19/64 in

-------------------------------
 7 km 283 m 47 cm 4 mm =

 4 mi 328 yd 2 ft 4 7/16 in
-------------------------------
 29 km 53 cm 8 mm =

 18 mi 49 yd 9 1/8 in
-------------------------------


 100 REM [ program MILE,KM ]
 110 FOR n=0 TO 23: READ m: POKE
USR "a"+n,m: NEXT n
 120 DATA 0,0,56,4,60,68,60,6,0,
16,20,24,48,16,12,0,8,16,56,64,5
6,4,120,0
 130 REM a=a b=i c=$
 140 CLS : BORDER 6
 150 PRINT INK 1;
   Program przelicza metryczne
  miary długości na mile, jardy,
 stopy i cale z dokładnością do
          1/64 cala"
 160 PRINT INK 2
   1 mile = 1760 yds = 5280 ft
   1 yd   = 3 ft     = 36 in
   1 ft   = 12 in"
 165 PRINT INK 0;
   1 mile = 1608.64 m
   1 km   = 1094 yd 3 19/64 in"
 170 FOR t=1 TO 5: BEEP .01,25:
BEEP .01,24: BEEP .015,30: NEXT
t
 180 PRINT INK 2;"---------------
-------------------"
 190 INPUT "kilometry ", LINE k$
: IF k$="" THEN LET klm=0
 191 IF k$<>"" THEN LET klm=VAL
k$
 192 INPUT "metry "; LINE m$: IF
 m$="" THEN LET mtr=0
 193 IF m$<>"" THEN LET mtr=VAL
m$
 194 INPUT "centymetry "; LINE c
$: IF c$="" THEN LET ctm=0
 195 IF c$<>"" THEN LET ctm=VAL
c$
 196 INPUT "milimetry ", LINE l$
: IF l$="" THEN LET mlm=0
 197 IF l$<>"" THEN LET mlm=VAL
l$
 200 LET mm=mlm+10*ctm+1000*mtr+
1000000*klm
 210 LET km=0: LET m=0: LET cm=0
: LET mm=INT mm
 240 IF mm>9 THEN LET cm=INT (mm
/10): LET mm=mm-10*INT (mm/10)
 250 IF cm>99 THEN LET m=INT (cm
/100): LET cm=cm-100*INT (cm/100
)
 260 IF m>999 THEN LET km=INT (m
/1000): LET m=m-1000*INT (m/1000
)
 270 PRINT " ";: IF km=0 THEN GO
 TO 290


 280 PRINT km;" km ";
 290 IF m=0 THEN GO TO 310
 300 PRINT m;" m ";
 310 IF cm=0 THEN GO TO 330
 320 PRINT cm;" cm ";
 330 IF mm=0 THEN GO TO 350
 340 PRINT mm;" mm ";
 350 LET mm=mm+10*cm+1000*m+1000
000*km
 360 PRINT "="
 370 LET mi=mm/1608640
 380 PRINT " ";
 390 IF INT mi=0 THEN GO TO 410
 400 PRINT INT mi;" mi ";
 410 LET yd=(mi-INT mi)*1760
 420 IF INT yd=0 THEN GO TO 440
 430 PRINT INT yd;" yd ";
 440 LET ft=(yd-INT yd)*3
 450 IF INT ft=0 THEN GO TO 470
 460 PRINT INT ft;" ft ";
 470 LET in=(ft-INT ft)*12
 480 IF in<1/64 THEN GO TO 170
 490 IF INT in=0 THEN GO TO 510
 500 PRINT INT in;" ";
 510 LET reszta=INT (64*(in-INT
in))/64
 520 IF reszta=0 AND in>=1 THEN
PRINT " in": GO TO 170
 530 FOR n=1 TO 5
 540 IF reszta*2^n=INT (reszta*2
^n) THEN PRINT (2^n*reszta);"/";
2^n;" in": GO TO 170
 550 NEXT n
 560 GO TO 170
9000 SAVE "KM/MILE" LINE 100: BE
EP .2,26: CLS : PRINT "Verify
 KM/MILE": VERIFY "": GO TO 100
```

Przedstawiamy zaktualizowaną listę (por. "IKS" nr. 1/87 oraz nr. 12/87) skrótów, które związane są z mikrokomputerem, łącznie z ich oryginalną interpretacją. Czekamy na inne, Waszym zdaniem, ważne skróty.

W jednym z najbliższych numerów "IKS-a" postaramy się przedstawić uaktualnioną listę skrótów wraz z ich tłumaczeniem (interpretacją) w języku polskim.

ACK - AFFIRMATIVE ACKNOWLEDGMENT,
ACU - AUTOMATIC CALLING UNIT,
ADA - ALL POINTS ADDRESSABLE,
ADLC - ADWANCED DATA LINK CONTROLLER,
ADP - AUTOMATIC DATA PROCESSING,
ADPCM - ADAPTIVE DIFFERENTIAL PCM,
AI - ARTIFICIAL INTELLIGENCE,
AL - AUTOLOAD,
ALE - ADDRESS LATCH ENABLE,
ALU - ARITHMETIC-LOGIC UNIT,
APA - ALL POINTS ADDRESSABLE MODE,
API - AUTOMATIC PROGRAM INTERRUPT,
ARQ - AUTOMATIC REQUEST FOR REPEATS,
ASCII - AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE,
AT - ADVANCED TECHNOLOGY,
AU - ARITHMETIC UNIT,
BASIC - BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE,
BCC - BLOCK CHARACTER CHECK,
BCD - BINARY CODED DECIMAL,
BCD - BLANK COLUMN DETECTION,
BDOS - BASIC DOS,
BER - BIT ERROR RATE,
BEX - BROADBAND EXCHANGE,
BF - BEST FIT ALGORITHM,
BFD - BEST FIT DECREASING ALGORITHM,
BIOS - BASIC INPUT-OUTPUT SYSTEM,
BIU - BUS INTERFACE UNIT,
BORAM - BLOCK-ORIENTED RAM,
BPI - BITS PER INCH,
BPS - BITS PER SECOND,
BSC - BINARY SYNCHRONOUS COMMUNICATIONS,
CAD - COMPUTER AIDED DESIGN,
CAI - COMPUTER-AIDED INSTRUCTION,
CAM - COMPUTER AIDED MANUFACTURE,
CAMAC - COMPUTER AIDED MEASUREMENT AND CONTROL,
CAN - CANCEL CHARACTER,
CAP - COMPUTER AIDED PUBLISHING,
CCP - CONSOLE COMMAND PROCESSOR,
CD ROM - COMPACT DISC ROM,
CGA - COLOR GRAPHICS ADAPTER,
CIM - COMPUTER INPUT MICROFILM,
CISC - COMPLEX INSTRUCTION SET COMPUTER,
CIU - COMPUTER INTERFACE UNIT,
CMOS - COMPLEMENTARY METAL-OXIDE SEMICONDUCTOR,
CNC - COMPUTER NUMERICAL CONTROL,
CPI - CHARACTERS PER INCH,
COM - COMPUTER OUTPUT MICROFILM,
CPL - CURRENT PRIVILEGE LEVEL,
CPM - CONTROL PROGRAM FOR MICROCOMPUTERS (CP/MONITOR),
CPM - CRITHICAL PATH METHOD,
CPS - CHARACTERS PER SECOND,
CPU - CENTRAL PROCESSING UNIT,
CRAM - CARD RANDOM ACCESS MEMORY,
CRC - CYCLICAL REDUNDANCY CHECK,
CRC - CYCLIC REDUNDANCY CHECK CHARACTER,
CROM - CONTROL AND READ-ONLY MEMORY,
CRT - CATHODE-RAY TUBE,
CSL - CONTROL AND SIMULATION LANGUAGE,
