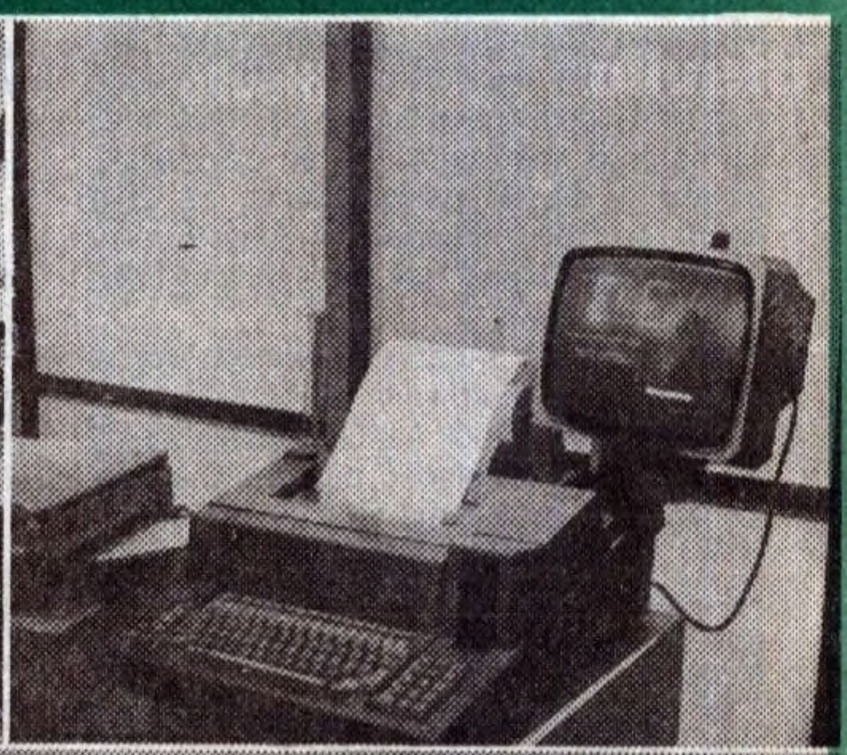
IKS

INFORMATYKA KOMPUTERY SYSTEMY

CENA 240 ZŁ

DODATEK DO „ŻOŁNIERZA WOLNOŚCI" Nr 7/8/1988 ISSN 0860 — 2794

Czy komputer zastąpi nauczyciela czyli „Informatyka w szkole"



Jak jubilat z jubilatem

Rozmowa z dyrektorem Centralnej Składnicy Harcerskiej mgr. Wojciechem Szanterem.

● *Czy uważa Pan, że trzydzieści lat istnienia jest dla przedsiębiorstwa wiekiem dojrzałym?*

— Trzydzieści lat, to z pewnością już daleko posunięta dojrzałość. Dla przedsiębiorstwa handlowego każdy rok istnienia na rynku ma olbrzymie znaczenie. Umacniamy przecież swoje miejsce i mamy nadzieję — zdobywamy nowych klientów...

● *Jeśli jest czym...*

— ...i gdzie.

● *To problem każdej firmy.*

— Dla nas los był łaskawy. Kiedy w 1958 roku Centralna Składnica Harcerska została wydzielona ze struktury spółdzielni „Społem" i rozpoczęła działalność jako samodzielna jednostka gospodarki uspołecznionej, faktycznie — nie mieliśmy nic. Wkrótce jednak ówczesny przewodniczący Rady Przyjaciół Harcerstwa, marszałek Marian Spychalski, przyczynił się do przeznaczenia na cele handlowe CSH znanego dziś budynku przy ul. Marszałkowskiej. I tak rozwinęliśmy działalność handlową.

● *Stając się z czasem monopolistą w sprzęcie i umundurowaniu harcerskim.*

— Nie tylko. Obok handlu prowadziliśmy od początku działalność produkcyjną. Nasze zakłady wytwarzają artykuły politechniczne. Oczywiście początkowo zajmowały się także przeróbkami sprzętu technicznego, pochodzącego z tak zwanego demobilu i przeznaczonego dla harcerzy.

● *Podobno zatrudnialiście wtedy jedynie „złote rączki" do wszystkiego.*

— Rzeczywiście podzespoły techniczne obejmowały bardzo szeroki wachlarz wyrobów, a ich „produkcja" była rzemieślnicza. Robiliśmy między innymi zabawki mechaniczne sterowane kablowo, mikrosilniki, sprzęt łączności...

● *I tak staliście się potentatem...*

— Nie. Tak się po prostu rozwijaliśmy. Rozszerzaliśmy działalność produkcyjną i rozbudowywaliśmy bazę handlową, dysponując wkrótce trzydziestoma sklepami na terenie całego kraju. Nie mamy też szans być monopolistą. Tylko w zaopatrzeniu ZHP w umundurowanie jesteśmy jedyni.

● *I dlatego jest ono tak drogie?*

— Nic podobnego. Cena umundurowania nie zależy od nas. Sprzedajemy uwzględniając jedynie koszt własny i marżę handlową, a ta nie zależy od nas.

● *To na czym zarabiacie?*

— Na sprzęcie.

● *Ale teraz jakby go więcej było w sklepach CSH.*

— To efekt importu, głównie z NRD. Tę współpracę rozpoczęliśmy w latach sześćdziesiątych. Sprowadzaliśmy artykuły techniczne, a przede wszystkim sprzęt modelarski, narzędzia, mikronapędy. Mniejsze szanse mamy na rozwinięcie eksportu, a przyczyna jest prozaiczna — brak producentów.

● *Importowany sprzęt jest bez wątpienia atrakcyjny, podobnie jak i jego cena...*

— Jakość kosztuje. Ale mamy i tańszy, na przykład obok narzędzi do obróbki drewna firmy Blacken-Decker wyroby z CELMY...

● *Pojawiające się niczym meteoryty...*

— Bo są niezłe i tanie. Zresztą nie jest ich tak mało — sprzedajemy przecież 25—30 proc. produkcji tej firmy. Prowadzimy również w dziewięciu punktach w kraju serwis narzędzi przez nas sprzedawanych.

● *A jeśli towar nie idzie...*

— Tak bywa. Wtedy obniżamy ceny i to znacznie, jak na przykład w wypadku nart. A jest to przecież sprzęt dobry. Na przecenie nie tracimy. Przeciwnie, uwalniamy zamrożony kapitał. Przecież handel polega na obrocie kapitałem.

● *Chyba nie zawsze jest to łatwe?*

— Nie przeżywamy kryzysu.

● *A na początku lat osiemdziesiątych?*

— W 1982 roku mieliśmy 2,8 mld obrotu, w 1988 realne jest osiągnięcie 22 mld!

● *Uwzględniając dewaluację...*

— Każdy rachunek daje pokaźny plus po naszej stronie.

● *To wynik ekonomiczny, ale również rozszerzacie swoją ofertę o sprzęt komputerowy.*

— Od 1985 roku najpierw w Warszawie, a później w innych miastach (w sumie w 9 sklepach) sprzedajemy komputery i osprzęt. Naszym celem było dostarczenie na polski rynek taniego, a tym samym dostępnego mikrokomputera.

● *Timexa, bardzo krytykowanego za jego stosunkowo małe możliwości...*

— Ale to nie przeszkodziło w sprzedaniu 18 000 egzemplarzy tego modelu. 90 procent Timexa kupiło ówczesne Ministerstwo Oświaty i Wychowania i kluby Młodych Mistrzów Techniki. A poza tym Timex miał zasadnicze dwie zalety: po pierwsze był zgodny z bardzo popularnym Spectrum, a po drugie i najważniejsze — po prostu był.

● *Do dzisiaj nie doczekaliśmy się natomiast mikrokomputera z prawdziwego zdarzenia, przeznaczonego dla szkół...*

— ...a bardzo chętnie byśmy nim handlowali.

● *Tymczasem, zamiast wyrobów z ELWRO na półkach w CSH gości firma PROSYSTEM.*

— I jak dotąd ten sprzęt sprzedaje się dobrze, choć komputery do najtańszego towaru przecież nie należą. W 1987 roku sprzęt komputerowy dał nam obrót 8 mld.

● *Przy takim obrocie pracownikom powodzi się chyba dobrze...*

— Średnie zarobki wynoszą 36 tysięcy, ale są oczywiście zróżnicowane. Spośród 1800 pracowników 350 pracuje w produkcji. Poza tym naszym celem nie jest maksymalizacja zysku — chcemy być dla młodzieży atrakcyjni, dlatego szkolimy załogę, inwestujemy, zabiegamy o towary innowacyjne...

● *I tylko takich życzę w całym następnym trzydziestoleciu istnienia.*

Dziękuję za rozmowę.

Wiesław CETERA

W NUMERZE:

W szponach Atari
— Okna — str. 4
Komputer i wideo — str. 8
Transputery — str. 9
Nuty z komputera — str. 10
Zadanie konikowe — str. 19
Masterword — str. 20
Kompilatory C — str. 22
Jak oszukać edytor ZX — str. 23
Tajemnica żółtego światła — str. 26
Gra — str. 27
Informatyczny słownik angielsko-polski — str. 29
Liga Myślących — str. 31

Dyrektor Wojciech Szanter przekazał redakcji komputer SVI-738 — sprowadzany do kraju przez firmę Prosystem i sprzedawany w sklepach Centralnej Składnicy Harcerskiej.

Informatyka w medycynie

Czy ryzyko zapadnięcia na chorobę serca jest większe u osoby prowadzącej higieniczny tryb życia, ale obciążonej genetycznie skłonnością do takich właśnie dolegliwości, czy też w wypadku osoby nie przestrzegającej wprawdzie zasad racjonalnego odżywiania i aktywnego wypoczynku, wolnej jednak od genetycznych obciążeń? Próby oceny rzeczywistej skali ryzyka z uwzględnieniem wszystkich realnych przyczyn zagrożenia wydają się tak trudne, że z pomocą może przyjść tylko... komputer!

Nikogo już dziś nie zaskakuje wykorzystanie komputerów w medycynie. Powszechnie stosuje się je przy przeprowadzaniu i analizie różnego rodzaju badań i testów. Okazuje się jednak, że na tym nie kończą się ich możliwości. Ostatnio pojawiły się próby spożytkowania techniki komputerowej do opracowywania indywidualnych ocen statystycznego stanu zagrożenia zdrowia.

Najpierw trzeba odpowiedzieć na sto pytań dotyczących trybu i warunków życia, chorób występujących w rodzinie, wyników badań medycznych. Odpowiedzi wprowadzone są do pamięci komputera, który dysponuje także danymi na temat stopnia umieralności i stanu epidemiologicznego. Korzystając z tych wiadomości, maszyna stawia diagnozę w postaci wydruku komputerowego. Zawiera on ogólną ocenę stanu zdrowia — bardzo dobry, dobry, słaby — oraz określenie „wieku zdrowotnego" w porównaniu z wiekiem rzeczywistym. Wydruk przedstawia także indywidualne ryzyko zachorowania w porównaniu z przeciętnym zagrożeniem daną chorobą wśród osób tej samej płci i tego samego wieku. Może też udzielić wskazówek, co robić, aby zdrowiej i dłużej żyć.

Wiele amerykańskich przedsiębiorstw — wśród nich również te największe, jak Ford czy IBM — wykorzystuje komputerowe diagnozy do określenia stanu zdrowotności załogi, co z kolei służy jako podstawa do opracowywania planów ochrony zdrowia zatrudnionych.

Środowisko lekarskie niechętnie jednak korzysta z usług elektronicznych „kolegów". Składa się na to kilka przyczyn. Nie chodzi tu bynajmniej tylko o pracochłonność wszystkich czynności związanych z przeprowadzeniem bardzo dokładnego i szczegółowego wywiadu oraz wprowadzeniem go do pamięci komputera. Istnieją także inne, o wiele chyba istotniejsze powody. Naukowe podstawy takich ocen nie wydają się zupełnie sprawdzone. W przypadku niektórych chorób z całą pewnością potwierdzono negatywny wpływ czynników mogących je wywoływać. Jest tak chociażby z rakiem płuc czy chorobami serca, chorobami związanymi z paleniem papierosów. Jednakże informacje dotyczące ewentualnych powiązań potencjalnych czynników chorobotwórczych, ze wskaźnikiem zapadania na inne dolegliwości, są już daleko mniej precyzyjne.

Komputerowe oceny opracowywane są przy wykorzystaniu danych statystycznych, a statystyka — jak powszechnie wiadomo — lubi kłamać. Podobnie jest i w tym wypadku. Obraz statystyczny uzyskany w wyniku badania dużej grupy w niewielkim tylko

Foto: „U.S. News and World Report"

stopniu może być wykorzystany do nakreślenia wizerunku indywidualnego pacjenta, odznaczającego się jedynym, jemu tylko właściwym kodem genetycznie uwarunkowanych mocniejszych i słabszych cech. Toteż komputerowa diagnoza jest raczej tylko ogólną, przybliżoną informacją na temat stanu zdrowia i możliwych rokowań na przyszłość. Zdaniem Edwina Hutchinsa, dyrektora uniwersyteckiego centrum w Atlancie, prowadzącego badania nad programem komputerowych diagnoz, ocena taka powinna być traktowana jako zbiór wskazówek bardziej co prawda wiarygodnych niż wróżby i horoskopy, ale nie pretendujących w żadnym wypadku do dokładnego i ścisłego przepowiadania przyszłości.

Częste są przecież przypadki, że ludziom nie dbającym zupełnie o swoje zdrowie, bez umiaru korzystającym z uciech życia, dopisuje zdrowie i dożywają długich lat. Zdarza się także, i to wcale nierzadko, że stosunkowo młodo umierają osoby, które — jeśli by sądzić z ich trybu życia — powinny żyć o wiele dłużej. Poza tym informacje komputera mogą być mylnie interpretowane. Pięćdziesięciolatek palący dwie paczki papierosów dziennie może odczytać ze swojego wydruku, że w jego wypadku ryzyko zachorowania na raka płuc jest dziesięć razy większe w porównaniu z przeciętnym zagrożeniem tą chorobą; elektroniczny „diagnostyk" nie powie już jednak, iż nie każdy palacz musi na pewno zachorować i że rak płuc rozwija się nie częściej niż u jednego z dziesięciu palaczy.

Należy jeszcze zwrócić uwagę na jeden niebezpieczny aspekt całego tego przedsięwzięcia. Otóż może się okazać, że człowiek zupełnie zdrowy, nie zagrożony żadną z ciężkich chorób, otrzyma diagnozę niezbyt dla niego pomyślną. Może to zostać spowodowane na przykład niedokładnym, czy niepełnym wywiadem, nieznajomością rodzinnych obciążeń genetycznych, czy wreszcie fałszywą interpretacją danych statystycznych, o co przecież nie tak trudno. Szkody powstałe w takim wypadku są trudne do przewidzenia. Pacjent nie tylko traci dobre samopoczucie, ale lęk spowodowany świadomością bliskiej choroby w znacznym stopniu wpływa na obniżenie jego kondycji, zarówno psychicznej, jak i fizycznej, co może doprowadzić do znacznego nawet utraty odporności organizmu i — jako konsekwencji — choroby, a nawet śmierci. Prawdopodobieństwo tak skrajnego przypadku nie jest wprawdzie duże, ale zawsze należy się liczyć z takim niebezpieczeństwem. Podstawową zasadą medycyny jest przecież „po pierwsze — nie szkodzić", toteż wszystkie czynniki mogące w najmniejszym nawet stopniu przyczynić się do osłabienia kondycji pacjenta powinny zostać wyeliminowane.

Oprócz wyżej wymienionych wątpliwości, nasuwa się jeszcze jedna. Nauka rozwija się w niezwykle szybkim tempie. Ośrodki prowadzące badania nad przyczynami różnych chorób opracowują wciąż nowe programy, służące do wykorzystania przy komputerowych diagnozach. Przyszłe osiągnięcia w dziedzinie genetyki i biochemii w połączeniu z dokładnym określeniem wpływu warunków życiowych, zaowocują z pewnością dużymi sukcesami na tym polu. Jednakże w miarę wzrostu dokładności i trafności ocen, trzeba sobie zadać pytania: jak dużo tak naprawdę chcemy wiedzieć? Jaki margines zostawić dla nadziei?

Małgorzata NOWOTNY

Nauka • Technika • Nauka • Technika

Poznawaniem tajników gaszenia ognia postanowili zająć się uczeni z Uniwersytetu w Tibilisi. W tym celu nie rozniecano jednak próbnych pożarów. Radzieccy uczeni do odpowiedzi na nurtujące ich pytanie wykorzystali komputer. Wpisali do zaprogramowanego "pożaru" umowne cząsteczki związku o średnicach rzędu kilkudziesięciu mikronów, będące odpowiednikiem proszków gaśniczych. W wyniku operacji symulacyjnych otrzymano odpowiedź, według której płonący w powietrzu metan gaszony jest wskutek zachodzących reakcji chemicznych węglowodorów z cząsteczkami soli. Natomiast czysty wodór okazał się nadzwyczaj podatny na termiczny kontakt z substancją gaśniczą. Tak więc to w jaki sposób gaszony jest pożar zależy przede wszystkim od rodzaju płomienia.

Amstrad

Kulki

```
10 GOTO 430
20 n=2
30 DIM x(n):DIM y(n):DIM a(n):DIM b(n)
40 GOSUB 370:s=INT(TIME/300):p=0:z=20:v=
12
50 MOVE 10,8: DRAWR 0,378:DRAWR 618,0:DR
AWR 0,-378:DRAWR -618,0
60 GOSUB 310
70 FOR i=1 TO n
80 LOCATE x(i),y(i):PRINT " ";
90 x(i)=x(i)+a(i):y(i)=y(i)+b(i)
100 LOCATE x(i),y(i):PRINT CHR$(231);
110 IF x(i)<=2 OR x(i)>=39 THEN a(i)=-a(
i)
120 IF y(i)<=2 OR y(i)>=24 THEN b(i)=-b(
i)
130 IF x(i)=z AND y(i)=v THEN 240
140 IF x(i)=18 AND y(i)<20 AND y(i)>6 TH
EN a(i)=-a(i)
150 IF x(i)=24 AND y(i)<16 AND y(i)>5 TH
EN a(i)=-a(i)
160 SOUND 1,x(i)*y(i),5,5
170 IF INKEY(71)=0 AND z>=3 THEN GOSUB 2
90:z=z-1
180 IF INKEY(63)=0 AND z<=38 THEN GOSUB
290:z=z+1
190 IF INKEY(29)=0 AND v>=3 THEN GOSUB 2
90:v=v-1
200 IF INKEY(31)=0 AND v<=23 THEN GOSUB
290:v=v+1
210 LOCATE z,v:PRINT CHR$(249);
220 NEXT
230 GOTO 60
240 FOR k=239 TO 60 STEP -50
250 SOUND 3,k,10:NEXT
260 FOR i=1 TO 100
270 INK 0,9,24:NEXT:INK 0,1
280 GOSUB 370:p=p+1:CLS:GOTO 50
290 LOCATE z,v:PRINT " ";:SOUND 3,500,1,
,,13
300 RETURN
310 t=INT(TIME/300)-s:LOCATE 6,1
320 IF t>=100 THEN 350
330 PRINT t:LOCATE 35,1:PRINT p
340 RETURN
350 LOCATE 15,1:PRINT "K O N I Ė C"
360 FOR i=1 TO 3000:NEXT:RUN
370 FOR i=1 TO n
380 x(i)=INT(RND*100):IF x(i)<5 OR x(i)>
35 THEN 380
390 y(i)=INT(RND*100):IF y(i)<5 OR y(i)>
20 THEN 390
400 IF x(i)=20 AND y(i)=12 THEN 380
410 a(i)=1:b(i)=-1
420 NEXT:RETURN
430 CLS:LOCATE 14,1:PRINT "K  U  L  K  I
 ";:LOCATE 20,12
440 PRINT CHR$(249)
450 LOCATE 20,5:PRINT "*"
460 LOCATE 20,19:PRINT ">"
470 LOCATE 13,12:PRINT "z"
480 LOCATE 27,12:PRINT "x"
490 MOVE 310,228:DRAWR 0,92
500 MOVE 310,204:DRAWR 0,-87
510 MOVE 297,213:DRAWR -80,0
520 MOVE 320,212:DRAWR 89,0
530 TAG:FOR x=620 TO 50 STEP -20
540 MOVE x,20:PRINT "   S T A R T
 S P A C J A      ";
550 SOUND 1,x,10
560 NEXT
570 SOUND 1,50,100
580 IF INKEY$ <> CHR$(32) THEN 580
590 MODE 1: RUN 20
```

C.S.

