IKS

INFORMATYKA KOMPUTERY SYSTEMY

CENA 120 zł

DODATEK DO „ŻOŁNIERZA WOLNOŚCI" NR 11/1988 ISSN 0860-2794

UWAGA:

Informatyczny słownik angielsko-polski — cd.

Atari — Kasetowy system operacyjny 2T06 TURBO

Jeśli ocenialibyśmy poziom informatyki i jej zastosowań na podstawie ilości i rozmachu imprez targowych poświęconych tej dyscyplinie to z pewnością znaleźlibyśmy się w ścisłej czołówce państw europejskich i na czele krajów socjalistycznych.

Niestety gala trwa tylko kilka dni i zaraz po niej pojawia się szara rzeczywistość. *Komputer 88*, *Infosystem 88*, *Targi Poznańskie*, *Softarg* i *Informacja 88* w Katowicach przyciągały do swoich stoisk kolorowymi monitorami. A że forma dla pospolitego "oglądacza" ważniejsza jest niż treść, kiedy na koronie katowickiego Spodka na ustawionch tam kilkudziesięciu monitorach zaczęto wyświetlać filmy, arena opustoszała.

Nie oznacza to oczywiście, aby podobne imprezy były zbędne. Informatykastworzyła zdrowy rynek (jeden z nielicznych) i wszelkiego rodzaju targi,wystawy czy pokazy pozwalają wyłowić nowości, a i w konsekwencji urealnić ceny. A jeśli już o nie chodzi to ceny sprzętu dyktowane są przez czarnorynkowy kurs dolara, zaś cena oprogramowania zależy od podaży, popytu i jego wartości.

Oprogramowanie jest coraz lepsze. Gołym okiem już widać jego profesjonalizm. W Katowicach, pomimo iż zabrakło tam wielu softwarowych potentatów nie brakowało ciekawych propozycji.

Oczywiście *Informacje 88* można (i słusznie) skrytykować. Ale jej największa zaletą jest fakt, że po prostu była. Rodzima informatyka z pewnością na tym nie ucierpi. Wypada mieć nadzieję, że organizatorzy w przyszłym roku znacznie uatrakcyjnią formę, a i zwiedzającym przedstawią zanacznie więcej wzstawców.

I jeszcze kilka słów o katowickim Spodku: najlepsze naszym zdaniem propozycje programowe opiszemy na naszych łamach. Już w następnym numerze przedstawimy oprogramowanie dziennikarzowi najbliższe - czyli polska wersja Deskop Publishing Rolanda Wacławka.

REDAKCJA

W NUMERZE

HEWLETT-PACKARD

Drukarka laserowa firmy Hewlett Packard model Laser Jet II jest w chwili obecnej najbardziej popularnym typem drukarki laserowej na świecie, nie tylko ze względu na niską cenę, ale przede wszystkim z powodu bardzo dużych możliwości użytkowych.

Standardowo jest ona wyposażona w pamięć 512 kB RAM z możliwością rozszerzenia do 1MB, 2MB, lub 4 MB. Jako jedna z nielicznych drukuje na papierze każdego rodzaju, foliach przeźroczystych, foliach i papierach samoprzylepnych typu JAC.

Przy zastosowaniu scanera można wykorzystywać ją jako kopiarkę z możliwością obróbki cyfrowej obrazu i druku. Znajduje szerokie zastosowanie w małej poligrafii do składania druku i zdjęć.

Jedną z nielicznych firm oferujących ten model drukarki laserowej jest znana z wysokiej jakości sprzętu służącego do opracowań graficznych przy zastosowaniu techniki komputerowej BIURO OBSŁUGI IMPORTU i EKSPORTU BOIiE mające siedzibę w Warszawie przy ulicy Chłodnej 35/37, pawilon 1a, telefon 24 78 18, tlx 817073 BOIiE PL.

Ostatnim szlagierem proponowanym przez firmę BOIiE jest stacja graficzna APOLLO w standardowej konfiguracji 4MB lub 8 MB RAM, 167 MB hard dysk i rozdzielczości 1024 x 1024 z 1 MB pamięci RAM na dwudziestocalowym kolorowym monitorze.

Na zorganizowanym w listopadzie br. pokazie sprzętu mikrokomputerowego nowej generacji oprócz drukarek laserowych, stacji graficznych, zaprezentowano różnego rodzaju zestawy komputerowe i kartę do obróbki cyfrowej obrazu. Atrakcyjna forma tej prezentacji, z wykorzystaniem dużych ekranów wizyjnych i profesjonalnych mikserów video, cieszyła się dużym zainteresowaniem.

LaserJet series II Printer

Nauka • Technika • Nauka

Firma Dell zapowiedziała rozpoczęcie, z końcem bieżącego roku, produkcji komputera będącego pierwszym odpowiednikiem IBM PS/2 jaki ukaże się na rynku. Jednakże zanim to nastąpi najlepszym komputerem firmy pozostanie najnowszy model o nazwie System 310, bazujący na mikroprocesorze 80386, z częstotliwością zegara 20 MHz, pracujący w standardzie AT. Jego pamięć statyczna RAM jest ograniczona do 32 kbajtów, jednakże zastosowanie procesora o częstotliwości 20 MHz i układu kontrolującego Intel 82385 powoduje, że jest on jednym z szybszych komputerów tej klasy. Jego cena bez wyposażenia dodatkowego 2699 dolarów USA.

Na amerykańskim rynku komputerowym coraz większą popularność zdobywa firma Elxsi, której siedziba znajduje się w San Jose w Kalifornii. Firma produkuje szeroki asortyment komputerów konkurujących wysoką jakością wykonania i niższą ceną z produktami takich renomowanych firm jak IBM czy Digital Equipment Corporation. W 1987 roku obroty jej osiągnęły wielkość 25,2 mln dolarów USA, co stanowi wzrost o prawie 9 mln w stosunku do roku poprzedniego, natomiast w roku bieżącym przewidywany jest wzrost o prawie 10 mln.

Komputery firmy Elxsi są najczęściej wykorzystywane do symulacji i modelowania w czasie rzeczywistym, a ich duża szybkość działania czyni je bardzo przydatnymi do obsługi dużych baz danych. Podstawę konstrukcji komputerów Elxsi stanowi bus danych pracujący z szybkością 320 Megabajtów na sekundę. Zastosowanie wspólnego busu pozwala na łatwą rozbudowę systemu poprzez wymianę procesora na szybszy i zwiększenie pamięci. W najnowszych komputerach Elxsi zastosowany jest 64-bitowy procesor równoległy 6460 i pamięć o pojemności 2000 megabajtów.

Wspomagający procesor neuronowy ANZA-Plus firmy Hecht-Nielsen Neurocomputers umożliwia komputerom IBM PC-AT lub kompatybilnym pracę jako komputer neuronowy wykonujący obliczenia w czasie rzeczywistym. Tak „przerobione" komputery znajdują zastosowanie szczególnie przy analizowaniu obrazów i sygnałów. Sieci neuronowe mogą być stosowane z dowolnymi kombinacjami nie więcej niż 2,5 miliona elementów obliczeniowych i połączeń między nimi. Maksymalna szybkość działania takiego systemu wynosi 10 milionów połączeń na sekundę. Wszystkie funkcje wiążące się z działaniem sieci neuronowej wykonywane są w pamięci procesora wspomagającego umożliwiając tym samym niezależną obsługę funkcji wejścia/wyjścia komputera.

Firma Hetch-Nielsen Neurocomputers (HNC) z San Diego uzyskała zamówienie od armii Stanów Zjednoczonych na zaprojektowanie sieci komputera neuronowego do analizowania i rozwiązywania problemów współczesnego pola walki. Na opracowanie projektu architektury wojskowego komputera sieci neuronowej firma otrzymała zaliczkę wysokości 50 tys. dolarów USA i pół roku na jego zaprezentowanie. Przewiduje się, że komputery sieci neuronowej znajdą zastosowanie głównie przy analizie danych z urządzeń kontrolnych, czujników oraz przy rozwiązywaniu problemów naukowych współczesnego pola walki, takich jak obróbka sygnałów radarowych, sonarowych czy obrazów. Ponieważ komputery sieci neuronowej tworzą własne algorytmy, ich sposób rozróżniania szczegółów jest podobny do ludzkiego, to mogą być przydatne w przewidywaniu zachowań przeciwnika na podstawie obserwacji z ćwiczeń. Wyraża się nadzieję, że komputery te poprzez obserwację czynności człowieka, będą w stanie nauczyć się sterowania robotami zastępującymi człowieka tam, gdzie jest to szczególnie niebezpieczne lub tam, gdzie wykonywane czynności są stale powtarzane. Przewiduje się także, że przyczynią się one do powstania nowych generacji broni, które będą eliminować zmęczenie ludzkie, samoczynnie naprawiać uszkodzenia powstałe w wyniku działań bojowych oraz uwzględniające zmiany zachodzące na polu bitwy.

Firma GTE Governement Systems Corp. otrzymała zamówienie wartości 7,3 min dolarów USA od dowództwa amerykańskich sił powietrznych na dostarczenie oprzyrządowania komputerowego wraz z oprogramowaniem dla elektronicznych systemów ćwiczebnych strategicznego lotnictwa bombowego. Za pomocą tych systemów załogi samolotów bombowych wykonują symulacje lotów bojowych, w trakcie których doskonalą się w wyszukiwaniu celów strategicznych i zwalczaniu obrony przeciwlotniczej przeciwnika.

Kasetowy system operacyjny 2TO6 TURBO

Wojciech ZABOŁOTNY

Program K.S.O. 2TO6 powstał jako system usprawniający programowanie i posługiwanie się programami narzędziowymi (SPEEDSCRIPT, SOUNDMACHINE i inne), dzięki zwiększeniu szybkości i wygody obsługi magnetofonu. Program umożliwia korzystanie z takich języków, jak: ASSEMBLER-EDYTOR, MAC-65, BASIC-XL, MICROSOFT-BASIC i innych, eliminując długie oczekiwanie na wgranie się używanego języka i uruchamianego programu. W praktyce jest możliwe wykorzystanie programu także do przyspieszonego wczytywania gier. Uzyskano szybkość transmisji około 3000 bit/sek. W obrębie rekordu, co daje efektywną prędkość transmisji około 2000 bit/sek (oznacza to, że na przykład gra „TOMAHAWK" ładuje się 4 minuty zamiast 20). Przy zastosowaniu dość dobrego magnetofonu (M 8010) i taśm z nośnikiem AGFA lub BASF program działa pewnie. Jeśli wystąpią kłopoty, należy skontrolować równomierność przesuwu taśmy i czystość głowicy. Przy używaniu magnetofonu innego typu niż M 8010 mogą się okazać konieczne następujące zmiany:

— rezystor RS należy dobrać tak, aby uzyskać właściwą wartość sygnału na wejściu magnetofonu przy zapisie,

— można pobierać sygnał z gniazda słuchawkowego magnetofonu, jeśli sygnał na wyjściu DIN jest zbyt słaby,

— można pobierać sygnał z gniazda słuchawkowego magnetofonu, jeśli sygnał na wyjściu DIN jest zbyt słaby.

Aby było możliwe przenoszenie programów między magnetofonami, konieczna jest także staranna regulacja skosu głowicy. Dla pełnego wykorzystania funkcji programu konieczne jest sterowanie przez komputer pracą silnika magnetofonu.

W dalszym ciągu artykułu będzie opisany interfejs dopasowujący sygnały komputera i magnetofonu oraz sterujący silnikiem oraz opis koniecznej przeróbki magnetofonu.

Niestety, nie jest możliwe przerobienie w prosty sposób magnetofonów „firmowych". Większość z nich zawiera kondensator (niezbędny do normalnej pracy) dołączony równolegle do głowicy uniwersalnej, co poprawia zapis i odczyt częstotliwości używanych w zapisie „firmowym", ale uniemożliwia zapis i odczyt impulsów używanych przez system TURBO. Oznacza to, że ci, którzy korzystają z magnetofonu firmowego, będą musieli dołączyć do zestawu zwykły magnetofon, ale jeśli ktoś używa zwykłego magnetofonu z dołączonym interfejsem (a jest to dość powszechne), może dołączyć do niego także interfejs **TURBO**.

Jak otrzymać program TURBO

Do artykułu dołączone są dwa listingi. Są to programy napisane w języku ATARI-BASIC. Aby wygenerować program K.S.O. 2TO6 należy wprowadzić do komputera PROGRAM1. Po uruchomieniu zapisze on na kasecie K.S.O. 2TO6 jako plik 3OOT (przed uruchomieniem dobrze by było zapisać ten program na kasecie instrukcją CSAVE).

Program 2 umożliwia otrzymanie programu kopiującego, pozwalającego przenosić pliki dowolnej długości między dyskietkami a taśmą w systemie TURBO.

Po uruchomieniu na dyskietce zostanie zapisany program kopiujący pod nazwą D: KOPIARKA, TUR.

Wgrywanie programu

1. Jeżeli chcesz pracować w języku ATARI-BASIC, to włącz komputer trzymając wciśnięty klawisz START. Jeżeli BASIC ma być wyłączony — trzymając OPTION I START.

2. W magnetofonie firmowym umieść kasetę z K.S.O. 2T06 i wciśnij klawisz PLAY (START w niektórych magnetofonach).

3. Naciśnij RETURN i program powinien się wczytać do komputera. Jeżeli był wyłączony, to na ekranie pojawi się napis READY, a o tym, że program wgrał się poprawnie, możesz się przekonać wprowadzając instrukcję DOS. Powrót do BASIC-a umożliwia opcja C menu K.S.O. (UWAGA! Jeżeli po wgraniu systemu nie był naciskany klawisz RESET, to przejście do K.S.O. i powrót do BASIC-a spowoduje skasowanie programu w BASIC-u).

UWAGA! Niektóre programy mogą działać błędnie, jeśli po wgraniu K.S.O. był używany klawisz RESET (dotyczy to zwłaszcza programów kopiowych z cartridge'ów, przykładem jest THE LAST STARFIGHTER).

Menu K.S.O.

K.S.O. 2TO6 umożliwia współpracę nie tylko z magnetofonem w trybie TURBO, ale także z magnetofonem firmowym. Dlatego w menu K.S.O. 2TO6 znajdują się opcje związane ze zwykłym trybem transmisji.

D — długie przerwy,

K — krótkie przerwy.

Powyższe funkcje służą do ustalenia rodzaju przerw międzyrekordowych w plikach ładowanych funkcjami R lub L z magnetofonu firmowego.

E — włączenie silnika,

H — zatrzymanie silnika.

Te funkcje dotyczą także magnetofonu firmowego. Ich zadaniem jest ułatwienie odnalezienia początku pliku na taśmie, co przy klasycznym trybie zapisu jest sprawą niezwykle ważną. Ponieważ tryb TURBO jest odporny na ustawienie taśmy daleko przed początkiem zbioru, a ponadto interfejs jest wyposażony w wyłącznik umożliwiający włączanie silnika niezależnie od sterowania przez komputer, nie wprowadziłem podobnych funkcji dla magnetofonu TURBO.

Następne funkcje dotyczą obu rodzajów magnetofonów:

R — załadowanie i uruchomienie programu w kodzie maszynowym,

L — załadowanie programu w kodzie maszynowym.

Wyczerpujące omówienie działania tych funkcji wymaga przedstawienia struktury plików wczytywanych przez system DOS.

Na początku takiego pliku znajdują się dwa bajty FF określające rodzaj pliku. Po nich następują segmenty pliku. Każdy segment na początku zawiera cztery bajty o specjalnym znaczeniu. Dwa pierwsze określają adres komórki, w której znajdzie się pierwszy bajt segmentu. Następne dwa określają adres, pod który zostanie wpisany ostatni bajt segmentu. Ilość bajtów segmentu, które są zapisane po tych czterech bajtach, jest oczywiście równa ADRES KONCA—ADRES POCZĄTKU. Niektóre programy generujące pliki w formacie DOS oddzielają segmenty dwoma bajtami $FF (np. KYAN PASCAL).

Specjalne znaczenie mają komórki 736,737,738 i 739. Jeśli wczytany segment zapisze niezerową wartość do komórek 738, 739, to K.S.O. uruchamia procedurę zaczynającą się od wpisanego tam adresu. Jeżeli owa procedura zakończy się instrukcją RTA, to będzie kontynuowane ładowanie pliku. Zawartość komórek 736, 737 jest badana po zakończeniu wczytywania pliku. Jeśli jest tam wpisana wartość niezerowa, to nastąpi uruchomienie programu od adresu podanego w tych dwóch komórkach. Jeśli nie, to gdy program był ładowany opcją L, nastąpi powrót do MENU K.S.O., jeśli zaś była użyta funkcja R., nastąpi skok na początek ostatniego wczytanego segmentu pliku. Przeciętnemu użytkownikowi wystarczy informacja, że funkcja R może służyć do ładowania większości programów wczytywanych normalnie przez loader „z wykrzyknikiem" lub BLC. Programy takie można próbować po przekopiowaniu ich (np. programem FCOPY) wczytywać w systemie TURBO.

C — powrót do cartridge'a

Ta funkcja umożliwia powrót do BASIC-a (lub programu umieszczonego na cartridge'u) po wywołaniu K.S.O. instrukcją DOS. Funkcja ta działa oczywiście tylko wtedy, gdy włączony jest BASIC lub włożony cartridge z innym programem.

— *zimny start systemu*

Ta funkcja nie wymaga komentarza.

Nazwy plików

Nazwa pliku może zawierać do 10 znaków. Użycie nazwy w instrukcji np. SAVE pokazane jest na poniższym przykładzie.

SAVE „D: s0123456789"

Znaki „D:" stanowią identyfikator urządzenia. Użyto identyfikatora takiego samego, jak dla stacji dysków ze względu na zgodność z takimi programami narzędziowymi jak SOUNDMACHINE czy MAGIC PAINTER. Znak „s" jest znakiem sterującym, ważnym tylko w trybie zapisu. Informuje on system, czy ma nastąpić rzeczywiście zapis pliku, czy jego weryfikacja. Umieszczenie na jego miejscu znaku „V" powoduje nie zapisywanie nowego pliku, lecz weryfikację tzn. porównanie tego, co ma być zapisane na taśmie z tym, co się na niej znajduje. W poniższych przykładach znakiem sterującym jest V (dla weryfikacji) albo spacja (dla zapisu). Niektóre programy (np. program graficzny KOALA) nie tolerują jednak spacji w nazwie pliku. Można wówczas dla zapisu użyć znaku sterującego będącego dowolną literą różną od „V".

PRZYKŁAD

Używamy programu SPEEDSCRIPT, chcemy zapisać znajdujący się w pamięci tekst na taśmę pod nazwą „TEKST-12". W tym celu naciskamy CONTROL+S, podajemy nazwę pliku „D: TEKST-12", naciskamy RETURN, włączamy ZAPIS i START w magnetofonie i naciskamy RETURN jeszcze raz. Po zapisaniu programu należałoby sprawdzić poprawność zapisu. Dyrektywa CONTROL+L wprowadza tekst z urządzenia zewnętrznego za kursor, usuwając tekst znajdujący się tam wcześniej. Gdybyśmy chcieli jej użyć do wczytania właśnie zapisanego tekstu, to przy ustawieniu kursora na końcu zapisywanego tekstu mogłoby zabraknąć wolnej pamięci. Natomiast po ustawieniu kursora na początku ewentualny błąd spowodowałby utratę całego tekstu. Na szczęście K.S.O. 2TO6 oferuje TRYB WERYFIKACJI.

Cofamy taśmę do początku pliku, naciskamy CONTROL+S podajemy, nazwę pliku „D: „TEKST-12", naciskamy RETURN, włączamy START w magnetofonie i naciskamy RETURN jeszcze raz. Ewentualne przekłamanie zostanie zgłoszone jako błąd 145, 143, 140 lub 136.

Posługiwanie się trybem weryfikacji w języku ATARI-BASIC (i niektórych innych) nastręcza pewne trudności: mianowicie instrukcja SAVE zapisuje także linię wprowadzoną w trybie bezpośrednim, a więc zapisana zostaje także podana nazwa pliku. Można sprawdzić, że zapisanie programu instrukcją SAVE „D: PROGRAM1" i zweryfikowanie go instrukcją SAVE „D: VPROGRAM1" zakończy się błędem 145. Przyczyną tego jest właśnie różnica między nazwami pliku użytymi w pierwszej i drugiej instrukcji. Nie oznacza to jednak, że instrukcja SAVE nie może współpracować z trybem weryfikacji. Podam jeden ze sposobów uniknięcia tego kłopotu: Należy wprowadzić obie instrukcje SAVE w jednej linii, a więc piszemy SAVE „D: PROGRAM1" SAVE „D: VPROGRAM1" i naciskamy RETURN, włączamy ZAPIS i START w magnetofonie i naciskamy ponownie RETURN. Program zostanie zapisany. Następnie cofamy taśmę do początku zapisanego właśnie zbioru i naciskamy RETURN. Teraz weryfikacja nastąpi w sposób właściwy.

Te kłopoty występują jednak tylko przy użyciu instrukcji SAVE. Znaki „0123456789" w pierwszym przykładzie oznaczają nazwę pliku i są zapisywane w jego nagłówku na taśmie. Nazwy krótsze są uzupełniane spacjami do 10 znaków, nazwy dłuższe są „obcinane".

Oto odpowiednie przykłady:
SAVE: „D: TEXT" zapisze na taśmie plik o nazwie „TEXT".
SAVE: „D: PROGRAM BARDZO CIEKAWY" zapisze plik o nazwie „PROGRAM BA".

Nazwa może zawierać dowolne znaki poza trzema zastrzeżonymi: *, /, @. Te znaki spełniają ważne funkcje przy odnajdywaniu na taśmie zbiorów przy odczycie lub weryfikacji.

ZNAK / oznacza, że dany znak nazwy nie będzie porównywany przy identyfikacji pliku.
Na przykład:
Instrukcja LOAD „D: PROGRAM15" może spowodować wczytanie zarówno programu „PROGRAM15", jak i programu „PROGRAMY5" (w zależności od tego, który będzie znaleziony wcześniej).

Znak @ oznacza, że identyfikację nazwy należy zakończyć na tym znaku.
Na przykład:
Instrukcja LOAD: „D: PR@" spowoduje wczytanie pierwszego programu o nazwie zaczynającej się od znaków „PR" (na przykład „PROGRAMY" „PRZYKŁADY" itp.).
Znak * powoduje, że jeżeli znaki występujące przed nim były zgodne, to zostaje wprowadzona pełna nazwa pliku z pytaniem czy jest on właściwy.
Przykład:

Instrukcja LOAD: „D: KW*" po napotkaniu na taśmie pliku o nazwie „KWIECIEŃ" spowoduje wyprowadzenie komunikatu „KWIECIEŃ (T/N)" i czekanie na decyzję użytkownika. Naciśnięcie klawisza „T" oznacza akceptację pliku i przejście do jego czytania (weryfikacji), naciśnięcie innego klawisza powoduje szukanie następnego pliku.

Ważne dla „Hackerow"

Kasetowy system operacyjny w swej aktualnej wersji umożliwia także niestandardowe wykorzystanie trybu TURBO — transmisja przebiega wtedy bez podziału na bloki, czyli znacznie szybciej. Poza tym program tak zapisany jest trudniejszy do skopiowania.

Oto informacje niezbędne do wykorzystania tych możliwości: Pod adresem 1792($700) znajduje się instrukcja skoku do procedury transmisji. Procedura ta wymaga jako danych wejściowych adresu początku bloku danych, adresu końca bloku danych, i kodu operacji. Informacje te wprowadzamy w następujący sposób:

— adres początku umieszczamy w komórkach $50, $51,

— w komórkach $52, $53 umieszczamy adres pierwszego bajtu po przesłanym bloku (bajt ten nie będzie już przesłany),

— kod operacji umieszczamy w akumulatorze i w komórce pamięci $704 w następujący sposób: niezerowa zawartość komórki 704 wymusza tryb weryfikacji bez względu na zawartość akumulatora, zero w akumulatorze oznacza zapis, a wartość różna od zera — odczyt.

Procedura transmisji zapisuje lub odczytuje wskazany blok danych uzupełniony sumą kontrolną.

Informację o wyniku operacji zawiera rejestr Y, podobnie jak przy innych operacjach I/O.

Struktura zapisu na taśmie

Do zapisu na taśmie wykorzystuje się system uproszczonej modulacji częstotliwości (podobnie jak w ZX SPECTRUM).

Występują trzy rodzaje impulsów generowanych na taśmie:

1 — impuls synchronizacji (stan wysoki przez 800 cykli zegara i stan niski przez 800 cykli zegara),

2 — impuls „1" (stan wysoki przez 400 cykli zegara, i stan niski przez 400 cykli zegara),

3 — impuls „0" (stan wysoki przez 200 cykli zegara i stan niski przez 200 cykli zegara).

Blok tworzony na taśmie przez procedurę transmisji składa się z 4096 impulsów synchronizacji, po których następują impulsy kodujące bajty danych (od najstarszego do najmłodszego bitu), a po nich bajt sumy kontrolnej utworzonej jako suma modulo 256 (nie EXOR!) wszystkich nadanych bajtów.

Przy korzystaniu z urządzenia D: na taśmie mogą być zapisane następujące bloki:

1 — blok nagłówka (zawsze na początku pliku),

2 — pełny blok danych (ilość tych bloków zależy od długości pliku),

3 — blok końca pliku (zawsze na końcu pliku).

BLOK NAGŁÓWKA ma długość 12 bajtów. Na początku zawiera bajty 0,255 stanowiące identyfikator nagłówka, a następnie 10 bajtów będących kodami ATASCII znaków nazwy pliku.

PEŁNY BLOK DANYCH ma długość 3074 bajtów. Na początku znajdują się bajty 0,12 (0+12*256=3072) informujące o ilości bajtów danych, a potem 3072 bajty danych.

BLOK KOŃCA PLIKU ma długość 3074 bajtów. Na początku znajdują się dwa bajty L,H informujące o ilości bajtów danych (L+255*H= ilość bajtów danych). Potem postępują bajty danych, a za nimi bajty uzupełniające blok do długości 3074 bajtów.

Oczywiście każdy z tych bloków poprzedzony jest impulsami synchronizującymi i zakończony bajtem sumy kontrolnej.

Przeróbki „sprzętowe"

Podstawowym urządzeniem koniecznym do korzystania z K.S.O. 2TO6 jest interfejs zmieniający analogowy sygnał z magnetofonu w cyfrowy sygnał doprowadzony do komputera.

Schemat tego układu i sposób jego realizacji opiszemy w kolejnym numerze IKS-a.

Dostępne oprogramowanie

Niektóre programy mogą pracować w systemie TURBO bezpośrednio po ich przekopiowaniu ze zwykłego magnetofonu. Niektóre wymagają dość daleko idących przeróbek.

Możliwe jest przerobienie na standard TURBO nawet tak długiego programu jak TOMAHAWK.

Wśród programów użytkowych istnieją specjalnie dostosowane do pracy w systemie TURBO; ASSEMBLER-EDYTOR i MAC85 (uzupełnione dodatkowo ramdyskami) VISICALC. Wiele języków (wśród nich znakomity TURBO-BASIC XL) nie wymaga żadnych przeróbek dla współpracy z systemem TURBO.

Istnieje ponadto program umożliwiający przenoszenie plików dowolnej długości między dyskami i taśmą w systemie TURBO oraz program umożliwiający kopiowanie plików o długości do 55 KB w systemie TURBO.

Oto niektóre (jest ich znacznie więcej) programy działające z TURBO K.S.O. 2TO6 **BEZ ŻADNYCH PRZERÓBEK**.

GRY: Alley Cat, Spindizzy, Combat Leader, Submarine Commander, Colossus Chess, Computer War, Pole Position, Moon Patrol, Behind Jaggi Lines, Boulder Dash, Montezumas's Revange. JĘZYKI: Pascal, Turbo Basic XL, ACTION!, Microsoft Basic, Assembler OSS.

Jak używać Kyan Pascal w systemie Turbo

System TURBO umożliwia korzystanie z KYAN PASCALA bez żadnych przeróbek (poza oczywiście skopiowaniem z dysku na taśmę w systemie TURBO). Wykorzystuje się tu możliwość pracy z dwoma magnetofonami. Można umieścić tekst źródłowy na magnetofonie normalnym i kompilować na magnetofon TURBO zleceniem PC/C-OD: Xnazwa. Nie każdy magnetofon firmowy zdąży zatrzymać się i ruszyć w czasie trwania przerwy między rekordami, mogą więc wystąpić kłopoty z wczytywaniem tekstu źródłowego.

Po wprowadzeniu zlecenia kompilacji kompilator zażąda dostępu do magnetofonu firmowego (ton pojedynczy) i po wciśnięciu klawisza RETURN wczyta tekst źródłowy. Jeśli nie zostaną wykryte błędy składniowe, to kompilator zażąda dostępu do magnetofonu TURBO (ton poczwórny) i po wciśnięciu klawisza RETURN zapisze na nim nagłówek pliku z kodem wynikowym, a następnie pojedynczym tekstem zasygnalizuje konieczność powtórnego wczytania pliku z tekstem źródłowym. Należy wówczas cofnąć taśmę z tekstem do początku zbioru. Po naciśnięciu RETURN nastąpi powtórne wczytanie tekstu źródłowego z równoczesnym generowaniem kodu wynikowego.

Aby uruchomić program należy wgrać go dyrektywą >D1:nazwa lub >D1:*. Po wczytaniu kodu komputer zażąda wprowadzenia zbioru b: biblioteki, który musi być skopiowany na taśmę pod nazwą D: XLIB. Po jegc załadowaniu program powinien się uruchomić.

Przy pracy z PASCALEM konieczne jest dokładne wprowadzanie nazw plików. Plik AUTORUN. SYS może być skopiowany pod dowolną nazwą, ale pliki edytora, kompilatora i biblioteki muszą być nazwane D:XED D:XPC i D:XLIB (X dowolna litera różna od V). Edytor i kompilator po wgraniu pliku AUTORUN muszą być ładowane dyrektywami >D1:ED i >D1:PC bez żadnych znaków specjalnych, także postać zlecenia kompilacji i wczytywania kodu należy zachować bez zmian (zamiast słowa „nazwa" należy wstawić litery tworzące nazwę programu).

Program do kopiowania z dyskietek

Przedstawiony program po załadowaniu ze stacji dysków (przez odpowiednią instrukcję DOS-a) umożliwia przenoszenie zbiorów dowolnej długości między taśmą w systemie TURBO i dyskiem (w obie strony), możliwa jest też weryfikacja, to znaczy porównanie zbioru na taśmie ze zbiorem na dysku. Przy operowaniu zbiorami na taśmie obowiązują te same reguły co przy K.S.O., z jedną różnicą: identyfikatorem magnetofonu TURBO jest T: a nie D:.

Oto przykłady użycia programu:

Aby skopiować zbiór LIB PASCALA trzeba podać:

Plik wejściowy: D:LIB
Plik wyjściowy: T:XLIB

Aby następnie sprawdzić poprawność zapisu:

Plik wejściowy: D:LIB
Plik wyjściowy: T:VLIB

lub

Plik wyjściowy: T:V

itp.

Wprowadzenie zamiast nazwy pliku wyjściowego lub wejściowego *1 spowoduje podanie zawartości dysku w stacji pierwszej (*2 — w drugiej itd.).

Aby skopiować plik TOMAHAWK z taśmy na dysk pod nazwę TOM.GRA należy podać:

Plik wejściowy: T:XTOMAHAWK
Plik wyjściowy: D:TOM.GRA

Uwagi

Procedura TURBO używa procesora do pomiaru czasu impulsów, w związku z tym konieczne było zablokowanie przerwań, czego konsekwencją jest zablokowanie klawisza BREAK. Przerwanie transmisji osiąga się przez równoczesne wciśnięcie klawiszy OPTION, SELECT i START.

TURBO nie toleruje starszych wersji systemu operacyjnego, dlatego też nie będzie pracować poprawnie z TRANSLATOREM, czy z innymi programami symulującymi stare wersje ROM-u. Oczywiście nie może pracować także na starych modelach ATARI.