CT - COMPUTER TOMOGRAPH,
CTI - COAXIAL TRANSCEIVER INTERFACE,
CU - CONTROL UNIT,
CVSD - CONTINUOUSLY VARIABLE SLOPE DELTAMODULATION,
DAA - DATA ACCESS ARRANGEMENT,
DAS - DATA ACQUISITION SYSTEM,
DBMS - DATABASE MANAGEMENT SYSTEM,
DCE - DATA COMMUNICATION EQUIPMENT,
DCN - DISTRIBUTED COMPUTER NETWORK,
DDCMP - DIGITAL DATA COMMUNICATIONS MESSAGE PROTOCOL,
DDD - DIRECT DISTANCE DIALING,
DDE - DIRECT DATA ENTRY,
DDL - DATA DESCRIPTION LANGUAGE,
DDS - DATA-PHONE DIGITAL SERVICE,
DEL - DELETE CHARACTER,
DES - DATA ENCRYPTION STANDARD,
DFA - DIGITAL FAULT ANALYSIS,
DG - DATAGRAM,
DI - DIGITAL INPUT,
DIC - DIGITAL INTEGRATED CIRCUIT,
DID - DIGITAL INFORMATION DISPLAY,
DIP - DUAL IN-LINE PACKAGE,
DLE - DATA LINK ESCAPE,
DMA - DIRECT ACCESS MEMORY,
DMAC - DIRECT MEMORY ACCESS CONTROLLER,
DML - DATA MANIPULATION LANGUAGE,
DNC - DIRECT NUMERICAL CONTROL,
DO - DIGITAL OUTPUT,
DOC - DIAGNOSTICS ON-CHIP,
DOR - DIGITAL OPTICAL RECORDING,
DOS - DISC OPERATING SYSTEM,
DPB - DISK PARAMETER BLOCK,
DPL - DESCRIPTOR PRIVILEGE LEVEL,
DRAM - DYNAMIC RANDOM ACCESS MEMORY,
DRAW - DIGITAL READ AND WRITE,
DTA - DISK TRANSFER AREA,
DTE - DATA TERMINAL EQUIPMENT,
DTE - DT-EQUIPMENT,
DU - DISK UNIT,
E^2 - ELECTRICALLY ERASABLE DEVICE,
EA - EFFECTIVE ADDRESS,
EAC - END-AROUND CARRY,
EARN - EUROPEAN ACADEMIC RESEARCH NETWORK,
EAROM - ELECTRICALLY ALTERABLE ROM,
EARS - ELECTRONIC AUDIO RECOGNITION SYSTEM,
EBCDIC - EXTENTED BINARY-CODED DECIMAL INTERCHANGE CODE,
EBNF - EXTENDED BACKUS-NAUR FORM,
ECC - ERROR CORRECTION CODES,
ECD - ENHANCED CONSOLE DRIVER,
ECGB - ENHANCED COLOR GRAPHICS BOARD,
ECMA - EUROPEAN COMPUTER MANUFACTURES' ASSOCIATION,
ECTL - EMITTER-COUPLED TRANSISTOR LOGIC,
EDC - ERROR DETECTION/CORRECTION,
EDP - ELECTRONIC DATA PROCESSING,
EDPC - ELECTRONIC DATA PROCESSING CENTER,
EDPE - ELECTRONIC DATA PROCESSING EQUIPMENT,
EDPM - ELECTRONIC DATA PROCESSING MACHINE,
EDR - EXTERNAL DATA REPRESENTATION,
EDSAC - ELECTRONIC DELAY STORAGE AUTOMATIC CALCULATOR,
EDT - EDITOR,
EDVAC - ELECTRONIC DISCRETE VARIABLE AUTOMATIC CALCULATOR,
EFL - ERROR FREQUENCY LIMIT,
EGA - ENHANCED GRAPHIC ADAPTER,
EIA - ELECTRONIC INDUSTRIES ASSOCIATION,
EIN - EUROPEAN INFORMATION NETWORK,
ELD - ELECTROLUMINESCENT DISPLAY,
ELSI - EXTRA-LARGE-SCALE INTEGRATION,
EMI - ELECTROMAGNETIC INTERFERENCE,
EMOS - ENHANCEMENT MOS,
ENQ - ENQUIRY,
EOB - END OF BLOCK,
EOF - END OF FILE,
EOI - NON-SPECIFIC END OF INTERRUPT,
EOR - END OF RECORD,
EOT - END OF THE TAPE,
EOT - END OF TRANSMISSION,
EPROM - ERASABLE PROGRAMMABLE READ ONLY MEMORY,
E^2PROM - ELECTRICALLY ERASABLE PROGRAMMABLE ROM,
ERA - ERASE CHARACTER,
EROM - ERASABLE ROM,
ERR - ERROR,
ES - EXTRA SEGMENT,
ESC - ESCAPE CHARACTER,
ESD - EXTERNAL SYMBOL DICTIONARY,
ESS - ELECTRIC SWITCHING SYSTEM,
ETB - END OF TRANSMISSION BLOCK CHARACTER,
ETX - END OF TEXT CHARACTER,
EU - EXECUTION UNIT,
EXE - EXECUTE,
FAT - FILE ALLOCATION TABLE,
FAX - FACSIMILE,
FC - FONT-CHANGE CHARACTER,
FCB - FILE CONTROL BLOCK,
FCFS - FIRST-COME FIRST-SERVED,
FCP - FILE CONTROL PROCESSOR,
FCS - FRAME CHECK SEQUENCE,
FD - FILE DESCRIPTION,
FDB - FILE DESCRIPTION BLOCK,
FDM - FREQUENCY DIVISION MULTIPLEXING,
FEP - FRONT END PROCESSOR,
FET - FIELD EFFECT TRANSISTOR,
F/F - FLIP-FLOP,
FF - FIRST FIT ALGORITHM,
FF - FORM FEED,
FFD - FIRST FIT DECREASING ALGORITHM,
FIB - FILE IDENTIFICATION BLOCK,
FIFO - FIRST-IN FIRST-OUT,
FILER - FILE MANAGER,
FILO - FIRST-IN LAST-OUT,
FINUFO - FIRST-IN-NOT-USED-FIRST-OUT,
FLOPS - FLOATING-POINT OPERATIONS PER SECOND,
FLX - FILE EXCHANGE UTILITY,
FM - FREQUENCY MODULATION,
FMT - FORMAT,
FPLA - FIELD-PROGRAMMABLE LOGIC ARRAY,

FPROM - FIELD-PROGRAMMABLE ROM,
FPU - FLOATING POINT PROCESSING UNIT,
FS - FILE SEPARATOR,
FSK - FREQUENCY SHIFT KEYING,
GE - GREATER OR EQUAL,
GEM - GRAPHICS ENVIRONMENT MANAGER,
GIGO - GARBAGE IN, GARBAGE OUT,
GKS - GRAPHICS KERNEL SYSTEM,
GPS - GENERAL PROBLEM SOLVING,
GSX - GRAPHICS SYSTEM EXTENSION,
GT - GREATER THEN,
HASP - HOUSTON AUTOMATIC SPOOLING PROGRAM,
HDAM - HIERARCHICAL DIRECT ACCESS METHOD,
HDLC - HIGH LEVEL DATA LINK CONTROL,
HDR - HIGH DENSITY RECORDING,
HGC - HERCULES GRAPHICS CARD,
HIC - HYBRID INTEGRATED CIRCUIT,
HISAM - HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD,
HLL - HIGH-LEVEL LANGUAGE,
IAD - IMAGE ACQUISITION AND DISPLAY,
IAS - IMMEDIATE ACCESS STORAGE,
IBM - INTERNATIONAL BUSINESS MACHINES,
IC - INTEGRATED CIRCUIT,
ICA - INTRA-APPLICATION COMMUNICATIONS AREA,
ICL - INTERNATIONAL COMPUTERS LTD.,
ID - IDENTIFY MARKER,
IDAS - INDUSTRIAL DATA ACQUISITION SYSTEM,
IDP - INTEGRATED DATA PROCESSING,
IDS - INTEGRATED DATA STORE,
IFF - INTERCHANGE FILE FORMAT,
IFIP - INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING,
ILBM - INTERLEAVE BIT MAP,
IMIS - INTEGRATED MANAGEMENT INFORMATION SYSTEM,