W szponach Atari (117)

Okna

Tomasz MROWIEC,
Ludwik PIELA

Podczas tworzenia programów musimy pamiętać, że korzystać z nich będą zwykle ludzie mający niewielkie doświadczenie informatyczne. Wymaga to zastosowania prostych i przystępnych metod komunikowania się z programem. Obsługa programu powinna być tak prosta i wygodna, aby każdy mógł się nim posługiwać po krótkim treningu. W tym celu powinien on spełniać następujące warunki:

— wyświetlać na ekranie wykaz funkcji, które może zrealizować, w postaci menu,

— komunikować się z użytkownikiem na zasadzie dialogu, sprawdzać poprawność wprowadzanych danych i poleceń,

— wyświetlać na ekranie wszystkie informacje potrzebne użytkownikom,

— podpowiadać, na życzenie, jakie czynności należy wykonać, aby zrealizować określone zadanie.

Interesującym i efektownym środkiem prezentowania na ekranie dodatkowych informacji, różnorodnych objaśnień i podpowiedzi jest technika okien. Przez pojęcie okna rozumiemy wyróżniony fragment ekranu przeznaczony do wyświetlania określonych informacji. Okna mogą być tworzone w dowolnych miejscach ekranu, mieć różne rozmiary, być umieszczone obok siebie, albo zasłaniać się częściowo lub całkowicie. W każdej chwili mogą być usunięte z ekranu, na którym pojawia się poprzednio jego zawartość.

Jeśli ktoś z czytelników nie widział jeszcze używania okien, stosowanych powszechnie w komputerach profesjonalnych, to nic straconego. Prezentowane programy demonstracyjne nie tylko pokażą, jak to wygląda, lecz także umożliwią tworzenie okien we własnych programach.

Zdajemy sobie sprawę, że najlepszy opis nie zastąpi najmarniejszego pokazu. Musimy jednak krótko opisać, jak wygląda używanie okien, zanim omówimy, na czym polega ich realizacja.

Z punktu widzenia użytkownika wygląda to następująco. W wybranym miejscu ekranu wyświetlany jest wyróżniony komunikat, niekiedy obramowany stosowną obwódką. Zasłania on informacje tekstowe lub graficzne, istniejące w tym obszarze ekranu. Na razie nie ma w tym nic interesującego. Podobny efekt można uzyskać środkami dostępnymi w języku BASIC. Najciekawsze ujawnia się dopiero w chwili usuwania okna. W jego miejsce pojawia się poprzednia zawartość ekranu.

W celu uzyskania takiego efektu musimy działać bezpośrednio na obszarze pamięci ekranu. Dlatego dokładnie musimy określić położenie okna nie we współrzędnych ekranu, lecz w terminach przesunięcia (liczby bajtów) początku okna (lewego górnego rogu) względem początku obszaru pamięci ekranu. Musimy również podać liczbę kolumn i wierszy zajmowanych przez okno.

Procedura tworzenia (otwierania) okien musi realizować następujące czynności:

— przechować, w zmiennej tekstowej, zawartość komórek pamięci obszaru ekranu, w którym będzie umieszczone okno,

— przepisać zawartość innej zmiennej tekstowej, przechowującej treść komunikatu umieszczonego w oknie, do wskazanego obszaru pamięci ekranu.

Dodatkowo musi ona wykonać jeszcze jedną, niewidoczną funkcję. Należy przekodować znaki, przechowywane w zmiennej tekstowej, z kodu ATASCII na kod wewnętrzny. Tylko w takiej postaci dadzą one czytelny tekst na ekranie.

W celu utworzenia okna musimy określić następujące parametry:

— położenie okna (lewy górny róg),

— wymiary (poziomy i pionowy),

— zawartość.

Następnie należy przechować zawartość obszaru pamięci ekranu, która będzie zakrywana przez okno. Kolejnym krokiem jest umieszczenie w tym obszarze pamięci, po uprzednim przekodowaniu, zawartości okna, którą przygotowaliśmy w zmiennej tekstowej.

Otwieranie kolejnego okna odbywa się w identyczny sposób. Usuwanie okna z ekranu (zamykanie) powoduje tylko odtworzenie poprzedniej zawartości używanego obszaru ekranu. Może to być początkowa zawartość ekranu, zawartość wcześniej utworzonego okna lub obie.

W przypadku tworzenia na ekranie wielu wzajemnie zasłaniających się okien należy pamiętać o właściwej kolejności ich zamykania, odwrotnej do otwierania.

Opisane czynności mogą być wykonane za pomocą procedury napisanej w języku BASIC, lecz czas jej realizacji może zniechęcić nas do używania okien. Dlatego najlepiej napisać ją w języku wewnętrznym komputera.

Program nr 1 zawiera dwie procedury w języku maszynowym. Pierwsza z nich, umieszczona w zmiennej tekstowej OTW$, służy do otwierania okna na ekranie. Realizuje ona następujące funkcje:

— wyznacza obszar ekranu przeznaczony na okno,

— przechowuje jego zawartość w wyznaczonym buforze,

— przekodowuje symbole umieszczone w oknie z kodu ATASCII na kod ekranu i umieszcza je w obszarze pamięci ekranu przydzielonym dla okna.

Sposób wywołania procedury tworzenia okna jest następujący:

X = USR(ADR(OTW$),A,S,W,
ADR(BUF$),ADR(O$))

gdzie: **OTW$** — zmienna tekstowa zawierająca procedurę w języku maszynowym,

A — adres lewego górnego rogu okna; adres bezwzględny komórki pamięci określającej ten róg,

S — szerokość okna w bajtach,

W — wysokość okna w liniach,

BUF$ — zmienna tekstowa służąca do przechowania zawartości obszaru ekranu zakrywanego oknem,

O$ — zmienna tekstowa zawierająca informacje wyświetlane w oknie.

Wielkości zmiennych tekstowych BUF$ i O$ muszą być jednakowe i nie mniejsze od liczby bajtów umieszczanych w obszarze okna. Nie ma żadnych ograniczeń na wielkość okna. Może ono zajmować nawet cały ekran.

Druga procedura w języku maszynowym, umieszczana w zmiennej tekstowej ZAM$, służy do zamykania okna. Realizuje ona następujące funkcje:

— wyznacza obszar ekranu przeznaczony na okno,

— odtwarza poprzednią zawartość tego obszaru na podstawie danych przechowywanych w zmiennej tekstowej.

Sposób wywołania tej procedury jest następujący:

X=USR(ADR(ZAM$),A,S,W,ADR(BUF$))

gdzie: **ZAM$** — zmienna tekstowa zawierająca procedurę w języku maszynowym,

A — adres bezwzględny lewego górnego rogu okna,

S — szerokość okna w bajtach,

W — wysokość okna w liniach,

BUF$ — zmienna tekstowa, w której przechowywana jest zawartość obszaru ekranu zakrywanego oknem.

W omawianym programie przedstawiony jest sposób przygotowania zawartości okna oraz dwa sposoby wywołania go na ekran. Za pierwszym razem za pomocą klawisza (HELP), za drugim poprzez wybranie odpowiedniej litery. Zamykanie obu okien następuje po naciśnięciu (HELP). Proszę zauważyć, że w celu zachowania pierwotnej zawartości ekranu w postaci niezmienionej, należy przestrzegać właściwej kolejności zamykania okien oraz używania tych samych zmiennych przechowujących zawartość obszaru ekranu zakrywanego oknem.

Charakterystyczną cechą prezentowanych procedur jest używanie dwóch odrębnych obszarów. Jednego do przechowywania zawartości wyświetlanego okna i drugiego do przechowywania zawartości obszaru ekranu zakrywanego oknem.

Proszę pamiętać, że obie procedury umożliwiają otwieranie i zamykanie okien tylko w trybie tekstowym (najlepiej GRAPHICS 0).

Przedstawione powyżej rozwiązanie nie jest jedynym sposobem tworzenia okien na ekranie. Poniżej prezentujemy inne. Umożliwia ono, podobnie jak poprzednie, zdefiniowanie okna o zadanej wielkości i umieszczenie go w dowolnym miejscu ekranu. Po przydzieleniu tekstu do zmiennej, to znaczy określeniu zawartości okna, następuje wymiana zawartości ciągu z wybranym obszarem pamięci ekranu. W tym celu należy wywołać dwa podprogramy w języku maszynowym. Pierwszy z nich dokonuje konwersji znaków z kodu ATASCII na kod wewnętrzny, a drugi dokonuje wymiany.

Powyższy sposób wykorzystany został w **programie nr 2.** W pierwszej kolejności umieszcza on procedurę konwersji w zmiennej tekstowej C$. Kody znaków przed i po konwersji przechowywane są w tym samym ciągu. Jeśli będzie to zmienna o nazwie D$, to wywołanie procedury kontrowersji ma następującą postać:

X=USR=(ADR(C$),ADR(D$))