Ewentualnych użytkowników systemu TURBO chciałbym ostrzec przed mogącymi się pojawić na giełdach wcześniejszymi wersjami (oznaczonymi iT...). Ze względu na inny sposób generowania sumy kontrolnej i niepełną diagnostykę błędów występuje w nich niebezpieczeństwo niewykrycia błędu transmisji (co na przykład przy kopiowaniu będzie prowadzić do wytworzenia błędnej kopii). Przy częstym korzystaniu z systemu jego wgrywanie może okazać się uciążliwe. Została w związku z tym opracowana także wersja umieszczona na cartridge'u, zgodna z wersją wgrywaną z kasety.

```
10 REM PROGRAM 1
20 REM Generuje K.S.O. 2T06 jako
plik BOOT na kasecie.
30 REM Nie mozna zmieniac numerow
 linii DATA !!!
40 GRAPHICS 0:? "▶▶PROSZE CZEKAC"
50 READ LB
60 DIM A$(LB),B$(22)
70 LICZN=0:CHKSUM=0
80 FOR I=1 TO LB
90 IF LICZN=0 THEN READ B$
100 GOSUB 270:A$(I)=CHR$(BYTE)
110 CHKSUM=CHKSUM+BYTE:CHKSUM=CHK
SUM-256*(CHKSUM>255)
120 LICZN=LICZN+1
130 IF LICZN<>10 AND I<>LB THEN 1
70
140 GOSUB 270
150 IF CHKSUM<>BYTE THEN A=10*INT
((I-1)/10)+320:? "BLAD DANYCH W L
INII ";A:LIST A:END
160 LICZN=0
170 NEXT I
180 POKE 764,255:GRAPHICS 0
190 ? " Prosze wlozyc do magnetof
onu kasete,"
200 ? " na ktorej ma byc zapisany
 program"
210 ? " w formie pliku BOOT, wcis
nac PLAY"
220 ? " i RECORD w magnetofonie i
 nacisnac"
230 ? " RETURN."
240 OPEN #1,8,128,"C:"
250 ? #1;A$;:CLOSE #1
260 END
270 B=ASC(B$(LICZN*2+1)):GOSUB 30
0:BYTE=Y
280 B=ASC(B$(LICZN*2+2)):GOSUB 30
0:BYTE=Y+16*BYTE
290 RETURN
300 Y=(B-48)*(B<58)+(B-87)*(B>96)
:RETURN
310 DATA 2056
320 DATA 0011f306c208a93c8d0248
330 DATA d318604c8d08000d2135d7
340 DATA 499199361bcad0fd60be50
350 DATA 0407200c07a9008d00d397
360 DATA a9008d1ad0ad1fd0f01154
370 DATA be0507200c07a9df8d0066
380 DATA d3a9088d1ad060a0804c2d
390 DATA 6208a20886300a48a000e9
400 DATA 9002a00220100768c630b2
410 DATA f008a20c200c074c3f071d
420 DATA 60a008ad1fd0f0d7ad0035
430 DATA d330f6a9008d1ad0aae8e0
440 DATA 301ead00d310f8a9088df4
450 DATA 1ad0ec0a07b00b6649ec31
460 DATA 0b07263088d0d460244992

470 DATA 30cfa08c4c6208209b0836
480 DATA a900a01085309848a004c8
490 DATA 20100768a8c630d0f38850
500 DATA d0f0a432a9008532a202ea
510 DATA 200c079848b132481865a5
520 DATA 31853168203b0768a8c82e
530 DATA f005c4004cce07e633c4e5
540 DATA 34d009a535c533d0064ce6
550 DATA e107eaeaea4cb407a53169
560 DATA a201200c07203b07a00142
570 DATA 4c62084c3607209b08a0e4
580 DATA 00a9ff8549a200ad1fd098
590 DATA f0eda d00d330f6a9008d51
600 DATA 1ad0e830ecad00d310f8c7
610 DATA a9088d1ad0ec0a0790d450
620 DATA c8d0daa432a90085329890
630 DATA 4820580768a8a530ae03ed
```

```
640 DATA 07d026913218653185311
650 DATA c8d002e633c434d0e2a513
660 DATA 33c535d0dc205807a0010c
670 DATA a530c531f002a08f4c62a6
680 DATA 08d132f0d8a091aec10821
690 DATA 9aa9408d0ed4a9228d006b
700 DATA d4a9388d02d3a9008d00b8
710 DATA d3a93c8d02d35860863444
720 DATA 8435a2fa8632a20e8633ba
730 DATA ae0307d006aad0034c90a1
740 DATA 074cf307a9008d0ed4858b
750 DATA 318d00d4a9088d1fd0a9f3
760 DATA 388d02d3a9608d00d3a99f
770 DATA 3c8d02d378bae8e88ec18e
780 DATA 086000a90c850ba9bf8528
790 DATA 0aa91a850f8de802a9fca5
800 DATA 850e8de702a9808d280a96
810 DATA a244a90aa01b4c86e4a949
820 DATA 31a00e20250ca210a90ce0
830 DATA 9d420320c8094cb00c18d3
840 DATA 900138664820 7e0ca20399
850 DATA a9009de002ca10faa21047
860 DATA a9039d4203a9209d440382
870 DATA a90b9d4503a9049d4a03b2
880 DATA ad280a9d4b0320c809208d
890 DATA 980990b5209809b0fba988
900 DATA 079d4203a90a9d4503a9b2
910 DATA 199d4403a9029d4803a9eb
920 DATA 009d490320c809386e1681
930 DATA 0aad170a9d4403ad180a0c
940 DATA 9d450338ad190aed170a07
950 DATA 9d4803ad1a0aed180a9d6c
960 DATA 4903fe4803d003fe49031e
970 DATA 20c809ade202d005ade305
980 DATA 02f00b20130aa9008de257
990 DATA 028de3024c3109186e16ed
1000 DATA 0aa210a9079d4203a90aee
1010 DATA 9d4503a9179d4403a90222
1020 DATA 9d4803a9009d490320c884
1030 DATA 09ad170a2d180ac9fff062
1040 DATA 02186038602056e4983096
1050 DATA 016048a90c9d420320564c
1060 DATA e4682c160a3024c988d059

1070 DATA 20ade002d00cade102d044
1080 DATA 07a548300f4cbf0cade01b
1090 DATA 028d170aade1028d180a0a
1100 DATA 6c170a48a948a00e2025c3
1110 DATA 0ca00068204c0c4cb00c57
1120 DATA 6ce20200bf0c0000500acc
1130 DATA 390bc10b6e0b100c120c8f
1140 DATA 000086438444a001a9006a
1150 DATA 9143a9ffc89143c8b1241f
1160 DATA c99bf0089143c8c00dd0b4
1170 DATA f360a9209143c8c00dd009
1180 DATA f960a9008d03078545851f1
1190 DATA 46a52a290cc904f035c9f6
1200 DATA 08f00160a002b124c956e5
1210 DATA f025a90420fcfd301ba2ad
1220 DATA f9a00e20290aa900a206f8
1230 DATA a00f20810820140c983058
1240 DATA 05a9808545604c840b38c3
1250 DATA 6646186647a90320fcfdf9
1260 DATA 30f0a901a206a00f2081bb
1270 DATA 08c080f0e39830f0a22d5d
1280 DATA a00b20290aa200bd2e0bf3
1290 DATA c92ff00dc92af016c940ea
1300 DATA f00addfa0ed0d3e8e00c40
1310 DATA d0e720140ca00184456001
1320 DATA a2028a48bdfa0e20b0f2fe
1330 DATA 68aae8e00cd0f1a2008ad1
1340 DATA 48bd1a0b20b0f268aae8b7
1350 DATA e006d0f1a9ff8dfc0220b1
1360 DATA fdf2295f09404820b0f27b
1370 DATA a99b20b0f268c954f0bcb2
1380 DATA 4ca10a2028542f4e29442f
1390 DATA a908a0c42086e4a24ca965
1400 DATA 08a0d02086e4a24ea90b0b
1410 DATA a0552086e4a446100a20ae
1420 DATA c20b98103ec088d026a03f
1430 DATA 01a5451020a900a891437f
1440 DATA e643d002e644a91ac54470
1450 DATA d0efa9fcc543d0e9a9003e
1460 DATA a2fca01a20810860a44689
1470 DATA 10188d270a20c20bc0001c
1480 DATA 300dcd270af008a091a929
1490 DATA 008545854660a44530033a
1500 DATA a08760a0009143e643d02e
1510 DATA 02e644eefa0ed003eefb0c
1520 DATA 0eadfb0ec90cf003a00139
1530 DATA 60a900a2fca01a20810843
1540 DATA 981004a900854520140ca2
1550 DATA 60a44588f003a08360ad96
1560 DATA fa0e0dfb0ed022a54710a2
1570 DATA 03a08860a901a2fca01a2f
1580 DATA 208108201c0c98309ca92d
1590 DATA 0ccdfb0ef005386647d0b9
1600 DATA d6a000b14348e643d00266
1610 DATA e64438adfa0ee9018dfaee
1620 DATA 0eadfb0ee9008dfb0e6899
1630 DATA a00160a9008dfa0e8dfb60
1640 DATA 0ea90e8544a9fc854360bb
1650 DATA a20084cc85cb1869029d1d
1660 DATA 4403986900 9d4503a000ea

1670 DATA b1cb9d4803c8b1cb9d4978
1680 DATA 03a90b9d42034c56e4851c
1690 DATA d484d520aad920e6d8a973
1700 DATA 0085cdf00520b0f2e6cd2f
1710 DATA a4cdb9800510f4498020cb
1720 DATA b0f24c770ca92020b0f2c7
1730 DATA e6cda906c5cdd0f360a987
1740 DATA 81a00e20250ca200a90557
1750 DATA 9d4203a90b9d4503a9209b
1760 DATA 9d4403a90e9d4803a900c7
1770 DATA 9d4903205 6e41004c08967
1780 DATA d016bc480388f01060a9e5
1790 DATA 5ea00e20250ca9ff8dfc73
1800 DATA 0220fdf2a2ff9aa90185ee
1810 DATA 09a900852ba90c852a20d4
1820 DATA 8eefa917a00d20250cadbc
1830 DATA 280a100aa928a00e2025cc
1840 DATA 0c4cee0ca920a00e2025da
1850 DATA 0ca951a00d20250ca9ff86
1860 DATA 8dfc0220fdf2295fa00850
1870 DATA d99f0ef0058810f830b43f
1880 DATA 980aa8b9a90e48b9a80eb0
1890 DATA 48603800cbe1f3e5f4eff7
1900 DATA f7f9a0f3f9f3f4e5eda0cc
1910 DATA eff0e5f2e1e3f9eaeef910
1920 DATA a0d7dae1e2a0b2d4b0b6b0
1930 DATA 9b9b2020202020205016
1940 DATA 727a657277793a20202063
1950 DATA cd009b9b7f7fa0cda0c536
1960 DATA a0cea0d5a09b9bcc2d4c34
1970 DATA 61646f77616e696520700c
1980 DATA 6c696b7520d22d4c6164f1
1990 DATA 6f77616e696520692073 90
2000 DATA 746172749bd32d5374610e
2010 DATA 72742070726f6772616d0c
2020 DATA 7520201b9d2d5a696d6e44
2030 DATA 792073746172749bcb2d9e
2040 DATA 4b726f746b696520707279
2050 DATA 7a6572777920c42d446c7b
2060 DATA 756769652070727a657278
2070 DATA 77799bc52d5572756368fc
2080 DATA 6f6d69656e696520202042
2090 DATA 20c82d5a6174727a796d58
2100 DATA 616e69659b202073696c18
2110 DATA 6e696b6120202020207b
2120 DATA 20202020207369 6c6e693a
2130 DATA 6b619bc32d506f77726fa8
2140 DATA 742064 6f20636172724724b
2150 DATA 696467652761 9b06004451
2160 DATA 4c5547494507004b524fba
2170 DATA 544b494515009b7ffd5a6d
2180 DATA 6c7920666f726d6174201b
2190 DATA 706c696b752021140 09b30
2200 DATA 7ffd426c6164207472618 5
2210 DATA 6e736d69736a693a2021fe
2220 DATA 009b20202020202 0a0cec7
2230 DATA e1e3e9f3eeee9eaa0e4ef3b
2240 DATA f7efeceef9a0ebece1f7a3
2250 DATA e9f3faa01c009b20202030
2260 DATA 20202020205 06f64616abe
2270 DATA 206e617a776520706c6968
2280 DATA 6b759b3f4c52534b4445e7
2290 DATA 481d43f908fc08fe09b954
2300 DATA 0ec10ec90ed10ef10edec4
2310 DATA 0ea9808d280a4cbf0ca97a
2320 DATA 008d280a4cbf0ca9348dba
2330 DATA 02d34cbf0ca93c8d02d3ed
2340 DATA a9038d0fd24cbf0cadf8c3
2350 DATA 03d0f8a9fd8d01d320efa4
2360 DATA 0e6cfabf6cfebfa9018d37
2370 DATA 44026cfcff00e4
```

```
10 REM PROGRAM 2
20 REM Zapisuje na dyskietce w pl
iku D:KOPIARKA.TUR
30 REM program przenoszacy pliki
miedzy systemem TURBO i dyskami
40 GRAPHICS 0:? "▶▶PROSZE CZEKAC"
50 READ LB
60 DIM A$(LB),B$(22)
70 LICZN=0:CHKSUM=0
80 FOR I=1 TO LB
90 IF LICZN=0 THEN READ B$
100 GOSUB 270:A$(I)=CHR$(BYTE)
110 CHKSUM=CHKSUM+BYTE:CHKSUM=CHK
SUM-256*(CHKSUM>255)
120 LICZN=LICZN+1
130 IF LICZN<>10 AND I<>LB THEN 1
70
140 GOSUB 270
150 IF CHKSUM<>BYTE THEN A=10*INT
((I-1)/10)+320:? "BLAD DANYCH W L
INII ";A:LIST A:END
160 LICZN=0
170 NEXT I
180 POKE 764,255:GRAPHICS 0
190 ? " Prosze wlozyc do stacji d
yskow dys- "
200 ? " kietke na ktorej ma byc z
apisany "
210 ? " program kopiujacy i nacis
nac"
220 ? " RETURN."
230 POKE 764,255:GET #16,A
240 OPEN #1,8,128,"D:KOPIARKA.TUR
"
250 ? #1;A$;:CLOSE #1
260 END
270 B=ASC(B$(LICZN*2+1)):GOSUB 30
0:BYTE=Y
280 B=ASC(B$(LICZN*2+2)):GOSUB 30
0:BYTE=Y+16*BYTE
290 RETURN
300 Y=(B-48)*(B<58)+(B-87)*(B>96)
:RETURN
310 DATA 1865
320 DATA ffff00209e224c8d2100d8
330 DATA 0d2135499199361bcad099
340 DATA fd60be0420200c20a900cd
350 DATA 8d00d3a9008d1ad0ad1f19
360 DATA d0f011be0520200c20a9c2
370 DATA df8d00d3a9088d1ad06089
380 DATA a0804c6221a20886300ae2
390 DATA 48a0009002a0022010204e
400 DATA 68c630f008a20c200c209e
410 DATA 4c3f2060a008ad1fd0f0dd
420 DATA d7ad00d330f6a9008d1aaa
430 DATA d0aae8301ead00d310f8e2
440 DATA a9088d1ad0ec0a20b00bdb
450 DATA 6688ec0b20263088d0d462
460 DATA 60248830cfa08c4c622168

470 DATA 30cfa08c4c6208209b0836
480 DATA a900a0108530 9848a004c8
490 DATA 20100768a8c630d0f38850
500 DATA d0f0a432a9008532a202ea
510 DATA 200c079848b132481865a5
520 DATA 31853168203b0768a8c82e
530 DATA f005c4004cce07e633c4e5
540 DATA 34d009a535c533d0064ce6
550 DATA e107eaeaea4cb407a53169
560 DATA a201200c07203b07a00142
570 DATA 4c62084c3607209b08a0e4
580 DATA 00a9ff8549a200ad1fd098
590 DATA f0eda d00d330f6a9008d51
600 DATA 1ad0e830ecad00d310f8c7
610 DATA a9088d1ad0ec0a0790d450
620 DATA c8d0daa432a90085329890
630 DATA 4820580768a8a530ae03ed
640 DATA 07d026913218653185311
650 DATA c8d002e633c434d0e2a513
660 DATA 33c535d0dc205807a0010c
670 DATA a530c531f002a08f4c62a6
680 DATA 08d132f0d8a091aec10821
690 DATA 9aa9408d0ed4a9228d006b
700 DATA d4a9388d02d3a9008d00b8
710 DATA d3a93c8d02d35860863444
720 DATA 8435a2fa8632a20e8633ba
730 DATA ae0307d006aad0034c90a1
740 DATA 074cf307a9008d0ed4858b
750 DATA 318d00d4a9088d1fd0a9f3
760 DATA 388d02d3a9608d00d3a99f
```

```
1710 DATA 207a617761727246f7363e6
1720 DATA 206479736b75206e9b50af
1730 DATA 6c696b2077656a73636994
1740 DATA 6f7779203a11009b506cb5
1750 DATA 696b2077796a7363696fb1
1760 DATA 7779203a18009b4b6f70d8
1770 DATA 696f77616e6965207a61bf
1780 DATA 6b6f6e637a6f6e65209be1
1790 DATA 06019b7d202020202020c0
1800 DATA 4b6f706961726b61204456
1810 DATA 59534b3c3e545552424f53
1820 DATA 544150459b9b2070726f24
1830 DATA 6772616d206f7072616300
1840 DATA 6f77616e792070727a650f
1850 DATA 7a9b2020202020202024
1860 DATA 2020202020202576f6a34
1870 DATA 6369656368612O5a6162ce
1880 DATA 6f6c6f746e65676f9b9b6b
1890 DATA 2020202020776572736a36
1900 DATA 6120322e362020504941 67
1910 DATA 53544f5720203139383 8ce
1920 DATA 9b9b555741474120212 1db
1930 DATA 2120416279207 56e696b0f
1940 DATA 6e6163206b6f6c697a6af4
1950 DATA 69207a65207374612d9b8c
1960 DATA 636a61206479736b6f777b
1970 DATA 206964656e747966696b62
1980 DATA 61746f72656d206d61673f
1990 DATA 6e65746f666f2d9b6e7575
2000 DATA 20545552424f206a657383
2010 DATA 7420543a2061206e696582
2020 DATA 20443a206a616b2077202d
2030 DATA 7379732d9b74656d696568
2040 DATA 204b2e532e4f2e205455c8
2050 DATA 52424f9b48a220a90b9da1
2060 DATA 4203a9009d48039d490360
2070 DATA 682056e460a210a9079d81
2080 DATA 4203a9009d48039d490340
2090 DATA 2056e460a001b1868d4dac
2100 DATA 27204025a903a2109d4295
2110 DATA 03a94c9d4403a9279d4523
2120 DATA 03a9069d4a03a9009d4b50
2130 DATA 032056e4301aa907a21059
2140 DATA 9d4203a9009d48039d49b2
2150 DATA 032056e4300620b0f24c53
2160 DATA 2827c088d0034cbe244c37
2170 DATA 2b25444d3a2a2e2a00e0b4
2180 DATA 02e102ae246b
```

```
1840 DATA 0c4cee0ca920a00e2025da
1850 DATA 0ca951a00d20250ca9ff86
1860 DATA 8dfc0220fdf2295fa00850
1870 DATA d99f0ef0058810f830b43f
1880 DATA 980aa8b9a90e48b9a80eb0
1890 DATA 48603800cbe1f3e5f4eff7
1900 DATA f7f9a0f3f9f3f4e5eda0cc
1910 DATA eff0e5f2e1e3f9eaeef910
1920 DATA a0d7dae1e2a0b2d4b0b6b0
1930 DATA 9b9b202020202020205016
1940 DATA 727a657277793a20202063
1950 DATA cd009b9b7f7fa0cda0c536
1960 DATA a0cea0d5a09b9bcc2d4c34
1970 DATA 61646f77616e696520700c
1980 DATA 6c696b7520d22d4c6164f1
1990 DATA 6f77616e69652069207390
2000 DATA 746172749bd32d5374610e
2010 DATA 72742070726f6772616d0c
2020 DATA 7520201b9d2d5a696d6e44
2030 DATA 792073746172749bcb2d9e
2040 DATA 4b726f746b696520707279
2050 DATA 7a6572777920c42d446c7b
2060 DATA 756769652070727a657278
2070 DATA 77799bc52d5572756368fc
2080 DATA 6f6d69656e696520202042
2090 DATA 20c82d5a6174727a796d58
2100 DATA 616e69659b202073696c18
2110 DATA 6e696b6120202020207b
2120 DATA 2020202020736 96c6e693a

2130 DATA 6b619bc32d506f77726fa8
2140 DATA 742064f20636172747 24b
2150 DATA 696467652761 9b06004451
2160 DATA 4c5547494507004b524fba
2170 DATA 544b494515009b7ffd5a6d
2180 DATA 6c7920665f726d6174201b
2190 DATA 706c696b752021140 9b30
2200 DATA 7ffd426c61642074726185
2210 DATA 6e736d69736a693a2021fe
2220 DATA 009b2020202020 0a0cec7
2230 DATA e1e3e9f3eee9eaa0e4ef3b
2240 DATA f7efeceef9a0ebece1f7a3
2250 DATA e9f3faa01c009b20202030
2260 DATA 2020202020506f64616abe

1670 DATA 009bfdfdfd2020202020a4
1680 DATA 20426c6164207472616e0c
1690 DATA 736d69736a693a202d0022
1700 DATA 9b2a6e20706f64616a65e8
```

```
770 DATA 3c8d02d378bae8e88ec18e
780 DATA 086000a90c850ba9bf8528
790 DATA 0aa91a850f8de802a9fca5
800 DATA 850e8de702a9808d280a96
810 DATA a244a90aa01b4c86e4a949
820 DATA 31a00e20250ca210a90ce0
830 DATA 9d420320c8094cb00c18d3
840 DATA 9001386648207e0ca20399
850 DATA a9009de002ca10faa21047
860 DATA a9039d4203a9209d440382
870 DATA a90b9d4503a9049d4a03b2
880 DATA ad280a9d4b0320c809208d
890 DATA 980990b5209809b0fba988
900 DATA 079d4203a90a9d4503a9b2
910 DATA 199d4403a9029d4803a9eb
920 DATA 009d490320c809386e1681
930 DATA 0aad170a9d4403ad180a0c
940 DATA 9d450338ad190aed170a07
950 DATA 9d4803ad1a0aed180a9d6c
960 DATA 4903fe4803d003fe49031e
970 DATA 20c809ade202d005ade305
980 DATA 02f00b20130aa9008de257
990 DATA 028de3024c3109186e16ed
1000 DATA 0aa210a9079d4203a90aee
1010 DATA 9d4503a9179d4403a90222
1020 DATA 9d4803a9009d490320c884
1030 DATA 09ad170a2d180ac9fff062
1040 DATA 02186038602056e4983096
1050 DATA 016048a90c9d420320564c
1060 DATA e4682c160a3024c988d059

1670 DATA b1cb9d4803c8b1cb9d4978
1680 DATA 03a90b9d42034c56e4851c
1690 DATA d484d520aad920e6d8a973
1700 DATA 0085cdf00520b0f2e6cd2f
1710 DATA a4cdb9800510f4498020cb
1720 DATA b0f24c770ca92020b0f2c7
1730 DATA e6cda906c5cdd0f360a987
1740 DATA 81a00e20250ca200a90557
1750 DATA 9d4203a90b9d4503a9209b
1760 DATA 9d4403a90e9d4803a900c7
1770 DATA 9d4903205 6e41004c08967
1780 DATA d016bc480388f01060a9e5
1790 DATA 5ea00e20250ca9ff8dfc73
1800 DATA 0220fdf2a2ff9aa90185ee
1810 DATA 09a900852ba90c852a20d4
1820 DATA 8eefa917a00d20250cadbc
1830 DATA 280a100aa928a00e2025cc
```

Losowy start generatora RND

Generator **RND** w ZX Spectrum posiada pewną niedogodność: po włączeniu komputera do sieci wartość wyjściowa generatora jest zerowana, czyli cały cykl generacji jest zawsze jedną i tą samą sekwencją:

```
  10 REM Krzysztof Pozniak ©1988
     losowy start generatora
     liczb pseudolosowych
  20 RANDOMIZE 1: REM ustawienie
                  stalego startu
                  generatora RND
  30 RANDOMIZE 256*PEEK 23672+PE
EK 23673: REM losowe ustawienie
          genetatora RND
  40 PRINT RND
  50 GO TO 20
```

Basic przewiduje możliwość ustawienia generatora za pomocą rozkazu **RANDOMIZE**, ale trzeba ją ustalić samodzielnie. Jeśli zależy nam na całkowicie losowym (i niezależnym od użytkownika) ciągu liczb losowych, to w standardzie **Basic'a** tego nie zrobimy.

Musimy skorzystać z komórek pamięci zliczających ramki (licznika ramek). Licznik zwiększa swoją wartość o 1 co 1/50 sekundy. Z dużą dokładnością możemy założyć, że ustalenie startu generatora **RND** na podstawie dwóch młodszych bajtów licznika zapewni całkowicie niepowtarzalny ciąg liczb.

```
pierwsze liczby generatora RND:

1 liczba  =  .0011291504
2 liczba  =  .08581543
3 liczba  =  0.43719482
4 liczba  =  0.79025269
5 liczba  =  0.2691803
6 liczba  =  0.18934631
7 liczba  =  0.20186904
8 liczba  =  0.14257813
9 liczba  =  0.69433594
10liczba  =  .075531006
11liczba  =  0.6658783
12liczba  =  0.94125366
13liczba  =  0.59408569
14liczba  =  0.55688477
15liczba  =  0.76686096
```

Krzysztof POŹNIAK

PROGRAM 64 ATARI

Loteryjka

W związku z rozpoczęciem publikowania na łamach „IKS-a" artykułów dotyczących języka „C", chciałbym przedstawić program napisany w Basic-u oraz jego implementację w „C". Początkujący zwolennicy języka „C" mogą w ten sposób obejrzeć i porównać efekty pracy obu języków. Aby nie rozbudowywać programu napisanego w „C", wykorzystano w nim funkcję czydalej (J) z drugiego numeru zeszytów programów „IKS-a". Zbiór wskazówek tego programu dla linkera składa się z następujących nazw:

```
LOTE c
AIO
PRINTF
GRAPHICS
DBC.OBJ
CZYDALEJ
```

Pierwszy wiersz, to oczywiście nazwa samego programu, którą można zmienić. Pozostałe związane są z biblioteką. Całość zostanie skomplikowana pod warunkiem uczynienia tego samego z funkcją czydalej (J).

Zadanie programu jest proste. Po wprowadzeniu ilości graczy, a następnie ich imion, komputer losuje pięć liczb w zakresie od 1 do 7 oraz oblicza ich sumę. Wygrywa ten, kto zdobędzie więcej punktów. Można do tej zabawy podejść od innej strony. Mechanizm jej przypomina trochę grę w kości. Będzie wygrywał ten, kto uzyska więcej par, trójek itd. o większej liczbie oczek.

J. J.

```
PG 500 REM ************************
YA 502 REM *                      *
VQ 504 REM *      LOTERYJKA       *
TE 506 REM *      * ATARI *       *
YS 508 REM *                      *
PI 510 REM ************************
JA 511 POKE 752,0
LL 512 PRINT CHR$(125):J=0:I=0
MS 514 DIM J$(255),I$(15):J$=""
TJ 515 DIM T$(1):OPEN #1,4,0,"K:"
AL 516 PRINT "ILU GRACZY ";:INPUT N
UD 517 PRINT
GY 518 FOR K=1 TO N:? "GRACZ ";K
VX 520 ? "IMIE ";:INPUT I$
TN 521 PRINT
BQ 522 J$(LEN(J$)+1)=I$:I=LEN(I$)+1
PP 523 FOR DL=I+(K*10) TO 9+(K*10)
ND 524 J$(LEN(J$)+1)=CHR$(32)
KU 525 NEXT DL
OR 526 I=LEN(I$)+1
IU 527 J$(LEN(J$)+1)=CHR$(44):J=J+I
HR 528 NEXT K
RY 536 FOR K=1 TO N:SOUND 0,99,10,5
DO 537 KS=(K-1)*10
MD 538 FOR KK=1+KS TO 9+KS
XU 539 PRINT J$(KK,KK);
MQ 540 NEXT KK
FN 542 SU=0
PL 543 FOR G=1 TO 5:SOUND 0,0,0,0
FF 544 W=1+INT(RND(0)*8):SU=SU+W
DV 546 PRINT CHR$(32);W;:NEXT G
CV 547 PRINT "   ";SU
JM 548 IF K=J THEN 550
HY 549 NEXT K
EE 550 ? :? "GRAMY DALEJ ?"
JW 551 POKE 752,1
KL 552 ? :? "          ";CHR$(212);"/
    ";CHR$(206);:GET #1,T
UE 554 IF T<>78 THEN ? :? :GOTO 536
OW 556 END
```

```
/*****************************/
/*                           */
/*         LOTERYJKA         */
/*           ATARI           */
/*                           */
/*****************************/
main()${
 int c, ilu, dl, i, j, k, g, su, t;
 char w, tab[100], imie[10];
  printf("\fIlu graczy (max 9)?");
  ilu=getchar();
  c=getchar();
for(dl=0;dl<=ilu-49;dl++)${
  printf("\nGracz %d ",dl+1);
  printf("\nImie  ");
  gets(imie);
  t=strlen(imie);
 for(i=0;i<t;i++)${
 j=i+dl*10;
 tab[j]=imie[i];
$}
 for(i=t;i<=9;i++)${
 j=i+dl*10;
 tab[j]=32;$}
$}
do${
  printf("\n");
  for(k=0;k<=ilu-49;k++)
$(sound(0,99,10,5);
 printf("\n");
 for(i=10*k;i<=10*k+9;i++)
 printf("%c",tab[i]);
 sound(0,0,0,0);
  su=0;
 for(g=1;g<=5;g++)${
  w=1+rnd(8);
  su=su+w;
 printf(" %d",w);
$}printf("  %d",su);
$}$}
 while(czydalej());
$}
```

Programy narzędziowe na CPC — AMSTRAD 6128

W niniejszym opracowaniu przedstawiamy niezbędne w pracy z komputerem CPC—AMSTRAD 6128 programy narzędziowe: DYSKMAPA, SEKTORED, KATALOG, DYSKCOPY i binarny program PROGBIN wykorzystywany przez każdy z wymienionych powyżej programów.

Program PROGBIN jest niezbędny przy operacjach wykonywanych na zbiorach. Warto wprowadzić i zachować na dyskietce jeszcze jeden program: JOB (wydruk 6), który ułatwi nam ściąganie prezentowanych programów do pamięci z jednoczesnym ich uruchomieniem.

Najpierw proponujemy wprowadzić i uruchomić komendą RUN program PROGBIN (wydruk 1). Jeśli wprowadzając program nie popełniłeś błędu pojawią się komunikaty:

OK — WSZYSTKIE DANE PRAWIDŁOWE

WŁÓŻ DYSKIETKĘ DO ZAPISU PROGRAMU BINARNEGO — WCIŚNIJ DOWOLNY KLAWISZ.

Jeśli postąpiłeś zgodnie z komunikatem to możesz uważać, że masz poza sobą wprowadzenie (i zachowanie na dyskietce) najbardziej żmudnego z wymienionych programów.