IMP - INTERFACE MESSAGE PROCESSOR,
IMPACT - IMPLEMENTATION PLANNING AND CONTROL TECHNIQUE,
IMR - INTERRUPT-MASK REGISTER,
INFO - INFORMATION FIELD,
INT - INTEGER,
INT - INTERRUPT,
I/O - INPUT/OUTPUT,
IOCS - INPUT/OUTPUT CONTROL SYSTEM,
IOS - INPUT/OUTPUT SYSTEM,
IP - IMAGE POSITIONING,
ISAM - INDEX-SEQUENTIAL ACCESS METHOD,
ISO - INTERNATIONAL STANDARDS ORGANISATION,
ISR - INTERRUPT SERVICE ROUTINE,
ISR - IN SERVICE REGISTER,
IT - INFORMATION TECHNOLOGY,
ITB - INTERMEDIATE TEXT BLOCK,
JCL - JOB CONTROL LANGUAGE,
JMP - JUMP,
KB - KILOBYTE,
KBD - KEYBOARD,
KSAM - KEYED SEQUENTIAL ACCESS METHOD,
LAN - LOCAL AREA NETWORK,
LCD - LIQUID-CRYSTAL DISPLAY,
LCFS - LAST COME-FIRST SERVED,
LF - LINE FEED,
LFU - LEAST FREQUENTLY USED REMOVAL,
LIC - LINEAR INTEGRATED CIRCUIT,
LIFO - LAST-IN FIRST-OUT,
LIOCS - LOGICAL INPUT/OUTPUT CONTROL SYSTEM,
LISP - LIST PROCESSING (LANGUAGE),
LPC - LINEAR PREDICTIVE CODING,
LRU - LEAST RECENTLY USED REMOVAL,
LSB - LOWER-SET SIGNIFICANT BIT,
LSI - LARGE SCALE INTEGRATION,
LT - LESS THAN,
LUT - LOOKUP TABLE,
MAC - MEDIUM ACCESS CONTROL,
MAR - MEMORY ADDRESS REGISTER,
MAU - MEDIUM ATTACHMENT UNIT,
MBR - MEMORY BUFFER REGISTER,
M.C. - MACHINE CYCLE,
MCU - MICROPROCESSOR CONTROL UNIT,
MDA - MONOCHROME DISPLAY/PRINTER ADAPTER,
MDI - MEDIUM DEPENDED INTERFACE,
MDR - MEMORY DATA REGISTER,
MFM - MODIFIED FREQUENCY MODULATION,
MGC - MONOCHROME GRAPHIC/PRINTER CARD,
MIC - MONOLITHIC INTEGRATED CIRCUIT,
MIC - MICROWAVE INTEGRATED CIRCUIT,
MICR - MAGNETIC INK CHARACTER READER,
MICR - MAGNETIC INK CHARACTER RECOGNITION,
MIDI - MUSICAL INSTRUMENT DIGITAL INTERFACE,
MIS - MANAGEMENT INFORMATION SYSTEM,
MMU - MEMORY MANAGEMENT UNIT,
MODEM - MODULATION/DEMODULATION DEVICE,
MOP - MULTIPLE ON-LINE PROGRAMMING,
MS DOS - MICROSOFT DOS,
MSI - MEDIUM-SCALE INTEGRATION,
MTTF - MEAN TIME TO FAILURE,
NAK - NEGATIVE ACKNOWLEDGEMENT,
NC - NUMERICAL CONTROL,
NCP - NETWORK CONTROL PROGRAM,
NDRO - NON-DESTRUCTIVE READOUT,
NIC - NETWORK INTERFACE CONTROLLER,
NIU - NETWORK INTERFACE UNITS,
NLQ - NEAR LETTER QUALITY,
NO-OP - NO-OPERATION INSTRUCTION,
NPR - NON-PROCESSOR REQUEST,
NRFD - NOT READY FOR DATA,
OCR - OPTICAL CHARACTER RECOGNITION,
ODS - OUTPUT DATA STROBE,
OEM - ORIGINAL EQUIPMENT MANUFACTURER,
OPR - OPERATION REGISTER,
PAM - PARTITIONED ACCESS METHOD,
PC - PERSONAL COMPUTER,
PC - PROGRAM COUNTER,
PCM - PULSE CODE MODULATION,
PCN - PERSONAL COMPUTER NETWORK,
PCU - PERIPHERAL CONTROL UNIT,
PDC - PROCESS DIGITAL CONTROLLER,
PDN - PUBLIC DATA NETWORK,
PDS - PROGRAM DEVELOPMENT SYSTEM,
PERT - PROGRAM EVALUATION AND REVIEW TECHNIQUE,
PGC - PROFESSIONAL GRAPHICS CONTROLLER,
PGS - PROFESSIONAL GRAPHICS SYSTEM,
PIC - PROGRAMMABLE INTERRUPT CONTROLLER,
PIC - PERSONAL IDENTIFICATION CODE,
PIF - PROGRAM INFORMATION FILE,
PIO - PROGRAMMABLE INPUT OUTPUT,
PIP - PERIPHERAL INTERCHANGE PROGRAM,
PIPS - PAN INFORMATION PROCESSING SYSTEM,
PL/1 - PROGRAMMING LANGUAGE 1,
PLA - PROGRAMMABLE LOGIC ARRAY,
PLC - PROGRAMMABLE LOGIC CONTROLLER,
PL/M - PROGRAMMING LANGUAGE FOR MICROPROCESSORS,
PL/Z - PROGRAMMING LANGUAGE FOR ZILOG,
PM - PHASE MODULATION,
PMD - POSTMORTEM DUMP,
POST - POWER-ON SELF TEST,
POS TERMINAL - POINT-OF-SALE TERMINAL,
PPI - PROGRAMMABLE PERIPHERAL INTERFACE,
PROM - PROGRAMMABLE READ-ONLY MEMORY,
PSN - PACKED SWITCHING NETWORK,
PSU - POWER SUPPLY UNIT,
PSW - PROCESSOR STATUS WORD,
PSW - PROGRAM STATUS WORD,
QBE - QUERY BY EXAMPLE,
QISAM - QUEUED INDEXED SEQUENTIAL ACCESS METHOD,
QUSAM - QUEUED SEQUENTIAL ACCESS METHOD,
QUTAM - QUEUED TELECOMMUNICATION ACCESS METHOD,
RACS - REMOTE-ACCESS COMPUTING SYSTEM,
RAM - RANDOM ACCESS MEMORY,
RCTL - RESISTOR-CAPACITOR TRANSISTOR LOGIC,
RECOL - RETRIEVAL COMMAND LANGUAGE,
REPROM - REPROGRAMMABLE MEMORY,
RGB - RED-GREEN-BLUE,
RISC - REDUCED INSTRUCTION SET COMPUTER,
RJE - REMOTE JOB ENTRY,
RMM - READ MOSTLY MEMORY,
ROM - READ-ONLY MEMORY,
ROMP - READ-ONLY MEMORY PROGRAMMABLE,
ROS - READ ONLY STORAGE,
RPC - REMOTE PROCEDURE CALL,
RPG - REPORT PROGRAM GENERATOR,
RPN - REVERSE POLISH NOTATION,
RPROM - REPROGRAMMABLE ROM,
RS - RECORD SEPARATOR,
RSX - RESOURCE SHARING EXECUTIVE,
RTC - REAL-TIME CLOCK,
RTL - RESISTOR-TRANSISTOR LOGIC,
RVI - REVERSE INTERRUPT,
RZ - RETURN TO ZERO (CODING),
SAA - SYSTEM APPLICATION ARCHITECTURE,
SAM - SEQUENTIAL ACCESS METHOD,
SATO - SELF-ALIGNED THICK OXIDE (TECHNOLOGY),
SBC - SINGLE-BOARD COMPUTER,
SCCS - SOURCE CODE CONTROL SYSTEM,
SCRAM - STATIC COLUMN RAM,
SDA - SOURCE DATA AUTOMATION,
SDI - SELECTIVE DISSEMINATION OF INFORMATION,
SDLC - SYNCHRONOUS DATA LINK CONTROL,
SGPA - STATIC GRAPH PARTITIONING ALGORITHM,
SI - SHIFT-IN CHARACTER,
SID - SOUND INTERFACE DEVICE,
SIMPP - SIMPLE IMAGE-PROCESSING PACKAGE,
SJN - SHORTEST JOB NEXT,