Druga procedura, która dokonuje wymiany zawartości wybranego obszaru ekranu,

```
Uwaga: Litery z lewej strony
numerów linii nie są częścią
programu, lecz sumą kontrolną
ułatwiającą wprowadzenie programu
za pomocą Edytora BASIC'u (IKS nr
4 z 1987 roku). Znaki, które są
podkreślone, należy wprowadzić w
negatywie. Znaki wypisane drukiem
rozszerzonym należy wprowadzić z
wciśniętym klawiszem CONTROL.

FK 1 REM ****************
FJ 2 REM *              *
XA 3 REM * Program nr 1 *
FL 4 REM *              *
FO 5 REM ****************
NL 6 REM
XN 10 DATA 104,104,133,204,104,13
   3,203,104,104,133,205,104,104,
   170,104,133,207,104,133,206,10
   4,133,209,104,133,208
FX 20 DATA 160,0,177,203,145,206,
   177,208,72,41,128,133,210,104,
   41,127,201,32,176,4,105,64,144
   ,7,201,96,176,3,56,233
RO 30 DATA 32,5,210,145,203,200,1
   96,205,144,218,24,165,203,105,
   40,133,203,165,204,105,0,133,2
   04,24,165,206
AA 40 DATA 101,205,133,206,165,20
   7,105,0,133,207,24,165,208,101
   ,205,133,208,165,209,105,0,133
   ,209,202,16,174,96,-1
NB 50 DATA 104,104,133,204,104,13
   3,203,104,104,133,205,104,104,
   170,104,133,207,104,133,206,16
   0,0
AG 60 DATA 177,206,145,203,200,19
   6,205,144,247,24,165,203,105,4
   0,133,203,165,204,105,0,133,20
   4,24,165,206,101
FN 70 DATA 205,133,206,165,207,10
   5,0,133,207,202,16,216,96,-1
NA 100 DIM OTW$(109),ZAM$(61),E$(
   1)
AS 110 FOR I=1 TO 200:READ A:ON (
   A=-1) GOTO 120:OTW$(I,I)=CHR$(
   A):NEXT I
VC 120 FOR I=1 TO 200:READ A:ON (
   A=-1) GOTO 130:ZAM$(I,I)=CHR$(
   A):NEXT I
OC 130 ADRE=PEEK(88)+256*PEEK(89)
SS 140 O1=ADRE+5+5*40
IK 150 O2=ADRE+10+10*40
BU 160 DIM O1$(200),O2$(200),BUF1
   $(200),BUF2$(200)
PF 200 O1$(1,7)="QRRRRRR"
HZ 204 O1$(8,14)="RRRRRRR"
MV 208 O1$(15,20)="RRRRRE"
JI 210 O1$(21,40)="|   WYKAZ KOME
   ND   |"
QI 220 O1$(41,47)="ARRRRRR"
DO 224 O1$(48,54)="RRRRRRR"
PJ 228 O1$(55,60)="RRRRRD"
MI 230 O1$(61,80)="| A=Dodaj  C=Z
   mien |"
ZG 240 O1$(81,100)="| D=Kasuj  E=
   Koniec|"
JA 250 O1$(101,120)="| P=Drukuj H
   =Pomoc |"

XF 255 O1$(121,127)="ZRRRRRR"
TJ 257 O1$(128,134)="RRRRRRR"
RH 258 O1$(135,140)="RRRRRC"
PX 260 O2$(1,7)="QRRRRRR"
IR 264 O2$(8,14)="RRRRRRR"
NN 268 O2$(15,20)="RRRRRE"
EG 270 O2$(21,40)="|     DODAJ
   |"
RA 280 O2$(41,47)="ARRRRRR"
EG 284 O2$(48,54)="RRRRRRR"
QB 288 O2$(55,60)="RRRRRD"
LX 290 O2$(61,80)="|Tekst wyjasni
   ajacy|"
RK 300 O2$(81,100)="| działanie k
   omendy|"
GY 310 O2$(101,120)="|     DODAJ
   |"
EK 320 O2$(121,127)="ARRRRRR"
TB 324 O2$(128,134)="RRRRRRR"
SG 328 O2$(135,140)="RRRRRD"
YP 330 O2$(141,147)="ZRRRRRR"
UZ 334 O2$(148,154)="RRRRRRR"
TG 338 O2$(155,160)="RRRRRC"
SK 1000 POSITION 0,0:IF PEEK(732)
   <>17 THEN 1000
NQ 1005 POKE 732,0
PD 1010 X=USR(ADR(OTW$),O1,20,6,A
   DR(BUF1$),ADR(O1$))
TY 1020 INPUT E$:IF E$<>"A" THEN
   2000
UN 1030 X=USR(ADR(OTW$),O2,20,7,A
   DR(BUF2$),ADR(O2$))
XG 1040 IF PEEK(732)<>17 THEN 104
   0
HD 1050 POKE 732,0:X=USR(ADR(ZAM$
   ),O2,20,7,ADR(BUF2$))
YT 2000 X=USR(ADR(ZAM$),O1,20,6,A
   DR(BUF1$))
MS 2010 GOTO 1000


FK 1 REM ****************
FJ 2 REM *              *
XU 3 REM * Program nr 2 *
FL 4 REM *              *
FO 5 REM ****************
NL 6 REM
MX 100 GRAPHICS 0:POKE 752,1:JA=1
   20
CK 110 DIM A$(961),D$(113),B$(241
   ),W$(116),C$(61)
FT 120 ? :? "WCZYTANIE W$":FOR I=
   1 TO 116:READ IT:W$(I,I)=CHR$(
   IT):NEXT I
LU 130 ? :? "WCZYTANIE C$":FOR I=
   1 TO 61:READ IT:C$(I,I)=CHR$(I
   T):NEXT I
GZ 140 A$=".":A$(960)=A$:A$(2)=A$
   :A$(961)=CHR$(0)
PE 150 A$(131,154)=" Program  de
   monstruje "
HC 160 A$(171,194)="sposob tworze
   nia okien "
CS 170 D$="**************  OKN
   A    **mogą pokazać** co chce
   sz ** "
UE 180 D$(63,113)="i      **    G
   DZIE   ** chcesz    *********
   ******"
IA 190 B$="_":B$(240)=B$:B$(2)=B$
   :B$(241)=CHR$(0)
XV 200 B$(35,55)=" Podprogramy te
    można  "
JD 210 B$(92,119)="używać w tryba
   ch tekstowych"
UU 220 B$(155,176)="Okna mogą być
    dowolne "
HK 230 B$(184,208)="w każdym miej
   scu ekranu"
AO 240 GRAPHICS 0:POKE 752,1:?
WF 250 X=USR(ADR(C$),ADR(A$))
YY 260 X=USR(ADR(C$),ADR(D$))
XG 270 X=USR(ADR(C$),ADR(B$))
DC 280 SOUND 0,0,2,2:GOSUB 370:GO
   SUB 340
FI 290 SOUND 0,2,2,2:GOSUB 340:GO
   SUB 370
FW 300 IF LG=0 THEN LIST 470,670:
   LG=1
PE 310 LICZ=LICZ+1:IF LICZ>=4 THE
   N GOSUB 380:GOSUB 390:GOSUB 38
   0:JA=10
DL 320 IF LICZ>=7 THEN GRAPHICS 0
   :? :? "KONIEC PRACY":END
PI 330 GOTO 280
ZE 340 IF JA=10 THEN L=41
BR 350 IF JA=120 THEN L=39
XE 360 X=USR(ADR(W$),ADR(D$),412,
   8,14,L):GOSUB 390:RETURN
CC 370 X=USR(ADR(W$),ADR(A$),0,24
   ,40,40):GOSUB 390:RETURN

FG 380 X=USR(ADR(W$),ADR(B$),285,
   8,30,40):GOSUB 390:RETURN
DP 390 POKE 540,JA:SOUND 0,0,0,0
WY 400 IF PEEK(540) THEN 400
```

```
ZC 410 RETURN
QG 420 DATA 104,104,133,206,104,1
   33,205
SG 430 DATA 104,141,1,6,104,141,0
SO 440 DATA 6,104,133,208,104,133
   ,207
LH 450 DATA 104,104,133,209,104,1
   41,3
NG 460 DATA 6,104,141,2,6,165,88
AO 470 DATA 133,203,165,89,133,20
   4,24
RF 480 DATA 165,203,109,0,6,133,2
   03
TZ 490 DATA 165,204,109,1,6,133,2
   04
AI 500 DATA 160,0,177,203,170,177
   ,205
GB 510 DATA 145,203,138,145,205,2
   00,196
LI 520 DATA 209,208,241,169,0,198
   ,207
KF 530 DATA 197,207,208,6,197,208
   ,240
SB 540 DATA 30,198,208,24,152,101
   ,205
CL 550 DATA 133,205,144,3,230,206
   ,24
SU 560 DATA 165,203,109,2,6,133,2
   03
VO 570 DATA 165,204,109,3,6,133,2
   04
MZ 580 DATA 24,144,197,96
MF 590 DATA 104,104,133,204,104,1
   33,203
KA 600 DATA 160,0,177,203,170,201
   ,0
MD 610 DATA 240,44,41,96,201,0,20
   8
ID 620 DATA 6,138,9,64,24,144,22
AQ 630 DATA 201,96,240,20,138,41,
   64
KJ 640 DATA 201,0,208,6,138,73,32
MA 650 DATA 24,144,5,138,9,32,73
WX 660 DATA 64,145,203,200,208,21
   0,230
HN 670 DATA 204,24,144,203,96


FK 1 REM ****************
FJ 2 REM *                *
YO 3 REM * Program nr 3 *
FL 4 REM *                *
FO 5 REM ****************
NL 6 REM
SH 100 GRAPHICS 0:POKE 752,1:POKE
    82,4
AD 110 ? :? "         OKNA TEKSTOW
   E "
RL 120 ? :? "     W TRYBACH GRAFI
   CZNYCH"
LJ 130 DIM W$(116),KOM$(11),ROB$(
   11*16),E$(263)
BN 140 ? :? "WCZYTANIE W$"
ME 150 FOR I=1 TO 116:READ A:W$(I
   ,I)=CHR$(A):NEXT I
ND 160 ? :? "WCZYTANIE E$"
VL 170 FOR I=1 TO 263:READ A:E$(I
   ,I)=CHR$(A):NEXT I
WR 180 G=8:W=40:GRAPHICS G+16:GOS
   UB 260
WM 190 G=7:W=40:GRAPHICS G+16:GOS
   UB 260
VO 200 G=6:W=40:GRAPHICS G+16:GOS
   UB 260
UN 210 G=5:W=20:GRAPHICS G+16:GOS
   UB 260
TX 220 G=4:W=10:GRAPHICS G+16:GOS
   UB 260
TS 230 G=3:W=10:GRAPHICS G+16:GOS
   UB 260
ZW 240 GRAPHICS 0:POKE 82,2:? :?
   "KONIEC PROGRAMU":END
NL 250 REM PODPROGRAM WYSWIETLANI
   A KOMUNIKATOW
JA 260 SETCOLOR 0,9,10:SETCOLOR 1
   ,3,8:SETCOLOR 2,9,0:SETCOLOR 4
   ,1,2
II 270 REM ZEROWANIE CIAGU
UB 280 ROB$=CHR$(0):ROB$(11*16)=R
   OB$:ROB$(2)=ROB$:KOM$=ROB$
OB 290 FOR L=1 TO 3
BE 310 KOM$="GR.? Typ  ":KOM$(4,4
   )=CHR$(G+48):KOM$(10,10)=CHR$(
   L+48):KOM$(11,11)=CHR$(0)
SS 320 REM KONWERSJA NA FORMAT GR
   AFICZNY
VH 330 X=USR(ADR(E$),ADR(ROB$),AD
   R(KOM$),LEN(KOM$)-1,L)
SX 340 REM UMIESZCZENIE NA EKRANI
   E
FT 350 X=USR(ADR(W$),ADR(ROB$),W*
   8*(L-1),8,20,W)
RA 360 REM
NT 370 SOUND 0,L*75,14,5:X=25*L:P
   OKE 20,0
DA 380 A=PEEK(20):POKE 708,45:IF
   A<X THEN POKE 708,40:GOTO 380
AA 390 POKE 708,40:SOUND 0,0,0,0:
   NEXT L:RETURN

TT 400 REM DANE DLA W$
OG 410 DATA 104,104,133,206,104,1
   33,205,104,141,1,6,104,141,0,6
NZ 430 DATA 104,133,208,104,133,2
   07,104,104,133,209,104,141,3,6
   ,104,141,2,6
AT 450 DATA 165,88,133,203,165,89
   ,133,204,24,165,203,109,0,6,13
   3
JA 470 DATA 203,165,204,109,1,6,1
   33,204,160,0,177,203,170,177,2
   05,145
OC 490 DATA 203,138,145,205,200,1
   96,209,208,241,169,0,198,207,1
   97,207,208
MU 510 DATA 6,197,208,240,30,198,
   208,24,152,101,205,133,205,144
   ,3,230
XF 530 DATA 206,24,165,203,109,2,
   6,133,203,165,204,109,3,6,133,
   204,24,144,197,96
HU 560 REM DANE DLA E$
BK 570 DATA 104,104,133,204,104,1
   33,203,104,133,212,104,133,211
   ,104,104,133
MK 590 DATA 216,10,133,217,104,10
   4,41,3,24,106,106,106,41,192,1
   33,207
XX 610 DATA 133,208,169,0,133,219
   ,169,4,133,220,169,0,168,145,2
   19,200
GL 630 DATA 192,16,208,249,169,25
   5,133,215,160,0,177,211,201,12
   8,144,4
TV 650 DATA 132,215,73,128,201,96
   ,144,4,160,3,176,19,201,64,144
   ,4
RS 670 DATA 160,1,176,11,201,32,1
   44,5,176,5,24,144,205,160,2,41
JA 690 DATA 31,10,10,10,133,213,2
   4,152,109,244,2,133,214,160,0,
   132
RU 710 DATA 209,132,210,162,0,134
   ,205,164,210,177,213,133,206,1
   64,209,165
BJ 730 DATA 206,48,6,177,219,5,20
   7,145,219,24,102,207,24,102,20
   7,6
IZ 750 DATA 206,232,224,4,208,231
   ,165,215,240,5,56,241,219,145,
   219,165
HE 770 DATA 208,133,207,230,209,2
   30,205,165,205,201,2,208,208,2
   30,210,164
UG 790 DATA 210,192,8,208,190,240
   ,3,24,144,160,165,203,72,165,2
   04,72
EN 810 DATA 160,0,177,219,145,203
   ,200,192,2,208,247,177,219,170
   ,200,177


XA 830 DATA 219,72,200,132,218,16
   4,217,138,145,203,104,200,145,
   203,24,165
LA 850 DATA 203,101,217,133,203,1
   44,2,230,204,164,218,192,16,20
   8,220,104
JB 870 DATA 133,204,24,104,105,2,
   133,203,144,2,230,204,230,211,
   208,2
KP 890 DATA 230,212,198,216,208,1
   77,96


FK 1 REM ****************
FJ 2 REM *                *
ZI 3 REM * Program nr 4 *
FL 4 REM *                *
FO 5 REM ****************
NL 6 REM
SH 100 GRAPHICS 0:POKE 752,1:POKE
    82,4
AD 110 ? :? "         OKNA TEKSTOW
   E "
RL 120 ? :? "     W TRYBACH GRAFI
   CZNYCH"
WB 130 DIM W$(116),KOM$(11),ROB$(
   22*16),E$(263),A$(11*16)
BN 140 ? :? "WCZYTANIE W$"
ME 150 FOR I=1 TO 116:READ A:W$(I
   ,I)=CHR$(A):NEXT I
ND 160 ? :? "WCZYTANIE E$"
VL 170 FOR I=1 TO 263:READ A:E$(I
   ,I)=CHR$(A):NEXT I
EB 175 G=15:W=40:GRAPHICS G+16:GO
   SUB 260
CY 176 G=14:W=20:GRAPHICS G+16:GO
   SUB 260
WR 180 G=8:W=40:GRAPHICS G+16:GOS
   UB 260
WM 190 G=7:W=40:GRAPHICS G+16:GOS
   UB 260
US 200 G=6:W=20:GRAPHICS G+16:GOS
   UB 260
UN 210 G=5:W=20:GRAPHICS G+16:GOS
   UB 260
TX 220 G=4:W=10:GRAPHICS G+16:GOS
   UB 260
TS 230 G=3:W=10:GRAPHICS G+16:GOS
   UB 260
TW 240 GRAPHICS 0:POKE 82,2:? :?
   "KONIEC PROGRAMU":END
NL 250 REM PODPROGRAM WYSWIETLANI
   A KOMUNIKATOW
GK 260 SETCOLOR 0,9,10:SETCOLOR 1
   ,3,6:SETCOLOR 2,9,0:SETCOLOR 4
   ,1,2
II 270 REM ZEROWANIE CIAGU
UZ 275 COLOR 1:PLOT W*3,0:FOR I=1
    TO 40:X=RND(0)*W*4:Y=RND(0)*W
   *2:DRAWTO INT(X),INT(Y):NEXT I
VW 280 ROB$=CHR$(0):ROB$(22*16)=R
   OB$:ROB$(2)=ROB$:KOM$=ROB$
OB 290 FOR L=1 TO 3
IZ 300 REM TWORZENIE OKNA
LX 305 A$=CHR$(0):A$(11*16)=A$:A$
   (2)=A$
MP 310 KOM$="Gr. ":KOM$(6-LEN(STR
   $(G)))=STR$(G)
ME 315 X=USR(ADR(E$),ADR(A$),ADR(
   KOM$),LEN(KOM$),L)
KM 320 ROB$=A$
ZG 325 WSK=(LEN(KOM$)-0)*16+1

MN 330 KOM$="Typ ":KOM$(5)=STR$(L
   )
MD 335 A$=CHR$(0):A$(11*16)=A$:A$
   (2)=A$
LV 340 X=USR(ADR(E$),ADR(A$),ADR(
   KOM$),LEN(KOM$),L)
XX 345 ROB$(WSK)=A$
XC 360 REM UMIESZCZANIE OKNA
OD 370 FOR I=1 TO 5
UC 380 X=USR(ADR(W$),ADR(ROB$),W*
   8+W/2-5,16,LEN(KOM$)*2,W)
XM 385 FOR J=15 TO 0 STEP -15/200
AG 386 SOUND 0,L*75,14,J
HO 387 NEXT J
CE 390 NEXT I:NEXT L:RETURN
BV 400 REM PODPROGRAM UMIESZCZANI
   A OKNA
OG 410 DATA 104,104,133,206,104,1
   33,205,104,141,1,6,104,141,0,6
NZ 430 DATA 104,133,208,104,133,2
   07,104,104,133,209,104,141,3,6
   ,104,141,2,6
AT 450 DATA 165,88,133,203,165,89
   ,133,204,24,165,203,109,0,6,13
   3
JA 470 DATA 203,165,204,109,1,6,1
   33,204,160,0,177,203,170,177,2
   05,145
OC 490 DATA 203,138,145,205,200,1
   96,209,208,241,169,0,198,207,1
   97,207,208
MU 510 DATA 6,197,208,240,30,198,
```

```
    208,24,152,101,205,133,205,144
    ,3,230
XF 530 DATA 206,24,165,203,109,2,
    6,133,203,165,204,109,3,6,133,
    204,24,144,197,96
CR 560 REM PODPROGRAM KONWERSJI Z
    NAKOW NA FORMAT CZTEROKOLOROWY
BK 570 DATA 104,104,133,204,104,1
    33,203,104,133,212,104,133,211
    ,104,104,133
MK 590 DATA 216,10,133,217,104,10
    4,41,3,24,106,106,106,41,192,1
    33,207
XX 610 DATA 133,208,169,0,133,219
    ,169,4,133,220,169,0,168,145,2
    19,200
GL 630 DATA 192,16,208,249,169,25
    5,133,215,160,0,177,211,201,12
    8,144,4
TV 650 DATA 132,215,73,128,201,96
    ,144,4,160,3,176,19,201,64,144
    ,4
RS 670 DATA 160,1,176,11,201,32,1
    44,5,176,5,24,144,205,160,2,41
JA 690 DATA 31,10,10,10,133,213,2
    4,152,109,244,2,133,214,160,0,
    132
RU 710 DATA 209,132,210,162,0,134
    ,205,164,210,177,213,133,206,1
    64,209,165
BJ 730 DATA 206,48,6,177,219,5,20
    7,145,219,24,102,207,24,102,20
    7,6
IZ 750 DATA 206,232,224,4,208,231
    ,165,215,240,5,56,241,219,145,
    219,165
HE 770 DATA 208,133,207,230,209,2
    30,205,165,205,201,2,208,208,2
    30,210,164
UG 790 DATA 210,192,8,208,190,240
    ,3,24,144,160,165,203,72,165,2
    04,72
EN 810 DATA 160,0,177,219,145,203
    ,200,192,2,208,247,177,219,170
    ,200,177
XA 830 DATA 219,72,200,132,218,16
    4,217,138,145,203,104,200,145,
    203,24,165
LA 850 DATA 203,101,217,133,203,1
    44,2,230,204,164,218,192,16,20
    8,220,104
JB 870 DATA 133,204,24,104,105,2,
    133,203,144,2,230,204,230,211,
    208,2
KP 890 DATA 230,212,198,216,208,1
    77,96
```

przechowywana jest w zmiennej W$. Wywołujemy ją następująco:

X=USR(ADR(W$), ADR(D$), O,W,S,L)

gdzie: **O** — wielkość przesunięcia lewego górnego rogu okna względem lewego górnego rogu ekranu (w bajtach),

W — liczba linii wyświetlanego tekstu (wysokość okna),

S — szerokość okna w znakach,

L — przesunięcie między początkiem każdej kolejnej linii, dla okna prostokątnego, w trybie GR.0,L powinno być równe 40, używając L=39 lub L=41 otrzymamy okno pochylone w jedną lub drugą stronę.

Podczas używania tych procedur należy pamiętać, aby w instrukcji USR dostarczyć odpowiednią liczbę parametrów. Nieprzestrzeganie tego może zablokować komputer.

Program nr 2 demonstruje kilka możliwych sposobów używania okien. Proszę zwrócić uwagę na wykorzystanie tej techniki do szybkiej wymiany zawartości całego ekranu. W tym celu zmienna tekstowa musi mieć wielkość 960 bajtów.

Powyższe procedury w prezentowanej postaci mogą być używane tylko w trybie tekstowym. W związku z tym interesujące może być szybkie wyświetlenie tekstu, najlepiej kolorowego, w dowolnym z trybów graficznych. Wtedy uwolnimy się od konieczności używania okna tekstowego, któremu na stałe przydzielone są cztery dolne linie ekranu.

Idea tworzenia okien w trybach graficznych jest identyczna jak w tekstowych. Różnica w realizacji polega na konwersji znaków. W trybie tekstowym należało zamienić kody ATASCII na kody ekranu. W trybie graficznym należy zamienić kody znaków na ich reprezentacje bitowe. Wykonuje to procedura przechowywana w zmiennej E$ (program nr 3).

Wywołujemy ją następująco:

X = USR(ADR(E$), ADR(ROB$),ADR (KOM$),LEN(KOM$)—1,K)

gdzie **E$** — zmienna tekstowa zawierająca procedurę konwersji,

ROB$ — pomocnicza zmienna tekstowa służąca do przechowywania reprezentacji bitowej znaków tekstu; musi być 16 razy dłuższa od tekstu,

KOM$ — zmienna tekstowa przechowująca wyświetlany tekst,

K — kod koloru wybrany dla okna.

Wyświetlenie okna na ekranie wykonujemy tą samą procedurą co w trybie tekstowym. Występują jednak pewne różnice w znaczeniu parametrów. Wynikają one z faktu, że ciąg do wyświetlania zawiera reprezentacje bitowe znaków. A każdy punkt elementarny definiowany jest przez dwa bity. Na jedną linię wyświetlanych symboli przypada 8 linii obrazu, których grubość zależy od używanego trybu graficznego.

Sposób wywołania tej procedury jest następujący:

X=USR(ADR(W$),ADR(ROB$),O,W,S,L)

gdzie: **O** — wielkość przesunięcia lewego górnego rogu okna względem lewego górnego rogu ekranu (w bajtach),

W — wysokość okna (w liniach obrazu),

S — szerokość linii w bajtach (ilość znaków w linii razy dwa),

L — przesunięcie między początkiem każdej kolejnej linii, jego wielkość zależy od używanego trybu graficznego.

Tworzenie okien w trybach graficznych nie jest, niestety, takie proste jak w tekstowych. Pewne komplikacje spowodowane są koniecznością konwersji ciągu znaków na jego odwzorowanie bitowe. Z tego powodu każdą linię tekstu umieszczonego w oknie musimy poddawać konwersji oddzielnie. Następnie ciąg wynikowy umieszczamy w zmiennej tekstowej, której zawartość zostanie wyświetlona w oknie. Wysokość okna musi być podana w liniach obrazu (8 linii na jedną linię tekstu). Szerokość linii musi być dwa razy większa od liczby wyświetlanych znaków, ponieważ każdy punkt elementarny opisywany jest za pomocą dwóch bitów.

W celu zapoznania się ze sposobami tworzenia okien należy dokładnie przeanalizować program nr 3, który wyświetla na ekranie okna zawierające jedną linię tekstu w rozmaitych trybach graficznych. Z kolei program nr 4 demonstruje sposób tworzenia okien zawierających dwie linie tekstu (także w trybach 14 i 15). W podobny sposób tworzy się okna zawierające trzy lub więcej linii tekstu.

Musimy przyznać, że nie udało się nam uzyskać spodziewanych efektów tworzenia okien w trybach graficznych. Jednak ze względu na nieprzekraczalny termin złożenia w redkacji kolejnego odcinka cyklu, wyniki naszych dalszych prac zamieścimy w następnym numerze „IKS-a".

Podczas przygotowywania tego odcinka korzystaliśmy z artykułów zamieszczonych w czasopiśmie ANTIC.