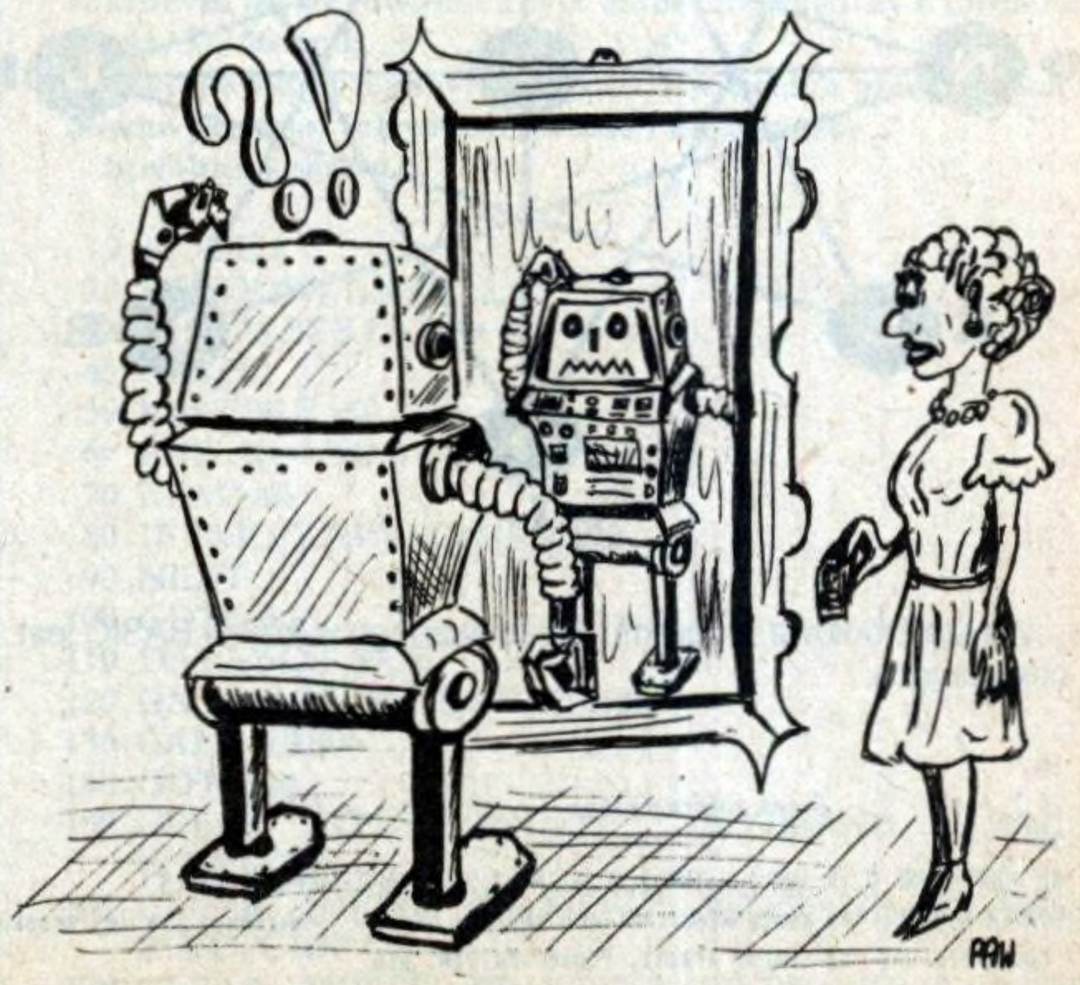
Programy narzędziowe DYSKMAPA, SEKTORED, KATALOG, DYSKCOPY.

Opublikujemy w kolejnych numerach „IKS-a"

```
10 RESTORE 200:A=&6000
20 FOR K=1 TO 256:REM 256 - liczba linii DATA zawierajacych po 20 bajtow kodu
30 SUMK=0
40 FOR I=1 TO 20
50 READ BAJT
60 SUMK=SUMK+BAJT
70 POKE A+I-1,BAJT
80 NEXT I
90 READ SUMA
100 IF SUMA<>SUMK THEN PRINT "BLEDNE DANE W linii ";K:STOP
110 A=A+20:NEXT K
120 CLS:LOCATE 1,1:PRINT "  OK  -  WSZYSTKIE dane PRAWIDLOWE"
130 LOCATE 1,12
140 PRINT "WLOZ DYSKIETKE DO ZAPISU PROGRAMU BINARNEGO - WCISNIJ DOWOLNY KLAWISZ"
150 IF INKEY$="" THEN 150
160 SAVE "progbin.bin",b,&6000,5120
200 DATA 195,31,16,195,159,16,195,93,16,195,177,16,0,193,198,194,199,195,200,196,2679
210 DATA 201,197,193,198,194,199,195,200,196,201,197,17,255,1,205,72,16,30,0,221,2988
220 DATA 86,0,223,90,16,33,13,16,229,223,81,16,225,48,11,58,81,190,119,35,1793
230 DATA 125,254,31,32,239,175,50,12,16,17,0,10,123,223,84,16,122,223,87,16,1855
240 DATA 201,108,197,7,114,202,7,3,198,7,99,199,7,221,86,0,221,94,2,221,2194
250 DATA 126,4,245,213,71,17,22,0,221,229,225,25,209,221,33,196,16,126,221,114,2534
260 DATA 0,221,119,2,221,35,221,35,221,35,221,35,43,43,16,237,241,254,9,40,2249
270 DATA 4,62,1,24,2,62,193,213,223,236,16,209,33,196,16,223,239,16,201,221,2390
280 DATA 86,0,221,78,2,221,110,4,221,102,5,30,0,223,242,16,201,221,86,0,2069
290 DATA 221,94,2,221,78,4,221,110,6,221,102,7,223,245,16,201,4,0,193,2,2171
300 DATA 4,0,198,2,4,0,194,2,4,0,199,2,4,0,195,2,4,0,200,2,1016
310 DATA 4,0,196,2,4,0,193,2,4,0,198,2,0,0,0,0,129,197,7,82,1020
320 DATA 198,7,102,198,7,78,198,7,0,0,0,0,0,0,0,0,229,229,229,229,1711
330 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
340 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
350 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
360 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
370 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
380 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
390 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
400 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
410 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
420 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
430 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
440 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
450 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
460 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
470 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
480 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
490 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
500 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
510 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
520 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
530 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
540 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
550 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
560 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
570 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
580 DATA 229,229,229,229,229,229,229,229,221,110,0,221,102,1,6,16,197,229,6,16,2957
590 DATA 126,205,61,19,62,32,205,90,187,35,16,244,6,3,205,90,187,16,251,225,2265
600 DATA 6,16,126,254,32,48,2,62,46,205,90,187,35,16,243,62,13,205,90,187,1925
610 DATA 62,10,205,90,187,193,16,204,201,245,15,15,15,15,205,70,19,241,230,15,2253
620 DATA 198,144,39,206,64,39,195,90,187,229,229,229,229,229,229,229,229,229,229,229,3681
630 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
640 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
650 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
660 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
670 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
680 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
690 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
700 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
710 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
720 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
730 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
740 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
750 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
760 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
770 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
780 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
790 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
800 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
810 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
820 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
830 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
840 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
850 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
860 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
870 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
880 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
890 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
900 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
910 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
920 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
930 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
940 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
950 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
960 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
970 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
980 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
990 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1000 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1010 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1020 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1030 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1040 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1050 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1060 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1070 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1080 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1090 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1100 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1110 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1120 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1130 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,231,4582
1140 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1150 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1160 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1170 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1180 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1190 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1200 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1210 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1220 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1230 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1240 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1250 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1260 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1270 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1280 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1290 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1300 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1310 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
```

```
1320 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1330 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1340 DATA 229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,229,4580
1350 DATA 229,229,229,229,237,140,243,59,32,206,196,53,115,173,29,21,27,156,251,36,2890
1360 DATA 232,149,163,36,118,179,89,36,225,224,194,61,188,100,58,18,209,46,252,100,2677
1370 DATA 115,118,164,66,17,71,8,215,17,236,159,230,36,71,212,4,30,180,204,96,2249
1380 DATA 246,11,109,178,22,97,203,198,16,115,85,73,47,204,159,157,148,216,103,21,2408
1390 DATA 126,68,194,95,149,22,226,191,21,82,176,176,178,118,138,139,132,247,99,199,2776
1400 DATA 208,72,175,149,167,229,196,149,64,43,158,247,116,88,46,142,133,83,134,32,2631
1410 DATA 193,102,59,188,24,252,209,52,47,172,211,230,74,195,100,82,3,35,214,124,2566
1420 DATA 177,157,161,148,121,218,6,145,210,85,82,216,11,188,22,39,97,113,58,76,2330

1430 DATA 107,180,41,79,164,88,230,116,252,159,248,113,68,99,93,26,17,243,216,65,2604
1440 DATA 235,145,86,64,91,105,254,85,162,137,188,177,68,73,169,128,110,233,139,165,2814
1450 DATA 112,243,140,40,249,217,80,7,85,167,190,75,89,175,170,210,162,40,183,153,2767
1460 DATA 173,98,3,48,82,173,122,35,47,102,182,68,210,238,29,53,242,14,204,82,2205
1470 DATA 62,156,167,169,31,137,140,37,214,215,8,68,119,135,152,78,182,130,200,198,2598
1480 DATA 74,174,166,37,243,200,240,28,147,105,109,46,200,99,168,143,216,13,234,198,2840
1490 DATA 111,116,38,67,126,188,210,235,198,236,122,210,233,172,29,41,108,115,252,173,2980

1500 DATA 100,180,214,34,75,25,79,126,45,64,21,59,161,102,118,249,90,224,13,79,2058
1510 DATA 57,220,192,160,65,80,1,150,0,204,212,19,206,46,107,34,120,174,52,196,2295
1520 DATA 18,170,32,94,59,15,19,61,13,139,165,182,148,135,64,178,235,105,255,223,2310
1530 DATA 79,161,86,12,163,197,178,220,5,128,58,133,182,135,96,134,238,138,122,103,2568
1540 DATA 59,140,195,18,25,27,37,70,130,61,166,190,10,106,21,206,167,20,55,78,1781
1550 DATA 49,53,201,164,167,49,57,29,217,232,97,91,156,188,156,168,204,241,88,227,2834
1560 DATA 119,207,253,93,59,15,122,41,168,145,43,247,5,133,157,34,158,200,4,24,2227

1570 DATA 22,100,203,88,82,3,214,71,201,45,234,106,95,198,55,241,65,135,231,112,2501
1580 DATA 88,193,84,57,25,17,174,163,140,110,47,104,139,111,198,197,161,236,244,240,2728
1590 DATA 79,186,105,183,2,113,243,107,122,100,129,176,154,161,217,6,126,25,97,246,2577
1600 DATA 4,173,165,31,190,137,247,11,215,173,247,234,69,67,48,24,211,212,60,59,2577
1610 DATA 71,179,169,251,165,174,246,103,42,100,224,146,236,232,209,11,209,91,254,41,3153
1620 DATA 213,175,124,111,208,9,227,184,253,234,40,2,234,226,126,100,200,78,234,7,2985
1630 DATA 219,65,198,79,31,119,46,89,98,254,70,246,109,233,11,61,226,76,165,78,2473
1640 DATA 44,38,15,183,206,189,148,143,12,117,152,246,164,100,51,220,89,105,165,243,2630

1650 DATA 36,85,141,175,9,178,47,44,22,141,41,169,39,45,146,206,148,173,109,11,1965
1660 DATA 70,230,113,172,236,107,142,2,143,215,6,180,112,35,221,31,219,178,183,104,2699
1670 DATA 53,248,131,107,48,139,159,175,15,73,168,2,210,133,226,55,159,251,118,16,2486
1680 DATA 150,61,206,114,47,117,119,37,187,180,120,120,13,11,237,50,102,131,208,75,2285
1690 DATA 129,27,146,147,252,254,141,251,55,21,181,152,112,123,18,146,19,3,161,242,2580
1700 DATA 143,122,189,216,220,232,76,30,81,57,219,199,148,222,122,17,149,186,243,18,2889
1710 DATA 199,215,238,201,9,3,216,185,253,60,162,165,169,15,119,239,61,7,19,158,2693
1720 DATA 22,175,254,67,248,211,2,104,254,207,168,227,114,62,115,139,19,176,8,83,2655

1730 DATA 73,68,49,84,142,196,169,53,186,251,44,250,40,27,51,151,215,179,222,113,2563
1740 DATA 116,189,243,214,140,27,205,253,220,96,172,103,196,126,52,148,237,146,147,187,3217
1750 DATA 6,255,13,175,57,65,116,232,39,18,218,67,69,9,93,203,14,208,223,50,2130
1760 DATA 55,49,217,105,167,179,48,180,170,58,109,17,90,32,216,44,79,30,100,101,2046
1770 DATA 230,178,185,216,1,195,254,108,233,223,208,136,73,224,236,150,30,22,237,137,3276
1780 DATA 10,56,27,10,205,147,53,101,204,22,180,54,42,118,123,210,106,20,63,246,1997
1790 DATA 38,4,132,160,53,104,101,181,244,107,129,213,123,108,209,254,185,221,48,47,2661
1800 DATA 225,254,68,133,205,230,174,133,26,21,16,160,65,252,26,88,30,104,5,232,2447

1810 DATA 119,233,159,16,121,51,202,202,180,81,250,129,198,101,85,28,98,243,95,91,2682
1820 DATA 32,153,161,77,14,68,95,39,86,12,114,147,151,183,139,176,204,202,32,24,2109
1830 DATA 7,10,182,233,184,164,16,9,197,4,129,39,202,194,45,139,41,216,146,165,2322
1840 DATA 74,187,192,76,89,101,1,23,127,29,245,24,83,159,19,94,194,231,24,107,2079
1850 DATA 91,29,106,39,13,54,109,81,240,73,146,9,130,250,202,41,96,16,119,168,2012
1860 DATA 127,113,252,86,214,111,131,61,17,77,165,236,240,72,206,46,30,74,39,73,2370
1870 DATA 80,132,189,86,183,217,43,243,209,10,33,47,200,131,213,148,92,152,12,4,2424
1880 DATA 159,114,132,52,206,205,108,185,169,54,44,169,2,98,56,205,226,203,92,211,2690
1890 DATA 121,12,148,107,74,96,204,174,232,224,212,163,23,27,70,101,154,227,102,20,2491

1900 DATA 45,200,31,160,104,28,88,75,92,189,211,218,236,174,127,88,48,205,141,142,2602
1910 DATA 105,51,209,80,83,125,119,100,111,131,146,62,189,130,14,212,26,149,220,59,2321
1920 DATA 224,3,123,216,191,38,186,30,50,238,200,226,242,215,143,15,245,159,159,18,2921
1930 DATA 158,122,157,136,5,134,187,136,174,48,161,25,241,33,19,70,209,146,240,223,2624
1940 DATA 12,14,68,128,161,157,234,129,114,157,110,68,196,60,199,26,46,0,210,148,2237
1950 DATA 78,122,50,186,100,15,212,198,34,114,88,179,232,245,134,82,192,57,156,151,2625
1960 DATA 163,122,30,190,225,20,118,156,153,209,42,250,139,143,127,17,205,236,148,11,2704

1970 DATA 73,143,63,165,0,75,29,36,135,150,60,85,253,63,248,13,252,244,55,237,2379
1980 DATA 48,208,129,54,129,6,88,248,22,4,55,58,164,88,27,95,31,43,23,235,1755
1990 DATA 92,84,49,1,194,44,144,93,155,47,120,39,230,82,242,104,6,85,218,202,2231
2000 DATA 231,24,64,237,194,103,98,170,161,101,88,146,74,228,181,233,179,176,143,12,2843
2010 DATA 77,61,236,5,199,201,54,29,30,4,11,59,146,21,178,247,67,198,97,110,2030
2020 DATA 249,217,230,68,110,197,130,102,201,141,234,206,19,122,99,187,184,203,117,86,3102

2030 DATA 173,151,245,246,76,200,105,196,144,130,22,153,32,116,44,240,209,13,44,83,2622
2040 DATA 15,150,246,153,189,178,20,244,34,28,181,223,233,187,168,162,242,181,138,83,3055
2050 DATA 188,196,37,114,104,92,210,184,119,187,208,248,83,193,82,251,6,47,47,131,2727
2060 DATA 218,51,80,24,179,115,50,184,45,60,30,226,88,4,216,166,131,95,175,107,2244
2070 DATA 177,45,199,109,248,4,197,44,3,171,72,209,146,46,53,152,128,117,144,164,2428
2080 DATA 26,95,52,52,236,193,153,1,60,159,224,231,70,29,73,198,155,243,97,43,2390

2090 DATA 232,238,111,124,63,208,172,151,85,220,199,211,77,15,130,236,174,219,84,110,3059
2100 DATA 139,137,107,215,209,179,165,60,18,125,3,228,122,100,184,185,148,253,227,58,2842
2110 DATA 180,1,35,189,140,235,36,18,202,153,239,255,101,196,250,130,119,50,143,39,2710
2120 DATA 4,200,60,22,30,17,119,164,144,75,108,225,103,85,127,19,2,253,112,51,1920
2130 DATA 37,8,248,23,198,233,30,88,246,50,165,126,134,187,203,51,245,178,81,128,2659

2140 DATA 219,55,5,251,17,42,29,59,230,178,216,109,49,127,90,18,20,140,30,148,2032
2150 DATA 129,128,119,64,250,106,150,135,42,21,92,154,95,201,247,242,120,80,238,6,2619
2160 DATA 5,69,224,179,44,242,91,88,242,80,164,166,82,127,91,81,124,63,240,210,2612
2170 DATA 15,116,136,195,239,131,230,39,41,93,193,169,228,86,169,191,135,127,172,199,2904
2180 DATA 197,79,46,226,113,127,219,229,168,157,67,224,132,95,130,186,125,34,236,214,3004
2190 DATA 27,87,27,115,112,2,172,155,23,220,253,48,15,157,253,179,73,252,33,216,2419
2200 DATA 187,149,47,227,73,148,8,24,91,247,211,54,116,7,232,64,196,248,182,157,2668
2210 DATA 189,26,105,106,41,105,117,99,198,193,225,10,133,38,211,21,188,232,155,98,2490
2220 DATA 202,103,189,70,252,192,176,143,253,134,94,221,91,140,166,60,83,228,141,52,2990
2230 DATA 51,239,212,207,227,243,134,19,192,113,208,17,232,108,177,132,237,153,25,238,3164
2240 DATA 235,117,230,123,156,241,190,252,120,97,179,105,239,80,51,52,0,0,0,0,2467
2250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2650 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2690 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2700 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2730 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2740 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2750 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Jacek Szaniawski



Nie martw się kochanie!
Za życia wyglądałeś jeszcze gorzej.

Lilavati (10)

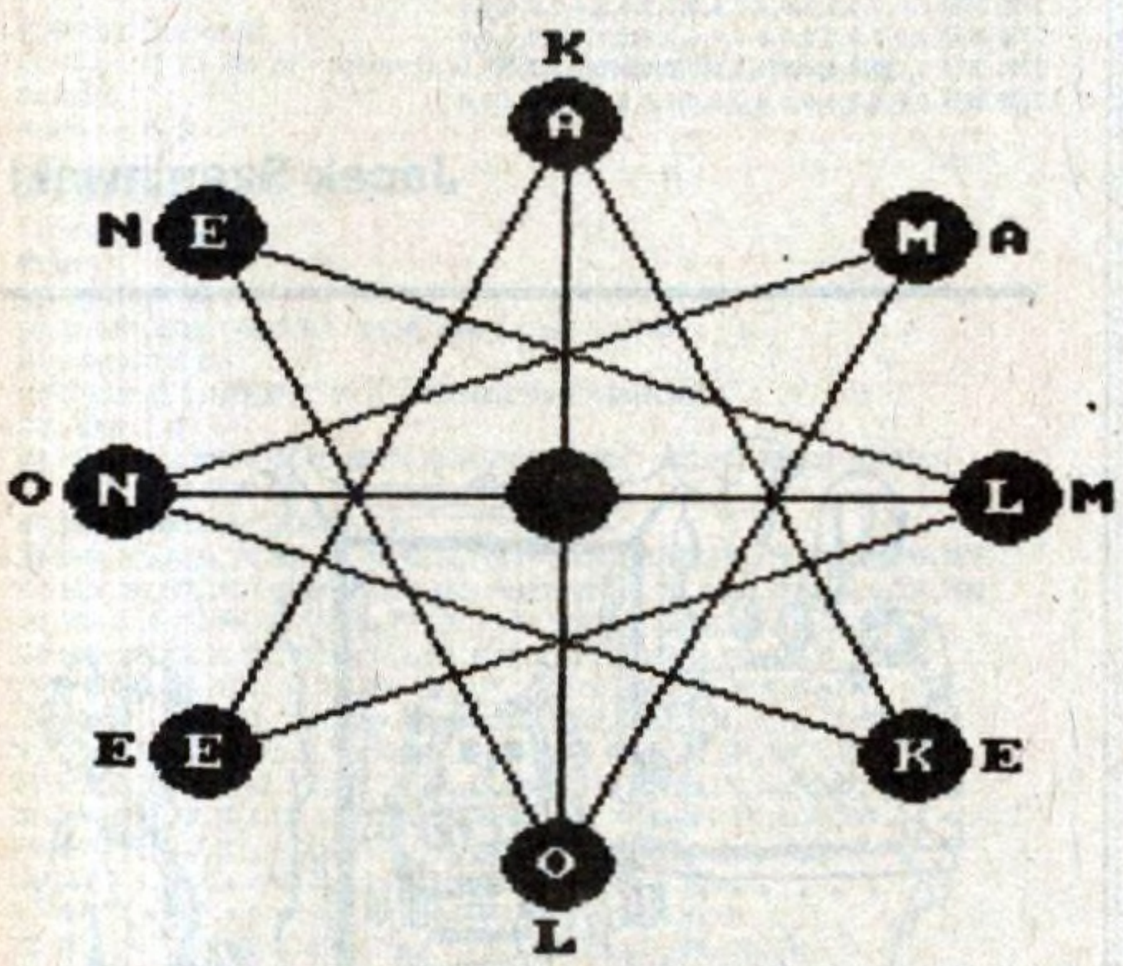
KAMELEON

Danuta KWASIŻUR, Mieczysław SKONIECZNY

W kolejnym, niejako jubileuszowym odcinku cyklu poświęconego rozwiązywaniu anegdot matematycznych przy użyciu mikrokomputera, prezentujemy prostą grę planszową przeznaczoną dla jednego gracza o nazwie „Kameleon". Zasady tej gry są bardzo proste, a prezentuje je program w początkowej fazie realizacji na ekranie monitora. Kopia tego obrazu ma następującą postać:

```
Na osmiu wierzcholkach gwiazdy osmio-
ramiennej i w jej srodku rozmieszczone
sa czarne krazki. Ponad krazkami stoja
litery, ktore w kierunku obiegu wskazow-
ki zegara skladaja poczawszy od wierz-
cholka gornego slowo KAMELEON.Oprocz
tego jest 8 luznych ruchomych krazkow,
na ktorych wypisane sa te same litery:
K,A,M,E,L,E,O,N. Krazki ruchome ustawio-
ne sa w przypadkowej kolejnosci na osmiu
zewnetrznych krazkach gwiazdy. Krazek
srodkowy pozostaje nie zajety. Gra pole-
ga na takim kolejnym przesuwaniu krazkow
ruchomych, aby wypisane na nich litery
znalazly sie obok takich samych liter
zapisanych zewnatrz gwiazdy, a wiec
utworzyly wyraz KAMELEON.
STEROWANIE RUCHEM KRAZKOW:
Kolejne naciskanie SPACJI powoduje migo-
tanie liter na kolejnych krazkach.
Nacisniecie klawisza ENTER spowoduje
przesuniecie krazka z migoczaca litera
na pusty krazek /jezeli takie przesunie-
cie jest mozliwe/.
NACISNIJ DOWOLNY KLAWISZ
```

Po zapoznaniu się z tekstem informacyjnym rozpoczyna się właściwy etap gry. Widok planszy i początkowego rozstawienia krążków jest widoczny na poniższym rysunku:



Wersja źródłowa programu opracowanego w języku BASIC jest następująca:

```
10 '
20 '              Ekran informacyjny
30 '
40 MODE 1:INK 0,14:INK 1,0:INK 2,6,26:INK 3,26:BORDER 14:PAPER 0:CLS
50 PEN 1:PRINT "Na osmiu wierzcholkach gwiazdy osmio-  ramiennej i w jej srodku
 rozmieszczone  sa czarne krazki. Ponad krazkami sto
ja  litery, ktore w kierunku obiegu wskazow-ki zegara skladaja poczawszy od wier
z-  cholka gornego slowo KAMELEON.Oprocz"
60 PRINT "tego jest 8 luznych ruchomych krazkow,  na ktorych wypisane sa te same
 litery:  K,A,M,E,L,E,O,N. Krazki ruchome ustawio-ne
 sa w przypadkowej kolejnosci na osmiuzewnetrznych krazkach gwiazdy. Krazek  sr
odkowy pozostaje nie zajety. Gra pole-";
70 PRINT "ga na takim kolejnym przesuwaniu krazkowruchomych, aby wypisane na nic
h litery  znalazly sie obok takich samych liter  za
pisanych zewnatrz gwiazdy, a wiec  utworzyly wyraz KAMELEON.
80 PEN 3:PRINT "STEROWANIE RUCHEM KRAZKOW:"
90 PEN 1:PRINT "Kolejne naciskanie SPACJI powoduje migo-tanie liter na kolejnych
 krazkach.  Nacisniecie klawisza ENTER spowoduje
   przesuniecie krazka z migoczaca litera  na pusty krazek /jezeli takie przesu
nie-cie jest mozliwe/."
100 PEN 3:PRINT "NACISNIJ DOWOLNY KLAWISZ"
110 CLEAR INPUT
120 a$=INKEY$
130 IF a$="" THEN GOTO 120
140 '
150 '          Segment sterujacy
160 '
170 GOSUB 920
180 GOSUB 640
190 PAPER 1:PEN 0
200 FOR k=2 TO 9
210 LOCATE wsp(k,1)+1,wsp(k,2)+1:PRINT LIT$(tabl(k))
220 NEXT k
230 PAPER 1:l=1
240 K=1
250 IF TABL(K)=1 THEN k=k+1:GOTO 250
260 PEN 2:LOCATE wsp(k,1)+1,wsp(k,2)+1:PRINT lit$(tabl(k))
270 CLEAR INPUT
280 a$=INKEY$
290 IF a$="" THEN GOTO 280
300 IF INKEY(6)=0 THEN GOTO 360
310 IF a$=" " THEN GOTO 330
320 SOUND 1,450:GOTO 280
330 PEN 0:LOCATE wsp(k,1)+1,wsp(k,2)+1:PRINT lit$(tabl(k))
340 IF k=9 THEN GOTO 240
350 k=k+1:GOTO 250
360 FOR v=1 TO 4
370 IF pow(k,v)=l THEN GOTO 400
380 NEXT v
390 SOUND 1,500:LOCATE £1,2,2:PRINT £1,"PRZESUNIECIE":PRINT £1," NIEMOZLIWE":FOR
 T=1 TO 1000:NEXT T:CLS £1:GOTO 250
400 PEN 0:LOCATE wsp(L,1)+1,wsp(L,2)+1:PRINT lit$(tabl(k))
410 LOCATE wsp(K,1)+1,wsp(K,2)+1:PRINT " ":TABL(L)=TABL(K):TABL(K)=1:L=K
420 GOSUB 770
430 ON BL+1 GOTO 180,240
```

W segmencie głównym programu znajdują się wywołania podprogramów, których rola i wersje źródłowe są następujące:

1. Definiowanie symboli własnych (kształt krążka) oraz sparametryzowane wpisywanie symbolu na ekran (wg wartości parametrów x,y będących współrzędnymi znakowymi).

```
440 '
450 '          Podprogram definicji symboli wlasnych
460 '
470 SYMBOL AFTER 125
480 SYMBOL 125,0,0,0,0,3,7,15,15
490 SYMBOL 126,0,0,60,255,255,255,255,255
500 SYMBOL 127,31,31,63,63,63,63,31,31
510 SYMBOL 128,248,248,252,252,252,252,248,248
520 SYMBOL 129,15,15,7,3
530 SYMBOL 130,255,255,255,255,255,60
540 SYMBOL 131,240,240,224,192
550 SYMBOL 132,0,0,0,0,192,224,240,240
560 RETURN
570 LOCATE x,y:PRINT CHR$(125);CHR$(126);CHR$(132)
580 LOCATE x,y+1:PRINT CHR$(127);CHR$(143);CHR$(128)
590 LOCATE x,y+2:PRINT CHR$(129);CHR$(130);CHR$(131)
600 RETURN
```

2. Losowe generowanie początkowego ustawienia liter na planszy.

```
610 '
620 '        Podprogram losowego ustawiania liter na planszy
630 '
640 los=2+INT(RND*8)
650 tabl(1)=1:tabl(2)=los
660 FOR K=3 TO 9
670 los=2+INT(RND*8)
680 FOR i=2 TO k-1
690 IF tabl(i)=los THEN GOTO 670
700 NEXT i
710 tabl(k)=los
720 NEXT k
730 RETURN
```

3. Badanie poprawnego rozwiązania zadania — zakończenie gry i ewentualne powtórzenie gry.

```
740 '
750 '        Podprogram badania poprawnosci zadania
760 '
770 BL=0
780 FOR J=1 TO 9
790 IF TABL(J)=J THEN GOTO 810
800 BL=1
810 NEXT J
820 IF BL=1 THEN RETURN
830 LOCATE £1,2,2:PRINT £1,"BRAWO!":PRINT £1," UDALO SIE!"
840 FOR J=0 TO 26:BORDER J:FOR T=1 TO 100:NEXT T:NEXT J
850 CLS £1:PRINT £1:PRINT £1,"SPROBUJESZ":PRINT £1,"JESZCZE RAZ? ":INPUT £1,"(T/
N) ",T$
860 IF T$="T" OR T$="t" THEN LOCATE 13,13:PRINT " ":CLS £1:RETURN
870 IF T$="n" OR T$="N" THEN STOP
880 GOTO 850
```

4. Rysowanie planszy gry.

```
890 '
900 '        Podprogram rysowania planszy
910 '
920 MODE 1
930 DIM wsp(9,2):DIM pow(9,4)
940 GOSUB 470
950 FOR i=1 TO 9:FOR j=1 TO 4:READ pow(i,j):NEXT j:NEXT i
960 DATA 2,4,6,8,1,5,7,0,6,8,0,0,1,7,9,0,8,2,0,0,1,3,9,0,2,4,0,0,1,3,5,0,6,4,0,0
970 FOR i=1 TO 9:READ wsp(i,1),wsp(i,2):NEXT i
980 DATA 12,12,12,2,20,5,22,12,20,19,12,22,4,19,2,12,4,5
990 DIM LIT$(9):FOR i=1 TO 9:READ LIT$(i):NEXT I
1000 DATA " ","K","A","M","E","L","E","O","N"
1010 PEN 1:GRAPHICS PEN 1
1020 FOR I=1 TO 9:X=WSP(I,1):Y=WSP(I,2):GOSUB 570:NEXT I
1030 FOR I=2 TO 4 STEP 2
1040 FOR K=5+(I-2) TO 7+(I-2)
1050 PLOT WSP(I,1)*16+9,(24-WSP(I,2))*16+7
1060 DRAW WSP(K,1)*16+9,(24-WSP(K,2))*16+7
1070 NEXT K
1080 NEXT I
1090 FOR K=3 TO 5
1100 PLOT WSP(8,1)*16+9,(24-WSP(8,2))*16+7
1110 DRAW WSP(K,1)*16+9,(24-WSP(K,2))*16+7
1120 NEXT K
1130 WINDOW £1,27,40,11,15:PAPER £1,1:CLS £1:PEN £1,3
1140 FOR K=3 TO 9 STEP 6
1150 PLOT WSP(6,1)*16+9,(24-WSP(6,2))*16+7
1160 DRAW WSP(K,1)*16+9,(24-WSP(K,2))*16+7
1170 NEXT K
1180 PEN 3:LOCATE 13,1:PRINT"K":LOCATE 23,6:PRINT"A":LOCATE 25,13:PRINT"M":LOCAT
E 23,20:PRINT"E":LOCATE 13,25:PRINT"L":LOCATE 3,20:P
RINT"E":LOCATE 1,13:PRINT"O":LOCATE 3,6:PRINT"N"
1190 RETURN
```

Rady i porady programisty (2)

Z. ŁUKASZEWICZ

Pisząc program rzadko zwracam uwagę na jego elegancję i rozwiązania konstrukcyjne, dążąc najprościej do osiągnięcia zamierzonego celu. Pamiętać jednak należy również o tym, że dobry program, to nie tylko taki program, który się „nie zacina", ale również jest czytelnie napisany. Przez czytelność i elegancję programu rozumiemy ograniczenie do minimum wierszy zawierających kolejno po sobie występujące instrukcje, instrukcje "go to", czyli unikanie zbędnych skoków. O tym i o wielu innych problemach wpływających na przejrzystość programów wielokrotnie pisano na łamach „IKS-a".

Jednym ze sposobów uniknięcia tych błędów jest posiadanie biblioteki krótkich, sprawdzonych podprogramów najczęściej używanych w pracy programisty-amatora. Takim przykładem jest podprogram umożliwiający sterowanie w programie w zależności od wybranego klawisza.