dokończenie w następnym numerze

INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

EDIT SESSION - sesja edycji,
EDIT WORD - słowo wydawnicze,
EDP - patrz: ELECTRONIC DATA PROCESSING,
EDPC - patrz: ELECTRONIC DATA PROCESSING CENTER,
EDPE - patrz: ELECTRONIC DATA PROCESSING EQUIPMENT,
EDPM - patrz: ELECTRONIC DATA PROCESSING MACHINE,
EDR - patrz: EXTERNAL DATA REPRESENTATION,
EDSAC - patrz: ELECTRONIC DELAY STORAGE AUTOMATIC CALCULATOR,
EDT - patrz: EDITOR,
EDUCABILITY - edukacja, nauczanie,
EDUCATION - patrz: EDUCABILITY,
EDUCATIONAL AIDS - pomoce naukowe,
EDUCATIONAL STATUS - poziom wykształcenia,
EDUCATIONAL SYSTEM - szkolnictwo,
EDULCORATE - usuwać błędną informację,
EDVAC - patrz: ELECTRONIC DISCRETE VARIABLE AUTOMATIC CALCULATOR,
EFFECTIVE - skuteczny, użyteczny, efektywny,
EFFECTIVE ADDRESS - adres efektywny,
EFFECTIVE FOR ... - obowiązujący na okres ...,
EFFECTIVE TIME - czas efektywny, czas użytkowy,
EFFECTIVE TRANSFER RATE - praktyczna szybkość przesyłania (danych),
EFFICIENCY - sprawność, współczynnik sprawności, także: wydajność, efektywność,
EFFICIENT - sprawny, wydajny, efektywny,
EFL - patrz: ERROR FREQUENCY LIMIT,
EGA - patrz: ENHANCED GRAPHIC ADAPTER,
EGOLESS PROGRAMMING - programowanie grupowe (technologia programowania),
EIA - patrz: ELECTRONIC INDUSTRIES ASSOCIATION,
EIGENVALUE - wartość własna,
EIGENVECTOR - wektor własny,
EIGHT'S COMPLEMENT - uzupełnienie do ośmiu, dopełnienie do ośmiu,
EIGHTY-COLUMN CARD - karta 80-kolumnowa,
EIN - patrz: EUROPEAN INFORMATION NETWORK,
EITHER-WAY - półdupleks,
EITHER-WAY CIRCUIT - kanał półdupleksowy,
EJECT - wyrzucać, wypychać, wyprowadzać, wydalać (np. karty z maszyny),
EJECTION - wyrzucanie, wypychanie, wyprowadzanie,
ELABORATE - wypracowany, starannie wykonany (we wszystkich szczegółach), złożony (np. mechanizm), skomplikowany,
ELABORATION - opracowanie, wypracowanie,
ELAPSED TIME CLOCK - licznik zegarowy,
ELECTRIC SENSING - odczytywanie elektryczne,
ELECTRIC SWITCHING SYSTEM - elektroniczny układ przełączania,
ELECTRICAL - elektryczny,
ELECTRICALLY - elektrycznie,
ELECTRICALLY ALTERABLE ROM - pamięć stała reprogramowalna elektrycznie, pamięć EAROM,
ELECTRICALLY ERASABLE DEVICE - urządzenie pamięciowe wymazywalne i programowalne elektrycznie,
ELECTRICALLY ERASABLE ROM - pamięć stała wymazywalna elektrycznie,
ELECTRICITY - elektryczność,
ELECTROMECHANIC - elektromechaniczny,
ELECTROMAGNETIC INTERFERENCE - zakłócenia elektromagnetyczne,
ELECTROMAGNETIC RELAY - przekaźnik elektromagnetyczny,
ELECTRONIC ANALOG COMPUTER - analogowa maszyna elektroniczna,
ELECTRONIC BRAIN - komputer elektroniczny, elektroniczna maszyna matematyczna, "mózg elektronowy",
ELECTRONIC DATA PROCESSING - elektroniczne przetwarzanie danych, EPD,
ELECTRONIC DATA PROCESSING CENTER - centrum elektronicznego przetwarzania danych,
ELECTRONIC DATA PROCESSING EQUIPMENT - elektroniczny sprzęt do przetwarzania danych,
ELECTRONIC DATA PROCESSING MACHINE - elektroniczna maszyna do przetwarzania danych,
ELECTRONIC DELAY STORAGE AUTOMATIC CALCULATOR - elektroniczny licznik automatyczny na liniach opóźniających,

ELECTRONIC DISCRETE VARIABLE AUTOMATIC CALCULATOR - elektroniczny licznik automatyczny zmiennych dyskretnych,
ELECTRONIC DOCUMENT - elektroniczny dokument, elektroniczny tekst,
ELECTRONIC INDUSTRIES ASSOCIATION - Organizacja Standardów Elektronicznych,
ELECTRONIC MAIL - elektroniczna poczta (środki przesyłania i ochrony wiadomości pomiędzy użytkownikami sieci komputerowej),
ELECTRONIC PACKAGING - montaż zespołów z mikroukładów elektronicznych,
ELECTRONIC POCKET-SIZE CALCULATOR - kieszonkowy kalkulator elektroniczny, minikalkulator,
ELECTRONIC SCULPTURING - modelowanie za pomocą komputera analogowego,
ELECTRONIC WORKSHEETS - tabele kalkulacyjne (programy odtwarzające na komputerze zadania planowania i kalkulowania),
ELECTRONICS - elektronika,
ELECTROSTATIC PLOTTER - ploter elektrostatyczny, pisak elektrostatyczny,
ELECTROSTATIC PRINTER - drukarka elektrostatyczna,
ELECTROSTATIC PRINTER-PLOTTER - patrz: ELECTROSTATIC PRINTER,
ELECTROTHERMAL PRINTER - drukarka termiczna,
ELEMENT - element, składnik, część składowa, człon,
ELEMENTARY - elementarny, jednostkowy,
ELEVATE - podnosić,
ELEVATION - podnoszenie, wzniesienie, także: rzut pionowy,
ELEVEN PUNCH - naddziurka w strefie jednostek (w 80-cio kolumnowej karcie dziurkowanej),
ELIGIBLE - gotowy do kontynuacji (np. do dalszej pracy),
ELIMINATION - eliminacja, usuwanie,
ELITE FACE - tryb drukowania z gęstością 12 znaków na cal,
ELONGATE - wydłużać (się),
ELSE - inaczej,
ELSI - patrz: EXTRA-LARGE-SCALE INTEGRATION,
EMBEDDING - wstawka, także: obejmujący, osłaniający,
EMBEDED - włączony, wstawiony, wbudowany (np. podprogram, procedura w ramach programu),
EMBODY - wbudować, włączyć, obejmować,