Tomasz MROWIEC
Ludwik PIELA

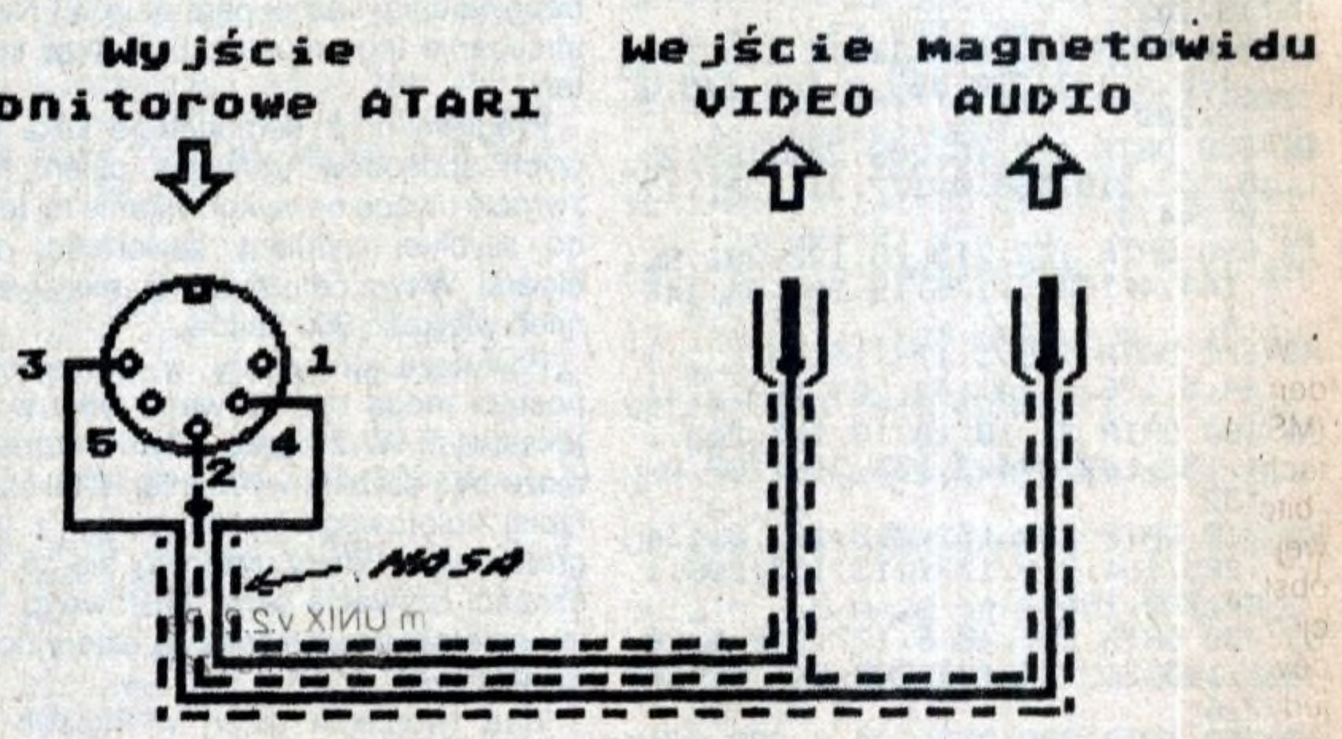
Atari

J. Janiec

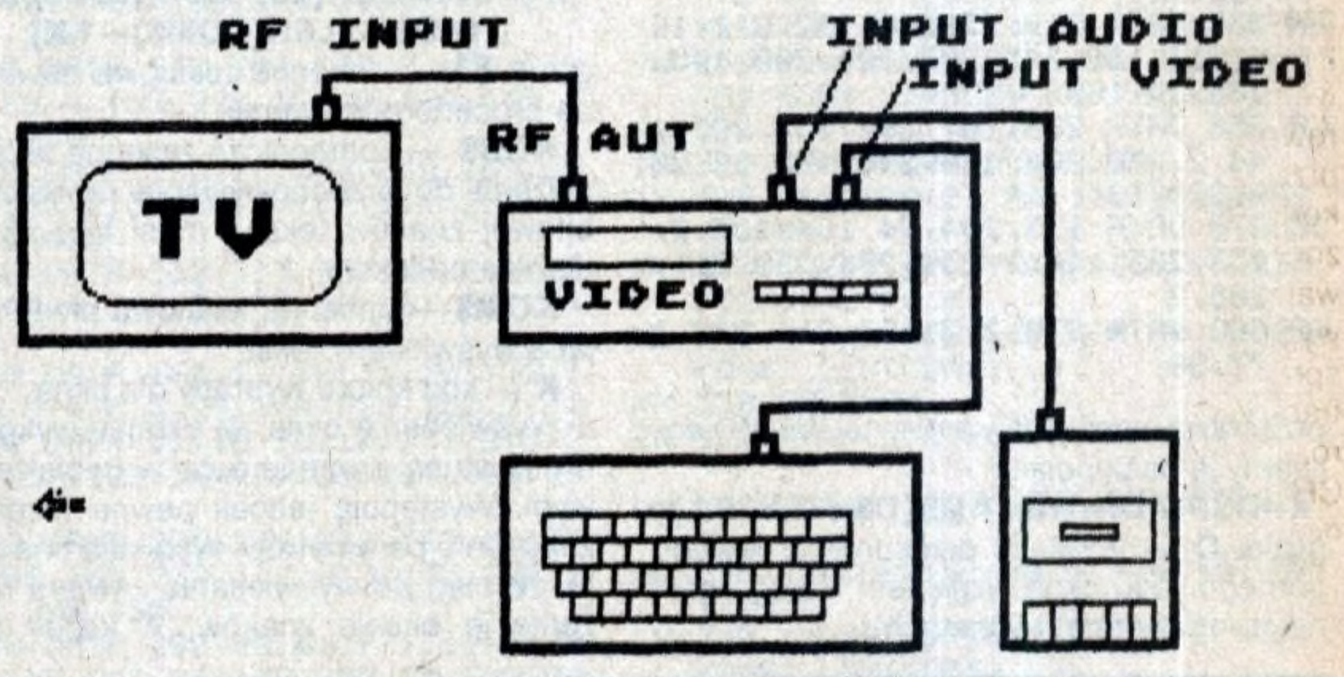
```
PG 500 REM **************************
YA 502 REM *                        *
HT 504 REM *       RACHUNKI         *
DI 505 REM *      *  Atari *        *
YP 507 REM *                        *
QH 509 REM **************************
BU 510 DIM B$(11),D$(3),U$(13)
QB 515 PRINT CHR$(125):P=0
WR 520 ? :? "R A C H U N K I"
FF 522 ? "==============="
YB 525 B$="B R A W O !":U$="P O P R
    A W !"
SM 530 FOR K=1 TO 10:D=INT(RND(0)*3
    )+1:A=INT(RND(0)*7)+2:B=INT(RND(
    0)*8)+1
DF 540 IF B>A THEN A=B
UE 545 IF D=1 THEN X=A+B:D$=" + "
XT 550 IF D=2 THEN X=A-B:D$=" - "
TE 555 IF D=3 THEN X=A*B:D$=" * "
VF 560 ? :? "POLICZ"
SB 565 ? :? A;D$;B;" = ";:INPUT Y:G
    OSUB 630
WW 570 IF X=Y THEN ? :? B$:GOTO 590
DQ 580 P=P+1:? :? U$:GOTO 560
HH 590 NEXT K
NT 600 ? :? P;" POPRAWEK"
RY 610 D$="":? :? "Jeszcze raz ?"
DW 615 ? :? "          ";CHR$(212);"/
    ";CHR$(206);:INPUT D$
TX 620 IF D$="T" THEN 515
ER 625 END :REM ********************
HD 630 FOR D=10 TO 1 STEP -1:SOUND
    0,100,10,D:NEXT D:SOUND 0,0,0,0
ZX 635 RETURN
```

KOMPUTER I VIDEO

W wielu domach, obok komputera drugim ważnym przedmiotem użytkowym jest magnetowid. Oba te sprzęty egzystują równolegle nie „wchodząc" sobie w drogę. Tylko czy tak musi być? Jako pierwsze wspólne ich przedsięwzięcie proponuję wykonać czołówkę do swoich kaset, coś w rodzaju exlibrisu. Zrobić to można za pomocą dowolnego komputera, a skopiować na każdy magnetowid. Nim do tego przystąpimy, powinniśmy wykonać kabel połączeniowy między komputerem a magnetowidem. Typy wtyków będą różne i zależne od stosowanego sprzętu. W wypadku stosowania do tworzenia grafiki komputera Atari, na jednym końcu kabla powinniśmy przylutować wtyk stereofoniczny GM-5, a w wypadku Spectrum typu mini „Jack". Drugi koniec kabla zaopatrujemy w dwa wtyki WW-1 (SANYO, FISHER) lub po jednym WW-1 i BNC (NEC). Długość kabla powinna być oczywiście tak dobrana by pozwoliła w sposób wygodny korzystać z zestawu, tzn. 1,2÷1,5 m. Schemat kabla do połączenia komputera Atari i dowolnego magnetowidu przedstawiono na rysunku 1.



To byłoby wszystko, jeśli chodzi o podłączenie komputera do magnetowidu. Drugi problem, to stworzenie czołówki „exlibrisu" swoich taśm. Metod jest wiele. Jedną z nich może być rozwiązanie programowe, to znaczy stworzenie programu z częścią graficzną, tekstową i ewentualnie dźwiękową. Nie jest to czynność łatwa, ale z pewnością dająca dużo satysfakcji. Inny sposób stworzenia czołówki, to wykorzystanie jednego z wielu programów graficznych. Tu chciałbym zaproponować program pod tytułem „MAGIC-PAITER". W przeciwieństwie do innych programów najczęściej o statycznej grafice, można stworzyć za jego pomocą efekt „płynącej tęczy". Może on zostać zwielokrotniony, gdy do kreślenia wykorzystamy kilka kolorów. Każdy z kolorów będzie tworzył odrębną „tęczę". Aby uzyskać w tym programie efekt „płynącej tęczy", należy po otrzymaniu na ekranie monitora zestawu możliwości, zainicjować Multicolor-effect, a z niego np. Set Color 0. Następnie z opcji szerokość linii (Select breadth of Stripes) wybrać dla przykładu najszerszą linię (Very wide Stripes) i dalej z wyboru rodzaju linii (Select kind of Stripes), przewijanie w górę (Stripes up). Teraz pozostaje nam tylko powrót do opcji Multicolor-effect i uruchomienie swojego dzieła za pomocą Start Effect. Jedynym mankamentem tego i podobnych programów jest brak dźwięku. Upartym, udźwiękowienie czołówki też nie powinno sprawić większego kłopotu. Potrzebne są w tym celu trzy rzeczy: magnetofon, program muzyczny i przewód koncentryczny do połączenia magnetofonu z magnetowidem i komputerem. Do utrwalenia stworzonej w komputerze melodii można użyć każdego magnetofonu kasetowego lub



szpulowego z wejściem na mikrofon i wyjściem do przegrywania. Melodię można stworzyć dowolnym programem muzycznym lub wykorzystać gotowe melodie. Z gotowych programów polecamy np: Music, Organy, Kompozytor.

Kiedy mamy przygotowany obraz i melodię, możemy przystąpić do przekopiowywania ich na taśmę magnetowidową. Przedtem jednak musimy stworzyć układ do kopiowania obrazu i dźwięku, którego schemat przedstawiono poniżej.

Życzymy powodzenia.

Janusz JANIEC

Chochlik dał znać o sobie. Na ostatniej stronie 6 nr „IKS-a" popełniliśmy błąd. Otóż w tekście „Też komputery..." pojemność informacyjna „kostki" wynosi oczywiście nie 32 bity a 32 Kb i analogicznie nie 64 bity a 64 Kb.

Przepraszamy!

Transputery — rodzina INMOS

Układy rodziny transputerowej zostały opracowane w pracowniach brytyjskiej firmy INMOS, znanej dotychczas z produkcji pamięci statycznych i dynamicznych RAM, wykonanych przy użyciu technologii 1,6 μ lub 1,2 μ CMOS. Dzisiaj najważniejszą ofertą INMOS są układy rodziny transputerowej oraz szeroki asortyment kart, modułów, pakietów i systemów rozszerzania układów transputerowych i mikrokomputerów PC. Przy ich produkcji INMOS stosuje 1,5 μ technologii CMOS. Oferta firmy obejmuje także sterowane mikroprocesorowo układy specjalistyczne. Są one często mylnie zaliczane do rodziny transputerów. Należą do nich kaskadowe, 32-stopniowe filtry cyfrowe (IMS A100) stosowane m.in w analizatorach stanów oraz specjalne przetworniki analogowo-cyfrowe (IMS G17x) dla potrzeb kolorowej grafiki komputerowej, systemów wideo i TV. Produkty INMOS bazujące na transputerach zestawiono w tabeli nr 1.

W poprzednim odcinku przedstawiłem jeden z nich, adapter linii komunikacyjnych IMS C012, dziś kilka słów o innych produktach. **IMS M212** jest specjalizowanym 16-bitowym transputerem o mocy obliczeniowej równej 10 MIPS[1], przeznaczonym do obsługi dysków typu WINCHESTER lub stacji dysków elastycznych łącznie do 4 napędów. IMS M212 zawiera w jednym układzie jednostkę procesora, 4 KB ROM, 2 KB RAM, kontroler dysku, układy interfejsów oraz system sterujący. Dodatkowo poprzez interfejs pamięci zewnętrznej może obsługiwać 64 KB DRAM. W pamięci ROM rezyduje system wspomagania obsługi napędu dyskowego, natomiast RAM wykorzystuje się jako bufor sektora lub dla zapamiętania parametrów, formatu zapisu i odczytu danych. Ponadto w RAM zapisuje się program przetwarzający dane z linii komunikacyjnych łączących M212 bezpośrednio z innymi transputerami lub poprzez adapter z konwencjonalnymi systemami mikroprocesorowymi. Trzeci interfejs jest sprzężony poprzez 2 jednokierunkowe, 8-bitowe porty z kontrolerem dysku. Kontroler ten wyposażony jest w dwa dwukierunkowe porty, używane też do kontroli takich funkcji dyskowych, jak ustawienie początkowe, selekcja, sprawdzanie słowa statusu itp. Możliwości transputera M212 wykorzystano w karcie rozszerzenia IMS B005, zawierającej WINCHESTER i dysk elastyczny oraz opcjonalnie dołączane duże stacje dysków poprzez interfejsy dodatkowo buforowane według standardu SA400/ST506.

IMSB007 jest podwójną Eurokartą przeznaczoną do realizacji funkcji graficznych. Zawiera ona 32-bitowy transputer IMS T414, wspomniany przetwornik IMS G170, 512 KB DRAM i podwójny port (512 KB) wideo RAM oraz buforowanie linii komunikacyjnych i wyjście RGB. Dodatkowe wspomaganie karty poprzez software w OCCAM zapewnia realizację podstawowych funkcji graficznych (przy kreśleniu linii i wielokątów) oraz generowanie alfanumerycznych tekstów ze skalowaniem. Obraz jest mapowany w pamięci z wykorzystaniem podwójnego ekranu dla szybkiego odświeżania zawartości wyświetlanego obrazu. Karta zapewnia wybór 262 lub 144 kodów umożliwiając grafikę 512x512 przy 8-bitowym kodzie opisującym dowolny punkt i jest przeznaczona dla monitorów o częstotliwości wybierania linii w zakresie od 20 do 50 kHz.

Moduły podstawowe, karty rozszerzania, pakiety specjalistyczne, a także urządzenia i systemy bazujące na technice transputerów wyposażone w odpowiednie oprogramowanie można znaleźć w ofertach wielu firm m.in. Transtech, Meiko, Sension, MikroWay, Nichie.

Kompletne oprogramowanie INMOS może być używane łącznie z systemem operacyjnym UNIX v.2.2. Pakiet stanowi integralną część systemu i zawiera transputer T1414 z 32-bitową magistralą lokalną, do której podłączono 2 MB dodatkowej pamięci DRAM i 64-256 KB EPROM oraz podwójny układ uniwersalnych ansychronicznych nadajników-odbiorników linii (DUART), dwa szeregowe porty RS232 oraz zwrotnicę linii komunikacyjnych. Karta posiada dwa złącza umożliwiające opcjonalne dołączanie modułów kompatybilnych z IMS B0xx. Ponadto na karcie zainstalowano niezbędne elementy i bloki dla systemu VME oraz adapter linii, który sprzęga (łącznie z DRAM) środowisko transputerowe z magistralą systemową.

Sension jest producentem transputerowych urządzeń TES, znacznie zwiększających parametry użytkowe i moc obliczeniową mikrokomputerów. TES są przeznaczone do bezpośredniego połączenia z mikrokomputerami IBM PC XT/AT i kompatybilnymi, a także stosując odpowiednie adaptery z innymi komputerami PC. Można w nich zainstalować maksymalnie 4 transputery T414-20 (opcjonalnie T800-20 i 4 MB DRAM. Ponadto firma oferuje system PARALLAX umożliwiając współbieżną wieloprocesorową pracę wykorzystując magistralę VME. PARALLAX jest uniwersalną otwartą konstrukcją pozwalającą w znacznym zakresie na konfigurowanie systemu oraz dalsze rozwijanie przy jednoczesnym bieżącym testowaniu poprawności współdziałania poszczególnych jego elementów. Cały system składa się z pewnej ilości nie powiązanych ze sobą subsystemów budowanych w oparciu o podwójne Eurokarty. Każdy subsystem może zawierać do 5 kart bazujących na 32-bitowych transputerach, PARALLAX jest wyposażony w moduły przetwarzające, pamięć masową, karty buforowanych interfejsów RS422 oraz optyczne linie komunikacyjne (dla transmisji danych), szybkie przetworniki analogowo-cyfrowe, pakiety zasilania, monitorowanie i analizy. Komunikacja jest trzypoziomowa. Dla realizacji sieci transputerowych dynamicznie rekonfigurowanych system posiada kartę zespołów programowanych 32-liniowych łącznie elektronicznych (IMS C004), adaptery i inne niezbędne elementy sieciowe.

W październiku 1987 roku powstało amerykańsko-brytyjskie konsorcjum NICHE. Produkty NICHE mają stanowić szereg integralnych systemów o budowie modularnej. Pierwszym pakietem, który opracowała Nichie Technology Limited jest rozwinięty system obliczeniowy NT-1000 ACP (rys. 2). Jest on przeznaczony do bezpośredniego włączenia w hardware amiteksurę mikrokomputerowych stacji roboczych Sun-3 i Sun-4 produkowanych przez Sun Microsystem. NT-1000 zawiera szeregowe i równoległe porty I/O, integralny szybki kontroler portów, złącza dla 32 modułów, pakiet konfiguracji sieci (na bazie układów C004) oraz interfejsy linii komunikacyjnych INMOS. Moduły posiadają 32-bitowy transputer i do 16 MB DRAM. Moc obliczeniowa stacji roboczych dla modelu Sun-3/160 wynosi 2 MIPS, Sun-3/260 — 4 MIPS, Sun-4/10 MIPS, zaś maksymalna moc obliczeniowa bloku NT/1000 jest równa 320 MIPS (48 MFLOPS). Zadanie wykonywane przez stację Sun-3/10 w ciągu 8 godzin przy zastosowaniu bloku NT-1000 może być zrealizowane w czasie 3 minut. Do stacji Sun można dołączyć 8 bloków NT-1000. Stację wyposażono w pełne oprogramowanie systemów transputerowych (TDS) oraz kompilatory języków FORTRAN, PASCAL, C, PROLOG oraz system operacyjny Sun-UNIX.

Stosowanie transputerów lub transputerowych kart rozszerzenia w konwencjonalnych mikrokomputerach klasy IBM PC XT/AT. DEC MicroVax czy komputerowych stacjach roboczych SUN może mieć tylko uzasadnienie w zwiększaniu możliwości użytkowych posiadanego sprzętu. Nie jest to jednak właściwe i pełne wykorzystanie potencjału i własności tych procesorów. Jest wiele przykładów, gdzie moc obliczeniowa decyduje o jakościowych walorach i zdolnościach techniczno-użytkowych urządzeń — na przykład skomplikowane obliczenia dla precyzyjnego kreślenia rysunków, krojów pisma, i innych znaków graficznych przez graficzne drukarki laserowe i plotery, grafika i animacja o bardzo dużej rozdzielczości, a także problemy trójwymiarowości obrazów.

Stosunek środowisk naukowych i inżynierskich do transputerów, ich znaczenia w rozwoju technik obliczeniowych i technologii układów mikro wyraża się w powołaniu przy SERC/DTI [2]/ towarzystwa fundacyjnego na rzecz popierania technik transputerowych oraz licznych regionalnych grup inicjatywnych. SERC/DTI Transputer Inititive skupia w swoich szeregach członkowskich ponad 1000 osób, przede wszystkim ze środowisk naukowych (z 40 placówek naukowo-uniwersyteckich i politechnicznych), działających w ramach 10 sekcji. Ich wkład w rozwój zagadnień technik komputerowych, to organizacja sesji i konferencji naukowych, popularyzacja wiedzy poprzez wydawnictwo materiałów, publikacji oraz podręczników programowania, organizacja kursów i, szkoleń a także prowadzenie prac naukowo-badawczych i wdrożeniowych. Wydaje się, że stosunek środowisk naukowych najlepiej wyraża wartość produktów firmy INMOS —

Tabela 1. Transputerowe produkty firmy INMOS

	Ozn. kat.	Opis
	1	2
Podstawowe układy rodziny INMOS	IMS T800	32-bit. transputer, 64-bit. FPU, 4 KB on-RAM[a]
	IMS T414	32-bit. transputer, 2 KB on-RAM[b]
	IMS T212	16-bit. transputer, 2 KB on-RAM[c]
	IMS M212	16-bit. disc procesor, 2 KB on-RAM, 2 KB on-ROM [b]
	IMS C004	zespół programowanej łącznicy elektronicznej[d]
	IMS C011	adapter linii z dwoma modami pracy[d]
	IMS C012	adapter linii[d]
Karty	IMS B001	T414 (T800 + 64 KB SRAM + 128 KB ROM, 2 porty RS232
	IMS B002	T414 + 2 MB DRAM + 128 KB ROM, 2 porty RS232
	IMS B003	cztery T414 (T800 + 1 MB DRAM, konfiguracja kwadratu
	IMS B004	karta do IBM PC XT (AT z T414 + 2 MB DRAM, IBM PC interfejs I/O
	IMS B005	M212 + 20 MB 3,5' Winchester + 1 MB 3,5' floppy disc
	IMS B006	M212 + 56 KB SRAM + 2 porty RS232 z możliwością zainstalowania do 9 T212 i 128 KB SRAM łącznie sterowaniem UART I EPROM
Pakiety	IMS B007	karta grafiki: T414 + G170 + 512 KB DRAM + KB VRAM
	IMS B009	cztery A100 + T212 + moduł dekodera adresu i generatora, interfejs IBM PC — 1280 MOPS[e] — opcjonalnie T414/T800, 1MB RAM
	IMS B010	T414 + 1 MB DRAM + C012 — karta dla komputera NEC PC-9801
	IMS B011	T414/T800 + 2 MB DRAM + 2 porty RS232, 2 miejsca dla modułów rozszerzenia INMOS, system magistrali VME
Moduły	IMS B203	10 B003 + 10 MB DRAM + D600 lub D701, możliwość rekonfiguracji poprzez wymianę karty na inną typu B0xx
	IMS B401	T414/T800 + 32 KB SRAM
	IMS B402	T212 + 8 KB SRAM
	IMS B403	T414/T800 + 1 MB DRAM
	IMS B404	T800 + 2 MB DRAM
	IMS B405	T800 + 8 MB DRAM
Transputerowe systemy/oprogramowanie	IMS D600	OCCAM 2 z pełnym edytorem, kompilatorem i konfiguratorem systemu dla komputera VAX/VAM 11/7xx firmy DEC mikroVAX
	IMS D700	OCCAM 2 z pełnym edytorem, kompilatorem i konfiguratorem
	IMS D 701	systemu dla mikrokomputera IBM PC XT AT
		D700 + B004
	IMS D703A	symulator A100 w OCCAM
	IMS D704A	D700 + D703A + B009
	IMS D711	D700 + kompilator C
	IMS D712	D700 + kompilator PASCAL
	IMS D713	D700 + kompilator FORTRAN-u

[a] — Zegar procesora: 17, 20, 30 MHz
[b] — Zegar procesora: 15, 20 MHz
[c] — Zegar procesora: 17, 20 MHz
[d] — Szybkość transmisji: 10/20 Mbitów/s.
[e] — operacje na sekundę

1/ MIPS — jednostka mocy obliczeniowej wyrażona w milionach instrukcji wykonywanych w ciągu sekundy.