```
10 '*****STEROWANIE PRZY UŻYCIU KLAWIATURY*****
20 RESTORE 80
30 FOR r%=1 TO n
40 READ s%
50 IF INKEY (s%)=0 THEN 90
60 NEXT
70 GOTO 20
80 DATA K1, K2, ..., Kn
90 ON r% GOSUB E1, E2, ..., En
```

W podprogramie rzeczywiste wartości zastąpiono parametrami, i tak:

n — liczba klawiszy użytych do sterowania;

K1, ..., Kn — numery klawiszy użytych do sterowania (numery umieszczone są na obudowie stacji dysków);

E1, ..., En — numery linii, w której zaczynają się podprogramy, które należy wykonać po wciśnięciu klawisza o numerze Kn.

Aby ułatwić zrozumienie przedstawiono poniżej przykład, w którym do sterowania użyto czterech klawiszy o numerach 64, 65, 57, 56 czyli „1", „2", „3", i „4".

Zastosowanie tego podprogramu w twoich programach na pewno poprawi ich przejrzystość i czytelność.

Spróbuj, sprawdź!

```
10 MODE 2
20 WINDOW£1,30,50,12,12
30 '**** PRZYKŁAD ***
40 CLS£1
50 RESTORE 110
60 FOR r%=1 TO 4
70 READ s%
80 IF INKEY (s%)=0 THEN 120
90 NEXT
100 GOTO 50
110 DATA 64,65,57,56
120 ON r% GOSUB 150,160,170,180
130 CALL &BB06
140 GOTO 40
150 CLEAR INPUT: PRINT£1," WYBRANO KLAWISZ
    1":RETURN
160 CLEAR INPUT: PRINT£1," WYBRANO KLAWISZ
    2":RETURN
170 CLEAR INPUT: PRINT£1," WYBRANO KLAWISZ
    3":RETURN
180 CLEAR INPUT: PRINT£1," WYBRANO KLAWISZ
    4":RETURN
```

INFORMATYKA W SZKOLE

Programowanie

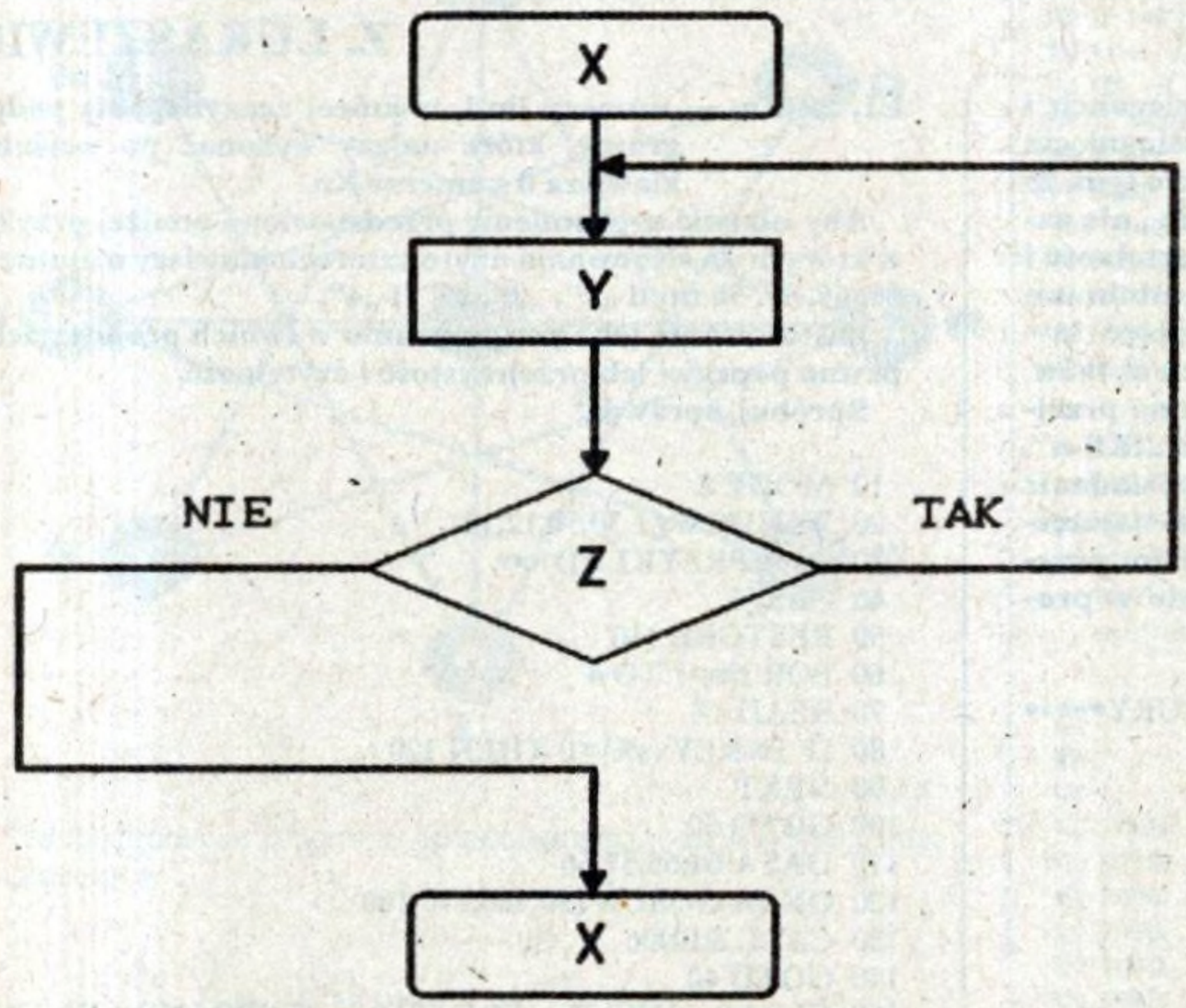
(Materiały do Podstaw Informatyki cz. 3)

Lektura poprzednich dwóch „odcinków" omawiających tematy naszej propozycji podstaw informatyki upoważnia już nas zapewne do przyjrzenia się niektórym problemom programowania komputerów. Komputerów? Tak. Komputerów, nie ma bowiem większego znaczenia, jak wielki jest to sprzęt, mikro, mini, stołowy czy też super lub hiper, ogólne zasady programowania są niezmienne.

Aby choć w części zasady te zrozumieć, koniecznie trzeba wiedzieć co to jest mikroprocesor, pamięć i oczywiście bajt! Wróćmy do naszego przykładu dodawania dwóch liczb. Załóżmy, że są one zapisane w pamięci (oczywiście RAM) komputera. Pierwsza liczba np.: 123 w komórce nr 100, druga natomiast np.: 056 w komórce nr 101. Pozostałe komórki pamięci są „puste" (w wypadku Spectrum, pozostaną wolne następujące komórki: od 0 do 99 i od 102 do 48 000). Usiłując dodać, za pomocą komputera, owe liczby (123+056) maszyna powinna wykonać określone czynności:

1. *Pobrać (odczytać) zawartość setnej komórki pamięci i przekazać (tę zawartość = 123) do procesora.*
2. *Pobrać drugi argument z komórki sto jeden (056) i przekazać do procesora.*
3. *Wysterować procesor tak by dodał wcześniej wprowadzone argumenty (123+056=179).*
4. *Przekazać z procesora wynik do komórki np.: 102.*

W efekcie wykonania podanych operacji wynik dodawania (179) będzie umieszczony w komórce nr 102. Operacje te (czynności), w podanym przykładzie, tworzą pewien „przepis" — przykład prostego algorytmu procesu dodania dwóch liczb przez komputer. Innymi słowy jest to opis kolejnych czynności, których wykonanie umożliwia nam osiągnięcie zamierzonego celu. Zatem algorytm ułatwia poprawne, jednoznaczne i czytelne sprecyzowanie czynności, tworzących pracę powierzoną maszynie. Czyli przed przystąpieniem do pisania programu celowe jest zbudowanie algorytmu. Zadanie to jest nieco ułatwione przez stosowanie umownych symboli graficznych.



x — treść opisująca czynność wykonywaną na początku lub na końcu algorytmu
y — czynności związane z przetwarzaniem informacji
z — sprawdzenie warunku i odpowiednie skierowanie do wykonywania kolejnych czynności

Rys. 1 Niektóre symbole graficzne stosowane w algorytmach

Nareszcie możemy przystąpić do próby napisania programu. Polega to na tłumaczeniu treści zawartych w algorytmie na taką postać, która jest „zrozumiana" przez maszynę. Podobnie jak tłumaczenie, osobie posługującej się tylko językiem angielskim, poleceń wypowiadanych w języku polskim. W praktyce nie jest to takie łatwe, procesor bowiem wykonuje tylko bardzo proste operacje — podobne do wykorzystanych w podanym wcześniej przykładzie. Mikroprocesor jest jak małe dziecko — należy mówić do niego, wyraźnie (bez błędów), bardzo prostym językiem — najlepiej pojedynczymi słowami: idź, dodaj, skocz, przesuń. Oczywiście owe polecenia mają swoje kody — „liczbowe nazwy" — na przykład operację dodawania wykona procesor na „polecenie" (rozkaz) 10000010. Zatem, aby dodać dwie liczby musimy „tylko" przekazać do procesora z dziesięć odpowiednich kodów operacji i „już" mamy wynik. Trzydzieści, dwadzieścia lat temu był to zresztą jedyny sposób programowania komputerów. Dzisiaj komputery są bardziej „życzliwe" („dorosły"), możemy bowiem opisywać nasz algorytm (pisać program) niemal w języku naturalnym, na przykład: PRINT A+B. Ten prosty zapis spowoduje dodanie przez komputer zawartości komórki A do komórki B i wyświetlenie wyniku na ekranie. Jeśli wcześniej do A wpisaliśmy 123 (LET A=123), a do B wprowadzono liczbę 056 (LET B=56), na ekranie ukaże się wynik: 179.

Tak wielkie zbliżenie języka programowania do tego jakim posługują się ludzie, możliwe stało się dzięki specjalnym **programom tłumaczącym.** Należy bowiem pamiętać, że mikroprocesory nadal „rozumieją" kody tylko bardzo prostych operacji. Zatem wspomniany „tłumacz" zamienia naszą jedną instrukcję np.: PRINT A+B na kilkadziesiąt kodów „rozumianych" przez procesor.

Innymi słowy mamy dwa rodzaje, a poprawniej — dwa poziomy języków. Język składający się z kodów (liczb) wymuszających proste operacje — jest to tak zwany **język wewnętrzny** (maszynowy). Bardzo **szybki,** elastyczny, ale kłopotliwy w użyciu. Mówi się o nim, że jest to język niskiego poziomu. Inny dla każdego typu procesora. Zważywszy jednak na niezmiennie **dużą użyteczność** języka wewnętrznego, a jednocześnie na kłopoty wynikające z konieczności stosowania kodów operacji w postaci liczb — powstał **assembler.** W języku tym posługujemy się literowy-

mi, bardziej czytelnymi niż liczby, kodami operacji (także adresów). Na przykład zamiast kodu 10000100, piszemy ADD (ang. dodaj), zamiast kodu 010000100, piszemy MOV (ang. MOVE — przesuń). W przykładach tych, mając na uwadze ich czytelność, pominięto kwestię adresacji, zainteresowanym polecam np.: „Mikrokomputery", A. J. Dirksen, WKiŁ, Warszawa 1985. Kończąc tę krótką prezentację assemblera należy pamiętać, że czas pisania programu w tym języku jest średnio osiem razy dłuższy od programowania w **językach wysokiego poziomu.** Jednak należy mieć na uwadze to, że zysk ten powstaje kosztem znacznego zmniejszenia szybkości działania programów.

Różne zastosowania komputerów wymusiły tworzenie odpowiednich języków wysokiego poziomu. Na przykład do oprogramowania obliczeń matematycznych stosuje się język FORTRAN (także w superkomputerach), do przetwarzania tekstów LISP, do budowy oprogramowania systemowego — język C, itp.

Warto w tym miejscu trochę więcej miejsca poświęcić wspomnianym „tłumaczom" języków wysokopoziomowych na język wewnętrzny komputera. Wykorzystuje się do tego celu, **kompilatory** lub **interpretery.** Kompitalor jest to (w znacznym uproszczeniu) program komputerowy, który zamienia instrukcje języka wysokopoziomowego na język wewnętrzny. Wprowadzamy zatem do komputera nasz program napisany np. w Basicu, tam (w pamięci komputera) czeka na niego kompilator. Następnie rozpoczyna się proces kompilacji po jej ukończeniu, w pamięci komputera pozostaje nasz program, ale już zupełnie w innej postaci — jako program napisany w języku wewnętrznym (zrozumiałym dla procesora). W takiej właśnie postaci jest on dalej wykonywany.

Natomiast **interpretery,** to także programy tłumaczące, ale „na żywo". Oznacza to, że **za każdym razem,** gdy uruchamiamy program napisany np.: w Basicu, jest on (w trakcie realizacji) tłumaczony (przez interpreter) instrukcja po instrukcji na język wewnętrzny i dopiero wykonywany przez komputer. Zatem wykonanie takiego programu jest znacznie dłuższe (wolniejsze), bowiem do okresu wykonania każdej instrukcji programu zawsze dodaje się czas jej tłumaczenia.

W większości domowych komputerów, właśnie w pamięci ROM, najczęściej zapisane mamy interpretery Basica, (były także, nieudane ze względu na brak zainteresowania próby z Pascalem, Logo i innymi programami.

Zważywszy na informacyjny zakres naszej nauki podstaw informatyki należy zwrócić uwagę na ryzyko, jakie niosła by obecnie sobą próba zbyt dokładnego poznania jakiegokolwiek języka programowania. Dlatego też, zważywszy na konieczność równoległego uczestniczenia PT Czytelników w zajęciach praktycznych, proponuję analizę funkcji **tylko kilku** instrukcji języka BASIC: PRINT, INPUT, DIM, GO TO, FOR, NEXT. Wcześniej jednak koniecznie powinniśmy poznać klawiaturę naszego komputera (klawisze funkcyjne, sposób pisania komend systemowych i instrukcji, itp.). Niestety pamiętać także trzeba, iż istnieją różne „dialekty" języka Basic. Oznacza to, że nie wszystkie nasze przykłady będą bezpośrednio dotyczyły różnych komputerów, dotyczą one bowiem, nadal najpopularniejszego u nas mikrokomputera Spectrum. W szczególności dotyczy to instrukcji DIM, która jest odmiennie interpretowana w Spectrum i innych komputerach np.: w Amstradzie. Proponuję w tej sytuacji przestudiować dokładniej niektóre fragmenty dokumentacji dostępnego komputera.

Wiemy zatem, że komputer dysponuje pamięcią, mikroprocesorem, szeregiem innych ciekawych elementów (mowa o tym będzie innym razem) — w sumie — zdolnością wykonywania (bardzo szybko) instrukcji w różnych językach programowania.

Przystępując do analizy niektórych zagadnień związanych z programowaniem komputerów należy przyjąć pewne założenie dotyczące interpretacji słowa „komórka". Wcześniej bowiem, dla ułatwienia, termin ten był jednoznaczny z bajtem. Jednak w językach wysokopoziomowych komórka staje się już wyróżnionym obszarem pamięci, składającym się z wielu bajtów. Istotnym ułatwieniem jest sposób identyfikacji owych komórek — mają one swoje (nadawane przez nas!) nazwy. Na przykład: IKS, nowa, P13, itp. Chcąc zatem cokolwiek zapisać do komórki (lub odczytać), posługujemy się jej nazwą, a nie numerem. Znakomicie ułatwia to pisanie programów. Szczególnie ten sposób **adresowania** miejsca w pamięci (jedna nazwa dla wielu bajtów) przydatny jest gdy chcemy posługiwać się wieloma danymi tego samego typu. Na przykład rejestrując w pamięci skróty informatyczne (zob. „IKS" nr 5/1988) budujemy tak zwaną **tablicę** o wielkości umożliwiającej zapamiętanie wielu skrótów. Np.: A$(a,b). Znak $ oznacza, że w tablicy o nazwie A zapisywane będą teksty, a — ile będzie skrótów, b — ile przewidujemy znaków dla każdego skrótu. Ten sposób gospodarki pamięcią okaże się bardzo przydatny. Aby można było korzystać z takiej tablicy należy wcześniej wykonać instrukcję DIM. Jej działanie można porównać z pracą mającą na celu dzielenie naszej działki na grządki: tu cebula, tam marchew, itp. Działka — to cała pamięć, grządki to jej części przydzielone różnym informacjom. DIM „pilnuje" porządku, np. by nie zrobić zbyt wielu grządek, bo po prostu działka okaże się zbyt mała. Stosując zatem DIM, należy podać ile danych chcemy wprowadzić — np. 200 skrótów i jak wielki jest każdy skrót (ile zajmuje znaków), np. dziesięć. Zapisujemy to następująco: DIM A$(200, 10).

Oznacza to, że w komputerze zostanie zarezerwowany obszar — nasza tablica A o wielkości 2000 bajtów z przeznaczeniem na dwieście skrótów, każdy po 10 znaków. Lub inaczej — w RAM zostanie zarezerwowane 200 komórek, każda po 10 znaków. Czyli komórki:

A$ (1, 10) — komórka nr 1
A$ (2, 10) — komórka nr 2
A$ (3, 10) — komórka nr 3
.
.
.
A$ (199, 10)
A$ (200, 10) — komórka nr 200.

Wróćmy teraz do naszego głównego tematu — programowanie. Na dobry początek wprowadzamy do pamięci komputera instrukcję:

```
INPUT „ALA", a
```

Po jej uruchomieniu (klawisz RETURN lub ENTER), na ekranie ukaże się słowo: ALA?, ze znakiem zapytania. Jest to efekt „bezmyślnego" odtworzenia przez maszynę tekstu w cudzysłowie (ALA) i komputer czeka na to byśmy napisali jakąś liczbę. Po napisaniu tej liczby, np. 18, zostanie ona zapamiętana w pamięci RAM. W którym miejscu (przecież pamięć jest bardzo duża)? W komórce o nazwie a. Jest to możliwe do realizacji w tak prosty sposób dzięki interpreterowi Basica. Szybko nadaje on, wymyślone przez nas nazwy kolejnym komórkom pamięci RAM. W naszym wypadku jedna z komórek będzie miała nazwę „a".

Gdy napiszemy: INPUT „ALA", nic nazwa naszej komórki brzmi „nic". Z doświadczenia wiadomo, iż zrozumienie relacji nazwy komórki i jej zawartości jest dosyć kłopotliwe. Dlatego omówienie tego tematu zostało powtórzone. Warto zresztą wrócić do tego tematu, jeśli nie pamiętamy w tej chwili zawartości komórki „a"?

Wobec powyższego, instrukcja INPUT może wyświetlić dowolny tekst „w uszach" (w cudzysłowie) oraz, co najważniejsze, wprowadzić do pamięci pod określony adres dowolną liczbę. Czy tylko liczbę? Mogą być także litery, ale wówczas, jak już było wcześniej — mówione — nazwę komórki musi kończyć znak $.

Na przykład: INPUT „Podaj nazwę miasta", m$

Po uruchomieniu tej instrukcji na ekranie ukaże się:

Podaj nazwę miasta?

w odpowiedzi piszemy:

Ciechocinek

zatem w komórce m$ będzie zapisana nazwa naszego pięknego uzdrowiska.

Podsumowując naszą wiedzę o instrukcji INPUT, należy podkreślić, iż komputer wykonując ją wypisuje dowolny tekst „w uszach" (najlepiej, gdy dotyczy on treści oczekiwanej przez komputer informacji) i czeka na przekazanie mu (przez nas) danych.

W ten sposób, niestety dosyć złożony, możemy stwierdzić, iż wiemy jak wprowadzić do komputera dowolną liczbę lub tekst. Mało tego — wiemy także, gdzie wprowadzone przez nas informacje są zapisane, np. w komórce: a, nic, m.

Wróćmy do przykładów. Każda instrukcja budowanego przez nas programu musi mieć swój numer (żeby interpreter Basica się nie pogubił, w jakiej kolejności ma wykonywać wprowadzane przez nas instrukcje). W praktyce instrukcje numeruje się od 10 z odstępem też 10. Jest to uzasadnione potrzebami, które szybko podczas pracy z komputerem okażą się istotne.

Gdy wprowadzimy do komputera instrukcje:

```
10 INPUT „Podaj swoje nazwisko", n$
```

20 INPUT „Podaj swoje imię", i$
po ich uruchomieniu i udzieleniu odpowiedzi (Abacki Tadeusz) w RAM zapisane zostanie nazwisko i imię. I co dalej? Po co to wszystko?

W rezultacie w komórce o adresie ACK — zapisana zostanie interpretacja tego skrótu, pod adresem ACU, ADA i innych podobnie.
Zatem pisząc:

Trzeba jednak bardzo szybko zaznaczyć, że program ten jest fatalny — INPUT powtórzone 249 razy? Stwierdzenie to sugeruje, że istnieją inne sposoby konstrukcji programów.

Programowanie

(Materiały do Podstaw Informatyki cz. 3)

Przejdźmy zatem do wykorzystania tak pracowicie wprowadzonych informacji. Służy temu, w sposób bardzo czytelny, instrukcja PRINT. Działa ona akurat odwrotnie od „INPUTa". Oznacza to, że gdy wykonamy instrukcję:

PRINT „Twoje nazwisko brzmi:", n
na ekranie ukaże się tekst:

Twoje nazwisko brzmi: Abacki

Czyli PRINT wypisuje „bezmyślnie" tekst „w uszach" oraz to co wcześniej wprowadziliśmy do komórki n.

Realizacja instrukcji: PRINT „Mieszkam w:", i będzie następująca:

Mieszkam w: Tadeusz

Bez sensu! Właśnie przykład ten wskazuje, że tekst w uszach musi mieć sens związany z treścią wypisywanej komórki. Zatem poprawnie powinno być:

PRINT „Mieszkam w:", m
Na ekranie ukaże się:

Mieszkam w: Ciechocinek

Prawie dobrze.

Uporaliśmy się zatem z tekstem wyświetlanym na ekranie (skąd się on bierze? I po co?) oraz danymi, które wprowadzamy, by móc je w dowolnej chwili odczytać. Można powiedzieć, że jest to bardzo odległe od praktyki. Czyżby?

Wyobraźmy sobie, że „Inputami" wprowadziliśmy do pamięci komputera oryginalną interpretację wszystkich skrótów informatycznych, np.:

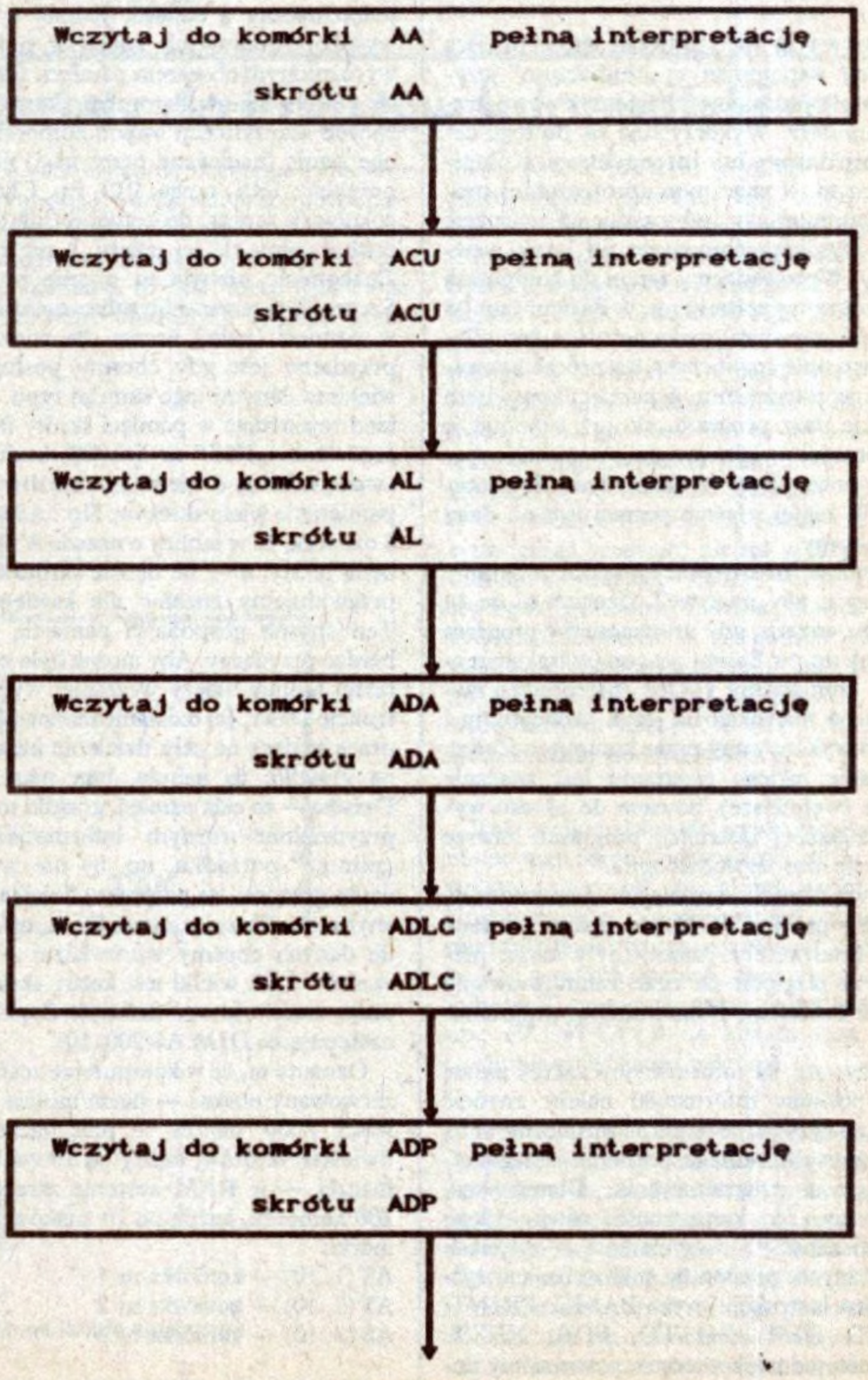
10 INPUT ACKS (pomijamy tekst w uszach)
20 INPUT ACUS
30 INPUT ADAS
40 INPUT ADLCS
.
.
.
249 INPUT WPR$

Po uruchomieniu naszego programu, korzystając z „IKS-a" nr 5 wpisujemy kolejno:

Affirmative Acknowledgment (AA)
Automatic Calling Unit (ACU)
All Points Addressable (ADA)
itd.

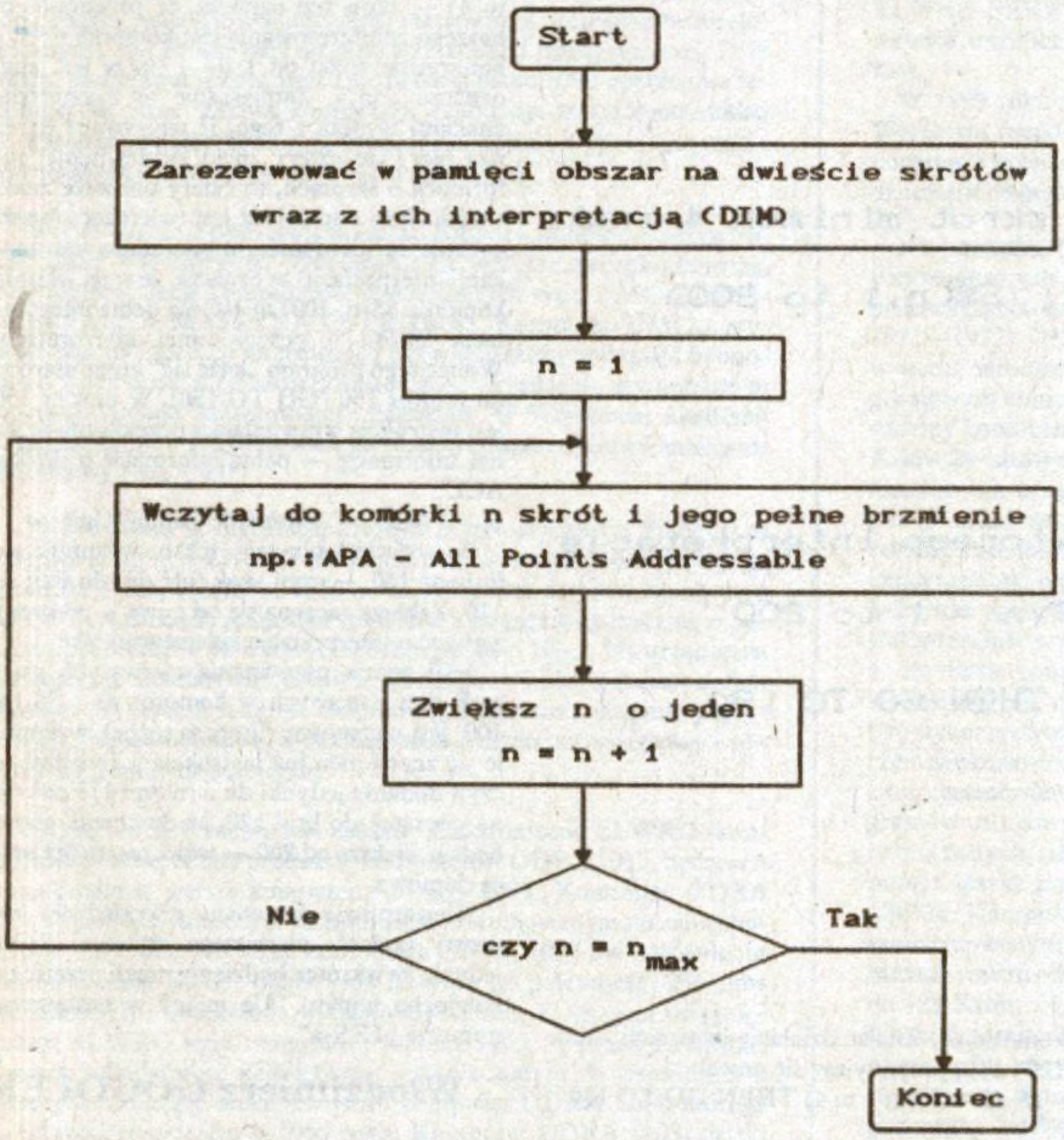
PRINT ACU
na ekranie ukaże się nam pełna interpretacja tego skrótu.

Wróćmy do algorytmu. Algorytm naszego dotychczasowego programu wygląda następująco:



Rys. 2

Wygląda to okropnie. Na szczęście można znacznie prościej, czytelniej:



Rys. 3

A jak to będzie wyglądało w postaci pro gramu?