EMBOSSMENT - zniekształcenie,
EMERGENCY - awaria, stan zagrożenia, niebezpieczeństwo, wypadek,
EMERGENCY CONDITION - sytuacja krytyczna, warunek krytyczny,
EMI - patrz: ELECTROMAGNETIC INTERFERENCE,
EMITTING - emitowanie,
EMOS - patrz: ENHANCEMENT MOS,
EMPHASIZED - wytłuszczone,
EMPIRICAL - empiryczny, doświadczalny,
EMPIRICAL PROBABILITY - prawdopodobieństwo a posteriori,
EMPLOY - zatrudniać,
EMPLOYER - pracownik,
EMPTY - pusty, próżny,
EMPTY LOOP - cykl pusty,
EMPTY SET - zbiór pusty,
EMPTY STRING - łańcuch pusty (o zerowej długości),

EMULATE - imitować, naśladować, także: współzawodniczyć,
EMULATION - emulacja, także: symulacja urządzenia, tworzenie odpowiednika,
EMULATOR - emulator,
ENABLE - umożliwiać,
ENABLE INTERRUPTS - włącz przerwania,
ENABLE PULSE - impuls zezwalający,
ENABLING A LINE - umożliwienie połączenia, nawiązanie połączenia,
ENCAPSULATE - hermetyzować,
ENCAPSULATED ELEMENT - mikroelement w obudowie,
ENCAPSULATION - obudowa, hermetyzacja (mikroelementu), także: przedstawić coś, opisać w sposób ukryty,
ENCIPHER - zaszyfrować, zakodować,
ENCIPHERING - szyfrowanie, kodowanie,
ENCLOSED HEREWITH - w załączeniu,
ENCLOSING - osłanianie, obejmowanie,
ENCLOSURE - załącznik,
ENCODE - kodować, zakodować,
ENCODER - koder, urządzenie kodujące, szyfrator,
ENCODING - kodowanie, szyfrowanie,
ENCOMPASS - obejmować,
ENCRYPTED MESSAGE - wiadomość zaszyfrowana,
ENCRYPTION - szyfrować, utajniać, także: szyfrowanie, kodowanie,
ENCRYPTION OF INFORMATION - utajnianie informacji,
END - koniec,
END ADDRESS - adres końcowy,
END-AROUND BORROW - ujemne przeniesienie cykliczne,
END-AROUND CARRY - przeniesienie cykliczne (z najwyższej pozycji liczby do najniższej),
END DISTORTION - zniekształcenie końcowe,
END-FILE MARK - znacznik końca pliku (zbioru),
ENDLESS FORM - formularz (druk) ciągły,
ENDLESS LOOP - pętla nieskończona,
ENDLESS TAPE - taśma "bez końca",
END MARK - znak końca,
END MODE - tryb zakończenia (rozkazu),
END OF BLOCK - znak końca bloku,
END OF FILE - znacznik końca zbioru,
END-OF-FILE INDICATOR - wskaźnik końca pliku (zbioru) danych,
END-OF-FILE LABEL - etykieta końca zbioru danych,
END OF LINE - koniec wiersza,
END OF MEDIUM - koniec nośnika,

ENGINEERING - technika, inżynieria,
ENGINEERING CYBERNETICS - cybernetyka techniczna,
ENGINEERING TIME - czas obsługi technicznej,
ENHANCE - uwydatniać, uwypuklać, wzmagać, zwiększać, wzmacniać,
ENHANCED - ulepszony, polepszony, udoskonalony,
ENHANCED COLOR DISPLAY - wyświetlacz ulepszonego koloru,
ENHANCED COLOR GRAPHICS BOARD - karta ulepszonej grafiki kolorowej,
ENHANCED GRAPHICS ADAPTER - adapter ulepszonej grafiki,
ENHANCED MODEL - model rozwinięty, model wzbogacony,
ENHANCED VERSION - wersja rozwinięta, wersja wzbogacona,
ENHANCEMENT - rozszerzanie, także: udoskonalanie, wzbogacanie,
ENHANCEMENT MOS - tranzystor MOS z kanałem wzbogaconym,
ENHANCEMENT TRANSISTOR - tranzystor polowy o kanale wzbogaconym, tranzystor ze wzbogaceniem,
ENLARGE - powiększać,
ENLARGEMENT - powiększenie,
ENQ - patrz: ENQUIRY,
ENQUEUE - ustawić w kolejce,
ENQUIRY - zapytanie (np. systemu),
ENQUIRY CHARACTER - symbol zapytania,
ENSAMBLE - zbiór (w matematyce),
ENSURE - zapewniać, upewniać się,
ENTER - umieszczać, wnosić, wprowadzać,
ENTERPRISE - przedsiebiorstwo, także: przedsięwzięcie,
ENTERPRISE ADMINISTRATOR - administrator dziedziny przedmiotowej,
ENTERPRISE DATA BASE - baza danych przedsiębiorstwa,
ENTITY - wyodrębniona całość, jednostka istnienia, istota, sens, sedno, także: objekt,
ENTRAP - zatrzymać, zamknąć,
ENTRY - komunikat wprowadzony do komputera, wejście, zapis (w katalogu), hasło (w słowniku), także: instrukcja, deklaracja w programie,
ENTRY BLOCK - blok wejścia,
ENTRY LABEL - patrz: ENTRY NAME,
ENTRY NAME - nazwa (punktu) wejścia,
ENTRY POINT - punkt wejścia, adres (punktu) wejścia,
ENUMERABLE - policzalny,
ENUMERATE - wyliczać,
ENUMERATED TYPE - typ wyliczeniowy,
ENUMERATION - numeracja, także: przeliczalność,
ENUMERATIVE ALGORITHM - algorytm wyliczeniowy, algorytm przeliczeniowy,
ENVELOPE - koperta, otoczka, powłoka, także: obwiednia,
ENVELOPE CURVE - obwiednia,
ENVELOPE DELAY - opóźnienie grupowe,
ENVELOPE FEEDBACK - sprzężenie zwrotne wewnętrzne,
ENVIRONMENT - zestaw maszynowy, warunki pracy, środowisko, otoczenie,
ENVIRONMENT DIVISION - dział zestawu maszynowego (w Cobolu),
ENVIRONMENTAL - środowiskowy,
ENVIRONMENTAL INPUT - wejście środowiskowe,
EOB - patrz: END OF BLOCK,
EOF - patrz: END OF FILE,
EOI - patrz: NON-SPECIFIC END OF INTERRUPT,
EOR - patrz: END OF RECORD,
EOT - patrz: END OF THE TAPE,
EOT - patrz: END OF TRANSMISSION,
EPROM - patrz: ERASABLE PROGRAMMABLE READ ONLY MEMORY,
E^2PROM - patrz: ELECTRICALLY ERASABLE ROM,
EPSS - sieć komunikacyjna EPSS,