2/ SERC/DTI — Science and Engineering Research Council (Departament of Trade and Industry — Rada Badań Naukowo-Technicznych) Dep. Handlu i Przemysłu. Fundacja na rzecz popierania technik transputerowych powstała przy SERC/DTI w lutym 1987 r.

W. M. S.

*****

Nuty z komputera — cz. 3

W trzeciej części artykułu zajmiemy się współpracą z całym zbiorem taktów. Jest to zagadnienie niezwykle istotne i (niestety) dosyć skomplikowane, zarówno od strony programowej, jak i użytkowej.

Na wstępie chcemy przekazać Czytelnikom kilka ważnych uwag:

UWAGA 1: **brzmienie, klawiatura dynamiczna** — oprócz wysokości dźwięku istnieje olbrzymia grupa parametrów opisująca przebieg dźwięku w czasie. Niestety, ich ilość i format zależą od typu syntezatora. Nowoczesne „klawisze" mają klawiaturę dynamiczną, stąd przesyłanie tylko wysokości dźwięku jest niewystarczające. Wynika z tego istotny wniosek: najlepszym wyjściem jest możliwość przesyłania w każdym kroku nie tylko wysokości dźwięku, ale i dowolnej ilości parametrów brzmieniowych. W przedstawianej wersji sekwencera nie będziemy tych parametrów programować, ale możliwość ich zapisu przewidzieliśmy (szczegóły podczas opisu „kompresji").

UWAGA 2: **listing** — łączymy z pierwszą częścią programu. Jeżeli linia o danym numerze już występuje, to należy ją wymienić na aktualnie zamieszczoną.

UWAGA 3: **zmiana skali dźwięków** — ze względu na przeprowadzaną kompresję zapisu taktu dźwięk **g7** (g siedmiokreślne) został przyporządkowany **pauzie**. Aktualna skala rozciąga się od **C2** do **fis7**. Konsekwencją tej zmiany jest przyporządkowanie w zapisie cyfrowym pauzy wartości 127.

UWAGA 4: **wprowadzanie nazw taktów** — procedura umożliwia wprowadzenie słowa dziesięcioznakowego:

1) możemy wprowadzić małe litery, cyfry, spację oraz znaki ASCII o kodach od 33 do 47
2) pierwszym znakiem musi być litera
3) funkcje dużych liter:

<**W**> — wprowadza nazwę: ukazuje się tekst „<t/n>" — wciśnięcie <t> wprowadza nazwę
<**D**> — kasuje ostatnio wprowadzony znak
<**P**> — przerwanie procedury

EDYTOR TAKTU (nowe funkcje)

klawisz <**m**> — umożliwia wprowadzenie podzielnika tempa (tzn. ile razy wolniej dany takt ma być wykonywany w stosunku do tempa podstawowego) parametry: 1) podzielnik

uwaga: opcjonalnie podzielnik jest ustawiany na wartość maksymalną: 255

klawisz <**u**> — pozwala na zapis taktu do pamięci (zmiennej tekstowej m$) i powrót do edytora głównego

klawisz <**U**> — powrót do edytora głównego bez zapisu taktu

klawisz <**n**> — pozwala podzielić edytowany takt na dwa niezależne takty
parametry: 1) nr kroku, od którego rozpoczynać się będzie drugi takt
2) nazwa tworzonego taktu

uwaga: jeżeli drugi takt będzie rozpoczynał się od kroku nr 1, to procedura podziału jedynie wprowadzi podaną nazwę taktu na miejsce starej

uwaga: po wykonaniu podziału następuje powrót do edytora głównego

EDYTOR GŁÓWNY

klawisz <**w**> — wywołanie taktu z pamięci (tzn. odczyt ze zmiennej tekstowej m$), wejście do edytora taktu
parametry: 1) nr taktu

klawisz <**u**> — utworzenie taktu
parametry: 1) nazwa taktu
2) ilość kroków
warunki: a) jest to pierwszy wprowadzany takt
3) następuje zgłoszenie funkcji <d> edytora taktu: należy podać parametry pierwszej ścieżki — wtedy nastąpi wejście do edytora taktu z pierwszą zadeklarowaną ścieżką
b) istnieje co najmniej jeden takt w pamięci
3) należy wybrać, czy takt jest tworzony od początku, czy wg ustawienia ście-

żek określonego taktu
wciśnięcie <n> — wykonanie procedury z punktu a) wciśnięcie <t> — spowoduje skorzystanie z wzorcowego taktu:
4) numer wzorcowego taktu

klawisz **<t>** — grupowa transpozycja w górę (przeprowadzana wg zawartości pola aktywności)
parametry: ilość półtonów transpozycji
klawisz **<T>** — jak wyżej — transpozycja w dół
klawisz **<k>** — kasowanie grupowe przeprowadzane na podstawie zawartości pola aktywności
uwaga: jeżeli nastąpi skasowanie wszystkich ścieżek, wtedy takt jest usuwany z pamięci
klawisz **<d>** — dane taktu — funkcja wypisuje podstawowe informacje o takcie i o ścieżkach (przynależność do instrumentu, źródła i typ zapisu)
parametry: 1) numer taktu
klawisz <I> — lista taktów — drukuje na ekranie ponumerowane takty, ich nazwy, ilość ścieżek, kroków i podzielnik tempa
uwaga: kolejne numery wypisywane przy taktach są obowiązujące podczas żądania: *podaj numer taktu*

```
   1 REM
**********************************

    EDYTOR UTWORU I TAKTU

**********************************

   7 DEF FN z$(y)=CHR$ (y-256*IN
T (y/256))+CHR$ INT (y/256)
   8 DEF FN z(y)=256*CODE m$(y+1
)+CODE m$(y)
  30 LET w$=CHR$ 0+CHR$ 0+CHR$ 0
+CHR$ 0: LET w$=w$+w$+w$+w$+CHR$
 0
  31 LET cike=10: LET cise=8
  40 LET t$="c0c1d0d1e0f0f1g0g1a
0a1h021001234567"
  50 LET m$=CHR$ 0
 103 LET pnk=(str-1)*mik+1: PRIN
T "  ";: LET knk=pnk+eik-1: FOR
l=pnk TO knk: PRINT " " AND l<10
0; INVERSE 1;"0" AND (l<10);l; I
NVERSE 0;: NEXT l
 172 LET str=FN m(str,INT ((ik-1
)/mik)+1): LET pnk=mik*(str-1)+1
1000 REM EDYTOR GLOWNY
1001 LET it=CODE m$(1)
1005 GO SUB 1910
1010 GO SUB 9198
1011 LET x$=INKEY$: IF x$="" THE
N GO TO 1011
1012 IF x$="w" AND it>0 THEN GO
SUB 1115: GO TO 1005
1014 IF x$="u" AND it<16 THEN GO
 SUB 1100: GO TO 1005
1020 IF x$="a" AND it>0 THEN GO
SUB 8900: GO TO 1010
1024 IF x$="k" AND it>0 THEN GO
SUB 1130: GO TO 1010
1025 IF x$="t" AND it>0 THEN GO
SUB 1140: GO TO 1010
1027 IF x$="T" AND it>0 THEN GO
SUB 1160: GO TO 1010
1029 IF x$="d" AND it>0 THEN GO
SUB 1170: GO TO 1010
1031 IF x$="i" AND it>0 THEN GO
SUB 1180: GO TO 1010
1090 GO TO 1005
1100 GO SUB 8975: IF q$="" THEN
RETURN
1101 IF k<=it THEN RETURN
1104 LET q=0: LET n$=q$: LET md=
256: LET z$="ilosc krokow:": GO
SUB 9190: IF q=0 THEN RETURN
1105 LET pt=255: LET e$="": LET
ik=q: IF it=0 THEN GO TO 1108
1106 LET z$="instr. w/q wzorca 1
-tak": LET q=1: GO SUB 9190
1107 IF q=1 THEN GO TO 1110
1108 LET is=0: LET q=1: LET eis=
1: GO SUB 9151: IF is=1 THEN LET
 nt=it+1: GO SUB 9000
1109 RETURN
1110 LET nt=q: GO SUB 1120: IF q
>0 THEN LET nt=it+1: FOR k=1 TO
ik: LET e$=e$+CHR$ 128: NEXT k:
LET n=FN z(2*q)+2: LET is=CODE m
$(n): LET n$=q$+m$(n+11+2*is TO
n+10+5*is): FOR k=1 TO is-1: LET
 e$=e$+e$: NEXT k: GO SUB 9000
1112 RETURN
1115 GO SUB 1120: IF q>0 THEN LE
T nt=q: GO SUB 9255: GO SUB 9000
1117 RETURN
1120 LET z$="podaj nr taktu:": L
ET md=it: LET q=0: GO TO 9190
1121 LET q=0: LET md=2: LET z$="
transp. 1-sc. 2-takt.": GO SUB 9
190: RETURN
1130 LET q=0: LET md=2: LET z$="
kasow. 1-sc. 2-takt.": GO SUB 91
90: IF q=0 THEN RETURN
1131 IF q=1 THEN GO SUB 8840: RE
TURN
1132 LET t=1
1133 FOR t=t TO it: IF w$(t)=CHR
$ 1 THEN GO SUB 8825: GO TO 1133
1134 NEXT t: RETURN
1140 LET q=0: LET md=126: LET z$
="transp w gore o:": GO SUB 9190
: IF q=0 THEN RETURN
1141 LET r=q: GO SUB 1121: IF q=
0 THEN RETURN
1142 IF q=2 THEN GO TO 1147
1143 LET t=1: GO SUB 8800
1144 IF t=0 THEN RETURN
1145 IF CODE m$(j+2)<>1 THEN GO
SUB 1150
1146 GO SUB 8810: GO TO 1144
1147 FOR t=1 TO it: IF w$(t)=CHR
$ 0 THEN GO TO 1149
1148 LET n=FN z(2*t): LET k=CODE
 m$(n+2): LET j=n+13+2*k: FOR g=
1 TO k: GO SUB 1150: LET j=j+3:
NEXT g
1149 NEXT t: RETURN
1150 LET q=n+FN z(n+11+2*g): GO
SUB 8830: LET b=w+q-1: IF m$(j+2
)=CHR$ 1 THEN RETURN
1151 FOR q=q TO b
1152 LET f=CODE m$(q): IF f>127
THEN LET f=f-128
1153 IF f=127 THEN GO TO 1156
1154 LET f=f+r: IF f<1 OR f>126
THEN LET f=127
1156 IF CODE m$(q)<128 THEN LET
m$(q)=CHR$ f: GO TO 1159
1157 LET m$(q)=CHR$ (f+128): LET
 q=q+1: IF CODE m$(q)<128 THEN G
O TO 1159
1158 LET q=q+1
1159 NEXT q: RETURN
1160 LET q=0: LET md=126: LET z$
="transp w dol o:": GO SUB 9190:
 IF q=0 THEN RETURN
1161 LET r=-q: GO SUB 1121: IF q
=0 THEN RETURN
1162 IF q=2 THEN GO TO 1147
1163 GO TO 1143
1170 GO SUB 1120: IF q=0 THEN RE
TURN
1171 CLS : LET n=FN z(2*q): PRIN
T "nazwa taktu: "; INVERSE 1;m$(
n+3 TO n+12)

1172 LET s=CODE m$(n+2): PRINT "
ilosc sciezek: ";TAB 19;s: PRINT
"ilosc krokow: ";TAB 19;CODE m$(
n+1)+1: PRINT "podzielnik metrum
```

```
: ";CODE m$(n)
1170 LET n=n+13+2*s: PRINT : PRI
NT "  instrument    zrodlo     ty
p": PRINT : FOR q=1 TO s: LET i=
12*CODE m$(n)-9: LET z=6*CODE m$
(n+1)-5: LET t=CODE m$(n+2): PRI
NT q;TAB 4;i$(i TO i+9);TAB 16;s
$(z TO z+5);TAB 23;("literowy" A
ND (t=0));("symbol." AND (t=1));
("cyfrowy" AND (t=2)): LET n=n+3
: NEXT q
1174 RETURN
1180 CLS : PRINT "nr   nazwa
 is    ik    pt": PRINT
1181 FOR t=1 TO it: LET n=FN z(2
*t): PRINT t;TAB 3; INVERSE 1;m$
(n+3 TO n+12);
1182 LET s=CODE m$(n+2): PRINT T
AB 15;s;TAB 20;CODE m$(n+1)+1;TA
B 25;CODE m$(n)
1183 NEXT t: RETURN
1185 CLS : LET n=FN z(2*q): PRIN
T "nazwa taktu: "; INVERSE 1;m$(
n+3 TO n+12)
1186 LET s=CODE m$(n+2): PRINT "
ilosc sciezek: ";TAB 23;s: PRINT
"ilosc krokow: ";TAB 23;CODE m$(
n+1)+1: PRINT "podzielnik taktow
ania: ";CODE m$(n)
1200 LET q=0: LET it=CODE m$(1):
 LET md=it: LET z$="wybierz takt
": GO SUB 9190: IF q=0 THEN GO
TO 9198
1910 CLS : PRINT TAB 7; INVERSE
1;"SEKWENCER v.1.0": PRINT : PRI
NT TAB 8; INVERSE 1;"EDYTOR GLOW
NY"
1912 PRINT TAB 5;"w - wywolanie
taktu"
1914 PRINT TAB 5;"u - utworzenie
 taktu"
1916 PRINT TAB 5;"t - transpozyc
ja w gore"
1917 PRINT TAB 5;"T - transpozyc
ja w dol"
1918 PRINT TAB 5;"k - kasowanie"
1930 PRINT TAB 5;"d - dane taktu
"
1932 PRINT TAB 5;"l - lista takt
ow"
1934 PRINT TAB 5;"a - aktywnosc"
1950 RETURN
8800 REM Wyszukiwanie sciezki
8801 FOR t=t TO it
8802 IF w$(t)=CHR$ 0 THEN GO TO
8813
8804 FOR i=1 TO ii
8805 IF w$(it+i)=CHR$ 0 THEN GO
TO 8812
8806 LET s=CODE i$((i-1)*12+1):
LET d=CODE i$((i-1)*12+2): FOR z
=1 TO s

8807 IF w$(it+ii+d)=CHR$ 0 THEN
GO TO 8811
8808 LET n=FN z(2*t): LET k=CODE
 m$(n+2): LET g=1
8809 FOR g=g TO k: LET j=n+10+2*
k+3*g: IF m$(j)=CHR$ i AND m$(j+
1)=CHR$ d THEN RETURN
8810 NEXT g
8811 LET d=d+1: NEXT z
8812 NEXT i
8813 NEXT t: LET t=0: RETURN
8820 REM skasow. sciezki w pam.
8822 GO SUB 8830: LET b=w+5: LET
 q=2*t+2: FOR l=t+1 TO it: LET m
$(q TO q+1)=FN z$(FN z(q)-b): LE
T q=q+2: NEXT l
8823 LET q=n+13: FOR l=1 TO k: L
ET m$(q TO q+1)=FN z$(FN z(q)-5-
w*(l>g)): LET q=q+2: NEXT l: LET
 m$(n+2)=CHR$ (k-1): LET q=n+11:
 LET e=q+2*k+3*g-2: LET b=n+FN z
(q+2*g)+5: LET k=k-1: LET m$=m$(
 TO q+2*g-1)+m$(q+2*g+2 TO e)+m$
(e+4 TO b-1)+m$(b+w TO ): RETURN

8825 REM skasowanie taktu w pam.
8826 LET u=FN z(2*t): LET r=FN z
(2+2*t)-FN z(2*t): FOR l=2 TO 2*
it STEP 2: LET m$(l TO l+1)=FN z
$(FN z(l)-r*(l>=(2+2*t))-2): NEX
T l
8827 LET m$=m$( TO 2*t-1)+m$(2+2
*t TO u-1)+m$(u+r TO )
8828 LET it=it-1: LET m$(1)=CHR$
 it: LET w$=w$( TO t-1)+w$(t+1 T
O ): RETURN
8830 REM oblicz. dlug. sciezki
8831 IF g<k THEN LET w=FN z(n+10
+2*g)-FN z(n+11+2*g): RETURN
8832 IF t<it THEN LET w=FN z(2*t
+2)-FN z(n+11+2*g)-n: RETURN
8833 LET w=LEN m$-FN z(n+11+2*g)
-n+1: RETURN
8840 REM proc. kasowania sciezek
8842 LET t=1: GO SUB 8800
8844 IF t=0 THEN RETURN
8846 IF k=1 THEN GO SUB 8825: GO
 SUB 8800: GO TO 8844
8848 GO SUB 8820: GO SUB 8809: G
O TO 8844
8849 RETURN
8900 REM Procedura Wyboru
8901 CLS : LET zu=1: LET pt=1: L
ET iwt=FN m(16,it): LET pi=1: LE
T iwi=FN m(16,ii): LET n=1: LET
iwz=FN m(16,CODE i$(1)): LET at=
0: LET ai=it: LET az=it+ii: LET
x=1: LET y=1: LET knc=iwt
8903 GO SUB 8940: GO SUB 8945: G
O SUB 8950: GO SUB 8960
8905 GO SUB 9198
8906 LET q$=INKEY$: IF q$="" THE
N GO TO 8906
8907 BEEP 0.01,20
8910 IF q$="6" AND y<knc THEN GO
 SUB 8965: LET y=y+1: GO SUB 896
0: GO TO 8906
8911 IF q$="7" AND y>1 THEN GO S
UB 8965: LET y=y-1: GO SUB 8960:
 GO TO 8906
8912 IF q$="8" AND x<3 THEN LET
knc=iwi*(x=1)+iwz*(x=2): GO SUB
8965: LET x=x+1: LET y=FN m(knc,
y): GO SUB 8960: GO TO 8906
8913 IF q$="5" AND x>1 THEN LET
knc=iwt*(x=2)+iwi*(x=3): GO SUB
8965: LET x=x-1: LET y=FN m(knc,
y): GO SUB 8960: GO TO 8906
8914 IF q$=" " AND x=1 THEN LET
w$(y+pt-1)=CHR$ NOT CODE w$(y+pt
-1): GO SUB 8940: GO TO 8906
8915 IF q$=" " AND x=2 THEN LET
w$(y+pt+it-1)=CHR$ NOT CODE w$(y
+pt+it-1): GO SUB 8945: GO TO 89
06
8916 IF q$=" " AND x=3 THEN LET
w$(y+az+pz-1)=CHR$ NOT CODE w$(y
+pz+az-1): GO SUB 8950: GO TO 89
06
8917 IF q$<>"k" AND q$<>"K" AND
q$<>"U" AND q$<>"U" THEN GO TO 8
921
8918 LET m=(q$>"a"): IF x=1 THEN
 GO SUB 8955: IF q=1 THEN FOR k=
1 TO iwt*m+it*NOT m: LET w$(k+(p
t-1)*m)=CHR$ (q$="U" OR q$="U"):
 NEXT k: GO SUB 8940: GO TO 8905
8919 IF x=2 THEN GO SUB 8955: IF
 q=1 THEN FOR k=1 TO iwi*m+ii*NO
T m: LET w$(k+it+(pi-1)*m)=CHR$
(q$="U" OR q$="U"): NEXT k: GO S
UB 8945: GO TO 8905
```

```
8920 IF x=3 THEN GO SUB 8955: IF
 q=1 THEN FOR k=1 TO iwz*m+(LEN
w$-it-ii)*NOT m: LET w$(k+(az+pz
-1)*m+((it+ii)*NOT m))=CHR$ (q$=
"U" OR q$="U"): NEXT k: GO SUB 8
950: GO TO 8905
8921 IF q$="c" THEN LET q=0: LET
 z$="1-kas. 2-ust. calosci": LET
 md=2: GO SUB 9190: IF q>0 THEN
FOR k=1 TO LEN w$: LET w$(k)=CHR
$ (q=2): NEXT k: GO TO 8903
8929 IF q$="p" THEN RETURN
8930 GO TO 8905
8940 PRINT AT 0,3;"TAKTY": LET a
x=1: FOR k=1 TO iwt: LET l=FN z(
(k+pt-1)*2)+3: PRINT AT k+1,1; I
NVERSE CODE w$(k+at);m$(l TO l+9
): NEXT k: GO TO 8970
8945 PRINT AT 0,13;"INSTRUMENTY"
: LET ax=13: FOR k=1 TO iwi: LET
 l=3+(k+pi-2)*12: PRINT AT k+1,1
6; INVERSE CODE w$(ai+k+pt-1);i$
(l TO l+9); INVERSE 0;(" *" AND z
u=k);(" " AND zu<>k): NEXT k: GO
 TO 8970
8950 PRINT AT 0,26;"ZRODLA": LET
 ax=26: FOR k=1 TO iwz: LET l=n+
(k-1)*6: PRINT AT k+1,26; INVERS
E CODE w$(az+k);s$(l TO l+5): NE
XT k: GO TO 8970

8955 LET q=1: LET z$=("kasowanie
" AND (q$="K" OR q$="K"))+("usta
wianie" AND (q$="U" OR q$="U"))+
(" kolumny" AND NOT m)+" 1-tak":
 GO TO 9190
8960 PRINT AT y+1,(x=2)*12+(x=3)
*25; FLASH 1;">": IF x=2 THEN LE
T zu=y: GO SUB 8945: LET uz=y: L
ET nri=y-1+pi: LET aii=1+(nri-1)
*12: LET n=6*(CODE i$(aii+1)-1)+
1: LET az=it+ii+CODE i$(aii+1)-1
: LET pz=1: LET iwz=FN m(16,CODE
 i$(aii)): GO TO 8950
8962 RETURN
8965 PRINT AT y+1,(x=2)*12+(x=3)
*25;" "
8966 RETURN
8970 FOR k=k TO 16: PRINT AT k+1
,ax;"          "+("      " AND ax<>26)
: NEXT k: RETURN
8975 REM wprow. nazwy taktu
8976 LET md=10: LET z$="nazwa ta
ktu: ": LET q$="": GO SUB 8980:
IF q$="" THEN RETURN
8977 FOR k=1 TO it: LET n=FN z(2
*k)+3: IF m$(n TO n+9)=q$ THEN G
O SUB 8979: RETURN
8978 NEXT k: RETURN
8979 PRINT AT 19,0; FLASH 1;" N
AZWA JUZ ISTNIEJE - Enter ":
PAUSE 1: PAUSE 0: RETURN
8980 GO SUB 9198: PRINT AT 19,0;
 INVERSE 1;z$;: PRINT ;" ";; IF
q$<>"" THEN PRINT INVERSE 1;q$;
8982 LET x$=INKEY$: IF x$="" THE
N GO TO 8982
8983 IF x$="P" THEN LET q$="": R
ETURN
8984 IF NOT (x$>="a" AND x$<="z"
 OR (x$="W" OR x$="D" OR x$>="
 AND x$<="9") AND q$<>"") THEN G
O SUB 9326: GO TO 8982
8985 IF x$="W" THEN PRINT ;" ";
FLASH 1;"(t/n)": PAUSE 1: PAUSE
0:: LET x$=INKEY$: LET q$=q$ AND
 (x$="t"): FOR k=LEN q$+1 TO 10*
(q$<>""): LET q$=q$+" ": NEXT k:
 BEEP .1,30: RETURN
8986 IF x$="D" THEN LET q$=q$( T
O LEN q$-1): GO TO 8980
8987 LET q$=q$+x$: IF LEN q$>md
THEN LET q$="": GO SUB 9326: GO
TO 8980
8988 PRINT INVERSE 1;x$;: GO TO
8982
9045 IF x$="d" THEN LET q=0: GO
SUB 9150: GO SUB 100: GO TO 9025
9046 IF x$="D" THEN LET q=by: GO
 SUB 9150: GO SUB 100: GO TO 902
5
9059 IF x$="u" THEN GO TO 9225
9060 IF x$="U" THEN GO TO 9228
9061 IF x$="m" THEN LET q=0: GO
SUB 9260: GO TO 9025
9062 IF x$="n" THEN LET q=0: GO
TO 9265

9090 GO TO 9025
9150 LET md=is+1: LET z$="dolacz
. sciez. nr:": GO SUB 9190: IF q
=0 THEN RETURN
9151 LET ans=q: GO SUB 9180: LET
 md=ii: LET q=0: LET z$="podaj n
r instumentu:": GO SUB 9190: IF
q=0 THEN RETURN
9152 LET ani=q: PRINT AT 3+q,2;
INVERSE 1;q;"-";i$(12*(q-1)+3 TO
 12*q): LET aai=1+12*(q-1): LET
ais=CODE i$(aai TO aai): LET asp
=CODE i$(aai+1 TO aai+1): GO SUB
 9185: LET md=ais: LET q=0: LET
z$="podaj nr zrodla:": GO SUB 91
90: IF q=0 THEN RETURN
9153 PRINT AT 3+q,20; INVERSE 1;
;q;"-";s$(6*(asp+q-2)+1 TO 6*(as
p+q-1)): LET ansi=q: LET md=3: L
ET q=0: LET z$="1-liter 2-graf 3
-cyfr": GO SUB 9190: IF q=0 THEN
 RETURN
9154 LET k=FN o(ans): LET z$="":
 FOR n=1 TO ik: LET z$=z$+CHR$ 1
28: NEXT n: LET e$=e$( TO k-1)+z
$+e$(k TO ): LET k=FN t(ans): LE
T n$=n$( TO k-1)+CHR$ ani+CHR$ (
asp+ansi-1)+CHR$ (q-1)+n$(k TO )
: LET eis=eis+(is<cise): LET is=
is+1: RETURN
9202 LET k=q: LET md=FN m(ik-q+1
,ik-1): LET q=0: LET z$="ile kro
kow kasowac:": GO SUB 9190: IF q
=0 THEN GO SUB 9198: RETURN
9210 LET md=is: LET z$="nr sciez
ki transpon.:": GO SUB 9190: IF
q=0 OR CODE n$(10+3*q)=1 THEN GO
 TO 9198
9211 LET k=q: LET q=0: LET md=12
6: LET z$="transpozycja w gore o
:": GO SUB 9190: IF q=0 THEN GO
TO 9198
9213 LET n=n+q: IF n>=127 OR n<0
 THEN LET n=128
9215 LET md=is: LET z$="nr sciez
ki transpon.:": GO SUB 9190: IF
q=0 OR CODE n$(10+3*q)=1 THEN GO
 TO 9198
9216 LET k=q: LET q=0: LET md=12
6: LET z$="transpozycja w dol o:
": GO SUB 9190: IF q=0 THEN GO T
O 9198
9225 LET q=1: LET z$="powrot z z
apisem? t=1:": GO SUB 9190: IF q
=0 THEN GO SUB 9198: GO TO 9025
9226 LET q=nt: IF q<=it THEN LET
 t=q: GO SUB 8825
9227 GO TO 9230
9228 LET q=1: LET z$="powrot bez
 zapisu? t=1:": GO SUB 9190: IF
q=0 THEN GO SUB 9198: GO TO 9025
9229 RETURN : REM
     zapis i kompres. bufora
9230 LET n=1: LET l=1: FOR m=1 T
O is: LET n$=n$( TO 8+2*m)+FN z$
(n+5*is+12)+n$(9+2*m TO ): LET k
```

```
=0
9231 LET h=CODE e$(l): LET h=h+(
h<=127)+127*(h>127): LET e$(n)=C
HR$ h: LET l=l+1: FOR i=2 TO ik:
 LET o=CODE e$(l): IF o>126 AND
o<>129 THEN LET o=127
9232 IF h=127 AND o=127 OR o=129
 THEN LET k=k+1: GO TO 9240
9233 IF k=0 THEN LET n=n+1: LET
e$(n)=CHR$ o: LET h=o: GO TO 924
0
9234 LET e$(n)=CHR$ (128+h): LET
 h=o: LET n=n+2: IF k<128 THEN L
ET e$(n-1)=CHR$ k: GO TO 9239
9235 LET e$(n-1)=CHR$ l: LET e$(
n)=CHR$ 128: LET n=n+1
9239 LET e$(n)=CHR$ o: LET k=0
9240 LET l=l+1: NEXT i: IF k>0 T
HEN LET e$(n)=CHR$ (128+h): LET
n=n+1: LET e$(n)=CHR$ k: IF k>=1
28 THEN LET n=n+1: LET e$(n)=CHR
$ 128
9241 LET n=n+1: NEXT m: LET e$=e
$(1 TO n-1): LET n$=CHR$ pt+CHR$
 (ik-1)+CHR$ is+n$
9242 LET it=CODE m$(1)
9243 IF q>it THEN FOR l=2 TO 2*i
t STEP 2: LET m$(l TO l+1)=FN z$
(FN z(l)+2): NEXT l: LET m$=m$(
TO 2*it+1)+FN z$(LEN m$+3)+m$(2*
it+2 TO )+n$+e$: GO TO 9246
9244 LET n=FN z(2*q): LET k=5*is
+13+LEN e$: FOR l=2 TO 2*it STEP
 2: LET m$(l TO l+1)=FN z$(FN z(
l)+k*(l>=2*q)+2): NEXT l
9245 LET m$=m$( TO 2*q-1)+FN z$(
n+2)+m$(2*q TO n-1)+n$+e$+m$(n T
O )
9246 LET it=it+1: LET m$(1)=CHR$
 it: LET w$=w$( TO q-1)+CHR$ 0+w
$(q TO ): RETURN
9255 REM [illegible]
9256 LET k=FN z(2*q): LET pt=COD
E m$(k): LET ik=CODE m$(k+1)+1:
LET is=CODE m$(k+2): LET n$=m$(k
+3 TO k+12)+m$(13+2*is+k TO 12+5
*is+k): LET n=k+13: LET n=k+FN z
(n): LET e$="": FOR l=1 TO is*ik
: LET i=CODE m$(n): LET e$=e$+CH
R$ (i-128*(i>127))+(i=127 OR i=25
5): IF i<128 THEN : GO TO 9259
9257 LET n=n+1: LET m=CODE m$(n)
: LET l=l+m: FOR j=1 TO m: LET e
$=e$+(CHR$ 129 AND i<255)+(CHR$
128 AND i=255): NEXT j: IF m>127
 THEN LET n=n+1
9259 LET n=n+1: NEXT l: RETURN
9260 LET md=255: LET z$="podziel
. metrum d.="+STR$ pt+":": GO SU
B 9190: IF q=0 THEN RETURN
9261 LET pt=q: RETURN
9265 LET z$="nr pierwsz. kroku":
 LET md=ik: GO SUB 9190: IF q=0
THEN GO TO 9025
9266 GO SUB 9198: GO SUB 8975: I
F q$="" THEN GO TO 9025
9267 IF k<=it THEN GO TO 9266
9268 IF n$(1 TO 10)=q$ THEN GO S
UB 8979: GO TO 9266
9269 IF q=1 THEN LET n$(1 TO 9)=
q$: GO TO 9226
9270 LET k=q: LET q=ik-k+1: LET
z$="": LET l=k-1: LET n=(ik-q)*i
s: FOR m=l TO n STEP ik-q: LET z
$=z$+e$(m+1 TO q+m): LET e$=e$(
TO m)+e$(q+m+1 TO ): NEXT m: LET
 ik=ik-q: LET zik=q
9271 LET x$=n$(11 TO ): GO SUB 9
226: LET e$=z$: LET n$=q$+x$: LE
T ik=zik: LET nt=it+1: GO TO 922
6
9337 LET m=m+cis+12*((2-VAL x$)*
n+(3+VAL x$)*NOT n): IF m>=127 T
HEN GO SUB 9326: LET m=128
9342 LET m=127*(x$="0")+1: RETUR
N
```