```
10 DIM S$(200,100)

20 FOR n = 1 to 200

30 INPUT "Podaj kolejny skrot minimum 4 znaki" i jego interpretacj

   S$(n,1 to 200)

40 NEXT n

50 STOP
```

Prawda, że prościej. W tym miejscu należy przedstawić kilka słów wyjaśnienia o znaczeniu kolejnych linii naszego nowego programu. Linia 10 DIM S$ (200, 100) to wspomniana wcześniej rezerwacja miejsca w pamięci dla wprowadzanych przez nas informacji — 200 skrótów (każdy może zajmować do 100 znaków). Kolejne linie naszego programu tworzą tak zwaną **pętlę**. Jej wykonanie polega na wielokrotnej realizacji czynności zapisanych między instrukcjami FOR i NEXT dla wypadków gdy n=1, n=2, n=3 aż do n=200 — czyli 200 razy. Komputer wykonując instrukcję zawartą w linii nr 20 ustala wartość n, początkowo wynosi ona jeden, i „idzie" dalej do linii 30. Jej wykonanie powoduje wyświetlenie na ekranie monitora tekstu:

Podaj kolejny skrót (minimum 4 znaki) i jego interpretację: w odpowiedzi piszemy np.:

ACU — Automatic Calling Unit,

tekst ten zostanie zapamiętany w komórce S$ (1, 100). Należy zwrócić uwagę, że skrót ten musi mieć co najmniej cztery znaki. Jeśli jest krótszy, należy uzupełnić go odstępami (o tak zwane „spacje"). Potrzebę tę uzasadnia konstrukcja dalszego ciągu naszego programu. Instrukcja NEXT n z linii nr 40 dodaje jedynkę do n i „na warsztat" dostaje się komórka S$ (2, 100), ponadto sprawdzany jest warunek czy n nie jest większe od 200, jeśli tak, to koniec programu, jeśli nie — program automatycznie wraca (pętla!) do linii 20 i proces wprowadzania danych powtarza się, ale już dotyczy to komórki S$ (2, 100).

Proszę zwrócić uwagę, że pięcioma instrukcjami zastąpiliśmy 249! Efekt pracy obu programów jest ten sam (niemal). Wskazuje to na rangę problemu poprawnego pisania programu. Programy wykonujące tę samą pracę mogą być napisane na dziesiątki sposobów, lepiej, gorzej. Wskazuje to na rangę problemu poprawnego pisania programu — kwalifikacji programisty.

Analizując rozwiązywanie naszego problemu — dokładnie widać, że dane są sprawnie wprowadzane do komputera. Umknęła nam jednak kwestia zasadnicza — po co to wprowadzanie? Aby móc szybko wyszukiwać wprowadzone w ten sposób informacje. W związku z tym nasz pięcioliniowy program powinniśmy uzupełnić o fragment (modół, część), którego wykonanie przez komputer powoduje, iż maszyna zapyta nas — użytkowników, jaki skrót chcemy rozszyfrować. Modół ten powinien także oczywiście wyszukiwać w pamięci owo rozszyfrowanie — pełną

INFORMATYKA W SZKOLE

interpretację podanego przez nas skrótu. Zatem nasz program, wraz z uzupełnieniem będzie wyglądał następująco:

```
10 DIM S$(200,100)
20 FOR n = 1 to 200
 30 INPUT "Podaj kolejny skrot minimum 4 znaki
" i jego interpretacje:"; S$(n,1 to 200)
 40 NEXT n
 50 STOP
110 INPUT "Podaj skrot, ktorego interpretacje
poszukujesz"; a$ 120 FOR n = 1 to 200
130 IF a$ = S$(n,1 to 4) THEN GO TO 150
140 NEXT n
150 PRINT S$(n,1 to 100)
160 GO TO 110
```

Gdy uruchomimy naszą nową część programu (należy napisać GO TO 110) instrukcja z linii 110 wypisze na ekranie tekst:

Podaj skrót, którego interpretację poszukujesz: i czeka na nasze pytanie. Podajemy je pisząc skrót, np.: ACU. Zostaje on wpisany do komórki a$. Znając działanie instrukcji z linii 120 i 140, przyjrzyjmy się nowej:

IF a$ = S$(n, 1 to 4) THEN GO TO 150

IF to po prostu pytanie „czy? (jeśli?)" zawartość komórki a$ (a tam jest nasze pytanie: skrót ACU) jest identyczna z zawartością pierwszych czterech znaków komórki S$ (n, 1 to 4) — zapis ten oznacza, że przedmiotem naszego zainteresowania jest komórka nr n, a konkretnie znaki od 1 do 4 (to w jęz. ang. oznacza „do"). Zamieszanie to z czterema znakami wynika z tego, iż wpisywane przez nas pierwsze cztery znaki każdej pełnej informacji o skrótach, to cztery pierwsze znaki skrótu. Jeśli odpowiedź jest twierdząca (znaki pytania są identyczne z początkowymi znakami interpretacji) to oznacza, że w tej właśnie komórce S$(n, 100) mieści się pełna interpretacja skrótu o poszukiwanej interpretacji. Wobec tego program „każe iść" komputerowi do linii nr 150 (GO TO 150). W linii nr 150 jest instrukcja wyświetlająca oczekiwaną przez nas informację — pełną informację o skrócie ACU:

ACU — Automatic Calling Unit

Po wyświetleniu tego tekstu wykonuje się linia nr 160 — czyli skok (idź do) do linii nr 110. Zabawa zaczyna się od nowa — możemy zapytać o interpretację kolejnego skrótu.

Jeśli wynik porównania pierwszych czterech liter zawartych w komórce a$ i S$ (n, 100) jest negatywny (litery są różne), wykonuje się znana nam już instrukcja w linii 140 — czyli dodanie jedynki do n (n=n+1) i powrót na początek do linii 120, aż do chwili, gdy n będzie większe od 200 — wówczas, to już praca domowa.

Niewątpliwie omawiany przykład nie jest prosty podczas pierwszego czytania. Sądzę jednak, że wkrótce będziemy mogli przejść do kolejnego tematu. Ale to już w następnym numerze „IKS-a".

Włodzimierz GOGOŁEK

...aż człowiek stworzył komputer (2)

Kontynuując temat podjęty w poprzednim numerze „IKS-a" przedstawimy osiągnięcia polskiej myśli technicznej, polskich twórców komputerów, przedstawimy wreszcie rys historyczny rozwoju informatyki w Polsce.

W 1921 roku po raz pierwszy w Polsce zastosowano maszyny liczące-analityczne do prac statystycznych.

W 1950 roku Zbigniew Pawlak zbudował pierwszą w Polsce cyfrową maszynę liczącą na przekaźnikach, którą zastosowano do celów dydaktycznych. W cztery lata później zbudowano pierwszy eksperymentalny, elektroniczny analizator równań różniczkowych specjalnych.

W 1958 roku zespół konstruktorów Zakładu Aparatów Matematycznych PAN zbudował pierwszy w Polsce komputer cyfrowy pierwszej generacji XYZ. Komputer XYZ był zbudowany na przerzutnikach dynamicznych lampowo-tranzystorowych i zawierał pamięć operacyjną rtęciową o pojemności 1024 słów 18-bitowych. Był modelem laboratoryjnym i wykonywał 800 dodawań na sekundę. W tym samym roku z inicjatywy Zakładu Aparatów Matematycznych Instytutu Matematycznego PAN założono pierwszy w Polsce usługowy ośrodek obliczeniowy na bazie komputera XYZ pod nazwą Biuro Obliczeń i Programów.

W 1959 roku w związku z produkcją sprzętu komputerowego utworzono Wrocławskie Zakłady Elektroniczne, a rok później w Zakładzie Doświadczalnym Instytutu Maszyn Matematycznych PAN zbudowano komputer ZAM 2. Otwierał on serię komputerów wyprodukowanych w kraju i zaliczanych do pierwszej generacji. Komputer ZAM 2 miał pamięć operacyjną o pojemności 1024 słów 18-bitowych, urządzenie wejściowe w postaci czytnika tasiemki dziurkowanej, a niektóre egzemplarze były także wyposażone w czytnik kart dziurkowanych. Urządzenie wyjściowe tworzyły dziurkowane tasiemki oraz pamięć zewnętrzna, zawierająca do 4 jednostek pamięci bębnowej, każda o pojemności 16 Ksłów. Wyniki obliczeń były odtwarzane z tasiemki dziurkowanej na dalekopisie. ZAM 2 pracował z szybkością 1000 dodawań na sekundę. Pracą sterował prosty system operacyjny, a w skład oprogramowania wchodziły dwa języki programowania SAS (skrót od pierwszych liter wyrazu pełnej nazwy: SYSTEM ADRESÓW SYMBOLICZNYCH) i SAKO (SYSTEM

AUTOMATYCZNEGO KODOWANIA), które umożliwiały obliczenia naukowo-techniczne, a nawet proste przetwarzanie danych. W tym samym mniej więcej okresie we Wrocławskich Zakładach Elektronicznych ELWRO zbudowano pierwszy model komputera ODRA 1001.

W 1961 roku opracowano translator dla pierwszego języka programowania SAKO, stworzonego w Instytucie Maszyn Matematycznych w Warszawie. Rok później w zakładach ELWRO we Wrocławiu skonstruowano komputer ODRA 1003, którego seryjna produkcja ruszyła od 1964 roku. Komputer ODRA 1003 był pierwszym małym komputerem (produkowanym do 1965 roku), przeznaczonym do obliczeń naukowo-technicznych i sterowania procesami technologicznymi. Był zbudowany na zasadach techniki tranzystorowej. Przy operacji dodawania pracował z szybkością około 1000 operacji na sekundę. W skład zestawu komputera wchodziły: arytmometr, układ sterowania, pamięć bębnowa, urządzenia wejścia-wyjścia (czytniki i dziurkarka tasiemki papierowej oraz dalekopis).

W 1963 roku na podstawie projektu Katedry Budowy Maszyn Politechniki Warszawskiej, Wrocławskie ELWRO rozpoczęło pierwszą, seryjną produkcję komputerów uniwersalnych typu UMC-1 (UNIWERSALNA MASZYNA CYFROWA). Komputer UMC-1 wykonywał 100 operacji na sekundę i był wyposażony w pamięć bębnową o pojemności 4096 słów 36-bitowych. Urządzenia zewnętrzne to czytnik tasiemki dziurkowanej i dalekopis. W Wojskowej Akademii Technicznej im. J. Dąbrowskiego skonstruowano cyfrowy analizator różnicowy JAGA 63.

W 1964 roku w Instytucie Maszyn Matematycznych PAN opracowano komputer ZAM 21 do obliczeń naukowo-technicznych. Jego produkcję rozpoczęto w 1966 roku. W skład zestawu komputera ZAM 21 wchodziła jednostka centralna z pamięcią operacyjną o pojemności 8 Ksłów 24-bitowych i cyklu pamięci 10 μs. Na urządzenia zewnętrzne składały się: monitor dalekopisowy, czytnik kart dziurkowanych, czytnik tasiemki dziurkowej, dziurkarka tasiemki papierowej, pamięć bębnowa, a później także dziurkarka wierszowa i pamięć na taśmie magnetycznej.

W 1965 roku Wrocławskie Zakłady Elektroniczne ELWRO skonstruowały i rozpoczęły produkcję komputerów ODRA 1013, będących udoskonaloną wersją komputera ODRA 1003. Komputer ODRA 1013 był kolejnym małym komputerem produkowanym po zaniechaniu w 1965 roku produkcji komputera ODRA 1003 i w odróżnieniu od niego miał pamięć operacyjną ferrytową o pojemności 256 słów 40-bitowych i zwiększoną wydajność obliczeń. Kolejnymi osiągnięciami ELWRO były : komputer ODRA 1103, nazwany elektronicznym kalkulatorem dziesiętnym, przeznaczonym do współpracy z maszynami licząco-analitycznymi i komputer ODRA 1204, którego produkcję rozpoczęto w 1968 roku. Komputer ODRA 1103, dzięki możliwości zastosowania operacji mnożenia i dzielenia wykorzystywano w stacjach maszyn licząco-analitycznych do przyśpieszania i usprawnienia obliczeń. Był wyposażony w pamięć operacyjną ferrytową oraz w pamięć bębnową. Szybkość pracy przy operacji dodawania lub odejmowania (liczby 8-cyfrowej) wynosiła 0,18 ms, mnożenia — 2 ms, dzielenia 20 ms. Komputer ODRA 1204 był komputerem drugiej generacji, przeznaczonym głównie do wykonywania obliczeń naukowo-technicznych oraz sterowania procesami technologicznymi w czasie rzeczywistym. W skład zestawu wchodziły: jednostka centralna z pamięcią operacyjną o pojemności od 4 do 64 Ksłów 24-bitowych i cyklu dostępu do pamięci równym 6 ms oraz z możliwością dołączenia do kanałów obsługujących urządzenia zewnętrzne do 8 urządzeń wejścia-wyjścia. Urządzenia zewnętrzne: pamięć bębnowa w postaci do 4 jednostek po 64 Ksłów 24-bitowych każda i czasie dostępu około 10 ms, czytnik tasiemki dziurkowanej, monitor w postaci elektrycznej maszyny do pisania, dziurkarka tasiemki papierowej, drukarka wierszowa. Szybkość działania przy operacji dodawania lub odejmowania liczb stałoprzecinkowych wynosiła 16 ms, a mnożenia — 80 ms.

W 1966 roku opracowano prototyp komputera ZAM 41 — pierwszego komputera do przetwarzania danych, wyprodukowanego w kraju w ramach rodziny komputerów ZAM.

ZAM 41 składał się z jednostki centralnej, zawierającej pamięć operacyjną o pojemności do 20 Ksłów, pamięci zewnętrznej taśmowej i bębnowej, urządzeń wejściowych w postaci czytników tasiemki dziurkowanej i kart dziurkowanych oraz urządzeń wyjściowych w postaci dziurkarki tasiemki papierowej i drukarki wierszowej. Szybkość działania wynosiła około 40 000 dodawań na sekundę, a organizacja komputerowa umożliwiała pracę wieloprogramową. Oprogramowanie pozwalało użytkownikom na korzystanie z następujących języków programowania: COBOL, ALGOL, SAKO, SAS, CEMMA i ZAM-GPSS. Łącznie wyprodukowano i zainstalowano od 1968 roku 16 egzemplarzy komputera ZAM 41.

W 1968 roku rozpoczęto prace koncepcyjne nad stworzeniem rodziny komputerów wspólnej produkcji państw socjalistycznych, nazywanej początkowo rodziną komputerów RIAD, a później rodziną komputerów Jednolitego Systemu. W tym samym roku utworzono Zakład Obsługi Technicznej Maszyn Matematycznych MERA-ELWRO-SERWIS, świadczący usługi w zakresie dostaw sprzętu, serwisu technicznego i oprogramowania oraz szkolenia użytkowników.

W 1969 roku w Zakładach Mechaniki Precyzyjnej w Błoniu pod Warszawą rozpoczęto produkcję drukarek wierszowych (montaż na podstawie licencji) oraz czytników tasiemki dziurkowanej CT-1001 i dziurkarki tasiemki papierowej D-102.

Na początku lat siedemdziesiątych we Wrocławskich Zakładach Elektronicznych ELWRO rozpoczęto produkcję komputerów do przetwarzania danych: ODRA 1304 (1970 rok), ODRA 1325 (lata 1971—1972), ODRA 1305 (1972 rok), pracujących do chwili obecnej w wielu ośrodkach w kraju. Komputer ODRA 1304 jest wieloprogramowym komputerem drugiej generacji produkowanym w ramach rodziny komputerów ODRA z pamięcią operacyjną o pojemności 32 Ksłów 24-bitowych i cyklu pamięci 6 μs. Ma możliwość dołączenia 9 kanałów dla urządzeń zewnętrznych m.in. kanału multiplekserowego. W skład zestawu komputera mogą wchodzić między innymi: czytnik kart dziurkowanych, czytnik tasiemki dziurkowanej, dziurkarka tasiemki papierowej, monitor operatorski, drukarka wierszowa, jednostki pamięci taśmowej. Szybkość operacji dodawania dla liczb stałoprzecinkowych wynosi 26 μs, mnożenia — 96 μs, skoku — 9 μs, a dla liczb zmiennoprzecinkowych szybkość operacji dodawania wynosi 160 μs, mnożenia — 450 μs. Komputer ODRA 1304 może być wykorzystywany zarówno do przetwarzania danych, jak i do obliczeń naukowo-technicznych. Do oprogramowania (zgodnego zresztą z oprogramowaniem komputerów firmy ICL serii 1900) oprócz programów usługowych i standardowych, zarówno dla potrzeb przetwarzania danych, jak i obliczeń naukowo-technicznych, wchodzą następujące języki programowania: PLAN, ALGOL, COBOL, FORTRAN. Komputer ODRA 1325 jest komputerem trzeciej generacji, skonstruowanym na obwodach scalonych TTL i produkowanym w ramach rodziny komputerów ODRA. Pamięć operacyjna wynosi od 8 do 128 Ksłów 24-bitowych, a cykl dostępu równy jest 1 μs. Do jednostki centralnej można podłączyć do 16 kanałów dla urządzeń zewnętrznych, w tym jednego kanału multiplekserowego. W skład zestawu mogą wchodzić wszystkie urządzenia komputera wieloprocesorowego, wieloprogramowego i wielodostępnego. W wypadku liczb stałoprzecinkowych szybkość wykonywania operacji dodawania wynosi 2,6 μs, mnożenia — 12 μs, skoku — 7 μs, a dla liczb zmiennoprzecinkowych szybkość operacji dodawania wynosi 9 μs, a mnożenia — 16 μs. Komputer ODRA 1325 jest przeznaczony do sterowania procesami technologicznymi, do obliczeń naukowo-technicznych i przetwarzania danych. Komputer ODRA 1305 jest także komputerem trzeciej generacji zbudowanym na obwodach scalonych TTL i produkowanym w ramach rodziny komputerów ODRA. Pamięć operacyjna może dochodzić do pojemności 256 Ksłów 24-bitowych, a cykl pamięci wynosi 1 μs. ODRA 1305 jest komputerem wieloprogramowym (może pracować do 16 programów), wielodostępnym (do 60 użytkowników) oraz dwuprocesorowym. Szybkość pracy dla operacji dodawania dwóch liczb stałoprzecinkowych wynosi 1,6 μs, mnożenia — 9 μs, a operacji skoku — 1 μs. Komputer ODRA 1305 jest w pełni zgodny funkcjonalnie i programowo z systemami komputerów serii ICL 1900, stąd można zarówno korzystać z oprogramowania tych komputerów, jak i komputerów ODRA 1304 i ODRA 1325. Jest przeznaczony głównie do przetwarzania danych, obliczeń naukowo-technicznych oraz do sterowania szybkimi procesami zachodzącymi w czasie rzeczywistym.

W 1970 roku w Wojskowej Akademii Technicznej zbudowano pierwszy komputer hybrydowy WAT 1001 oraz grafoskop. W skład elementów analogowych komputera hybrydowego WAT 1001 wchodzą między innymi: wzmacniacze całkujące i sumujące, pamięci analogowe, generatory funkcji nieliniowych, układy mnożące, generatory zmiennych losowych, układ samodzielnego sterowania cyfrowego itp. Komputer WAT 1001 może pracować w trybie pracy ciągłej z ręcznym i automatycznym sterowaniem oraz w trybie pracy powtarzalnej. Wprowadzanie programu odbywa się za pomocą wymiennych tablic programowania oraz klawiatury. Wyniki obliczeń można wy-

...aż człowiek stworzył komputer (2)

prowadzić na ekran monitora, drukarkę rejestratory lub woltomierz. Do układu sterowania można podłączyć komputer cyfrowy, który może sterować zarówno blokami operacyjnymi, jak i programem liczenia. Komputer WAT 1001 jest przeznaczony do syntezy i analizy układów dynamicznych opisywanych równaniami różniczkowymi zwyczajnymi lub cząstkowymi oraz do obliczania układów równości algebraicznych.

W 1970 roku w Zakładach Wytwórczych Przyrządów Pomiarowych ERA w Warszawie opracowano model i prototyp minikomputera K-202. Był to pierwszy w Polsce komputer na układach scalonych. Głównym projektantem i konstruktorem był Jacek Karpiński.

W 1972 roku w Instytucie Maszyn Matematycznych PAN w Warszawie opracowano minikomputer MOMIK 8b. MOMIK 8b jest zbudowany na obwodach scalonych TTL i zawiera pamięć operacyjną o pojemności 2, 4, 8—16 Kstów 8-bitowych i czas cyklu równy 1,8 μs. Pracuje z szybkością od 150 tysięcy do 0,5 miliona operacji na sekundę. Wyposażony jest w kanał multipleksowy i kanał bezpośredniego dostępu. Do kanałów można maksymalnie podłączyć 32 urządzenia zewnętrzne. MOMIK 8b jest wykorzystywany jako jednostka centralna w minikomputerach rodziny minikomputerów MERA 300. W tym samym roku we Wrocławskich Zakładach Elektronicznych ELWRO zmontowano pierwszy w Polsce egzemplarz komputera Jednolitego Systemu o symbolu R-30, produkowany następnie od 1975 roku seryjnie już przez Centrum Komputerowych Systemów Automatyki i Pomiarów MERA ELWRO (dawne Wrocławskie Zakłady Elektroniczne ELWRO) jako komputer IS-1032. Komputer IS-1032 jest przeznaczony do przetwarzania danych oraz rozwiązywania obszernych i skomplikowanych zadań naukowo-technicznych. Pamięć ma pojemność od 128 do 1024 Kbajtów. IS-1032 jest komputerem wieloprogramowym i wielodostępnym. Ma jeden kanał multiplekserowy, umożliwiający podłączenie do 256 urządzeń wejściowo-wyjściowych, 3 kanały selektorowe, umożliwiające podłączenie do kanału 8 jednostek sterujących maksymalnie do 256 urządzeń wejściowo-wyjściowych oraz procesowo. Czas dostępu do pamięci operacyjnej wynosi 0,5 μs, a cykl podstawowy komputera wynosi 300 μs.

W 1973 roku rozpoczęto w Zakładach Wytwórczych Przyrządów Pomiarowych ERA produkcję minikomputera systemu MERA 300 i komputera biurowego MERA 302.

W 1976 roku na 48 Międzynarodowych Targach Poznańskich Centrum Komputerowych Systemów Automatyki i Pomiarów MERA-ELWRO wystawiło przewoźny Ośrodek Obliczeniowy, w którego wyposażeniu znajdował się komputer ODRA 1325, umieszczony w typowym kontenerze.

W 1981 roku powstało i ukonstytuowało się Polskie Towarzystwo Informatyczne.

Dłużej zatrzymaliśmy się na produkcji lat siedemdziesiątych — komputerów serii ODRA, gdyż były one i są nadal udaną konstrukcją sprzętową z bardzo bogatym oprogramowaniem.

A co przyniosły lata osiemdziesiąte — niestety przede wszystkim znaczne pogłębienie się luki technologicznej między krajami przodującymi w dziedzinie techniki komputerowej a Polską. W Polsce brak jest układów dużej skali integracji, mikroprocesorów, pamięci półprzewodnikowych, a w szczególności pamięci dyskowych. Aktualnie w Polsce produkuje się mikroprocesor MCY 7880 oparty na przestarzałym 8-bitowym mikroprocesorze Intel 8080, pamięci zewnętrzne na dyskach elastycznych, choć należy zauważyć, że produkcja ta nie zaspokaja wewnętrznego popytu i nie jest najwyższej jakości. W związku z tymi trudnościami produkcja mikrokomputerów nie odpowiada potrzebom. W ostatnich latach powstało wiele prywatnych zakładów (część z nich powstała w kooperacji z firmami zachodnimi), które zaczęły montować mikrokomputery z elementów produkowanych w innych krajach (najczęściej krajów Dalekiego Wschodu), bądź zajęły się tylko pośrednictwem w sprzedaży zestawów mikrokomputerowych. Z mikrokomputerów produkowanych obecnie czy w najbliższej przyszłości należy wymienić: MERITUM I, MERITUM II, COMPAN 8, MK 4501/2, KRAK-86, cała seria mikrokomputerów ELWRO, czy w końcu MAZOVIA 1016. Aktualnie nie wszystkie z wymienionych mikrokomputerów są produkowane, inne okazały się po prostu niewypałem. Zatrzymajmy się przy mikrokomputerach ELWRO 800 Junior i MAZOVIA 1016.

Model mikrokomputera osobistego ELWRO 800 Junior opracowano, z myślą o zastosowaniu go w polskich szkołach, w Instytucie Automatyki Politechniki Poznańskiej w Poznaniu oraz w Instytucie Komputerowych Systemów Automatyki i Pomiarów we Wrocławiu. ELWRO 800 Junior zawiera: 8-bitowy mikroprocesor Z80 A lub jego odpowiednik U880 produkowany w NRD, stronicowaną pamięć operacyjną, kolorową grafikę (obraz może być 16-kolorowy i może migotać), prosty generator dźwięku, pakiet lokalnej sieci komputerowej, sterownik pamięci na dyskietkach, łącze magnetofonowe oraz dodatkowe układy wejścia-wyjścia. Format zapisu informacji na taśmie i adresy urządzeń zewnętrznych są identyczne z mikrokomputerem ZX Spectrum, co zapewnia ELWRO 800 Junior dostęp do obszernej biblioteki oprogramowania. ELWRO 800 Junior jest wyposażony w dyskowy system operacyjny CP08, w pełni zgodny z systemem CP/M wersja 2.2. Można więc używać wielu programów narzędziowych i użytkowych. Do ELWRO 800 Junior przez wbudowane łącze równoległe o standardzie Centronics i szeregowe RS-232 C można podłączyć drukarkę znakową lub graficzną, manipulator drążkowy (joystick), pióro świetlne, myszkę i wiele innych urządzeń. Do końca 1986 roku planowano wyprodukować 500 sztuk tego komputera, do końca 1987 roku 5000 sztuk, a produkcja docelowa (tylko kiedy to nastąpi) miała wynosić 30 000 sztuk rocznie.

Drugim mikrokomputerem osobistym, na który warto zwrócić uwagę, jest długo oczekiwany, a montowany przez spółkę z ograniczoną odpowiedzialnością „Mikrokomputery" w pełni zgodny z IBM PC/XT mikrokomputer MAZOVIA 1016. MAZOVIA 1016 ma 16-bitowy mikroprocesor INTEL 8086 lub jego odpowiednik K1810WM86 produkowany w ZSRR, pamięć operacyjną o pojemności 256 Kbajtów (może być rozszerzona do 640 Kbajtów), pamięć stałą ROM o wielkości 48 Kbajtów. Podstawowym dyskowym systemem operacyjnym jest spolszczony MS DOS. Twórcy MAZOVII 1016 przystosowują do potrzeb polskiego użytkownika również inne systemy operacyjne jak: CP/M86 czy XENIX. Poza tym do sprzedaży przygotowano spolszczone oprogramowanie typu: bazy danych, programy kalkulacyjne, edytory tekstów itp. MAZOVIA 1016 ma tworzyć podstawowe wyposażenie wyższych uczelni oraz instytutów naukowo-badawczych. Do końca 1986 roku planowano wyprodukować 500 sztuk tych komputerów, a w 1987 roku około 5000 sztuk. Do chwili obecnej produkcji nie uruchomiono.

Należy zauważyć, że zarówno konstrukcja ELWRO 800 Junior, jak i MAZOVII 1016 są już przestarzałe w porównaniu ze sprzętem tej klasy produkowanym w przodujących w dziedzinie techniki komputerowej krajach świata.

J. S.

Komputerowe piractwo — niebezpieczeństwo XXI wieku

Jerzy RAJCH

Okazuje się, iż komputer — spełniający tak wiele pożytecznych czynności — może być też narzędziem przestępstwa. I to nie przestępstwa drobnego czy pospolitego, ale takiego, o którym przed paroma laty nikomu się nawet nie śniło. To wielki problem dalszego rozwoju komputeryzacji, który może zaważyć na obliczu wielu istotnych spraw świata w XXI wieku.

Przed paroma miesiącami niezwykle zaszokowany poczuł się premier Belgii, Wilfried Martens. Za pośrednictwem prasy dowiedział się bowiem, że w pamięci „Bistela", czyli rządowego systemu informacji komputerowej — od Belgian Information System by Telephon — zupełnie swobodnie porusza się komputerowy pirat.

Prasa nazwała pirata tajemniczym „X". Wyposażony w normalny komputer osobisty łatwo złamał wszystkie systemy zabezpieczające. Kluczem do tego były inicjały premiera: W. M. Dzięki temu mógł śledzić na ekranie wymianę korespondencji między członkami rządu, miał dostęp do najświeższych wiadomości agencyjnych, notowań giełdowych, prognoz gospodarczych i informacji dotyczących sytuacji ekonomicznej kraju. Ów „X" wprowadził też do systemu pewne poprawki blokujące tym dostęp do informacji niektórym ministrom. Na marginesie, oni sami tego wcale nie zauważyli! Być może sprawa nadal pozostałaby nie odkryta, gdyby pirat się nie znudził i swych możliwości nie zademonstrował dziennikarzom.

Premier Martens wprawdzie zapewnił, że komputer w pamięci nie posiadał żadnych tajnych informacji, ale opinia publiczna wcale nie była przekonana, czy bystry posiadacz komputera nie wszedł w posiadanie jakiejś tajemnicy państwowej. Rządowi specjaliści skwapliwie zapowiedzieli skonstruowanie takiego systemu zabezpieczającego, który uniemożliwi podobną sytuację. Widać zapomnieli, że analogiczny przypadek wydarzył się trzy lata wcześniej i wówczas też zapewniano o nowym niezawodnym zabezpieczeniu...

Dużo poważniejsze następstwa miało błahe z pozoru wydarzenie z lipca 1986 roku. Wówczas do komputera-matki firmy Philips za pomocą tajnego kodu „włamał się" ktoś nieupoważniony. Firma, by rozwikłać zagadkę, kto i po co to uczynił, postanowiła przeprowadzić śledztwo. Okazało się, że piraci nie zmienili programu ani części jego danych, ale zaatakowali cały system i do ich urządzeń wprowadzili wiele tajnych programów, które nie tylko modyfikowały programy Philipsa, ale mogły też je niszczyć.

W takiej sytuacji sprawę przejęła brygada finansowa paryskiej policji. Szybko odkryła piętnastocyfrowy kod piratów. Jego pięć pierwszych cyfr wskazywało, że pochodzą oni z RFN. Z kolei policja zachodnioniemiecka stwierdziła, iż podany kod identyfikacyjny należy do... uniwersytetu w Heidelbergu. Zaś władze uczelni nic nie wiedziały o takim wykorzystaniu ich komputera.

Ale i piraci popełnili błąd. Porozumiewając się za pomocą komputera między sobą użyli skrótu CERN. A to oznacza Centrum Badań i Poszukiwań Nuklearnych z siedzibą w Genewie. Prowadzący śledztwo poszli tym śladem i udali się do Szwajcarii. Tam czekała na nich niespodzianka. Fizycy pracujący w centrum wyjawili, że kłopoty z piratami mają już od ponad roku. A ich komputery Vax są połączone z 300 laboratoriami na całym świecie.

Stało się jasne, że piraci uzyskali dostęp do ścisłych tajemnic zachodniego świata nauk fizycznych. Obawy te potwierdził Interpol. Wreszcie przestali być tajemnicą. Tym „komputerowym postrachem" okazali się być członkowie oficjalnego klubu CCC — Chaos Computer Club, z siedzibą w Hamburgu i skupiającego prawie stu ludzi. W swym programie deklarują — ich status złożony jest i zarejestrowany w sądzie — że w informatycznym świecie panuje olbrzymi chaos w zakresie bezpieczeństwa. By tę tezę udowodnić, rozpracowywali najbardziej zawiłe programy.

Tu zaczęły się kłopoty i dyskusje między zachodnioniemieckim a francuskim wymiarem sprawiedliwości. W RFN nie istniały jeszcze przepisy prawne karzące za tego typu przestępstwa informatyczne. We Francji też nie, ale policja zakwalifikowała te czyny jako... włamanie i zniszczenie. Nie mogąc się dogadać Francuzi kontynuowali śledztwo na terenie swego kraju. W krótkim czasie wyłapali prawie dwustu piratów grasujących we francuskiej sieci komputerowej.

Tymczasem członkowie Chaos Computer Club rozpoczęli ofensywę i w ciągu pół roku „włamali się" do 135 komputerów ważnych instytucji naukowych — i nie tylko — świata. Między innymi rozszyfrowali w NASA program wahadłowca kosmicznego i dane związane z „gwiezdnymi wojnami".

Ale niemieckie prawodawstwo poszło z duchem czasu. Podczas rewizji w siedzibie klubu znaleziono kilka wręcz wybitnych pirackich programów oraz kilkusetstronicowy spis z kodami i hasłami wejściowymi do olbrzymiej ilości komputerów. Członkowie klubu bronili się, że dopięli tego , co zamierzali. Zwrócili uwagę na luki, które mogą zagrażać całej ludzkości. Bo skoro można zawładnąć w ten sposób tajnymi informacjami czy badaniami lub wpłynąć na wystrzelenie czy lot rakiety jądrowej — to jak tę sytuację inaczej nazwać?

Śledztwo trwa. Nie ma bowiem pewności, jak wielu podobnych, a bardzo zdolnych piratów, będzie chciało pobawić się w cudzym oprogramowaniu. Pewnych systemów zabezpieczających, jak widać, nie ma. To problem do jak najszybszego rozwikłania i przez prawników, i informatyków.

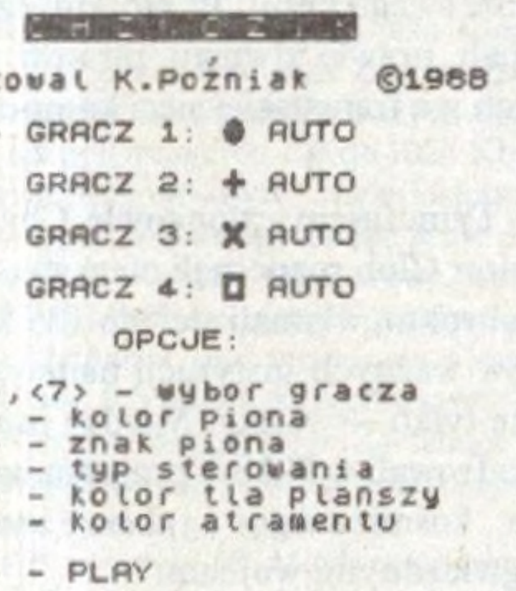
CHIŃCZYK

Krzysztof POŹNIAK

Gra znana jest chyba wszystkim, ale dla porządku przypomnę jej zasady.

W grze bierze udział od 2 do 4 graczy. Zadaniem każdego z nich jest jak najszybsze dotarcie do własnego pola końcowego. O ruchu decyduje rzut kostką. Po rozpoczęciu rozgrywki piony graczy znajdują się w polach startowych. Wyrzucenie kostką 1 lub 6 oczek uprawnia jeden z pionów do wejścia na planszę. Piony poruszają się o tyle pól do przodu, ile oczek wyrzucono kostką. Nie wolno wymijać innych pionów (nawet własnych). Jeżeli ruch zostanie zakończony na polu, na którym stoi pion przeciwnika, to pion przeciwnika uważany jest za zbity i wraca na swoje pole startowe. Po wyrzuceniu 6 oczek i wykonaniu ruchu na planszy (a nie wyjścia z pola startowego) gracz nabywa prawo do powtórnego rzutu.

Po uruchomieniu programu na ekranie ukazuje się plansza atrybutów:



Gracz może wybrać znak, kolor i typ sterowania każdym pionem.

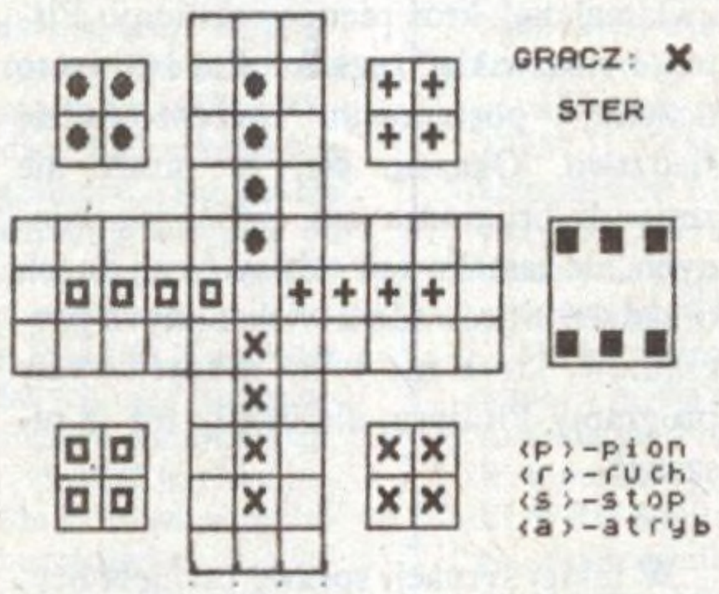
Typy sterowania:

AUTO — gracz automatyczny sterowany przez komputer,

STER — gracz sterowany przez użytkownika z klawiatury,

STOP — gracz wyłączony z rozgrywki.

Uruchomienie gry odbywa się po naciśnięciu klawisza ⟨**p**⟩. Na ekranie ukazuje się plansza gry z początkowym ustawieniem pionów:



Gracze „rzucają" kostką po kolei. Jeśli gracz ma tryb **AUTO**, to posunięcie odbywa się automatycznie, jeśli tryb **STER**, to nad graczem kontrolę przejmuje użytkownik:

1) nad kostką pokazuje się napis: LOSOWANIE. Wciśnięcie dowolnego klawisza zatrzymuje ten proces i wyświetla wylosowaną liczbę oczek;

2) użytkownik w tym momencie może posłużyć się klawiszem: ⟨**p**⟩ — wybór piona (wybrany pion jest wyświetlany w trybie FLASH), ⟨**r**⟩ — wykonanie ruchu wybranym pionem.

UWAGA: klawiszem ⟨**p**⟩ mamy dostęp jedynie do pionów mogących wykonać ruch.

Jeśli żaden pion nie może się ruszyć, to na dole ekranu ukazuje się napis: RUCH NIEMOŻLIWY i gracz traci kolejkę.

Funkcje dodatkowe:

klawisz ⟨**a**⟩ — zmiana atrybutów pionów i planszy. Po zmianie gra jest kontynuowana.

klawisz ⟨**s**⟩ — ostateczne przerwanie gry.

Nie wyjaśnię dokładnie, jak porusza się pion w trybie **AUTO**. Zapewniam jednak, że nie jest to ruch całkowicie losowy (pomijając oczywiście rzut kostką). Ruch każdego piona określany jest pewnym priorytetem. Wybiera się ruch o najwyższym priorytecie. Tylko wtedy, gdy najwyższy priorytet posiada kilka pionów, wybór jednego z nich jest losowy. Zdradzę jedynie, że źródło decyzji ukryte jest w liniach 1200—1260.