EQUAL - równać się, być równym,
EQUAL ODDS - równe szanse,
EQUALITY - równość,
EQUALITY SIGN - znak równości, (identyczności),
EQUALIZATION - korygowanie,
EQUALIZE - wyrównywać, zrównywać,
EQUALIZER - korektor,
EQUALIZING - wyrównywanie,
EQUALLY - w równym stopniu,
EQUALLY LIKELY CASES - przypadki równie prawdopodobne,
EQUALS SIGN - znak równości, "=",
EQUATE - przyrównywać, także: sprowadzać do wspólnego mianownika,
EQUATES - stałe przyporządkowania,
EQUATION - równanie, także: wzór matematyczny,
EQUATION SOLVER - analizator wielomianów algebraicznych,
EQUIANGULAR - równokątny,
EQUI-BOUNDED - wspólnie ograniczony,
EQUIDISTANT - w równych odstępach,
EQUILATERAL - równoboczny,
EQUILIBRATE - równoważyć,
EQUILIBRIUM - równowaga,
EQUINUMEROUS - równoliczny,
EQUIP - wyposażać,

EQUIPMENT - wyposażenie, sprzęt,
EQUIPMENT COMPATIBILITY - wymienność sprzętu,
EQUIPMENT FAILURE - awaria sprzętu,
EQUIPOLLENT - równoważny,
EQUIPONDERATE - równoważyć,
EQUIPOTENT - równoliczny,
EQUIPROBABLE - jednakowo prawdopodobny, o tym samym stopniu prawdopodobieństwa,
EQUIVALENCE - równoważność, równowartość,
EQUIVALENCE ELEMENT - element równoważności,
EQUIVALENT - równoważny, równowartościowy, równorzędny,
EQUIVALENT CIRCUIT - układ zastępczy,
ERA - patrz: ERASE CHARACTER,
ERASABILITY - wymazywalność,
ERASABILITY OF STORAGE - kasowalność danych,
ERASABLE - dający się wymazać,
ERASABLE PROGRAMMABLE READ ONLY MEMORY - pamięć stała, którą można programować i kasować (np. kasowanie elektryczne),
ERASABLE ROM - pamięć stała kasowalna, pamięć ROM kasowalna,
ERASABLE STORAGE - pamięć kasowalna,
ERASE - wycierać, wymazywać, kasować,
ERASE CHARACTER - znak wymazujący, wycierający, znak błędu,
ERASE KEY - klawisz anulowania, klawisz kasowania,
ERASE ROUTINE - podprogram zerujący, program zerujący, podprogram kasujący,
ERASE TIME - czas wymazywania (w monitorach z lampą pamięciową),
ERASING - wymazywanie, usuwanie,
ERASURE - miejsce wymazane, przekreślenie w tekście, także: wycieranie (informacji gumką),
E-REGISTER - patrz: EXTENSION REGISTER,
EROM - patrz: ERASABLE ROM,
ERR - patrz: ERROR,
ERRATIC - błędny, pomyłkowy,
ERRONEOUS - omyłkowy, błędny,
ERRONEOUSLY - omyłkowo, przez omyłkę,
ERROR - błąd, uchyb,
ERROR BURST - grupa błędów, pakiet błędów występujących z tego samego powodu,
ERROR BYTE - bajt błędu,
ERROR CHECK - kontrola błędu,
ERROR-CHECKING CODE - kod wykrywający błędy, kod sprawdzający,
ERROR CONDITION - przypadek wystąpienia błędu,
ERROR CONTROL - kontrola błędów,
ERROR CORRECTING - korekcja błędów,
ERROR CORRECTING CODE - patrz: ERROR CORRECTION CODE,
ERROR-CORRECTING COMPILER - kompilator z automatyczną korekcją błędów,
ERROR CORRECTION - korekcja błędów,
ERROR CORRECTION CODE - kod korygujący błędy, kod samokorygujący,
ERROR DETECTING CODE - kod detekcyjny (wykrywający błędy),
ERROR DETECTION - wykrywanie błędu,
ERROR DETECTION/CORRECTION - detekcja i korekcja błędów,
ERROR DIAGNOSTIC - komunikaty o błędach, doniesienia o błędach,
ERROR FLAG - flaga błędu, znacznik błędu,
ERROR-FREE - bezbłędny, wolny od błędów,
ERROR FREQUENCY LIMIT - częstość graniczna błędów,
ERROR GAP - odstęp między błędami,
ERROR HANDLING - obsługa błędu,
ERROR HANDLING ROUTINE - podprogram obsługi błędów,
ERROR INDICATE LIGHT - świetlny sygnał błędu,
ERROR INDICATION - wykazywanie błędów,
ERROR INTERRUPT - przerwa w wykonywaniu programu w skutek błędu, który nie może być automatycznie skorygowany,
ERROR MEAN-SQUARE - błąd średniokwadratowy,
ERROR MESSAGE - komunikat błędu,
ERROR PRINTOUT - wykaz błędów,
ERROR PROBABILITY - prawdopodobieństwo popełnienia błędu,
ERROR-PRONE - łatwo ulegający błędom,
ERROR PROPAGATION - powtarzanie (powielanie) błędu,
ERROR PROTECTION - ochrona przed popełnieniem błędu,
ERROR RATE - wskaźnik błędów (np. przy odczycie taśmy),
ERROR RECOVERY - korekcja błędów,
ERROR ROUTINE - program korekcyjny,
ERROR SEARCH PROGRAM - program wyszukiwania błędów,
ERROR TAPE - taśma błędów,
ERROR TRAPPING - wychwytywanie błędów,
ES - patrz: EXTRA SEGMENT,
ESC - patrz: ESCAPE CHARACTER,
ESCALATION - wzrost, wzmaganie się, narastanie,
ESCAPE - unikać, uciekać, także: ucieczka, przejście do innej opcji,
ESCAPE CHARACTER - znak funkcyjny,
ESCAPE CODE - kod kierujący, kod sterujący,
ESCAPE KEY - klawisz wyjścia, klawisz ucieczki (pozwala na przerwanie wykonywanej pracy na komputerze),

Pole minowe

Ta gra przeznaczona jest dla tych, którzy lubią rozwiązywać zagadki logiczne. Jest to adaptacja gry ze „SPECTRUM" do możliwości graficznych „MERITUM". Zasady gry podane są po jej uruchomieniu. Poziom trudności regulowany jest deklarowaną przez gracza ilością min. Ilość ta praktycznie nie powinna przekraczać 100, bo przejście może być niemożliwe, choć na polu zmieścić się może 280 min.