klawisz <**a**> — wywołanie pola aktywności
pole składa się z trzech kolumn: taktów, instrumentów i źródła
b) uaktywniony element jest wypisywany na ekranie w trybie inwersyjnym
c) uaktywniać możemy niezależnie takty, instrumenty i ich źródła
d) po polu poruszamy się kursorem
funkcje pola aktywności: klawisz <**6**> — kursor w dół
klawisz <**7**> — kursor w górę
klawisz <**5**> — kursor w lewo
klawisz <**8**> — kursor w prawo
<**SPACE**> — włączanie/wyłączanie elementu, przy którym znajduje się kursor
klawisz <**k**> — kasowanie aktywności uwidocznionego na ekranie fragmentu kolumny, przy której znajduje się kursor
klawisz <**u**> — jak <**k**>, ale ustawienie aktywności
klawisz <**K**> — kasowanie aktywności całej kolumny, przy której ustawiony jest kursor
klawisz <**U**> — jak <**K**>, ale ustawienie aktywności
klawisz <**c**> — dotyczy całego zbioru:
parametr: 1 — kasowanie
2 — ustawianie

uwaga: kolumna „źródła" wyświetla źródła instrumentu, po którego nazwie jest wyświetlona gwiazdka

uwaga: funkcje <**w**>, <**t**>, <**T**>, <**k**>, <**d**>, <**l**>, <**a**> są wykonywane pod warunkiem istnienia przynajmniej jednego taktu

uwaga: funkcje <**t**>, <**T**>, <**k**> jako drugi parametr mają wybór trybu funkcji: 1 — funkcja wykonywana jest na poszczególnych ścieżkach pod warunkiem zgodności taktu, instrumentu i źródła
2 — funkcja dotyczy całego taktu bez względu na zadeklarowane instrumenty i ich ścieżki

INFORMACJE DLA CIEKAWSKICH

zapis taktu w pamięci, procedura kompresji:

Treść taktu aktualnie edytowanego (podczas pracy edytora taktu) znajduje się w zmiennej tekstowej **e$**. Na podstawie opisu z poprzedniej części wynika, że jest to zapis pochłaniający dużą ilość pamięci, chociaż wygodny od strony edycyjnej.

Dlatego w momencie zapisu taktu do pamięci (czyli wprowadzenia go do zmiennej tekstowej **m$**) ulega on kompresji.

Na takie postępowanie zdecydowaliśmy się, aby zaoszczędzić jak najwięcej pamięci i w ten sposób umożliwić wpisanie dużej ilości taktów. Duże marnotrawstwo powodują dźwięki przedłużone, ciągi pauz, czy całe puste ścieżki.

Mechanizm jest prosty: jeżeli dźwięk trwa dłużej niż jeden krok, to zostaje zapisana wysokość dźwięku i informacja, na ile kroków jest on przedłużony. Jako informację o przedłużeniu dźwięku wykorzystujemy 8 (najstarszy) bit słowa wysokości dźwięku (nie jest on używany przy zapisie wysokości dźwięku).

Format zapisu przedstawia rysunek 1:

Przykładowe zapisy przedstawiają rysunki 2 i 3:

uwaga: jak wspomnieliśmy we wstępie, w prezentowanej wersji nie przewidujemy stosowania parametrów brzmieniowych i dlatego ich ilość stale wynosi 0

pamięć taktów:

Zbiór taktów jest trzymany w zmiennej tekstowej **m$**.
it — ilość taktów
pt — podzielnik tempa
ik — 1 — zmniejszona o jeden ilość kroków
is — ilość ścieżek

Krzysztof PÓŹNIAK
Michał ROLLINGER
Jarosław ZIEMBICKI

INFORMATYKA W SZKOLE

Jak uczyć informatyki w szkole? Ten problem budzi gorące namiętności informatyków, nauczycieli i uczniów. I nie jest to spór teoretyczny, coraz częściej bowiem w szkolnych pracowniach pojawiają się komputery. Od dwóch lat mamy nawet program nauczania tego uzupełniającego przedmiotu, jakim jest informatyka. Ten program budzi jednak kontrowersje, na przykład, że jego nierozsądna realizacja doprowadzić może do uczynienia z informatyki „sztuki dla sztuki" pozbawiając w ten sposób szans na wpisanie tej dyscypliny w codzienność.

Jaki powinien być zatem program nauczania tego przedmiotu? A może nie należy tej dyscypliny traktować zbyt samodzielnie i stworzyć raczej program jej zastosowania w nauczaniu fizyki, matematyki, chemii...?

Poszukajmy odpowiedzi na te pytania, a od początku roku szkolnego towarzyszyć będziemy wszystkim uczniom w poznawaniu właśnie informatyki.

Redakcja

Rola komputera w edukacji

Uchwała Prezydium Rządu z 5 listopada 1984 r. zakłada wprowadzenie do wszystkich szkół średnich do roku 1990 mikrokomputerów wraz z oprzyrządowaniem. Za pomocą tych urządzeń możliwe jest kształcenie informatyczne naszej młodzieży oraz wspomaganie nauczania innych przedmiotów. Komputery zastosowane właściwie są w szkole cennymi środkami dydaktycznymi, rozwijającymi aktywność umysłową uczniów, wspomagają one i uatrakcyjniają proces dydaktyczny. By komputer mógł dobrze spełniać swoją funkcję w szkole, nauczyciel musi być zaopatrzony w dobre, sprawdzone programy edukacyjne. Obserwujemy obecnie olbrzymie, jak zawsze, zaangażowanie polskich nauczycieli w rozwój oświaty przez tworzenie świetnych programów edukacyjnych. Bardzo istotną sprawą jest ich upowszechnienie, by zainteresowani nauczyciele i uczniowie mogli wspólnie z nich korzystać. W tej dziedzinie przychodzi nam z dużą pomocą czasopismo wojskowe „Żołnierz Wolności", wydając specjalny dodatek komputerowy „IKS", który na swoich gościnnych łamach umieszcza również programy edukacyjne. Nie czynią tego niestety inne czasopisma komputerowe. Najliczniejszą grupę czytelników tych czasopism stanowi młodzież, a młodzież uczęszcza do szkół i poszukuje tych programów. Nieocenione usługi w upowszechnianiu programów edukacyjnych wyświadcza nam Polskie Radio, zwłaszcza audycje Rozgłośni Harcerskiej emitowane w 4 programie. Wielu uczniów ma swoje własne komputery i oni przynoszą do szkoły, za moją namową, programy edukacyjne zarejestrowane na kasetach magnetofonowych, które ukazały się w prasie (niestety, ciągle jeszcze za mało) lub nagrane bezpośrednio za pomocą magnetofonu z radia. Jeśli program mi się podoba, przegrywam go na swoją kasetę lub dyskietkę; w ten sposób je kolekcjonuję. Nauczyciele, wbrew mylnie panującym opiniom, nie mają dużo wolnego czasu, dlatego uważam, że w zdobywaniu tak bardzo potrzebnych programów edukacyjnych powinni nam pomagać nasi uczniowie, dla których będzie to połączenie przyjemnego z pożytecznym. Również wykorzystany jest do tego celu szkolny komputer. Jeśli wszyscy nauczyciele, zajmujący się wprowadzeniem komputerów do dydaktyki, będą zachęcać dostępnymi środkami swoich uczniów do kolekcjonowania tak potrzebnych programów, to żaden egzemplarz „IKS-a" nie będzie leżał w kiosku, w oczekiwaniu na swego czytelnika. Istnieją szkoły, które mają komputery, ale z nich nie korzystają z braku programów edukacyjnych.