```
  1 REM ************************
  2 REM
  3 REM                    ©1988
  4 REM     Krzysztof Poźniak
  5 REM
  6 REM       C H I Ń C Z Y K
  7 REM    gra dla 1 - 4 osob
  8 REM
  9 REM ************************

  10 REM [redefiniowanie znakow]
  12 RESTORE 9900: DIM i(4,8): D
IM k$(7,7): DIM p(2,40): DIM b(4
,2,4): DIM s(4,2,4)
  14 LET ink=7: LET paper=0
  15 INK ink: PAPER paper: BORDE
R paper: CLS
  16 LET k=USR "a": FOR k=k TO k
+31: READ a: POKE k,a: NEXT k
  20 FOR m=1 TO 40: FOR n=1 TO 2
: READ a: LET p(n,m)=a: NEXT n:
NEXT m

  21 FOR m=1 TO 4: FOR n=1 TO 4:
 FOR k=1 TO 2: READ a: LET b(m,k
,n)=a: NEXT k: NEXT n: NEXT m
  22 FOR m=1 TO 4: FOR n=1 TO 4:
 FOR k=1 TO 2: READ a: LET s(m,k
,n)=a: NEXT k: NEXT n: NEXT m
  23 FOR k=1 TO 7: READ a$: LET
k$(k)=a$: NEXT k
  24 FOR k=1 TO 4: LET i(k,3)=7
 LET i(k,4)=CODE "●"+k-1: LET i(
k,5)=10*k-9: NEXT k
  25 LET k=USR "r": FOR k=k TO k
+7: FOR l=0 TO 24 STEP 8: READ a
: POKE k+l,a: NEXT l: NEXT k
  40 REM [wyswietlanie planszy]
  41 FOR k=1 TO 4: LET i(k,5)=0:
 NEXT k
  42 LET stop=0: LET auto=4
  45 GO SUB 8000
 100 REM [petla gry]
 110 LET koniec=0: DIM a(4,4): D
IM d(40): DIM k(4,4): FOR k=1 TO
 4: LET i(k,1)=4: LET i(k,2)=-4:
NEXT k
 112 GO SUB 9800
 115 LET x=1
 120 IF i(x,5)<2 THEN PRINT AT 2
,23;"GRACZ: "; INK i(x,3);CHR$ i
(x,4);AT 4,25; INK ink;("AUTO" A
ND NOT i(x,5));("STER" AND i(x,5
))
 125 LET w=0: GO SUB 9000: LET r
uch=0
 130 IF i(x,5)=0 THEN GO SUB 100
0
 140 IF i(x,5)=1 THEN GO SUB 200
0
 150 IF x$="s" THEN GO TO 40
 160 IF x$="a" THEN LET pamx=x:
GO SUB 8000: GO SUB 9800: LET x=
pamx: GO TO 120
 170 IF ruch THEN GO SUB 2200: I
F w=6 THEN GO TO 125
 180 IF koniec=3-stop THEN GO SU
B 200: GO TO 40
 190 LET x=x+1: IF x>4 THEN GO T
O 115
 195 GO TO 120
 200 REM [koniec rozgrywki]
 210 CLS : PRINT AT 2,8; INVERSE
 1;"C H I N C Z Y K";AT 5,8; FLA
SH 1;"gra  zakonczona"
 220 FOR k=1 TO 4: LET q=i(k,5):
 IF q<=2 THEN GO TO 240
 230 PRINT AT 3+q-1,5;"miejsce "
;q-2;" - gracz : "; INK i(k,3);C
HR$ i(k,4)
 240 NEXT k: PRINT #0;TAB 4; INV
ERSE 1;"NACISNIJ DOWOLNY KLAWISZ
": PAUSE 1: PAUSE 0: RETURN
1000 REM [losowanie i ruch]
1005 LET x$=INKEY$: IF x$="s" OR
 x$="a" THEN RETURN
1010 LET w=INT (6*RND)+1: GO SUB
 9000
1020 GO SUB 8300: IF NOT ruch TH

EN GO TO 2400
1030 GO SUB 1100
1040 LET k=INT (4*RND)+1: GO TO
2150
1100 REM [sprawdzanie ruchu]
1110 LET s=1
1120 FOR q=1 TO 4: IF i(q,8)>0 T
HEN GO SUB 1200
1130 NEXT q
1140 FOR q=1 TO 4: IF i(q,8)<s T
HEN LET i(q,8)=0
1150 NEXT q: RETURN
1200 REM [ruch jednego piona]
1205 LET i(q,8)=1: LET e=i(q,7)
AND (i(q,7)<>i(x,6))
1210 IF e>0 THEN IF d(e)=0 THEN
LET e1=a(x,q): IF (x=1 AND e1>=2
1) OR (x=2 AND NOT (e1>=11 AND e
1<=30)) OR (x=3 AND e1<=20) OR (
x=4 AND e1>=11 AND e1<=30) THEN
LET i(q,8)=2: GO TO 1250
1212 IF e>0 THEN IF d(e)<>0 THEN

  IF i(d(e),2)<i(x,2) THEN IF e<
>1 AND e<>11 AND e<>21 AND e<>31
 THEN LET i(q,8)=3: GO TO 1250
1214 IF e>0 THEN IF d(e)<>0 THEN
 IF i(d(e),2)>=i(x,2) THEN LET i
(q,8)=4+(i(d(e),2)>=i(x,2)+1): G
O TO 1250
1216 IF e<0 THEN IF a(x,q)>0 THE
```

```
N LET i(q,8)=4: GO TO 1250
1218 IF e=0 THEN IF d(i(x,6))=0
AND w=1 THEN LET i(q,8)=3: GO TO
 1250
1220 IF e=0 THEN IF d(i(x,6))>0
AND w=1 THEN IF i(d(i(x,6)),2)<i
(x,2) THEN LET i(q,8)=4: GO TO 1
250
1222 IF e=0 THEN IF d(i(x,6))>0
AND w=1 THEN IF i(d(i(x,6)),2)>=
i(x,2) THEN LET i(q,8)=4: GO TO
1250
1224 IF e=0 THEN IF d(i(x,6))>0
AND w=6 THEN IF i(d(i(x,6)),2)>=
i(x,2) THEN LET i(q,8)=3: GO TO
1250
1226 IF e>0 AND w=6 THEN LET i(q
,8)=2: GO TO 1250
1228 IF e<0 THEN IF a(x,q)<0 THE
N LET i(q,8)=3: GO TO 1250
1250 IF i(q,8)>s THEN LET s=i(q,
8)
1260 RETURN
2000 REM [illegible]
2010 GO SUB 8200
2020 IF x$="s" OR x$="a" THEN RE
TURN
2030 GO SUB 9000
2035 GO SUB 8300: IF NOT ruch TH
EN GO TO 2400
2038 LET k=4: GO SUB 2120
2040 LET x$=INKEY$: IF x$="" THE
N GO TO 2040
2045 BEEP .05,20
2050 IF x$="s" OR x$="a" THEN RE
TURN

2052 IF x$="p" THEN GO SUB 2100

2054 IF x$="r" THEN RETURN
2090 GO TO 2040
2100 REM [illegible]
2110 GO SUB 8400
2120 GO SUB 2150
2130 GO TO 8410
2150 REM [illegible]
2152 LET k=k+1-4*(k=4)
2154 IF i(k,8)=0 THEN GO TO 2152
2156 RETURN
2200 REM [illegible]
2210 IF a(x,k)=0 THEN LET e=d(i(
k,7)): GO TO 2350
2220 IF a(x,k)>0 THEN LET d(a(x,
k))=0
2230 IF a(x,k)<0 THEN LET k(x,-a
(x,k))=0
2240 FOR l=1 TO w: GO SUB 8420:
GO SUB 8370: GO SUB 8430: NEXT l
2260 IF a(x,k)>0 THEN LET e=d(a(
x,k)): LET d(a(x,k))=x: IF e<>0
THEN GO SUB 2300
2270 IF a(x,k)<0 THEN LET k(x,-a
(x,k))=1: IF a(x,k)=i(x,2) THEN
LET i(x,2)=i(x,2)+1: IF NOT i(x,
2) THEN LET koniec=koniec+1: LET
 i(x,5)=koniec+2
2290 RETURN
2300 REM [illegible]
2310 LET i(e,1)=i(e,1)+1: LET pa
mx=x: LET pamk=k
2320 FOR s=1 TO 4: IF a(e,s)=a(x
,k) THEN LET x=e: LET k=s: LET a
(x,k)=0: GO SUB 8430: LET x=pamx
: LET k=pamk: RETURN
2330 NEXT s: STOP
2350 REM [illegible]
2352 LET w=1: GO SUB 8420: LET a
(x,k)=i(x,6): LET i(x,1)=i(x,1)-
1: LET d(a(x,k))=x: GO SUB 8430:
 IF e<>0 THEN GO SUB 2300
2354 RETURN
2400 REM [illegible]

2410 PRINT #0;AT 1,8; INK ink; P
APER paper; FLASH 1;"RUCH NIEMOZ
LIWY": BEEP .03,20: PAUSE 50: PR
INT #0;AT 1,8;"                ":
 RETURN
8000 REM [illegible]
8010 PAPER paper: INK ink: BORDE
R paper: CLS : PRINT TAB 8; INVE
RSE 1;"C H I N C Z Y K"
8015 PRINT AT 1,16;",";AT 2,1;"
opracowal K.Pozniak    ©1988"
8020 FOR x=1 TO 4: GO SUB 8100:
NEXT x: LET x=1
8030 PRINT AT 12,12;"OPCJE:"
8032 PRINT AT 14,4;"<6>,<7> - wy
bor gracza";TAB 4;"<k> - kolor p
iona";TAB 4;"<z> - znak piona";T
AB 4;"<s> - typ sterowania";TAB
4;"<t> - kolor tla planszy";TAB
4;"<a> - kolor atramentu";TAB
4;"<p> - PLAY"
8035 PRINT AT 2*x+2,6;">"
8040 LET x$=INKEY$: IF x$="" THE
N GO TO 8040
8045 BEEP .05,20
8050 IF x$="6" THEN PRINT AT 2*x
+2,6;" ": LET x=x+1-4*(x=4): GO
TO 8035
8051 IF x$="7" THEN PRINT AT 2*x
+2,6;" ": LET x=x-1+4*(x=1): GO
TO 8035
8052 IF x$="k" THEN LET i(x,3)=i
(x,3)+1-8*(i(x,3)=7): GO SUB 810
0: GO TO 8040
8053 IF x$="z" THEN LET i(x,4)=i
(x,4)+1-4*(CHR$ i(x,4)="□"): GO
SUB 8100: GO TO 8040
8054 IF x$="s" THEN LET q=i(x,5)
: LET stop=stop-(q=2)+(q=1): LET
 auto=auto-(q=0)+(q=2): LET i(x,
5)=q+1*(q<=2)-3*(q=2): GO SUB 81
00: GO TO 8040
8055 IF x$="t" THEN LET paper=pa
per+1-8*(paper=7): GO TO 8010

8056 IF x$="a" THEN LET ink=ink+
1-8*(ink=7): GO TO 8010
8057 IF x$="p" AND stop>=3 THEN
BEEP .1,20: BEEP .1,30: PRINT #0
;TAB 2; FLASH 1;"TA ROZGRYWKA NI
E MA SENSU !": PAUSE 100: INPUT
""
8058 IF x$="p" AND stop<3 THEN R
ETURN
8060 BEEP .1,30: GO TO 8040
8100 REM [illegible]
8110 PRINT AT 2*x+2,8;"GRACZ ";x
;": "; INK i(x,3);CHR$ i(x,4); I
NK ink;(" AUTO" AND (i(x,5)=0));
(" STER" AND (i(x,5)=1));(" STOP
" AND (i(x,5)>=2))
8120 RETURN
8200 REM [illegible]
8210 PRINT AT 7,23; FLASH 1;"LOS
OWANIE"
8220 LET w=RND: LET x$=INKEY$: I
F x$="" THEN GO TO 8220


8230 LET w=1+INT (6*w): PRINT AT
 7,23;"      ": RETURN
8300 REM [illegible]
8310 FOR k=1 TO 4
8320 IF a(x,k)<i(x,2) THEN GO TO
 8328
8322 IF a(x,k)=0 AND (w=6 OR w=1
) AND d(i(x,6))<>x THEN LET i(k,
7)=i(x,6): GO TO 8330
8324 IF a(x,k)<>0 THEN GO SUB 83
50: IF i(k,7)<>a(x,k) THEN GO TO
 8330
8328 LET i(k,7)=a(x,k): LET i(k,
8)=0: GO TO 8335
8330 LET i(k,8)=1: LET ruch=1
8335 NEXT k: RETURN
8350 REM [illegible]
8352 LET pam=a(x,k): FOR q=1 TO
w: GO SUB 8360: IF a(x,k)<>pam T
HEN NEXT q: LET i(k,7)=a(x,k): L
ET a(x,k)=pam: RETURN
8354 LET i(k,7)=pam: LET a(x,k)=
pam: RETURN
8360 REM [illegible]
8362 GO SUB 8370
8364 IF a(x,k)<0 AND a(x,k)>=-4
THEN IF NOT k(x,-a(x,k)) THEN RE
TURN
8365 IF a(x,k)>0 THEN IF d(a(x,k
))=0 OR (q=w AND d(a(x,k))<>x) T
HEN RETURN
8366 LET a(x,k)=pam: RETURN
8370 REM [illegible]
8372 IF a(x,k)=40 AND x<>1 THEN
LET a(x,k)=1: RETURN
8373 IF a(x,k)+1=i(x,6)+40*(x=1)
 THEN LET a(x,k)=-1: RETURN
8374 IF a(x,k)>0 THEN LET a(x,k)
=a(x,k)+1: RETURN
8375 IF a(x,k)<0 THEN LET a(x,k)
=a(x,k)-1: RETURN
8400 REM [illegible]
8402 LET q=0: LET v=1: GO TO 845
0


8410 REM [illegible]
8412 LET q=1: LET v=1: GO TO 845
0
8420 REM [illegible]
8422 LET q=0: LET v=0: GO TO 845
0
8430 REM [illegible]
8432 LET q=0: LET v=1
8450 REM procedura wykonawcza
8452 IF a(x,k)<0 THEN PRINT AT b
(x,1,-a(x,k)),b(x,2,-a(x,k)); IN
VERSE NOT v; INK i(x,3); FLASH q
;CHR$ i(x,4)
8454 IF a(x,k)=0 THEN PRINT AT s
(x,1,k),s(x,2,k); INK i(x,3); FL
ASH q;(" " AND NOT v);(CHR$ i(x,
4) AND v)
8456 IF a(x,k)>0 THEN PRINT AT p
(1,a(x,k)),p(2,a(x,k)); INK i(x,
3); FLASH q;(" " AND NOT v);(CHR
$ i(x,4) AND v)
8458 RETURN

9000 REM [illegible]
9020 PRINT AT 9,25;k$(w+1,1);AT
9,27;k$(w+1,2);AT 9,29;k$(w+1,3)
;AT 11,27;k$(w+1,4);AT 13,25;k$(
w+1,5);AT 13,27;k$(w+1,6);AT 13,
29;k$(w+1,7)
9030 RETURN
9800 REM [illegible]
9805 CLS
9810 FOR k=0 TO 3: PLOT 68,171-1
6*k: DRAW 48,0: NEXT k
9811 FOR k=0 TO 3: PLOT 4,167-16
*k: DRAW 176,0: NEXT k
9812 FOR k=0 TO 2: PLOT 68,43-16
*k: DRAW 48,0: NEXT k
9813 PRINT #0;AT 0,6; INK ink; P
APER paper;"[illegible]"
9814 FOR k=0 TO 3: PLOT 4+k*16,5
9: DRAW 0,48: NEXT k
9815 FOR k=0 TO 3: PLOT 68+16*k,
0: DRAW 0,171: NEXT k
9816 FOR k=0 TO 3: PLOT 132+k*16
,59: DRAW 0,48: NEXT k
9820 FOR k=0 TO 2: PLOT 132+k*16
,123: DRAW 0,32: NEXT k: FOR k=0
 TO 2: PLOT 132,123+k*16: DRAW 3
2,0: NEXT k
9821 FOR k=0 TO 2: PLOT 132+k*16
,11: DRAW 0,32: NEXT k: FOR k=0
TO 2: PLOT 132,11+k*16: DRAW 32,
0: NEXT k
9822 FOR k=0 TO 2: PLOT 20+k*16,
11: DRAW 0,32: NEXT k: FOR k=0 T
O 2: PLOT 20,11+k*16: DRAW 32,0:
 NEXT k
9823 FOR k=0 TO 2: PLOT 20+k*16,
123: DRAW 0,32: NEXT k: FOR k=0
TO 2: PLOT 20,123+k*16: DRAW 32,
0: NEXT k
9830 PLOT 197,106: DRAW 45,0: DR
AW 0,-45: DRAW -45,0: DRAW 0,45
9832 FOR x=1 TO 4: GO SUB 9850:
NEXT x: GO SUB 9840
9833 IF auto=4 THEN PRINT AT 0,2
3; INVERSE 1;"D E M O"
9834 PRINT AT 17,23;"<p>-pion";A
T 18,23;"<r>-ruch";AT 19,23;"<s>
-stop";AT 20,23;"<a>-atryb"
9835 RETURN
9840 REM [illegible]
9841 FOR x=1 TO 4: FOR k=1 TO 4:
 GO SUB 8430: NEXT k: NEXT x
9843 RETURN
9850 REM [illegible]
9851 FOR k=1 TO 4: PRINT AT b(x,
1,k),b(x,2,k); PAPER i(x,3); INK
 paper;CHR$ i(x,4): NEXT k
9852 RETURN
9900 REM [illegible]
9901 DATA BIN 00011000
9902 DATA BIN 01111110
9903 DATA BIN 01111110
9904 DATA BIN 11111111
9905 DATA BIN 11111111
9906 DATA BIN 01111110
9907 DATA BIN 01111110

9908 DATA BIN 00011000
9910 REM [illegible]
9911 DATA BIN 00011000
9912 DATA BIN 00011000
9913 DATA BIN 00011000
9914 DATA BIN 11111111
9915 DATA BIN 11111111
9916 DATA BIN 00011000
9917 DATA BIN 00011000
9918 DATA BIN 00011000
9920 REM [illegible]
9921 DATA BIN 11000011
9922 DATA BIN 11100111
9923 DATA BIN 01111110
9924 DATA BIN 00111100
9925 DATA BIN 00111100
9926 DATA BIN 01111110
9927 DATA BIN 11100111
9928 DATA BIN 11000011
9930 REM [illegible]
9931 DATA BIN 11111111
9932 DATA BIN 11111111

9933 DATA BIN 11000011
9934 DATA BIN 11000011
9935 DATA BIN 11000011
9936 DATA BIN 11000011
9937 DATA BIN 11111111
9938 DATA BIN 11111111
9950 REM [illegible]
9951 DATA 1,11,1,13,3,13,5,13,7,
13,9,13,9,15,9,17,9,19,9,21,11,2
1,13,21,15,19,13,17,13,15,13,13,
15,13,17,13,19,13,21,13,21,11
9952 DATA 21,9,19,9,17,9,15,9,13
,9,13,7,13,5,13,3,13,1,11,1,9,1,
9,3,9,5,9,7,9,9,7,9,5,9,3,9,1,9
9955 REM [illegible]
9956 DATA 3,11,5,11,7,11,9,11
9957 DATA 11,19,11,17,11,15,11,1
3
9958 DATA 19,11,17,11,15,11,13,1
1
9959 DATA 11,3,11,5,11,7,11,9
9960 REM [illegible]


9961 DATA 3,3,3,5,5,3,5,5
9962 DATA 3,17,3,19,5,17,5,19
9963 DATA 17,17,17,19,19,17,19,1
9
9964 DATA 17,3,17,5,19,3,19,5
9970 REM [illegible]
9971 DATA "          "
9972 DATA "    ■     "
9973 DATA "   ■■■    "
9974 DATA "  ■■■■■   "
9975 DATA " ■■■■■■■  "
9976 DATA "■■■■■■■■■ "
9977 DATA "■■■■■■■■■ "
9980 REM [illegible]
9981 DATA BIN 00000000,BIN 00001
000,BIN 00001000,BIN 00001000
9982 DATA BIN 00000000,BIN 00001
000,BIN 00001000,BIN 00001000
9983 DATA BIN 00000000,BIN 00001
000,BIN 00001000,BIN 00001000
9984 DATA BIN 00000000,BIN 00001
000.BIN 00001000.BIN 00001000

9985 DATA BIN 11111111,BIN 00001
111,BIN 11111111,BIN 11111000
9986 DATA BIN 00000000,BIN 00000
000,BIN 00000000,BIN 00000000
9987 DATA BIN 00000000,BIN 00000
000,BIN 00000000,BIN 00000000
9988 DATA BIN 00000000,BIN 00000
000,BIN 00000000,BIN 00000000
9990 REM [illegible]
9991 CLEAR : SAVE "chinczyk" LIN
E 1: VERIFY "chinczyk"
```

Jak płynnie i szybko przesuwać...

Danuta KWASIŻUR, Mieczysław SKONIECZNY

BASIC mikrokomputera AMSTRAD pozwala na „płynny" ruch dowolnych obiektów na ekranie dzięki odpowiedniemu zastosowaniu instrukcji TAG. MOVE oraz właściwie zorganizowanej pętli FOR. To narzędzie satysfakcjonuje użytkownika przy małych wymiarach obiektu (powierzchnia kilku znaków). Im większy obiekt tym satysfakcja coraz mniejsza. „Płynność" okazuje się ruchem skokowym (każdy wiersz jest przesuwany osobno), co daje się zauważyć na ekranie, a i szybkość takiego ruchu pozostawia wiele do życzenia.

Mankamenty te można usunąć stosując odpowiednio sparametryzowane podprogramy w języku wewnętrznym. Chcemy zaprezentować i omówić 4 podprogramy, które przesuwają dowolnie zadeklarowany obiekt w 4 kierunkach:

— do góry (ruch pionowy)
— na dół (ruch pionowy)
— w lewo (ruch poziomy)
— w prawo (ruch poziomy)

Ilustracją zastosowania ww. podprogramów jest program demonstracyjny, który steruje ruchem obiektu dość dużego (6x9 znaków) po ekranie mikrokomputera w 4 kierunkach. Oczywiście wymiar obiektu jest zmienny, co nie ogranicza stosowania podprogramów. Dla czytelników, którzy pokuszą się o wprowadzenie programu demonstracyjnego z klawiatury do pamięci mikrokomputera i zainicjują jego działanie, kształt ruchomego obiektu będzie niespodzianką (patrz instrukcje 50—150)

W wierszu 40 określono wymiary ruchomego obiektu graficznego:

X1 — szerokość obiektu (w znakach)
Y1 — wysokość obiektu (w znakach)

W naszym wypadku X1=9, Y1=6, a więc obiekt ma powierzchnię 54 znaków. Deklarowany obszar, który zostanie przesunięty, jest traktowany jak prostokąt. Poszczególne fragmenty programu realizują następujące czynności:

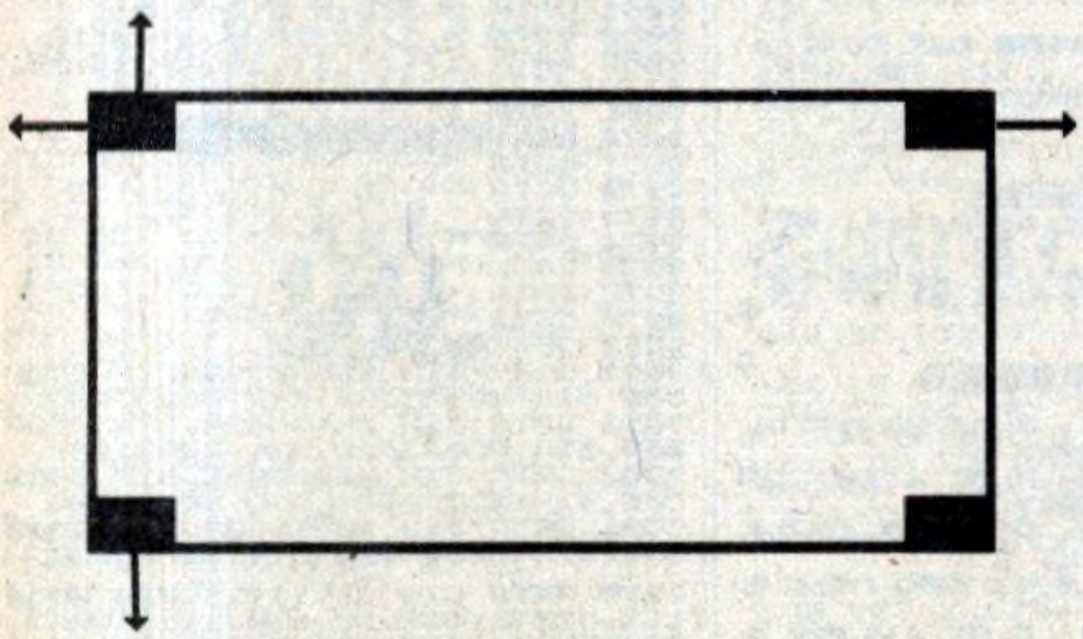
210—270 — wywołanie podprogramu na ruch poziomy w prawo. Zmienne x,y w wierszu 240 określają współrzędne znakowe prawego, górnego rogu obiektu.

280—360 — wywołanie podprogramu na ruch pionowy w górę. Zmienne x,y w wierszu 310 określają współrzędne znakowe lewego, górnego rogu obiektu.

410—480 — wywołanie podprogramu na ruch poziomy w lewo. Zmienne x,y w wierszu 370 określają współrzędne znakowe lewego, górnego rogu obiektu.

500—580 — wywołanie podprogramu na ruch pionowy w dół. Zmienne x,y w wierszu 390 określają współrzędne znakowe lewego, dolnego rogu obiektu.

Współrzędne, którego rogu należy podać dla odpowiedniego ruchu, pokazuje poniższy rysunek (strzałkami oznaczono kierunek ruchu).



Zmienna L we wszystkich wypadkach określa, o ile znaków należy przesunąć obiekt.

Pomocniczą rolę odgrywa podprogram:

```
10 MODE 1:INK 0,0
20 SPEED INK 20,20:INK 1,26,6:INK 2,14,26
30 PAPER 0:BORDER 0:CLS:PEN 1
40 y1=6:x1=9
50 LOCATE 7,9:PRINT CHR$(213);CHR$(143);CHR$(143);CHR$(215)
60 LOCATE 8,10:PRINT CHR$(213);CHR$(143);CHR$(143);CHR$(215)
70 LOCATE 9,11:PRINT CHR$(213);CHR$(143);CHR$(143);CHR$(215)
80 LOCATE 9,12:PRINT CHR$(214);CHR$(143);CHR$(143);CHR$(212)
90 LOCATE 8,13:PRINT CHR$(214);CHR$(143);CHR$(143);CHR$(212)
100 LOCATE 7,14:PRINT CHR$(214);CHR$(143);CHR$(143);CHR$(212)
110 GRAPHICS PEN 1
120 PLOT 192,222:DRAWR 46,46:PLOT 192,222:DRAWR 46,-46
130 PEN 2:LOCATE 16,12:PRINT"INFORMATYKA"
140 LOCATE 16,13:PRINT"KOMPUTERY"
150 LOCATE 16,14:PRINT"SYSTEMY"
160 FOR t=1 TO 500:NEXT
170 x=7:y=9:l=6:GOSUB 1020
180 GOSUB 410
190 x=1:y=14:l=11:GOSUB 1020
200 GOSUB 500
210 GOSUB 730
220 POKE &7F78,y1:POKE &7F81,2*x1
230 POKE &7F8D,2*x1-1:POKE &7F9A,79+2*x1
240 x=9:y=20:l=31:GOSUB 1020
250 POKE &7F7B,w1:FOR Z=w2+1 TO 2*l
260 FRAME:POKE &7F7A,Z:CALL &7F76
270 FOR t=1 TO 5:NEXT:NEXT
280 GOSUB 590
290 POKE &7F01,y1*8:POKE &7F5A,2*x1
300 POKE &7F60,2*x1
310 x=32:y=20:l=19:GOSUB 1020
320 POKE &7F74,w2:POKE &7F75,w1
330 FOR B=1 TO 1
340 FOR C=1 TO 8
350 POKE &7F02,c:CALL &7F00
360 FOR t=1 TO 2:NEXT:NEXT:FRAME:NEXT
370 x=32:y=1:l=31:GOSUB 1020
380 GOSUB 410
390 X=1:Y=6:L=19:GOSUB 1020:GOSUB 500
400 GOTO 210
410 GOSUB 950
420 POKE &7F78,y1:POKE &7F81,2*x1
430 POKE &7F8D,257-2*x1:POKE &7F9A,81-2*x1
440 POKE &7F7B,w1
450 FOR Z=w2 TO W2-2*l+1 STEP -1
460 POKE &7F7A,Z:CALL &7F76
470 FRAME:FOR t=1 TO 5:NEXT:NEXT
480 RETURN
490 X=1:Y=6:L=19:GOSUB 1020
500 GOSUB 800
510 POKE &7F01,y1*8:POKE &7F5A,2*x1
520 POKE &7F60,2*x1
530 POKE &7F74,W2:POKE &7F75,W1+56
540 FOR B=1 TO L
550 FOR C=1 TO 8
560 POKE &7F02,c:CALL &7F00
570 FOR t=1 TO 2:NEXT:NEXT:FRAME:NEXT
580 RETURN
```

który przelicza współrzędne znakowe x,y na adresy tych znaków. Oczywiście w mikrokomputerze AMSTRAD CPC 6128 adres początkowy ekranu wynosi 49152.

Podprogramy na ruch pionowy

Podprogram na ruch w górę ma następującą postać:

```
590 RESTORE 610:FOR i=&7F00 TO &7F73
600 READ v:POKE i,v:NEXT
610 DATA 1,0,1,42,116,127,16,93
620 DATA 229,124,198,56,103,17,80,0,237,82,84
630 DATA 93,225,197,229,213,205,95,127,209,98,107
640 DATA 34,116,127,225,193,13,40,46,6,7,84,93,124
650 DATA 198,8,103,197,229,213
660 DATA 205,95,127,209,225,193,13,40,26,16,236
670 DATA 84,93,124,214,56,103,213,17,80,0,237,90
680 DATA 209,197,229,213,205,95,127,209,225,193
690 DATA 24,207,54,0,84,93,19,1,0,0,237,176,201
700 DATA 1,0,0,237,176,201,124,214,8,87
710 DATA 93,229,98,107,34,116,127,225,24,187,201
720 RETURN
```

Podprogram na ruch w dół ma następującą postać:

```
730 RESTORE 750:FOR i=&7F76 TO &7FA1
740 READ v:POKE i,v:NEXT
750 DATA 1,8,0,33,39,192,84
760 DATA 93,19,197,1,0,0,237,184,193,35,54,0
770 DATA 13,40,6,17,0,8,25,24,234,5,40,12,124
780 DATA 214,56,103,17,0,0,25,14,8,24,219,201
790 RETURN
```

Jedno wywołanie podprogramu daje efekt na ekranie przesunięcia obiektu o jedną linię (1/8 znaku) odpowiednio w górę lub w dół. Adres początkowy (ustalony po wyjściu z podprogramu 1020) obiektu jest ładowany do adresów & 7F74 i & 7F75 za pomocą instrukcji POKE (wiersze 320 i 530 programu demonstracyjnego). Jak widać w wierszach 800 i 590 oba podprogramy są wymiennie ładowane w ten sam obszar pamięci określony adresami & 7F00 — & 7F73.

Podprogramy na ruch poziomy

Podprogram na ruch w prawo ma postać:

```
800 RESTORE 820:FOR i=&7F00 TO &7F73
810 READ v:POKE i,v:NEXT
820 DATA 1,0,1,42,116,127,16,93,229
830 DATA 124,214,56,103,17,80,0,237,90,84
840 DATA 93,225,197,229,213,205,95,127,209
850 DATA 98,107,34,116,127,225,193,13,40,46
860 DATA 6,7,84,93,124,214,8,103,197,229,213
870 DATA 205,95,127,209,225,193,13,40,26,16,236
880 DATA 84,93,124,198,56,103,213,17,80,0,237,82
890 DATA 209,197,229,213,205,95,127
900 DATA 209,225,193,24,207
910 DATA 54,0,84,93,19,1,0,0,237,176,201
920 DATA 1,0,0,237,176,201,124,198,8,87
930 DATA 93,229,98,107,34,116,127,225,24,187,201
940 RETURN
```

Podprogram na ruch w lewo ma postać:

```
950 RESTORE 970:FOR i=&7F76 TO &7FA1
960 READ v:POKE i,v:NEXT
970 DATA 1,8,0,33,36,192,84,93
980 DATA 27,197,1,0,0,237,176,193,43,54,0
990 DATA 13,40,6,17,0,7,25,24,234,5,40,12,124
1000 DATA 214,56,103,17,0,0,25,14,8,24,219,201
1010 RETURN
```

Jedno wywołanie podprogramu daje efekt na ekranie przesunięcia obiektu o pół znaku odpowiednio w lewo lub w prawo. Adres początkowy jest analogicznie ustalony w podprogramie 1020 i ładowany za pomocą instrukcji POKE pod odpowiednie adresy. Oba podprogramy są wymiennie ładowane w ten sam obszar pamięci (& 7F76 — &7FA1).

Wszystkie opisane wyżej podprogramy przesuwania wykorzystują następujące rejestry:

bc — przechowuje liczbę, określającą wielkość obiektu;
hl — przechowuje adresy obiektu;
de — roboczy do obliczania adresów.

Wywołanie odpowiedniego podprogramu w języku wewnętrznym w programie użytkowym ma postać:

CALL adres startowy

gdzie adres startowy jest początkiem obszaru rozmieszczania podprogramu w pamięci (odpowiednio & 7F00 — ruch pionowy i & 7F76 — ruch poziomy).

Odpowiednie efekty na ekranie otrzymuje się przez umiejętne zorganizowanie pętli FOR wykorzystującej parametry w1 i w2 (otrzymane z podprogramem 1020) oraz L.

```
1020 A=(Y-1)*80+(x-1)*2
1030 W1=INT(a/256)
1040 W2=a-W1*256
1050 w1=192+w1
1060 RETURN
```

Efekt spowolnienia ruchu może użytkownik uzyskać przez odpowiednie zastosowanie pętli FOR w postaci:
FOR i=1 to T: NEXT
gdzie T jest uzależniony od wymogów ruchu.

Na zakończenie życzymy zainteresowanym czytelnikom cierpliwości we wprowadzaniu programów oraz ciekawych efektów przy formowaniu własnych, odpowiednio zdefiniowanych obiektów.

Danuta KWASIŻUR
Mieczysław SKONIECZNY

ZX-81

Rakieta w cel

W rejon bronionego przez ciebie odcinka pola walki nadleciał nieprzyjacielski helikopter. Ustaw więc odpowiednio kąt wyrzutni, np. 40°, 70° itp. i odpal rakietę klawiszem „ENTER". Z tej walki wyjdzie cało tylko jeden, gdyż w wypadku, kiedy nie trafisz w helikopter, zrzuci on na Twoją pozycję bombę.

J. J.

```
  10 PRINT AT 20,27;"▟"
  15 INPUT B
  20 LET C=27*COS(PI*B/180)*COS(
PI*B/180)
  25 FOR A=1 TO 19
  30 GOSUB 100
  35 LET N=27-INT(SQR(A/19)*C)
  37 A=1 THEN GOTO 50
  40 PRINT AT 21-A,27-INT(SQR((A
-1)/19)*C);" "
  45 PRINT AT 19,27;" "
  50 PRINT AT 20-A,N;"."
  55 IF A=19 AND A=N THEN GOTO 1
20
  60 IF A=19 THEN GOTO 70
  65 NEXT A
  70 FOR A=20 TO 27
  75 GOSUB 100
76 IF A=27 THEN GOTO 130
80 NEXT A
  100 PRINT AT 0,A-2;"   "; AT 1
,A-3;"   "
  105 PRINT AT 0,A-1;" - "; AT 1
,A-2;"--▣"
  110 PRINT AT 0,A-1;"-▟-"; AT 1
,A-2;"*-▣"
  115 RETURN
  120 PRINT AT 0,18;"   "
  125 FOR A=1 TO 20
  127 PRINT AT A-1,17;"   "; AT
 A,17;"***"
  128 IF A=20 THEN GOTO 140
  129 NEXT A
  130 FOR A=3 TO 20
  132 PRINT AT A-1,27;" "; AT A,
27;"*"
  134 IF A=20 THEN STOP
  135 NEXT A
  140 PRINT AT 5,10;"RAKIETA TRA
FILA"
```

PROGRAM 64

ZX-81

POLE MINOWE

Twoim zadaniem jest przejście przez pole minowe do punktu oznaczonego na ekranie literą „X". Po każdym ruchu gwiazdki wskutek drgań, następuje eksplozja dziesięciu min oznaczonych literą „M". Jeżeli któraś z min wybuchnie w miejscu znajdowania się gwiazdki, następuje zakończenie gry, gdy nie, możesz zabawę kontynuować. Ruch gwiazdce nadaje się za pomocą klawiszy: 5, 6, 7, 8. Gdy przestanie zadowalać cię stopień trudności, możesz zmienić ilość wybuchających min. Dokonuje się tego przez zwiększenie lub zmniejszenie cyfry 10 w wierszu o etykiecie 25.

J. J.

```
  1 REM ****************************
  2 REM *      POLE MINOWE         *
  3 REM *          ZX-81           *
  4 REM ****************************
  5 LET A=7
  10 LET B=7
  15 PRINT AT 7,14;"X"
  20 PRINT AT A,B;"*"
  25 FOR I=1 TO 10
  30 LET X=INT(RND*14+1)
  35 LET Y=INT(RND*14+1)
  40 IF (X=7 AND Y=14) OR X=7 A
ND Y=0) THEN GOTO 30
  45 PRINT AT X,Y;"M"
  50 IF X=A AND Y=B THEN GOTO 1
10
  55 NEXT I
  60 INPUT A$
  65 PRINT AT A,B;" "
  70 IF A$="8" THEN LET B=B+1
  75 IF A$="5" THEN LET B=B-1
  80 IF A$="6" THEN LET A=A+1
  85 IF A$="7" THEN LET A=A-1
  90 IF A=7 AND B=14 THEN GOTO
100
  95 GOTO 20
  100 PRINT "WYGRALES"
  105 STOP
  110 PRINT "JUZ PO TOBIE !"
```

ATARI

Drugie prawo dynamiki

J. GOLLA

```
0 REM *** DRUGIE PRAWO DYNAMIKI
 ***
1 DIM A$(200):A$="-------------
-------"
2 A$(21,100)="....DRUGA ZASADA..
.......DYNAMIKI.....      ------
------"

6 GRAPHICS 2:POKE 752,1:POKE 710
,0:POSITION 0,3
7 FOR T=1 TO 76:SOUND 0,RND(0)*5
0,12,15:? #6;A$(T,T);:SOUND 0,0,
0,0
8 FOR C=1 TO 1500 STEP T:NEXT C:
NEXT T
9 ? "      NACISNIJ DOWOLNY KLAW
ISZ"
10 OPEN #1,4,0,"K:":GET #1,S
20 GRAPHICS 1+16:? #6;"    KLAWI
SZ NR 1     CZESC TEORETYCZNA":?
 #6;" "
30 ? #6;"    KLAWISZ NR 2     CZ
ESC SYMULACYJNA":TRAP 20
40 GET #1,S:S=S-48:ON S GOTO 100
,212
50 GOTO 40

100 REM ________________
105 REM CZESC TEORETYCZNA
110 REM ________________
115 GRAPHICS 0:POKE 710,0:POKE 8
2,1:POKE 752,1:POKE 709,0
120 ? CHR$(125):? CHR$(127);"  D
RUGA ZASADA DYNAMIKI";CHR$(29)
125 ? "Jezeli na cialo dziala st
ala sila,to"
130 ? "porusza sie ono wzgledem
inercjalnego  ukladu";
135 ? " odniesienia ruchem jedno
stajnie zmiennym z przyspie";
140 ? "szeniem wprost pro- porcj
onalnym do sily,";
145 ? "a odwrotnie pro-  porcjon
alnym do masy ciala."
150 ? "Kierunek i zwrot wektora
przy";
155 ? "spieszeniasa zgodne z kie
runkiem i";:? " zwrotem wekto-";
160 ? "ra sily.":? :? :? :? CHR$
(127);"NACISNIJ DOWOLNY KLAWISZ"
165 FOR T=0 TO 10 STEP 0.3:POKE
709,T:NEXT T:GET #1,S
170 ? CHR$(125):? CHR$(127);" DR
UGA ZASADA DYNAMIKI"
175 ? CHR$(127);"SILA=MASA*PRZYS
PIESZENIE"
176 ? CHR$(127);CHR$(127);"_   _
"
180 ? CHR$(127);CHR$(127);"F=m*a
";CHR$(29)
185 ? CHR$(127);"PRZYSPIESZENIE=
SILA/MASA";CHR$(29)
186 ? CHR$(127);CHR$(127);"_ _"
190 ? CHR$(127);CHR$(127);"a=F/m
";CHR$(29)
195 ? "Przebyta DROGA jest wpros
t proporcjo-  nalna do";
200 ? " przyspieszenia w stalych
 prze-dzialach czasowych."
205 ? :? :? :? :?
210 ? CHR$(127);"NACISNIJ DOWOLN
Y KLAWISZ":GET #1,S
212 GRAPHICS 2
215 ? #6;"przyspieszenie jest
 proporcjonalne          do sily"
:REM Tekst napisz w INVERSE VIDE
O
220 ? #6;"PRZYSPIESZENIE JEST OD
```

```
WROTNIE PROPORCJO-    NALNE DO M
ASY"
225 POKE 710,0:POKE 752,1
230 ? "   DOWOLNY KLAWISZ:WYBOR
SYMULACJI."
235 ? CHR$(127);"RETURN TO START
 SYMULACJI."
240 GET #1,S:IF S=155 THEN ON A
GOTO 1000,2000
245 IF L=0 THEN POSITION 2,7:? #
6;"masa jest stala":REM Tekst na
pisz w INVERSE VIDEO:A=2:GOSUB 2
70
250 L=-1-L
255 IF L=0 THEN POSITION 2,7:? #
6;"SILA JEST STALA":A=1:GOSUB 26
5
260 GOTO 240
265 POKE 708,15:POKE 711,3:RETUR
N
270 POKE 711,15:POKE 708,3:RETUR
N
1000 REM ________________
1010 REM SILA JEST STALA
1020 REM ________________
1040 GRAPHICS 0:POKE 710,0:POSIT
ION 12,2:? "SILA JEST STALA";CHR
$(253)
1050 POSITION 14,4:? "F1=F2=F3=1
N":IF X=1 THEN X=0:GOTO 1400
1060 POSITION 2,20:? "MASA WOZKO
W MOZE PRZYJMOWAC"
1070 ? "WARTOSC 1,2,3,4 lub 5 kg
"
1080 POSITION 2,15:? "PODAJ MASE
 NR 1 ?";CHR$(253);:GET #1,M1
1090 M1=M1-48:IF M1>5 OR M1<1 TH
EN ? :? " ZLA WARTOSC":GOTO 1080
1100 POSITION 10,8:? "M1=";M1;"k
g"
1110 POSITION 2,16:? "PODAJ MASE
 NR 2 ?";CHR$(253);:GET #1,M2

1120 M2=M2-48:IF M2>5 OR M2<1 TH
EN ? :? " ZLA WARTOSC":GOTO 1110
1130 POSITION 17,8:? "M2=";M2;"k
g"
1140 POSITION 2,17:? "PODAJ MASE
 NR 3 ?";CHR$(253);:GET #1,M3
1150 M3=M3-48:IF M3>5 OR M3<1 TH
EN ? :? " ZLA WARTOSC":GOTO 1140
1160 POSITION 25,8:? "M3=";M3;"k
g":? CHR$(29);CHR$(29)
1170 ? "CZY MASY SA PODANE PRAWI
DLOWO(T/N) ?";CHR$(253);:GET #1,
S
1180 IF S=84 THEN GOTO 1200
1190 GOTO 1040
1200 H=0:IF M1<=M2 AND M1<=M3 TH
EN M=M1:GOTO 1220
1210 M=M3:IF M2<=M3 THEN M=M2
1220 GRAPHICS 7:SETCOLOR 4,4,4
1230 COLOR 1:SETCOLOR 0,1,0:PLOT
 1,10
1240 DRAWTO 159,10:PLOT 1,40:DRA
WTO 159,40:PLOT 1,70:DRAWTO 159,
70
1250 FOR T=18 TO 159 STEP 14.1:P
LOT T,11:PLOT T,41:PLOT T,71:NEX
T T
1260 PP=PEEK(106)-32:POKE 106,PP
-32:POKE 54279,PP
1270 ? CHR$(127);"PROSZE POCZEKA
C 10 SEKUND!":POKE 559,62
1280 PL=PP*256+1024:FOR I=PL TO
PL+1024:POKE I,0:NEXT I:POKE 532
77,3
1290 POKE 704,8:POKE 705,15:POKE
 706,1
1300 POKE 53256,1:POKE 53257,1:P
OKE 53258,1
1310 POKE 53248,50:POKE 53249,50
:POKE 53250,50
1320 RESTORE 1510:P=PL+51:FOR I=
P TO P-6*M1-5 STEP -1
1321 READ B:POKE I,B:NEXT I
1330 RESTORE 1510:P=PL+367:FOR I
=P TO P-6*M2-5 STEP -1
1331 READ B:POKE I,B:NEXT I
1340 RESTORE 1510:P=PL+683:FOR I
=P TO P-6*M3-5 STEP -1
1341 READ B:POKE I,B:NEXT I:GOSU
B 6000
1350 ? CHR$(28);"KAZDY KLAWISZ T
O START SYMULACJI  "
1360 ? "SPACJA TO WYKONANIE OBLI
CZEN CWICZ.":GET #1,S
1370 IF S=32 THEN X=1:POKE 106,P
P+16:POKE 53277,0:GOTO 1040
1380 H=0:GOSUB 7020:GOTO 1540
1390 REM ________________
1400 REM BLOK OBLICZENIOWY
1405 REM ________________
1410 POKE 53248,0:POKE 53249,0:P
OKE 53250,0
1420 POSITION 8,6:? "m1=";M1;"kg
","m2=";M2;"kg","m3=";M3;"kg":?
:? :?
1450 GOSUB 9000
1455 POSITION 7,14:? "a1=";INT((
1/M1)*100)/100;
1456 ? " a2=";INT((1/M2)*100)/10
0;
1457 ? " a3=";INT((1/M3)*100)/10
0;" [m/s^2]"
1460 POSITION 7,16:? "S1=";(M*10
0)/M1:POSITION 14,16:? " S2=";(M
*100)/M2
1465 POSITION 22,16:? " S3=";(M*
100)/M3
1470 POSITION 30,16:? "       ";
? :?
1480 ? CHR$(127);"KLAWISZ N TO N
OWA SYMULACJA"
1490 ? "    INNY KLAWISZ TO POWR
OT DO MENU"
1495 GET #1,S:IF S=78 THEN GOTO
1040
1500 GOTO 20
1510 DATA 66,66,231,165,231,0,25
5,255,255,255,255,0,255,255,255,
255
1520 DATA 255,0,255,255,255,255,
255,0,255,255,255,255,255,0
1530 DATA 255,255,255,255,255,0,
0,0
1540 ? CHR$(28);CHR$(28);"KAZDY
KLAWISZ ZATRZYMUJE SYMULACJE"
1560 D=H^2
1570 POKE 53248,50+D*(1/M1):POKE
 53249,50+D*(1/M2)
1575 POKE 53250,50+D*(1/M3)
1580 IF (1/M)*D+40>177 THEN 1350
1590 IF PEEK(764)<>255 THEN FOR
T=1 TO 100:NEXT T:POKE 764,255:G
OTO 1350
1600 H=H+0.44:GOTO 1560
1900 REM ________________
1950 REM MASA JEST STALA
1960 REM ________________
2000 GRAPHICS 0:POKE 710,0:POSIT
ION 12,2:? "MASA JEST STALA";CHR
$(253)
2010 POSITION 12,4:? "m1=m2=m3=1
[kg]":IF X=1 THEN X=0:GOTO 3400
2015 POSITION 2,20:? "SILA DZIAL
AJACA NA WOZEK MOZE"
2016 ? "PRZYJMOWAC WARTOSC 0,1,2
,3,4 lub 5[N]"
2020 POSITION 2,15:? "PODAJ SILE
 NR 1 ?";CHR$(253);:GET #1,F1

2022 F1=F1-48:IF F1>5 OR F1<0 TH
EN ? :? " ZLA WARTOSC":GOTO 2020
2025 POSITION 5,8:? "F1=";F1;"[N
]"
2030 POSITION 2,16:? "PODAJ SILE
 NR 2 ?";CHR$(253);:GET #1,F2
2032 F2=F2-48:IF F2>5 OR F2<0 TH
EN ? :? " ZLA WARTOSC":GOTO 2030
2035 POSITION 15,8:? "F2=";F2;"[
N]"
2040 POSITION 2,17:? "PODAJ SILE
 NR 3 ?";CHR$(253);:GET #1,F3
2044 F3=F3-48:IF F3>5 OR F3<0 TH
EN ? :? " ZLA WARTOSC":GOTO 2040
2050 POSITION 25,8:? "F3=";F3;"[
N]":? ""
2060 ? "CZY SILY SA PODANE PRAWI
DLOWO(T/N) ?";CHR$(253);:GET #1,
S
2070 IF S=84 THEN GOTO 2100
2075 GOTO 2000
2100 H=0:IF F1>=F2 AND F1>=F3 TH
EN F=F1:GOTO 3000
2300 F=F3:IF F2>=F3 THEN F=F2
3000 GRAPHICS 7:SETCOLOR 4,4,4
3004 COLOR 1:SETCOLOR 0,1,0:PLOT
 1,10
3005 DRAWTO 159,10:PLOT 1,40:DRA
WTO 159,40:PLOT 1,70:DRAWTO 159,
70
3007 FOR T=18 TO 159 STEP 14.1:P
LOT T,11:PLOT T,41:PLOT T,71:NEX
T T

3010 PP=PEEK(106)-32:POKE 106,PP
-32:POKE 54279,PP
3015 ? CHR$(127);"PROSZE POCZEKA
C 10 SEKUND!":POKE 559,62
3020 PL=PP*256+1024:FOR I=PL TO
PL+1024:POKE I,0:NEXT I:POKE 532
77,3
3021 POKE 704,15:POKE 705,15:POK
E 706,15
3022 POKE 53256,1:POKE 53257,1:P
OKE 53258,1
3023 POKE 53248,50:POKE 53249,50
:POKE 53250,50
3030 RESTORE 4000:P=PL+27:FOR I=
P TO P+34:READ B:POKE I,B:NEXT I
3040 RESTORE 4000:P=PL+343:FOR I
=P TO P+34:READ B:POKE I,B:NEXT
I
3050 RESTORE 4000:P=PL+659:FOR I
=P TO P+34:READ B:POKE I,B:NEXT
I
3060 GOSUB 7050
3070 ? CHR$(28);"KAZDY KLAWISZ T
O START SYMULACJI  "

3080 ? "SPACJA TO WYKONANIE OBLI
CZEN CWICZ.":GET #1,S
3090 IF S=32 THEN X=1:POKE 106,P
P+16:POKE 53277,0:GOTO 2000
3100 H=0:GOSUB 7020:GOTO 4560
3399 REM ________________
3400 REM BLOK OBLICZENIOWY
3401 REM ________________
3402 POKE 53248,0:POKE 53249,0:P
OKE 53250,0
3410 POSITION 8,6:? "F1=";F1;"[N
]","F2=";F2;"[N]","F3=";F3;"[N]"
,:? :?
3417 GOSUB 9000
3420 POSITION 8,16:? "S1=";(F1*1
00)/F:POSITION 16,16:? " S2=";(F
2*100)/F
3421 POSITION 24,16:? " S3=";INT
(((F3*100)/F)*100)/100
3422 POSITION 9,14:? "a1=";F1:PO
SITION 16,14:? " a2=";F2
3423 POSITION 24,14:? " a3=";F3;
" [m/s^2]":? :? :? :?
3450 ? CHR$(127);"KLAWISZ N TO N
OWA SYMULACJA"
3460 ? "    INNY KLAWISZ TO POWR
OT DO MENU":GET #1,S:IF S=78 THE
N 2000
3470 GOTO 20
4000 DATA 255,255,255,255,255,25
5,255,255,255,255,255,255,255,25
5,255

4010 DATA 255,255,255,255,255,23
1,165,231,66,66,0,0,0,0,0,0,0,0,
0,0,0,0
4560 ? CHR$(28);CHR$(28);"KAZDY
KLAWISZ ZATRZYMUJE SYMULACJE"
4565 D=H^2
4600 POKE 53248,50+D*F1:POKE 532
49,50+D*F2:POKE 53250,50+D*F3
4610 IF F*D+40>178 THEN 3070
4615 IF PEEK(764)<>255 THEN FOR
```

```
T=1 TO 100:NEXT T:POKE 764,255:G
OTO 3070
4520 H=H+0.13:GOTO 4565
6000 COLOR 2:RESTORE 6010:FOR T=
1 TO 15:READ A,B,C,D:PLOT A,B
6001 DRAWTO C,D:NEXT T:RETURN
6010 DATA 18,6,70,6,68,5,69,5,68
,4,68,4,68,7,69,7,68,8,68,8
7000 DATA 18,36,70,36,68,35,69,3
5,68,34,68,34,68,37,69,37,68,38,
68,38
7001 DATA 18,66,70,66,68,65,69,6
5,68,64,68,64,68,67,69,67,68,68,
68,68
7020 COLOR 0:A=4:GOSUB 7030:A=34
:GOSUB 7030:A=64:GOSUB 7030:RETU
RN
7030 FOR T=A TO A+4:PLOT 0,T:DRA
WTO 90,T:NEXT T