Życzę przyjemnej zabawy.

Bolesław MRUGAŁA

```
10 RANDOM:CLS
20 DIM M(15,20)
30 PRINT CHR$(23)
40 PRINT@460,"P O L E  M I N O W E"
50 FOR D=1 TO 600:NEXT D
60 CLS:PRINT"TWOJE ZADANIE POLEGA NA PRZEJSCIU PRZEZ POLE Z MINAMI
DO PRZECIWLEGLEJ BRAMY. JEZELI STANIESZ NA POLU SASIADUJACYM
Z POLEM ZAMINOWANYM TO USLYSZYSZ SYGNAL. JEZELI STANIESZ NA
POLU Z MINA - SYGNALU JUZ NIE USLYSZYSZ.
80 PRINT@384,"PRZEMIESZCZAJ SIE ZA POMOCA KLAWISZY:
          I - W GORE
          M - W DOL
          J - W LEWO
          K - W PRAWO

100 INPUT"ILOMA MINAMI MA BYC ZAMINOWANE POLE";MN
105 CLS
110 FOR I=4 TO 121
120 SET(I,2):SET(I,45):NEXT I
130 FOR J=2 TO 17
140 SET(4,J):SET(5,J):SET(120,J):SET(121,J):NEXT J
150 FOR J=30 TO 45
160 SET(4,J):SET(5,J):SET(120,J):SET(121,J):NEXT J
170 MI=0
180 MI=MI+1
190 IF MI>MN THEN 230
200 IX=RND(19):IG=RND(14)
210 IF M(IG,IX)=1 THEN 200
220 M(IG,IX)=1:GOTO 180
230 X=20:Y=7:XS=7:YS=20:XN=XS:YN=YS
240 PRINT@Y*64+3*X,CHR$(191);CHR$(179);CHR$(191);CHR$(19);
260 A$=INKEY$
270 IF A$=""THEN 260
280 IF A$="J"THEN YN=YS-1:GOTO 320
290 IF A$="K"THEN YN=YS+1:GOTO 320
300 IF A$="I"THEN XN=XS-1:GOTO 320
310 IF A$="M"THEN XN=XS+1:GOTO 320
315 GOTO 260
320 IF XN<1 OR XN>14 THEN XN=XS:GOTO 260
330 IF XN>5 AND XN<10 THEN 350
340 IF YN<1 OR YN>19 THEN YN=YS:XN=XS:GOTO 260 ELSE 400
350 IF YN>20 THEN YN=YS:XN=XS:GOTO 260
360 IF YN=0 THEN PRINT@6," B R A W O ";ELSE 400
370 PRINT@XS*64+YS*3+1,CHR$(191);CHR$(19);
380 PRINT@XN*64+YN*3,CHR$(191);CHR$(179);CHR$(191);CHR$(19);
390 GOTO 500

400 PRINT@XS*64+YS*3+1,CHR$(191);CHR$(19);
410 PRINT@XN*64+YN*3,CHR$(191);CHR$(179);CHR$(191);CHR$(19);
420 XS=XN:YS=YN
430 IF M(XN,YN)=1 THEN GOTO 1000
435 A=0:IF YN=20 THEN 440 ELSE IF M(XN,YN+1)=1 THEN A=A+1
440 IF M(XN,YN-1)=1 THEN A=A+1
445 IF M(XN+1,YN)=1 THEN A=A+1
447 IF M(XN-1,YN)=1 THEN A=A+1
448 IF A=0 THEN 260
450 FOR J=1 TO A
460 FOR I=1 TO 5:NEXT I
470 FOR I=1 TO 10
480 OUT 254,1:OUT 254,0:NEXT I
490 NEXT J:GOTO 260
500 FOR I=1 TO 14
510 FOR J=1 TO 19
520 IF M(I,J)=1 THEN PRINT@I*64+J*3+1,CHR$(140);CHR$(19);
530 NEXT J: NEXT I
540 PRINT@966," JESZCZE RAZ? - NACISNIJ SPACJE ";
550 IF INKEY$=" " THEN 600 ELSE 550
600 CLS
610 FOR I=1 TO 20
620 FOR J=1 TO 14
630 M(J,I)=0:NEXT J:NEXT I
640 GOTO 100
1000 FOR J=1 TO 10:NEXT J
1010 FOR I=1 TO 11
1020 PRINT@XN*64+YN*3,CHR$(128);CHR$(140);CHR$(128);CHR$(19);
1030 FOR J=1 TO 10:NEXT J
1040 PRINT@XN*64+YN*3,CHR$(166);CHR$(128);CHR$(153);CHR$(19);
1050 FOR J=1 TO 10 :NEXT J
1060 NEXT I:GOTO 500
```

Liga Myślących

Zadanie 1

Znaleźć trzy najmniejsze liczby naturalne n takie, że między n oraz n+10 nie ma żadnej liczby pierwszej, jak również trzy najmniejsze liczby naturalne m takie, że między 10m a 10(m+1) nie ma żadnej liczby pierwszej.

Zadanie 2

Znaleźć wszystkie liczby naturalne a, dla których liczba $a^{10} + 1$ jest podzielna przez 10.

Zadanie 3

Na stole leżą cztery kule o jednakowym promieniu r, stykające się jedna z drugą. We wnękę utworzoną przez nie położono piątą kulę o tym samym promieniu. Znaleźć odległość najwyżej położonego punktu piątej kuli od płaszczyzny stołu.