Kształcenie informatyczne zależy od postaw i przygotowania nauczycieli. Dla wielu z nich, szczególnie starszych stażem, komputer na lekcji, to tajemniczy przyrząd, pochłaniający nauczycielowi wiele czasu. Każda nowość ma swoich zwolenników i przeciwników. Komputer wymaga od nauczyciela zapoznania się z nową dziedziną wiedzy, co pochłania wiele czasu. Sam również pokonywałem barierę psychologiczną i lęk przed nieznanym. Przecież 25 lat uczyłem bez komputera i wcale nie odczuwałem jego braku. Mamy w szkołach wielu zapaleńców, którzy wciągają do stosowania komputerów innych nauczycieli. Próby pisania własnych programów przez nauczycieli, to zjawisko naturalne. Z własnego doświadczenia wiem, i to musimy sobie uświadomić, że wiedzę informatyczną o wiele łatwiej jest opanować ludziom młodym. Dlatego nasi uczniowie powinni szybko prześcignąć swoich mistrzów w umiejętności posługiwania się komputerem i pisaniu programów. Dla nas nauczycieli, stanowić to będzie źródło satysfakcji z dobrze wykonanej pracy. Pisanie programu jest bardzo czasochłonne, dlatego nauczyciel może być autorem scenariusza, polecając pisanie samego programu uczniowi. W technikach napisanie programu edukacyjnego może być tematem pracy końcowej z przygotowania zawodowego. Uważam, że programy edukacyjne nie powinny wychodzić poza już i tak przeciążone programy nauczania poszczególnych przedmiotów. W czasopiśmie „Mikroclan", wydawanym na pięknym papierze, znalazłem program na Spectrum „W pracowni fizycznej", traktujący o drganiach, które nie występują w żadnym programie szkolnym, ani wyższej uczelni i nie występują... w przyrodzie, jedynie w wyobraźni programisty. Komputer rysuje wszystko, nawet rzeczy, które nie istnieją, jeśli jest źle zaprogramowany. Jest on szybki i dokładny, ale nie potrafi myśleć. Myślenie jest domeną nauczyciela-programisty. Komputer może być niewłaściwie wykorzystany w dydaktyce. Sprowadza on wtedy ucznia do roli niemyślącego wykonawcy, odbierając mu inicjatywę i pomysłowość.

W pierwszym szeregu entuzjastów wprowadzenia komputerów do szkół są nauczyciele fizyki. Są oni zainteresowani każdą możliwością wprowadzenia nowych technik do dydaktyki. Komputer na lekcjach fizyki ma szerokie pole działania. Może być zastosowany do różnych celów. Służy do modelowania zjawisk fizycznych, analizując je numerycznie, do szybkich obliczeń i przeprowadzania analizy wyników doświadczeń, dając dużą oszczędność czasu na lekcji. Wyniki pomiarów przedstawia łatwo na wykresie o zmiennej skali w celu przejrzystszego odwzorowania badanej zależności. Za pomocą komputera można symulować na ekranie różne doświadczenia lub działanie urządzeń niedostępnych w szkole, np. akceleratory, spektograf masowy, reaktor jądrowy. Nie wolno jednak zastąpić symulacją komputerową rzeczywistych eksperymentów, przeprowadzanych przyrządami fizycznymi na lekcji. Komputer może pomóc w uzupełnieniu i utrwaleniu przeprowadzonych wcześniej doświadczeń. Symuluje on doświadczenia w warunkach idealnych (bez tarcia) lub ekstremalnych. Za pomocą komputera, dzięki jego możliwościom graficznym, szybkości i niezawodności, można uzyskać duży postęp w nauczaniu fizyki. Najefektywniejszym wykorzystaniem komputera do badania zjawisk fizycznych jest sprzężenie go z obserwowanym obiektem. Komputer spełnia wtedy dwie funkcje: jest przyrządem pomiarowym i analizatorem otrzymanych wyników. Za pomocą komputera nauczyciel może sprawdzić wiadomości uczniów, używając do tego testu z omawianego działu. Wystawianie uczniom ocen przez komputer ożywia lekcje powtórzeniowe, najmniej lubiane przez uczniów i nauczycieli. Przy użyciu programu z instrukcją DATA sprawdzam wyniki obliczeń, dokonanych przez uczniów, jeśli każdy z nich otrzymał inne dane, co uniemożliwia odpisywanie od sąsiada.

Program edukacyjny powinien, moim zdaniem, być tak napisany, by zawierał w odpowiednim miejscu wszystkie wskazówki i polecenia

dla obsługującego go, by używać go mógł każdy nauczyciel lub uczeń, nawet nie znający zasad programowania. Dodatkowy tekst w publikacji prasowej jest wtedy zbędny, nie zostaje on przecież umieszczony na taśmie magnetofonowej lub dyskietce. Jeśli program przewiduje wprowadzenie do komputera pewnych danych za pomocą instrukcji INPUT, należy wtedy koniecznie w programie umieścić tekst objaśniający, w jakich granicach winny być te dane, by symulacja komputerowa zmieściła się na ekranie. Sposób przekazywania wiedzy za pomocą komputera jest efektowny, bardziej koncentruje uwagę ucznia. Po kilkuletnim stosowaniu na lekcjach komputerów MERITUM (szkolny) oraz SPECTRUM i ATARI (własne), zauważyłem, że uzyskuję zdecydowanie lepsze wyniki nauczania. Przeprowadziłem kilkakrotnie następujący eksperyment pedagogiczny: ucząc w 4 równoległych klasach, w 2 z nich utrwaliłem tematy z optyki geometrycznej — obrazy w zwierciadłach i soczewkach, po narysowaniu ich na tablicy i pokazaniu na ławie optycznej, za pomocą symulacji komputerowej. Wyniki przeprowadzanych sprawdzianów w tych klasach były dużo lepsze. Gdy na lekcji używany jest komputer, uczniowie nie słyszą często dzwonka, ogłaszającego oczekiwaną zwykle przerwę, a do opuszczania pracowni muszę ich zmuszać.

Prawdziwą przyjemność w stosowaniu komputera na lekcji odczułem jednak dopiero po zastosowaniu stacji dysków do ATARI (własna). Rodzice powinni zdać sobie sprawę z tego, że komputer w domu może w edukacji dziecka lub młodzieńca wyrządzić wiele szkody, jeśli używany będzie jedynie do zabawy. Kiedy uczeń odrobi lekcje, gdy całymi godzinami będzie ślęczał przed ekranem telewizora, pochłonięty grami komputerowymi.

Głównym celem obecności komputera w szkole jest przygotowanie do życia i działania w nowoczesnym społeczeństwie. Pamiętajmy o tym, że ci uczniowie, którzy dziś zasiadają w ławkach naszych szkół, będą pracować w XXI wieku, a ich narzędziem pracy z pewnością często będzie komputer. Brak umiejętności w posługiwaniu się komputerem w niedalekiej przyszłości będzie tak dotkliwy, jak obecnie brak umiejętności pisania i czytania.

mgr Jan GOLLA
nauczyciel fizyki
w Zespole Szkół Chemiczno-Mechanicznych
w Gliwicach

Jak uczyć informatyki

Zważywszy na duże zainteresowanie informatyką w szkole z jednej strony, a niewielkie doświadczenie w stosowaniu komputerów (jako przedmiotu i narzędzia) w klasach z drugiej, proponujemy wzajemną pomoc w tym zakresie. Zapraszamy przede wszystkim nauczycieli, którzy prowadzą podstawy informatyki w szkole średniej lub uczelni humanistycznej. Interesują nas zarówno rzeczywiste programy zajęć, jak i konkretne rozwiązania metodyczne nauczania informatyki — szczególnie programy wykorzystujące komputer, jako narzędzie i jako przedmiot nauczania.

Najlepsze propozycje zostaną opublikowane i godziwie nagrodzone:

I nagroda 50 tys. zł
II nagroda 30 tys. zł
III nagroda 20 tys. zł

PROGRAM LICEUM OGÓLN
(profil podstawowy i matem
ELEMENTY INFOR
(uzupełniający przedmi

CELE KSZTAŁCENIA I WYCHOWANIA

Zasadniczym celem zajęć z elementów informatyki jest nauczanie metod rozwiązywania za pomocą komputera prostych problemów na poziomie programu liceum ogólnokształcącego, dostosowanych do wiedzy i umiejętności uczniów. W trakcie tych zajęć uczniowie powinni poznać podstawy programowania oraz zdobyć praktyczne umiejętności posługiwania się szkolnym sprzętem informatycznym i jego oprogramowaniem. Dodatkowym celem nauczania elementów informatyki jest stworzenie warunków sprzyjających korzystaniu z komputera podczas uczenia się matematyki, fizyki, chemii, pracy-techniki i innych przedmiotów.

TREŚCI KSZTAŁCENIA I WYCHOWANIA

WIADOMOŚCI

1. Obsługa mikrokomputera

Zapoznanie się ze szkolnym mikrokomputerem. Konfiguracja mikrokomputera. Prawidłowe połączenie części składowych. Podstawowe zasady obsługi mikrokomputera. Włączenie. Układ klawiatury. Monitor ekranowy. Wskaźnik na ekranie. Wprowadzenie komend i tekstów. Kasowanie pomyłek. Rozumienie typowych odpowiedzi i reakcji mikrokomputera. Wprowadzanie i uruchamianie programu z pamięci zewnętrznej. Przerywanie wykonywania programu. Możliwe nieprawidłowości pracy mikrokomputera i sposoby postępowania w razie ich wystąpienia.

2. Praktyczne zastosowanie mikrokomputera

Poznanie działania wybranych programów, pokazujących typowe zastosowanie mikrokomputera: symulowanie procesów lub działania urządzeń, układanie i przekształcanie obrazów na ekranie, zbieranie i przechowywanie informacji, wykonywanie prostych obliczeń i zestawień rachunko-

wych, zapisywanie i redagowanie itp. Pokazanie uczniom sposobów wania się takimi programami; s wpływania na przebieg symulowan cesów, doboru parametrów symu urządzeń, komponowanie obrazó kształtu formularza w zestawien chunkowych, dobieranie formy re nego tekstu.

Analiza działania wybranych pro Określanie, jakie problemy mozna zywać za pomocą danego progra jak wygląda proces rozwiązywani mu. Zbadanie, czy da się wyodręb problemy, które są rozwiązywane osobno oraz czy da się ustalić, ja ków cząstkowych jest składany os wynik rozwiązania całego problem

3. Tworzenie rysunków na

Podstawowe komendy graficzn giwanie się wskaźnikiem (znaczni ekranie. Układ współrzędnych ek sowanie odcinków, punktów i łuk tworzenie z nich prostych rysun ekranie. Operowanie kolorami rysu

4. Procedury

Programowanie czynności, któ być wielokrotnie wykonywane. T procedur. Parametry procedur i po nie się nimi. Procedury z obliczan cią (funkcje, operacje). Wyrażenia dy posługiwania się funkcjami i wyra przy wykonywaniu obliczeń. Przeka wartości między procedurami; zmie balne i lokalne. Współpraca proce cedury rekurencyjne. Procedura re ca (edytor); posługiwanie się nią prawianiu procedur. Zapisywanie p na stałe w pamięci zewnętrznej. Wo nie procedur z pamięci zewnętrzne krokomputera w celu ich wykonan prawienia.

5. Styl programowania

Metoda zstępująca tworzenia procedur: od ogólnych założeń i pr zarządzającej wykonaniem całości cedur zapewniających wykonanie ności szczegółowych.

Opanowywanie i utrwalanie p wych nawyków przy pisaniu proced nowywanie i utrwalanie podsta umiejętności programowania struk go, w szczególności: czytelnego za

A W SZKOLE

NOKSZTAŁCĄCEGO
natyczno-fizyczny)
RMATYKI
iot nauczania)

e tekstów
w posługi-
sposobów
nych pro-
lowanych
w, doboru
niach ra-
edagowa-

ogramów.
a rozwią-
amu oraz
ia proble-
bnić pod-
kolejno i
ąk z wyni-
stateczny
u.

ekranie
ne. Postu-
ikiem) na
ranu. Ry-
ków oraz
nków na
unku i tła.

óre mają
Tworzenie
osługiwa-
ą wartoś-
. Przykła-
rażeniami
kazywanie
enne glo-
edur. Pro-
redagują-
przy po-
procedur
/czytywa-
ej do mi-
ja lub po-

zestawu
procedury
ci do pro-
ie czyn-

prawidło-
dur. Opa-
awowych
kturalne-
apisywa-

nia algorytmów, stosowania czytelnych nazw procedur i zmiennych oraz dobierania przejrzystych schematów współdziałania procedur. Używania w procedurach konstrukcji strukturalnych: powtarzania, instrukcji warunkowych i rekurencji.

Niezmienniki procedur: wartości, które są takie same przed wykonaniem procedury i po jej wykonaniu. Zależność rezultatów wykonania procedury od wartości początkowych zmiennych, mających wpływ na jej wykonanie. Przykłady niezmienników: współrzędne wskaźnika na ekranie, kierunek wskaźnika, kolor kreski. Tworzenie procedur tak, by pewne wielkości były z zasady niezmiennikami; korzyści wynikające z takiego podejścia przy planowaniu współpracy procedur. Pisanie przez uczniów własnych zestawów procedur do tworzenia złożonych rysunków.

6. Nieelementarne metody grafiki

Tworzenie procedur graficznych tak, aby ułatwić przekształcenia i modyfikacje rysunków, na przykład zmianę skali lub położenia elementów rysunku, rysowanie scen trójwymiarowych z różnych punktów widzenia. Zastosowanie list do opisywania elementów rysunku i operacji na listach umożliwiających programowanie przesunięć, obrotów i zmian skali.

7. Działania na tekstach

Działania na zdaniach, przedstawianych jako ciąg wyrazów, zapisany w formie listy. Operacje składania wyrazów w zdanie (listę). Wydobywanie ze zdania wyrazów: pierwszego, ostatniego, dowolnie wybranego. Przykłady procedur działających na listach i zasady tworzenia takich procedur. Użycie rekurencji przy przeglądaniu lub modyfikowaniu list. Tworzenie losowo generowanych zdań i danego zasobu wyrazów. Programowanie prostych, sensownych odpowiedzi na zdania wprowadzane z klawiatury. Projektowanie i programowanie prostych dialogów z komputerem, opartych na zasadzie rozpoznawania tekstów. Działania na słowach, rozumianych jako ciągi znaków.

8. Ćwiczenia samodzielne

Programowanie rozwiązań wybranych przez uczniów problemów. Utrwalanie i pogłębianie przyswojonych wiadomości. Analiza i ocena uzyskanych rozwiązań i stosowanych metod.

ORIENTACYJNY PRZYDZIAŁ GODZIN

1. Obsługa mikrokomputera	2/	3%
2. Praktyczne zastosowania mikrokomputera	6/	8%
3. Tworzenie rysunków na ekranie	4/	5%
4. Procedury	12/	16%
5. Styl programowania	12/	16%
6. Nieelementarne metody grafiki	12/	16%
7. Działania na tekstach	12/	16%
8. Ćwiczenia samodzielne	15/	20%
Razem godzin	75/	100%

PROGRAM LICEUM OGÓLNOKSZTAŁCĄCEGO — profil podstawowy i matematyczno-fizyczny (ELEMENT INFORMATYKI — uzupełniający przedmiot nauczania), to tytuł owocu niewątpliwie ogromnej pracy dziewięcioosobowego zespołu autorów. Czytanie tego opracowania potwierdza ogromną wiedzę jego twórców. Wydaje się jednak, że przeważyły akademickie (politechniczne?) doświadczenia autorów. Proponowany materiał bowiem, przy znacznym wysiłku nauczycieli i uczących się (najlepiej studentów), można opanować zapewne nie szybciej niż w 200—250 godzin, a nie w zakładanym czasie 75 godzin.

Skąd ta opinia? Czy aby rzeczywiście „zasadniczym celem zajęć z elementów informatyki jest nauczanie metod rozwiązywania za pomocą komputera prostych problemów na poziomie programu liceum ogólnokształcącego, dostosowanych do wiedzy i umiejętności uczniów"?[1] Wydaje się, że stwierdzenie to stwarza niebezpieczeństwo uznania, iż celem lekcji informatyki jest nauczanie ograniczonych możliwości komputerów. W każdym razie uczniowie mogą odnieść wrażenie, że komputery służą do rozwiązywania prostych zadań. PROGRAM nie przewiduje prezentacji rzeczywistych, możliwych i przyszłych zastosowań informatyki. Szkoda także, że nie zostało zdefiniowane znaczenie dostosowania „do wiedzy i umiejętności uczniów". Czym mierzyć owo „dostosowanie" — klasą? Wielkością miasta? Czy kwalifikacjami nauczyciela WF?

Jako inny cel kształcenia i **wychowania** (podkr. wł.) proponowana jest nauka programowania (o języku nie wspomniano, na wszelki wypadek, ani słowem). Rodzi to kolejną wątpliwość: „nauka programowania"? Kto potrafi nauczyć programowania (bo po co nauka, skoro niemożliwe jest nauczenie?) w szkole średniej w 75 godzin?

Podczas definiowania celów kształcenia i wychowania — wychowanie pozostało w tytule. Zapomniano o emocjach, „sympatiach", granicach, o których należy pamiętać poznając komputery. Podobnie jeśli idzie o treści kształcenia i wychowania (?) — wychowanie pozostało w tytule.

Praktyczne zastosowanie komputerów, które (wg PROGRAMU) powinny być zrozumiale (z zachowaniem wymagań dydaktyki) przedstawione uczniom, to „tylko":
— symulowanie procesów działań urządzeń,
— układanie i przekształcanie obrazów na ekranie,
— zbieranie i przechowywanie informacji,
— wykonywanie prostych obliczeń i zestawień rachunkowych, m.in. dobór kształtu formularza w zestawieniach rachunkowych,
— zapisywanie i redagowanie tekstów,
— analiza działania wybranych programów,
— itp.!

Jest to zaledwie jeden z 8 (ośmiu) punktów treści kształcenia i wychowania przewidzianych na 75 godzin. Darując sobie analizę pozostałych punktów, warto zwrócić uwagę na „Zagadnienia związane z realizacją programu". Podkreślono tu bardzo ważny problem edukacji komputerowej w ogóle — przygotowanie nauczycieli. Ale już za chwilę napisano, że: „komputer, używany w szkole, powinien mieć... pamięć zewnętrzną na nośnikach magnetycznych lub możliwość przyłączenia magnetofonu kasetowego" — czyżby magnetofon kasetowy był pamięcią z optycznym nośnikiem informacji? Na marginesie tej uwagi, ciekawy jestem, czym kierowali się autorzy, pisząc o magnetofonie kasetowym, a nie o magnetofonie w ogóle — przecież wszędzie tam, gdzie pasuje magnetofon kasetowy, można wstawić szpulowy — prawda?

Trochę ironicznie brzmi niepokój o efekt — „zniechęcenie uczniów i nauczycieli do dalszych przedsięwzięć tego typu".