7040 ? CHR$(28);"SILY DZIALAJA D
O ZATRZYMANIA WOZKOW":RETURN
7050 COLOR 2:RESTORE 8000:PLOT 1
8,6:DRAWTO 18+14*F1,6:A=14*F1:B=
4
7051 GOSUB 7100
7055 RESTORE 8000:PLOT 18,36:DRA
WTO 18+14*F2,36:A=14*F2:B=34
7056 GOSUB 7100
7060 RESTORE 8000:PLOT 18,66:DRA
WTO 18+14*F3,66:A=14*F3:B=64
7061 GOSUB 7100:RETURN
7100 FOR T=1 TO 5:READ C:PLOT 18
+A,B:DRAWTO C+A,B:B=B+1:NEXT T
7101 RETURN
8000 DATA 18,19,20,19,18
9000 ? "DROGI WOZKOW SA WYRAZONE
 W LICZBACH"
9001 ? "NIEMIANOWANYCH.DROGA NAJ
SZYBSZEGO"
9002 ? "WOZKA ZOSTALA PRZYJETA J
AKO 100.":? :RETURN
```

* SUMA KONTROLNA / ETYKIETA *

IP	0	AX	1	GX	2	ID	6
ZC	7	ZB	8	XD	9	PH	10
MY	20	KY	30	PO	40	SN	50
QH	100	CA	105	QJ	110	AW	115
BR	120	BY	125	BY	130	HE	135
NK	140	NR	145	TN	150	AN	155
CF	160	OF	165	LA	170	JW	175
MG	176	RR	180	PA	185	FI	186
WO	190	AZ	195	EV	200	OK	205
FT	210	TR	212	DN	215	EA	220
VJ	225	ZD	230	KF	235	ZM	240
FV	245	NE	250	JD	255	NV	260
PX	265	SE	270	CA	1000	ZO	1010
CG	1020	RC	1040	TO	1050	RY	1060
PR	1070	ZF	1080	PC	1090	KI	1100
CY	1110	CF	1120	RA	1130	HT	1140
MG	1150	XE	1160	LW	1170	YH	1180
PR	1190	LF	1200	VX	1210	RB	1220
CS	1230	GE	1240	RJ	1250	UW	1260
IW	1270	SY	1280	EI	1290	KG	1300
QV	1310	EE	1320	DR	1321	YP	1330
DU	1331	BC	1340	LN	1341	RX	1350
FR	1360	OS	1370	WO	1380	UH	1390
UM	1400	UC	1405	WC	1410	TC	1420
AY	1450	OY	1455	HW	1456	JZ	1457
TY	1460	TI	1465	PE	1470	HT	1480
ZO	1490	PN	1495	PN	1500	DJ	1510
MK	1520	EL	1530	TJ	1540	BU	1560
JP	1570	ZQ	1575	MY	1580	YD	1590
HP	1600	PJ	1900	YG	1950	QB	1960
EU	2000	XI	2010	PN	2015	UC	2016
PZ	2020	LL	2022	AT	2025	UO	2030
QK	2032	WH	2035	ZD	2040	VR	2044
GK	2050	OZ	2060	YC	2070	OP	2075
UO	2100	WF	2300	QT	3000	CX	3004
GK	3005	RU	3007	UF	3010	IW	3015
SE	3020	JE	3021	KQ	3022	RG	3023
FD	3030	TX	3040	DT	3050	CP	3060
RZ	3070	FT	3080	IA	3090	AS	3100
VT	3399	UO	3400	TO	3401	WJ	3402
TZ	3410	BQ	3417	VY	3420	JA	3421
VB	3422	OQ	3423	HM	3450	ZR	3460
QI	3470	OB	4000	BG	4010	TS	4560
CR	4565	XK	4600	DI	4610	AW	4615
MU	4620	CF	6000	FZ	6001	SI	6010
GL	7000	BX	7001	IB	7020	YL	7030
MY	7040	YI	7050	AP	7051	RG	7055
BJ	7056	JQ	7060	JC	7061	WX	7100
PF	7101	PJ	8000	CP	9000	RM	9001
LC	9002						

ŻEGLOWANIE

Nowa pasja, jaka ogarnęła Long Terry zaczęła się pięknego majowego dnia na lekcji historii cywilizacji ziemskiej. Wiemy, że nie należą one do najchętniej wysłuchiwanych przez uczniów szkoły elementarnego przygotowania do życia. Być może stary profesor Duck Long, mówiący monotonnym głosem i niezbyt często ilustrujący tezy wykładu eksponatami nie umiał zapalić do nich swoich uczniów. Tym razem jednak zamiast chorego profesora po włączeniu monitorów uczniowie VI oddziału zobaczyli wysokiego eleganckiego mężczyznę o sylwetce sportowca. Ten mając przed sobą pulpit, na którym włączyły się numery identyfikacyjne słuchaczy, uśmiechnął się szelmowsko i rozpoczął wykład. Mówił o sposobach pokonywania przestrzeni wodnych na różnego rodzaju pływadłach bez silników. Większość uczniów nie dawała temu wiary, jak to bowiem może być możliwe, żeby gdzieś w odległych latach od ich XXX w. ludzie płynąc z szybkością przysłowiowego żółwia mogli pokonywać olbrzymie przestrzenie oceanów i mórz.

Kiedy po zajęciach dzielili się swymi uwagami Tom Caurt zaproponował: słuchajcie możemy przecież spróbować zbudować coś takiego do pływania i sprawdzić. Praktycznie pomysł został przyjęty bez sprzeciwu.

No ...i zaczęło się. Najpierw za pomocą komputerów zgromadzili wiadomości o żeglowaniu i żaglowcach, później rozpoczęli opracowywanie dokumentacji. Nie należało to do najszybszych i bezkolizyjnych zadań. Raz po raz musieli pokonywać bariery. Bo oto nie można było zdobyć drewna na budowę łodzi. Wszystkie drzewa w myśl konwencji o ochronie przyrody były objęte prawem. Jack wymyślił użycie serganu masy plastycznej z ubiegłego wieku, imitującej drewno, ale okazało się, że nigdzie jej już nie produkują. I kiedy sytuacja wydawała się bez wyjścia Tom Frank, który był z rodzicami na wycieczce, przywiózł wiadomość, że czterdzieści mil od miasta stoją opuszczone domy wykonane z tego materiału. Konieczne było uzyskanie zgody na rozebranie i transport. Koniec końców poszło łatwiej niż przypuszczali. Materiał zwieźli i trzeba było przystąpić do prac. Okazało się, że żaden z robotów nie potrafi wykonywać czynności związanych z obrabianiem serganu. Dobrze, że ojciec Jacka jest programistą, napisał i zaprogramował dwa roboty i praca ruszyła. Zawsze po lekcjach cały oddział VI był w swojej stoczni (takie starożytne słowo). Chłopcy przywiązali się do miejsca robót. Gdzieś na początku czerwca wszystko było gotowe. Długa na dwanaście metrów łódź z trzema masztami gotowa była do przetransportowania na wodę. Na wszelki wypadek zainstalowano na niej wyposażenie ratownicze, takie jakie mają pływoloty, ster automatyczny, wspomagane komputerem identyfikatory. I wtedy okazało się, że nie ma żagli i olinowań. Po długich poszukiwaniach zamiast figurującego w zapisach płótna żeglarskiego użyto stersonu, materiału używanego do ochrony przed deszczem.

To było wielkie święto, kiedy łódź została spuszczona na wodę. Okazało się, że jest bardzo przyjemnie, kiedy kołysze się na falach. Przez najbliższe dni wszyscy uczyli się komend i czynności związanych z obsługą pływadła. Instrukcje te zdobył gdzieś niezawodny jak zwykle Jack. I wreście nadszedł dzień pierwszego rejsu. Padły komendy, naciągnięto żagle i nic, żadnego ruchu. Ich pływadło nie chciało się ruszyć. W końcu jednak wypłynęli budząc olbrzymią sensację. Nauczyli się powoli manewrować swoim żaglowcem, wzmacniając lub osłabiając nadmuch i szło. Najgorzej było przy wykonywaniu zwrotów. Z początku dopływali do jakiejś wyspy i tam opuszczali żagle, później zaczepiali liny i obracali statek, podnosili żagle i płynęli z powrotem. Fajne to były przejażdżki. Rodzice z początku z pobłażaniem przyglądali się ich wyczynom, później z coraz większym zainteresowaniem. W tym czasie chłopcy nauczyli się manewrowania swoim statkiem. Ponieważ kończył się rok szkolny postanowili odbyć długi rejs. Gromadzili zapasy, zbierali wiadomości o warunkach żeglowania. Podczas jednego z rejsów, dopłynęli do domu wykorzystując wiatry. I to co niejasne stało się jasnym. Kiedy swą decyzję wypłynięcia w długi rejs zgłosili Radzie Obywatelskiej natychmiast znaleźli się dziennikarze, którzy chcieli razem z nimi płynąć. Na szczęście podczas jednej z przejażdżek większość zachorowała na starą, jak świat chorobę morską i mieli ich z głowy.

Rejs ich był transmitowany na bieżąco. Prawie w każdym dzienniku pokazywano ich jak sobie radzą, a radzili sobie dobrze. Kiedy powrócili do swego miasta czekali na nich rodzice i znajomi. Ale nie to było najważniejsze, nabrali tężyzny fizycznej, sprawności, jakiej nie miał nikt z ich kolegów. Rada Lekarska wydała wtedy oświadczenie zmieniające dotychczasowy zakaz wykonywania prac fizycznych, dopuszczając je, gdyż rozwijają możliwości człowieka.

Od ich powrotu nie było prawie dnia, aby ich statek nie był zwiedzany przez wycieczki szkolne, inżynierów, techników. Dało się zauważyć również i to, że uruchomiono produkcję sergentu, a także budowę podobnych pływadeł. Te wkrótce zdominowały całe wybrzeże, co z kolei doprowadziło do konieczności wydania określonych przepisów porządkowych.

Tylko chłopcy z VII już wtedy oddziału myśleli o nowym pomyśle. Miał nim być statek podwodny z XX w. Kiedy to piszę, przystąpili właśnie do jego budowy.

Martin CARR

INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

H

HACKER - 1. programista, potrafiący pisać programy bez wstępnego zapoznawania się ze szczegółowym opisem, a także potrafiący nanosić poprawki do działających programów, nie posiadając dokumentacji, 2. użytkownik systemu komputerowego (sieci komputerowej), zajmujący się poszukiwaniem sposobów dostępu do zabezpieczonych danych (włamywacz do zabezpieczonych programów, systemów lub baz danych),
HALF - połowa, połówka,
HALF-ADDER - półsumator, sumator jednocyfrowy,
HALFADJUST - zaokrąglać,
HALF-BYTE - półbajt,
HALF CARRY - przeniesienie pomocnicze,
HALF DUPLEX - półdupleks, dupleks częściowy (niejednoczesna komunikacja w obu kierunkach),
HALF DUPLEX CHANNEL - kanał półdupleksowy,
HALF-DUPLEX CIRCUIT - patrz: HALF DUPLEX CHANNEL,
HALF-DUPLEX DATA PATH - półdupleksowy tor przesyłania danych,
HALF-DUPLEX OPERATION - system pracy półdupleksowej,
HALF-SUBTRACTOR - układ odejmujący jednocyfrowy,
HALF TITLE - tytuł wstępny, przedtytuł,
HALF-TITLE PAGE - strona przedtytułowa,
HALF-TONE - 1. półton, 2. półtonowy,
HALF-TONING - wytwarzanie półtonów,
HALF-WORD - półsłowo,
HALT - zatrzymanie (np. pracującego komputera),
HALT INSTRUCTION - rozkaz stopu, rozkaz zatrzymania,
HALVE - dzielić na połowę,
HALVING - dzielący na pół,
HAMMER - młotek (np. drukarki),
HAMMERLOCK - zastawka blokująca młotek drukarki,
HAMMER SPRING TENSION - naciąg sprężyny młotka (drukarki),
HAMMING CODE - kod Hamminga,
HAMMING DISTANCE - odstęp Hamminga, dystans Hamminga,
HANDBOOK - podręcznik, poradnik (książka),
HANDBOOK FOR PROGRAMMED LEARNING - podręcznik programowany,
HANDFUL - garść (czegoś), partia (np. kart),
HAND-HELD - test podręczny,
HAND-HELD COMPUTER - komputer podręczny,
HANDHELD MICROCOMPUTER - mikrokomputer podręczny,
HANDLE - wykonywać, mieć do czynienia, manipulować,
HANDLER - 1. podprogram współdziałania z urządzeniami zewnętrznymi, manipulator, 2. program obsługi sytuacji wyjątkowej,
HANDLING - manipulowanie, obsługiwanie, przenoszenie, operowanie,
HANDSHAKE - dwuszynowa metoda kontroli przesyłania informacji,
HANDSHAKING - 1. przesyłanie z potwierdzeniem, 2. uzgadnianie transmisji, 3. procedura synchronizująca (transmisję),
HANDSHAKING SIGNALS - sygnały uzgodnienia,
HANDWRITTEN - napisany ręcznie,
HANDY - poręczny, podręczny,
HANGING INDENT - występ, przesunięcie w lewo (np. tekstu),
HANGUP - zawieszenie (stan systemu komputerowego, w którym nie wyprowadza on już komunikatów i nie reaguje na zewnętrzne przerwania),
HANG-UP - przerwanie (wykonywania programu),
HAPHAZARD - 1. przypadek, 2. przypadkowy, dorywczy,
HAPPEN - zdarzać się,
HARD - 1. twardy, sztywny, 2. stały, niezmienny,
HARD ADDER - przyrostowy sumator sztywny,
HARD COPY - trwała postać zapisu informacji, wydruk na drukarce ekranu monitora,
HARD-COPY TERMINAL - końcówka drukująca, terminal drukujący,
HARD DATA - dane w postaci liczb i wykresów, (w odróżnieniu od opisów jakościowych),
HARD DISC - dysk twardy,
HARD DISK - patrz: HARD DISC,
HARD ERROR - błąd stały,
HARD PAGE BREAK - "twarde" zakończenie strony (w systemach edycji tekstów, przejście do nowej strony, stosowane przy rozpoczynaniu np. nowego rozdziału),
HARD-SECTORED DISK - dysk ze stałym rozmieszczeniem sektorów (dysk magnetyczny o stałym podziale na sektory, rozmieszczenia których nie można zmienić programowo),
HARD SECTORING - odpowiedni zapis na ścieżce kolejnych sektorów poprzez umieszczenie obok otworu indeksowego dyskietki kolejnych otworów, określających położenie początkowe sektorów,
HARD SPACE - "twarda" spacja (spacja wprowadzona do tekstu przez użytkownika, w odróżnieniu od spacji wprowadzanej automatycznie podczas justowania),
HARD TO DISTINGUISH - trudny do odróżnienia,
HARD TO REACH - trudno dostępny,
HARDWARE - sprzęt komputerowy, hardware,
HARDWARE BLOCK PRINT ROUTINES - procedury drukowania zawartości pakietu blokami fizycznymi,
HARDWARE CHECK - kontrola realizowana tylko za pomocą sprzętu,
HARDWARE COMPATIBILITY - wymienność sprzętowa,
HARDWARE-COMPATIBLE - wymienny sprzętowo,
HARDWARE CONTROL - sterowanie układowe,
HARDWARE DIVISION - dzielenie maszynowe,
HARDWARE ENVIRONMENT - środki sprzętowe (sprzęt komputerowy wykorzystywany w trakcie pracy programu),
HARDWARE ERROR - błąd sprzętowy (błąd spowodowany złą pracą sprzętu),
HARDWARE-IMPLEMENTED PROCESS SCHEDULING - sprzętowe sterowanie przebiegiem procesu,
HARDWARE INTERRUPT - przerwanie sprzętowe (przerwanie od urządzenia zewnętrznego np. po błędzie wykonywanego rozkazu, itp.),
HARDWARE MODEL - model fizyczny,
HARDWARE MULTIPLICATION - mnożenie sprzętowe (wykonanie operacji mnożenia rozkazem procesora a nie przez podprogram),
HARDWARE PROGRAM COUNTER - wbudowany licznik rozkazów,
HARDWARE-SOFTWARE TRADE-OFF - podział zadań między sprzęt a oprogramowanie,
HARDWARE SPRITE - sprzętowe środki tworzenia dynamicznego zobrazowania graficznego,
HARDWARE STACK - stos sprzętowy,
HARDWARE SUPPORT - wsparcie sprzętowe, pomoc sprzętowa, realizacja sprzętowa,
HARDWIRED - 1. sprzętowo zaszyty, 2. realizowany środkami sprzętowymi, 3. z wbudowanym konstrukcyjnie układem sterowania lub programowania,
HARDY - mocny, odporny, wytrzymały,
HARMFUL - szkodliwy,
HARMLESS - niegroźny, nieszkodliwy,
HARMONIC - harmoniczny,
HARMONIC DISTORION - zniekształcenie harmoniczne,
HARTLEY - jednostka miary informacji równa informacji reprezentowanej przez jedną cyfrę dziesiętną,
HASH - informacja pozbawiona praktycznego znaczenia, znajdująca się w pamięci,
HASH ADDRESSING - adresowanie z haszowaniem (patrz: HASHING),
HASH DATA - dane zbędne, dane pomyłkowe,
HASHING - haszowanie (sposób organizacji struktur danych, zapewniający efektywne wyszukiwanie i aktualizację danych, tzw. tablice HASH),
HASHING ALGORITHM - algorytm haszowania (patrz: HASHING),
HASHING FUNCTION - funkcja rozmieszczania, funkcja haszowania,
HASH TABLE - tablica przewodnik,
HASH TOTAL - suma kontrolna,
HASP - patrz: HOUSTON AUTOMATIC SPOOLING PROGRAM,
HASTEN COMPLETION - przyspieszyć zakończenie (np. prac programowych),
HAVE A BEARNING ON ... - wywierać wpływ na ...,
HAVE AN EFFECT ON ... - wpływać na ...,
HAVE A RESTRAINING EFFECT - działać hamująco,
HAVE AT ONE'S DISPOSAL - rozporządzać (czymś),
HAZINESS - nieostrość (np. obrazu na ekranie monitora),
HDAM - patrz: HIERARCHICAL DIRECT ACCESS METHOD,
HDLC - patrz: HIGH LEVEL DATA LINK CONTROL,
HDLC STATION - stacja protokołu HDLC,
HDR - patrz: HIGH DENSITY RECORDING,
HD - patrz: HIGH DENSITY,
HD TV - telewizja wysokiej jakości,
HEAD - głowica, także: nagłówek,
HEAD COMPUTER - komputer główny,
HEADER - nagłówek, etykieta początkowa,
HEADER CARD - karta inicjująca, karta wskazująca,
HEADER ENTRY - pozycja tytułowa,
HEADER LABEL - etykieta początku zbioru danych, zapis początku,
HEADER LABEL CHECK - sprawdzanie etykiety nagłówkowej (np. na taśmie magnetycznej),
HEADER RECORD - zapis (rekord) nagłówkowy,
HEAD GAP - szczelina głowicy magnetycznej, także: przerwa między głowicą magnetyczną a nośnikiem danych,

HEADING - patrz: HEADLINE,
HEADLINE - nagłówek,
HEAD MOVEMENT - przesuw głowicy,
HEAD POSITIONING TIME - czas szukania, czas ustawiania głowic,
HEAD SPACING - rozstaw głowic magnetycznych,
HEAD STACK - zespół głowic magnetycznych dla zapisu wielościeżkowego,
HEAD-TO-TAPE GAP - przerwa między głowicą magnetyczną a taśmą,
HEAP - 1. stos (dynamiczny obszar pamięci), 2. wielka ilość, 3. gromadzić,
HEAP MANAGER - program zarządzający stertą (dynamicznym obszrem pamięci programu), program dynamicznego rozdziału obszaru pamięci,
HEAVY - 1. intensywny, 2. ciężki, 3. silny,
HEAVY LOSS - znaczna strata, poważna strata,
HEIGHT - wysokość,
HEIGHT-BALANCED TREE - drzewo o zrównoważonej wysokości,
HEIGHT OF TREE - wielkość drzewa,
HELP - tekst objaśniający, objaśnienie, opis, help,
HELP LIBRARY - biblioteka tekstów objaśniających, biblioteka opisów (objaśnień),
HELP LINE - wiersz (na ekranie monitora) z tekstem objaśniającym,
HERCULES GRAPHIC CARD - karta grafiki Hercules,
HESITATION - opóźnienie, przerwa w przetwarzaniu,
HETEROGENEOUS - niejednorodny, różnorodny,
HETEROGENEOUS COMPUTER NETWORK - niejednorodna sieć komputerowa,
HEURISTIC - heurystyka,
HEURISTIC PROGRAM - program heurystyczny,
HEURISTIC ROUTING - heurystyczny wybór trasy,
HEWLETT-PACKARD COMPANY - firma amerykańska, produkująca między innymi mini- i mikrokomputery, przyrządy pomiarowe oraz tworząca oprogramowanie komputerowe,
HEX - patrz: HEXADECIMAL,
HEXADECIMAL - szesnastkowy,
HEXADECIMAL DIGIT - cyfra szesnastkowa (tzn. jedna z wielkości: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F),
HEXADECIMAL FORMAT - format szesnastkowy (przedstawienie danych w systemie szesnastkowym),
HEXADECIMAL NOTATION - szesnastkowy system liczenia,
HEXADECIMAL NUMBER - liczba szestnastkowa,
HEXADECIMAL NUMBER SYSTEM - system szesnastkowy, system heksadecymalny,
HEX FILE - plik (zbiór) typu szesnastkowego,
HGC - patrz: HERCULES GRAPHIC CARD,
HIBERNATING PROCESS - proces zatrzymujący, proces zawieszający,
HIBERNATING TASK - zadanie zatrzymujące, zadanie zawieszające,
HIBERNATION - stan oczekiwania, stan wyczekiwania,
HIC - patrz: HYBRID INTEGRATED CIRCUIT,
HIDAM - patrz: HIERARCHICAL INDEXED DIRECT ACCESS METHOD,
HIDDEN - 1. ukryty (np. plik), 2. ukryć, schować,
HIDDEN LINE - linia ukryta (np. linie niewidoczne w dwuwymiarowym rzucie bryły),
HIDDEN-LINE REMOVAL - usunięcie linii niewidocznych (w grafice komputerowej sposób odzwierciedlenia trójwymiarowego obiektu),
HIDDEN SURFACE - niewidoczna powierzchnia,
HIDDEN-SURFACE REMOVAL - usunięcie niewidocznych powierzchni (w grafice komputerowej sposób odzwierciedlenia trójwymiarowego obiektu),
HIDE - ukryć, schować,
HIERARCHICAL ACCESS METHOD - hierarchiczna metoda dostępu,
HIERARCHICAL ADDRESSING - hierarchiczna adresacja (sposób przedstawienia obiektu w sieci komputerowej przy pomocy złożonego identyfikatora, odzwierciedlającego strukturę sieci i drogę dostępu),
HIERARCHICAL COMPUTER NETWORK - hierarchiczna sieć komputerowa,
HIERARCHICAL DATA BASE - hierarchiczna baza danych,
HIERARCHICAL DIRECT ACCESS METHOD - hierarchiczna metoda bezpośredniego dostępu,
HIERARCHICAL INDEXED DIRECT ACCESS METHOD - hierarchiczna metoda bezpośredniego dostępu z wykorzystaniem indeksacji,
HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD - hierarchiczna indeksowo-sekwencyjna metoda dostępu,
HIERARCHICAL NETWORK - sieć hierarchiczna,
HIERARCHICAL SEQUENTIAL ACCESS METHOD - hierarchiczna sekwencyjna metoda dostępu,
HIERARCHICAL STORAGE - hierarchiczna pamięć komputerowa,
HIERARCHY - kolejność ważności, hierarchia,
HIERARCHY OF SETS - hierarchia zbiorów,
HIGH - 1. duży, 2. większy (przy porównywaniu dwóch wielkości),
HIGH BIT - górny bit (najbardziej znaczący bit),
HIGH BOUND - ograniczenie górne (pliku), kres górny,
HIGH DENSITY - wysoka gęstość (sposób zapisu informacji na dysku),
HIGH DENSITY RECORDING - zapis o dużej gęstości,
HIGH-DENSITY STORE - pamięć o dużej gęstości zapisu,
HIGH-DUTY - o dużej wydajności, wysokosprawny,
HIGHER-LEVEL LANGUAGE - język wyższego poziomu,
HIGHER ORDER EQUATION - równanie wyższego rzędu,
HIGH LEVEL - wyższy poziom,
HIGH-LEVEL COMPILER - kompilator wyższego rzędu,
HIGH LEVEL DATA LINK CONTROL - protokół HDLC,
HIGH-LEVEL GOAL - kres wysokiego poziomu,
HIGH-LEVEL LANGUAGE - język wysokiego poziomu,
HIGH-LEVEL PROTOCOL - protokół wysokiego poziomu,
HIGHLIGHT - uwypuklać, uwydatniać, podświetlać, rozjaśniać, błyszczenie (migotanie lub podświetlanie elementu obrazu),
HIGHLIGHTING - wydzielenie, uwydatnienie, wyodrębnienie (w grafice komputerowej wyodrębnienie części tekstu lub fragmentu obrazu na ekranie monitora poprzez podświetlenie, migotanie, itp.),
HIGH-ORDER DIGIT - cyfra o najwyższej wartości (np. w liczbie),
HIGH-ORDER POSITION - najstarsza pozycja (pierwsza z lewej strony pozycja bajtu, słowa, czy wiersza),
HIGH-PERFOMANCE - szybko działający,
HIGH-QUALITY - wysokiej jakości,
HIGH-RESOLUTION MODE - tryb graficzny wysokiej rozdzielczości,
HIGH-SPEED CARRY - przenoszenie szybkie,
HIGH-SPEED MEMORY - pamięć szybka, pamięć o dostępie szybkim,
HIGH-SPEED MEMORY BLOCK - blok pamięci ultraszybkiej,
HIGH-SPEED PAPER FEED - szybkie zasilanie drukarki papierem,
HIGH-SPEED PRINTER - drukarka szybka,
HIGH-SPEED STORAGE - patrz: HIGH-SPEED MEMORY,
HIGH-SPEED STORE - patrz: HIGH-SPEED MEMORY,
HIGHWAY - szyna, magistrala,
HIGH-YIELD - wysoka wydajność,
HIRED LINE - łącze dzierżawione,
HISAM - patrz: HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD,
HIT - pozytywny wynik porównania dwóch wartości,
HITACHI - firma japońska, produkująca sprzęt komputerowy kompatybilny z wyrobami firmy IBM,
HIT-ON-THE-FLY PRINTER - drukarka wierszowa drukująca w ruchu,
HIT RATE - współczynnik zgodności, współczynnik zbieżności,
HLL - patrz: HIGH-LEVEL LANGUAGE,
HLS MODEL - model "barwa-jaskrawość-nasycenie" (w grafice komputerowej charakterystyka koloru za pomocą trzech parametrów: barwy, jaskrawości i nasycenia),
HOLD - przechowywać (np. w pamięci), zatrzymywać, znajdować się, mieścić (w sobie), także: uważać że, utrzymywać,
HOLD-DOWN - przytrzymywać (np. naciśnięty klawisz),
HOLDER - oprawka, uchwyt,
HOLDING - 1. zabezpieczanie, przechowywanie (np. danych), 2. stan posiadania (np. taśm magnetycznych w bibliotece),
HOLD TIME - czas utrzymywania informacji (np. na wejściach bloków sekwencyjnych),
HOLE - dziurka (np. w karcie),
HOLE COUNT - liczenie dziurek (np. na karcie),
HOLE DENSITY - gęstość dziurek,
HOLLERITH CARD - 80-kolumnowa karta dziurkowana z nadrukiem w kodzie Hollerith'a,
HOLLERITH CODE - kod Hollerith'a (kod wykorzystywany do przedstawiania informacji tekstowej na karcie dziurkowanej),
HOLLERITH CONSTANT - wielkość tekstowa (w języku Fortran),
HOLOGRAPHIC MEMORY - pamięć holograficzna,
HOLOGRAPHIC STORAGE - pamięć holograficzna,
HOLOGRAPHIC STORE - patrz: HOLOGRAPHIC STORAGE,
HOLOGRAPHY - holografia,
HOME - początek, pozycja początkowa, (w grafice komputerowej, określenie prawego, górnego rogu ekranu monitora),
HOME ADDRESS - adres własny (np. pole ścieżki dysku, zawierające adres tej ścieżki),
HOME BLOCK - blok początkowy,
HOME BUCKET - porcja macierzysta,
HOME COMPUTER - komputer domowy,
HOME LOCATION - komórka podstawowego obszaru, obszar podstawowy,
HOME POSITION - pozycja główna, pozycja podstawowa,
HOME RECORD - zapis (rekord) główny (pierwszy zapis w pliku), zapis (rekord) odniesienia (pierwszy w łańcuchu zapisów)

LIGA MYŚLĄCYCH

ZADANIE 1

Pięciu studentów, powtarzających trzeci rok studiów wybiera losowo — każdy niezależnie od pozostałych, jedną z trzech równoległych grup. Jakie jest prawdopodobieństwo, że:

1) wszyscy studenci znajdą się w pierwszej grupie,

2) wszyscy studenci znajdą się w tej samej grupie,

3) w ustalonej grupie znajdzie się dokładnie trzech studentów?

ZADANIE 2

Ile średnio powinno przypadać rodzynków na bułeczkę aby z prawdopodobieństwem nie mniejszym niż 0,99 twierdzić, że w bułeczce znajduje się choćby jeden rodzynek?

ZADANIE 3

Dwaj piechurzy wyszli w jednakowym czasie naprzeciw siebie i spotkali się po trzech godzinach i dwudziestu minutach. W jakim czasie każdy z nich przejdzie odległość, jaka dzieli miejsca, z których wyszli, jeżeli pierwszy przyszedł na miejsce, z którego wyszedł drugi, o 5 godzin później?

ZADANIE 4

Po przejściu w ciągu godziny 3,5 km, pasażer idący do pociągu, zorientował się, że idąc dalej z tą samą prędkością spóźni się na pociąg o 1 godzinę. Dlatego pozostałą drogę przeszedł z prędkością 5 km/h, przybywając na stację, na 30 minut przed odejściem pociągu. Wyznaczyć drogę, jaką przebył pasażer.

ZADANIE 5

Dwóch turystów idzie naprzeciw siebie — jeden z miejscowości A, a drugi z miejscowości B. Pierwszy turysta wyszedł z miejscowości A o 6 godzin później i przy spotkaniu okazało się, że przeszedł o 12 km mniej niż drugi. Kontynuując dalej podróż z tą samą prędkością, pierwszy turysta przyszedł do miejscowości B po 8 godzinach, a drugi do A po 9 godzinach. Wyznaczyć odległość AB oraz prędkość obu turystów.

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

GIEŁDA POMYSŁÓW

ZX-81

Szukanie dziury

```
1 REM **********************
2 REM *  SZUKANIE DZIURY   *
3 REM *        ZX-81       *
5 REM **********************
10 FOR A=1 TO 32
15 PRINT CHR$ 7;
20 NEXT A
25 IF A=32 THEN GOTO 30
30 LET S=0
35 LET J=INT RND*16
40 GOSUB 145
45 PRINT "
"
50 PRINT AT 9,7; "ODLEGLOSC?"
55 INPUT A
60 PRINT AT 9,7; "                "
65 PRINT AT 17,J;""
70 J=J+INT(A/3+RND)
75 GOSUB 145
80 LET S=S+1
85 PRINT AT 5,12; "ILOSC PODE
JSC: ";S
90 IF J<27 THEN GOTO 50
95 IF J>27 THEN GOTO 120
100 PRINT AT 17,27; "" ;AT 18,
27;"0"
105 LET U=J-S*2
110 PRINT "ILOSC PUNKTOW:"; U
;"- ZA TRAFIENIE"
115 GOTO 125
120 PRINT "BEZNADZIEJNIE"
125 FOR G=1 TO 100
130 NEXT G
135 CLS
140 RUN
145 PRINT AT 17,J; "O"
150 RETURN
```

KRZYŻÓWKA NR 10

LUNONAUTA		NAPÓJ ALKOHOLOWY	WIELKA ILOŚĆ	SZAKLA		GRUPA, ZESPÓŁ		NOGA SSAKA	RADIOWY LUB TELEFONICZNY	MIASTO W PD. TURCJI	EKSKOMUNIKA
UNICESTWIENIE	S	A	M	O	Z	A	G	Ł	A	D	A
ARCHAIK, PALEOZOIK	E	R	A	ARKUSZ POPRAWEK		AWANTURA	R	A	B		
MAŁY LAS	L	A	S	E	K	BEZDĘTNOWA	O	P	O	N	A
ZWIERZĘ Z RODZINY O TEJ SAMEJ NAZWIE	P	E	K	A	R	I	BRZEGOWIEC	M	A	N	
BIEG RZEKI	N	OBSZAR PRZED PORTEM / TRÓJCHLOROETYLEN		R	E	D	A	NA CZEŚĆ JUBILATA	E	WIDZI W NOCY	
MIESZANINA, STOP	O	GLINOWIEC / DUŻY DESZCZ		T	A	L	POKARMŻER	D	N		
MIASTO W WIETNAMIE, OŚR. ADMINISTR. PROW. BINH TRI	N	U	R	T	ŻOŁNIERZ POLICJI WOJSKOWEJ		A		T		
AUTOR POW. „KRÓLEWICZ I ŻEBRAK"	A	L	I	A	2	PIENIĄDZ NASZYCH SĄSIADÓW			DO KONSERWACJI		NIEJEDEN NA GŁOWIE
KANAŃSKI BÓG PŁODNOŚCI	H	U	E	ODMIANA GIPSU	WŁAŚCICIEL STATKU	A	R	M	A	T	O	R
MIŁA WOŃ	T	W	A	I	N	UZDROWISKO W SZWAJCARII	A				
MAŁA ALEKSANDRA		A	A	L	PREZENT UPOMINEK	D	A	R	PIERWIASTEK O LA. 85		TĘŻEC	
WYCIĘTY ZNAK NA CZYMŚ	A		A	L	A	GRUPA SPOŁECZ. W INDII.	K	A	S	T	A	
ODMIANA JABŁONI	A		B	DO JARZYN	R	JESIENNY KWIAT / LAMPA Z KLOSZEM	A	S	T	E	R	
JEDNOSTKA DŁUGOŚCI	R	WILLA SZYMANOWSKIEGO	A	T	M		DAWNE LICZYDŁA	T	ODMIANA GRUSZY		JEST W BANKU	
JAŁMUŻNA	K	O	K	S	A	PŁASKI, OKRĄGŁY TALERZYK Z HOSTIĄ		A				K
	M	E	T	R		P	A	T	E	R	A	
	D	A	T	E	K		BRAT CZECHA I LECHA		R	U	S	
	T	PORADA	R	A	D	A	AFRYKAŃSKA LUB AZJATYCKA					A

W rozwiązaniu krzyżówki wystarczy podać, ile razy w diagramie występują litery: I, K, S. Rozwiązanie zadania należy przesłać pod adresem redakcji na kartach pocztowych, w terminie do końca stycznia, naklejając kupon „IKS-a". Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.

Nauka • Technika

Nowoczesna technika w strukturze kolejowej, to przede wszystkim wysokiej klasy łączność sygnalizacyjna — czyli zabezpieczenie ruchu coraz szybszych pociągów i szeroko podjęta informacja, realizowana dzięki wykorzystaniu komputerów. Instytuty badawcze kolejnictwa, zespoły naukowców z każdym rokiem oferują coraz to nowsze, bardziej korzystne ekonomicznie i technicznie rozwiązania z użyciem m.in. mikroprocesorów i światłowodów.

Zastosowanie techniki komputerowej do kontroli biegu pociągów i łączności między dyspozytorniami i maszynistami zapewnia kolei efektywniejszą działalność. Zdalne pomiary z lokomotywy umożliwiają wychwycenie każdej najmniejszej niedokładności w poszczególnych urządzeniach hamulcowych w składzie badanego pociągu. Zastosowanie elektronicznego systemu wykrywania wadliwego działania urządzeń hamulcowych wpływa na osiąganie przez pociągi lepszych efektów eksploatacyjnych. Przekazywanie informacji do komputera w lokomotywie zapewnia nie tylko efektywną kontrolę biegu pociągu, ale umożliwia zdalne śledzenie trasy, sygnałów podawanych na semaforach, zamykanych rogatek na przejazdach itp. Biegiem pociągu kieruje maszynista, nadzoruje go komputer, który jednocześnie podaje podobne dane do dyspozytora, posiadającego łączność radiową z maszynistą.

Kierunki podawanej drogi z komputera w lokomotywie do dyspozytora nie tylko umożliwiają zlokalizowanie pociągu na trasie, ale także automatyczne nastawianie drogi przebiegu oraz uzyskanie pełnej kontroli nad pociągami poruszającymi się w danym kierunku. Umożliwia to wprowadzenie na trasę w bezpiecznych odstępach dodatkowych pociągów towarowych.

Handlowcy w Stanach Zjednoczonych oferują dzieciom komplét do demonstracji zjawiska nadprzewodnictwa. To, co nie tak dawno mogła obserwować tylko mała grupa naukowców zatrudnionych w kilku najlepszych ośrodkach badawczych świata, obecnie bawi i uczy młodzież szkolną.

Prototyp zabawki skonstruowali specjaliści z amerykańskiego Uniwersytetu Wisconsin. W skład zestawu wchodzi naczynie z przezroczystego tworzywa połączone ze zbiornikiem na ciekły azot oraz mała płytka o średnicy 25 mm. Krążek ten wykonany został ze związków ltru, baru i miedzi z tlenem. Po obniżeniu temperatury w plastykowym naczyniu, płytka zaczyna unosić się w powietrzu, zaś mierniki badające własności próbki rejestrują zmiany w przepływie prądu efektonicznego oraz zachodzące przeobrażenia pola magnetycznego.

Producent zabawki dołącza do niej także rękawice i okulary ochronne, wszak eksperymentator ma do czynienia z ciekłym azotem. Model przedstawiający nadprzewodnictwo można nabyć za 25 dolarów.

„IKS" — dodatek „Żołnierza Wolności". Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarra, Romuald Giąb, 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. 1965. Nr ind. 361682. U-3