Zadanie 4

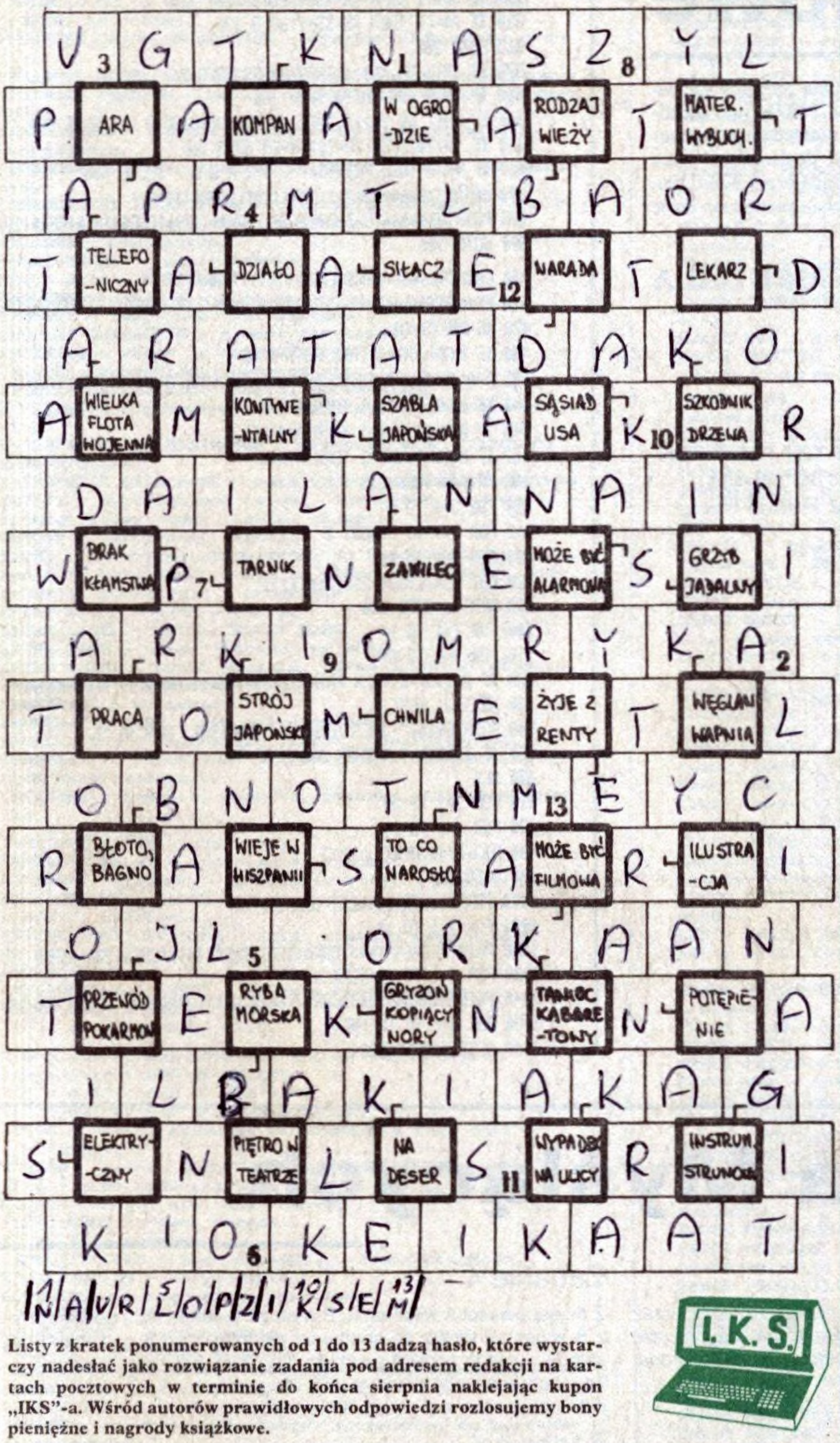
Z miejscowości A w kierunku B wyszedł piechur. Po a godzinach z z B wyjechał kolarz na spotkanie piechura i po b godzinach od swego wyjazdu spotkał piechura. Ile czasu potrzeba kolarzowi, a ile piechurowi, aby przebyć całą drogę między A i B, jeżeli kolarz potrzebuje na to o c godzin mniej niż piechur?

Zadanie 5

Wydajność warsztatu A wynosi m procent wydajności warsztatów B i C, a wydajność warsztatu B wynosi n procent wydajności warsztatów A i C. Jaki procent wynosi wydajność warsztatu C w stosunku do wydajności warsztatów A i B?

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

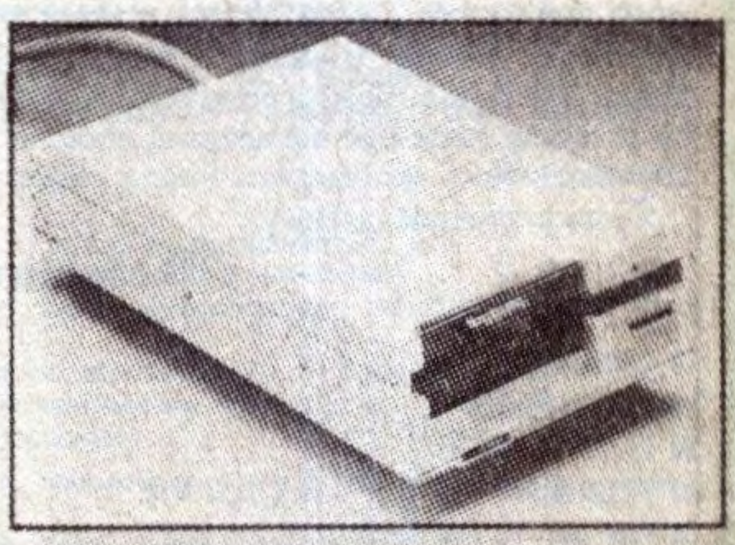
KRZYŻÓWKA NR 5

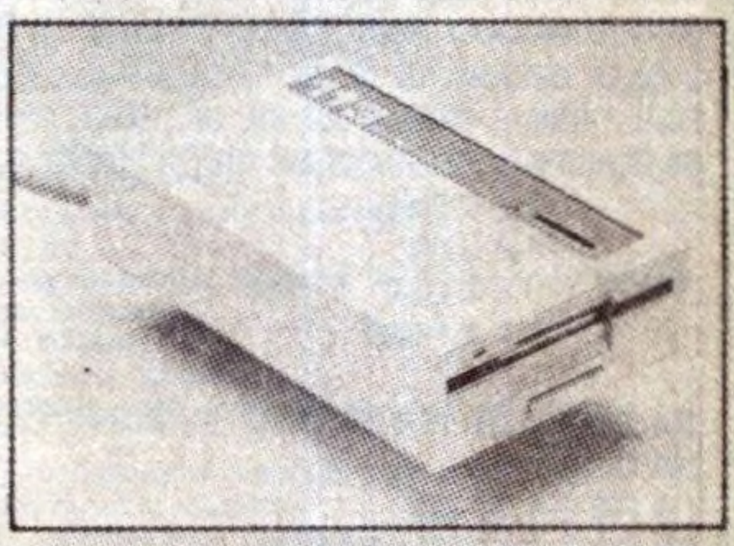


Listy z kratek ponumerowanych od 1 do 13 dadzą hasło, które wystarczy nadesłać jako rozwiązanie zadania pod adresem redakcji na kartach pocztowych w terminie do końca sierpnia naklejając kupon „IKS"-a. Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.

MIKROCIEKAWOSTKI

Bondwell





„IKS" — dodatek „Żołnierza Wolności". Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarra, Romuald Głąb, Włodzimierz Gogołek, Janusz Janiec, Henryk Krasuski, Ireneusz Miernik, Ludwik Piela, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe", Warszawa ul. Grzybowska 77. Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 1350. Nr ind. 361682.

U-23