Autorzy PROGRAMU twierdzą, że „rozsądek nakazuje", by jednak program ten zweryfikować (a jednak!)... w odpowiednich szkołach wyposażonych między innymi w specjalistyczne laboratoria informatyczne. Cóż z takiej weryfikacji, skoro jej wyniki mają być reprezentatywne dla wszystkich szkół (także tych bez laboratoriów).

Całkowicie się nie zgadzam z zaleceniem: „Należy unikać podawania wiadomości o charakterze teoretycznym, jak szczegóły bu-

dowy komputerów. Jak zatem wytłumaczyć uczniom problem miniaturyzacji, szybkości, niezawodności, źródła gubienia informacji (RAM) i jej pamiętania (ROM), ekonomii eksploatacji sprzętu itp.?

Zalecana jest natomiast analiza przykładowych programów. Autorzy nie zaryzykowali jednak choćby jakiejkolwiek egzemplifikacji tej propozycji, np.: najlepiej, według naszych doświadczeń, nadaje się do tego celu analiza programu sortowania. (Wiele przykładów zawiera prasa informatyczna).

Niestety, lekturę PROGRAMU utrudnia także „hermetyczny język", np. „Uzyskane wiadomości powinny być niezwłocznie wykorzystane i utrwalone w pracy na komputerze" — to jest PROGRAM, nie brudnopis.

Natomiast przykłady zawarte w PROGRAMIE wyraźnie nobilizują poziom licealistów — „przykład algorytmu jedzenia kaszki:

jedz kaszkę (czynność do wykonania)
jeśli talerz jest pusty, to koniec (warunek zakończenia)
zjedz łyżkę kaszki (czynność podstawowa każdego kroku)
jedz kaszkę (rekurencja)" (s. 12)

Doświadczenie uczy, że każdy początek jest trudny — nowe rodzi się w bólach (bolesne było czytanie PROGRAMU) — niemniej stanowi podstawę do tworzenia rzeczy innych, czasami lepszych. Stąd też w najbliższym numerze przedstawimy naszą propozycję PROGRAMU. Sprawdzoną w praktyce. Dlatego PROGRAM uzupełni „konspekt" na wszystkie zajęcia, które przewidziane są na około 60 godzin. Planujemy, że konspekt ten będzie przedstawiony w „odcinkach" co miesiąc — każda część będzie obejmowała sześciogodzinne zajęcia — wykłady lub ćwiczenia. Uzupełnieniem będą uwagi państwa oraz zespołu redakcji.

Włodzimierz GOGOŁEK

Co nas interesuje

(zajęcia z informatyki)

Sądzę, że potwierdzeniem naszego zainteresowania informatyką (jestem słuchaczem II roku Wojskowej Akademii Politycznej) były jedne z ostatnich zajęć z tego przedmiotu.

Celowym wydaje się podkreślenie oryginalnej ich formy. Słuchacze bowiem stając w roli wykładowcy starali się przez 5 minut zainteresować odbiorców dowolnie podjętą tematyką z dziedziny informatyki. Ze względu na ograniczenia czasowe głos zabrało 9 słuchaczy. Jako pierwszy „przejął pałeczkę" od wykładowcy kol. Kos. Tematem jego wystąpienia była prezentacja możliwości dysku twardego typu „Winchester". Referujący przytoczył parametry jednego z typów dysku, podkreślając jego dużą pojemność (360 MB) przy jednocześnie małych rozmiarach. Wspomniał o zastosowaniu nowych cienkich głowic manganowych, a także o cieńszej i wydajniejszej powłoce magnetycznej dysku. Te i inne zalety dysku plus niska cena jednostkowa w stosunku do jego pojemności stawiają firmę IBM w roli potentata w tej dziedzinie.

Z kolei głos zabrał kol. Dela prezentując sposoby wykorzystywania komputera w sferze teleinformacji. Mówił on między innymi o zastosowaniu mikrokomputera domowego, a także o możliwościach dialogu za pośrednictwem ekranów telewizyjnych. Ponadto przedstawił nowości brytyjskiej firmy PRESTER z dziedziny systemu teleinformatycznego.

Natomiast kol. Kosiński „przykuł" uwagę pozostałych słuchaczy problemem płytek krzemowych. Obecnie możemy mówić o toczącej się rywalizacji w dziedzinie konstruowania najbardziej doskonałych płytek krzemowych między Japonią (TOSHIBA) i Europą Zachodnią (SIEMENS-PHILIPS). Kol. Kosiński mówił także o problemach w zdobyciu surowca tj. krzemu, co powoduje nagły wzrost cen na pamięci komputera. Ponadto bardzo ciekawie wypowiedział się na temat promieniowania synktronowego tj. tzw. promieniowania X, które daje możliwość budowania pamięci RAM do 64 megabitów.

Następne 5 minut przeznaczone zostało dla kol. Górskiego. Podjął on temat dotyczący zastosowania komputera w rolnictwie. Współczesne rolnictwo, to także odpowiednio „uzbrojone" komputery, które informują człowieka, np. jakie składniki posiada próbka danej gleby i co należy na niej zasiać lub też jakie jest zapotrzebowanie na nawozy sztuczne itd. W sumie komputer staje się dziś wręcz nieodzownym doradcą dla rolnika.

Kolejnym referującym był kol. Rzeczycki. Swoje wystąpienie oparł o artykuł z tygodnika „NEWSWEEK" dotyczący przyszłościowego zastosowania komputera. Mówił on o współczesnym trendzie doskonalenia komputera na wzór mózgu człowieka, a także o jego zastosowaniu na rzecz przewidywania ludzkich działań. Warto jednak pamiętać, że mózg człowieka posiada ok. 100 mln połączeń między neuronami, natomiast współczesny komputer ma tylko 100 połączeń, a więc tyle co mózg mrówki. Jednakże generalnym celem jest zminimalizowanie roli programisty na rzecz samodzielności maszyny. Negatywnym natomiast jest fakt, iż najnowsze zdobycze w tej dziedzinie znajdują swoje zastosowanie przede wszystkim w przemyśle zbrojeniowym.

Kol. Kasprzak natomiast zabrał głos w kwestii zastosowania procesora i pamięci w komputerze. Wystąpienie to było oparte o najnowsze dane w tej dziedzinie. Wspomniano między innymi o Conection Machine, a więc o komputerze wyposażonym w wiele procesorów. Możliwość zastosowania jak największej liczby procesorów w komputerze ma na celu jak najszybsze i dokładne dokonanie określonej operacji (np. w dziedzinie grafiki). Programista komunikuje się z takim komputerem za pośrednictwem tradycyjnego komputera, korzystając ze standardowego języka programowania.

Możliwościami zastosowania komputera w dziedzinie marynistyki zajął się kol. Sykulak — oficer marynarki wojennej. Przedstawiona przy okazji plansza zobrazowała, iż każdy współczesny statek (okręt), to potencjalne „siedlisko" komputerów. Zastosowanie ich zaś dotyczy: dowodzenia statkiem czy okrętem, jego bezpieczeństwa, nawigacji i eksploatacji, w tym optymalizacji zużycia paliwa. W tym wypadku komputer okazuje się wielkim sprzymierzeńcem człowieka.

W dalszej części zajęć ciekawe wystąpienie zaprezentował kol. Tessikowski. Poruszył on kwestię możliwości wykorzystania komputera do gry w szachy. Już w końcu lat pięćdziesiątych pojawiły się pierwsze programy tego typu dla maszyn cyfrowych. Faktem jest, iż właśnie „królewska gra" okazała się doskonałym polem do badań nad sztuczną inteligencją. „Elektroniczny szachista" jednak nie jest doskonały, nie potrafi wygrać z arcymistrzami tej gry. Nie oznacza to jednak, że tak musi być, gdyż wraz z postępem w dziedzinie elektroniki komputer staje się coraz bardziej znaczący, a jego możliwości nie są dzisiaj jeszcze w pełni znane.

I wreszcie jako ostatni zabrał głos kol. Minkina. Skoro zajęcia odbywały się w Wojskowej Akademii Politycznej, nie można było pominąć tematyki militarnej, a ściślej wykorzystania komputera w siłach zbrojnych. Kol. Minkina zajął się kwestią komputeryzacji w armii Stanów Zjednoczonych. Przedstawił on słuchaczom między innymi możliwości systemu taktyczno-operacyjnego (Tactical Operation System) przeznaczonego do uzyskiwania, diagnozowania i przekazywania informacji. System ten może być wykorzystywany np. w kierowaniu ogniem artylerii, obronie powietrznej, kierowaniu tyłami czy też w dowodzeniu taktycznym. Znamienny jest fakt, iż już dzisiaj mówi się w USA o tzw. elektronicznym polu walki. Zaś dowództwo SZ USA ma opracowanych 8 planów wojny światowej i 60 planów działań w sytuacjach krytycznych, gdzie dane mogą być przekazywane w ciągu kilku minut. I wszystko za sprawą komputera. Jest to jednak w tym wypadku przerażające.

Niewątpliwie, omawiane zajęcia osiągnęły swój zasadniczy cel, tzn. poszerzyły wiedzę, a także zainteresowania problematyką dotyczącą roli komputera w życiu człowieka. Wypowiedzi słuchaczy nie były, co prawda stricte fachowe, ale takie nie miały być. Były natomiast odpowiednio dostosowane do odbiorcy, dzięki czemu w swej wymowie, w tym wypadku, były bardziej wyraziste i zrozumiałe.

Sądzę, iż taka forma zajęć powinna być częściej stosowana, bowiem prowadzących i słuchających mobilizuje oraz bardziej przybliża tego typu wiedzę i w takim zakresie, niż wykład tradycyjny.

Zbigniew CYWIŃSKI

Zadanie konikowe

Danuta KWASIŻUR,
Mieczysław SKONIECZNY

Dzisiejszy odcinek cyklu rozwiązywania anegdot matematycznych chcemy poświęcić zadaniu konikowemu na szachownicy. Podanie mówi, że uczony indyjski SISSA-NASIR, twórca szachów, zażądał od władcy, oczarowanego nową grą, pozornie skromnej zapłaty. Chciał dostać tyle zboża, ile otrzyma, gdy poprzez 64 pola szachownicy podwajane będzie 1 ziarno położone na pierwszym polu. Jak łatwo odgadnąć, jest to suma pierwszych 64 wyrazów postępu geometrycznego, którego wyrazem początkowym jest 1, a ilorazem 2. Władca Indii, człowiek bardzo bogaty, nie był niestety w stanie uhonorować autora gry. Dlaczego — cierpliwi Czytelnicy mogą sami policzyć, ile powinno wynosić honorarium wynalazcy.

Zadanie konikowe jest sformułowane następująco:
jak można skokami konika szachowego obiec wszystkie 64 pola szachownicy, korzystając z każdego pola tylko 1 raz.

Zagadnieniami skoków konika po szachownicy zajmowało się wielu znanych matematyków. Stworzyli oni szereg metod rozwiązania przedstawionego wyżej zadania, z których najbardziej znane to:

1. Metoda Eulera (koniec wieku XVII).
2. Metoda ramowa Moora (1843 rok).
3. Metoda ramowa Moivre'a (początek XVIII wieku).
4. Metoda podziału na ćwiartki Rogeta (połowa XIX wieku).

Zainteresowanych szczegółami Czytelników odsyłamy do rozdziału 6 „Lilavati".

Liczba różnych rozwiązań tego zadania, co wydaje się wręcz niemożliwe, jest ogromna — przekracza 31 milionów. Będzie to chyba zachętą dla Czytelników, którzy zechcą wprowadzić program do pamięci mikrokomputera i spróbować swoich sił.

CHARAKTERYSTYKA PROGRAMU

Po zainicjowaniu pracy, program przedstawia na ekranie zasady gry w następującej postaci:

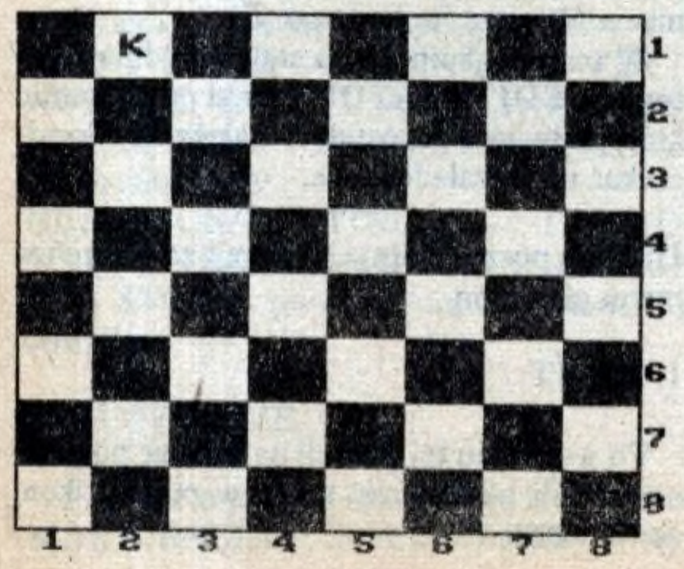
```
PROGRAM - ZADANIE KONIKOWE

Twoje zadanie polega na obejsciu wszystkich 64 pol szachownicy skokami konika
szachowego. Obowiazuje zasada, ze kazde pole moze byc wykorzystane tylko jeden
raz. Proba ponownego wykorzystania pola jest sygnalizowana komunikatem:
                        "TO POLE JUZ WYKORZYSTANO"
Jezeli bedziesz probowal wykonac ruch, jakim nie porusza sie konik, to program
zareaguje komunikatem:
                        "TAK NIE PORUSZA SIE KONIK"
Za chwile ukaze sie szachownica. W oknie gornym jest podawany nr ruchu. Aby
przesunac konika musisz podac nr kolumny i wiersza kazdorazowo wciskajac po
liczbie klawisz ENTER ( okienko srodkowe ). Numeracja jest widoczna obok szacho-
wnicy ( na dole nr kolumny, a z prawej strony nr wiersza ).

Przed podaniem wspolrzednych pierwszego ruchu na szachownicy nie widac konika.
W nastepnych ruchach aktualna pozycja konika jest sygnalizowana litera K.

Pola juz raz wykorzystane sa oznaczane kwadratem w srodku pola
```

Następnie po naciśnięciu dowolnego klawisza pojawia się plansza szachownicy i użytkownik może rozpocząć grę. Sytuację po 1 ruchu przedstawia poniższy rysunek:



Program opracowany w języku BASIC mikrokomputera AMSTRAD CPC 6128 ma następującą wersję źródłową:

CHARAKTERYSTYKA PODPROGRAMÓW

Do programu głównego należy dołączyć 3 podprogramy spełniające następujące funkcje:

1. Czyszczenie 2 okienka i sprawdzanie poprawności wprowadzonych danych (nr wiersza i kolumny).

```
10 DIM s(8,8)
20 'Ekran informacyjny
30 MODE 2:INK 0,26:INK 1,0:PAPER 1:PEN 0:CLS:BORDER 0
40 WINDOW £1,1,80,1,3:PAPER £1,0:PEN £1,1:CLS£1
50 WINDOW £2,1,80,23,25:PAPER £2,0:PEN £2,1:CLS£2
60 LOCATE £1,26,2:PRINT£1,"PROGRAM - ZADANIE KONIKOWE"
70 LOCATE 1,5:PRINT"Twoje zadanie polega na obejsciu wszystkich 64 pol
 szachownicy skokami konika  szachowego. Obowiazuje zasada, z
e kazde pole moze byc wykorzystane tylko jeden  raz. Proba ponownego w
ykorzystania pola jest sygnalizowana komunikatem:"
80 PRINT"                              "TO POLE JUZ WYKORZYSTANO"
                  Jezeli bedziesz probowal wykonac ruch, jaki
m nie porusza sie konik, to program  zareaguje komunikatem:"
90 PRINT"                              "TAK NIE PORUSZA SIE KONIK"
                  Za chwile ukaze sie szachownica. W oknie go
rnym jest podawany nr ruchu. Aby     przesunac konika musisz podac nr
kolumny i wiersza kazdorazowo wciskajac po"
100 PRINT"liczbie klawisz ENTER ( okienko srodkowe ). Numeracja jest w
idoczna obok szacho-wnicy ( na dole nr kolumny, a z prawej str
ony nr wiersza )."
110 PRINT"Przed podaniem wspolrzednych pierwszego ruchu na szachownicy
 nie widac konika.  W nastepnych ruchach aktualna pozycja koni
ka jest sygnalizowana litera K."
120 PRINT:PRINT"Pola juz raz wykorzystane sa oznaczane kwadratem w sro
dku pola"
130 LOCATE £2,26,2:PRINT£2,"NACISNIJ DOWOLNY KLAWISZ"
140 CLEAR INPUT:A$=INKEY$
150 IF A$="" GOTO 140
160 'Deklaracja kolorow i okien w trybie 1
170 MODE 1:INK 0,26:INK 1,0:INK 2,6:INK 3,19
180 PAPER 2:BORDER 6:CLS
190 WINDOW £1,28,40,1,3:PEN £1,1:PAPER £1,0:CLS£1
200 WINDOW £2,28,40,5,10:PEN £2,1:PAPER £2,0:CLS£2
210 WINDOW £3,28,40,12,16:PEN £3,1:PAPER £3,0:CLS£3
220 'Zerowanie tablicy s
230 FOR i=1 TO 8
240 FOR j=1 TO 8:s(i,j)=0:NEXT:NEXT
250 ruch=1
260 'Rysowanie szachownicy
270 PAPER 0:PEN 1:FOR i=0 TO 6 STEP 2
280 FOR i1=6*i+1 TO 6*i+3
290 FOR i2=1 TO 4
300 PRINT CHR$(143);CHR$(143);CHR$(143);"    ";
310 NEXT i2:PRINT: NEXT i1
320 FOR i1=6*i+4 TO 6*i+6
330 FOR i2=1 TO 4
340 PRINT "    ";CHR$(143);CHR$(143);CHR$(143);
350 NEXT i2:PRINT: NEXT i1:NEXT i
360  PAPER 2:PRINT" 1  2  3  4  5  6  7  8"
370 FOR i=1 TO 8 :LOCATE 25,i*3-1:PRINT USING"£";i:NEXT
380 'Czynnosci wstepne do pierwszego ruchu
390  PEN 3:PEN £1,2:PEN £2,2:PEN £3,2
400 ob=0:GOSUB 720:suk=0
410 SYMBOL AFTER 255
420 SYMBOL 255,199,206,220,240,240,220,206,199
430 'Kolejny ruch
440 CLS £1:LOCATE £1,2,2
450 PRINT £1,"NR RUCHU:";USING"££";ruch
460 GOSUB 600
470 ob=2:GOSUB 720:IF blad=2 GOTO 440
480 ob=3:GOSUB 720
490 IF blad=3 GOTO 440
500 'Sprawdzenie koloru pola, na ktore ma skoczyc konik
510 IF (in=1 OR in=3 OR in=5 OR in=7) AND (jn=1 OR jn=3 OR jn=5 OR jn=
7) THEN PAPER 1:GOTO 530
520 IF (in=2 OR in=4 OR in=6 OR in=8) AND (jn=2 OR jn=4 OR jn=6 OR jn=
8) THEN PAPER 1 ELSE PAPER 0
530 IF ruch<>1 THEN LOCATE (i-1)*3+2,(j-1)*3+2:PRINT CHR$(143)
540 LOCATE (in-1)*3+2,(jn-1)*3+2:PRINT CHR$(255)
550 ruch=ruch+1
560 i=in:j=jn:s(i,j)=1
```

```
570 GOSUB 1080
580 IF suk=1 GOTO 20
590.GOTO 440
```

2. Obsługa błędów
Umownie możliwym błędom nadano kody 0 — 3.
Wystąpienie każdego z tych kodów powoduje zmianę zawartości tekstu wyświetlanego w oknie nr 3. Interpretacja kodów jest następująca:
0 — brak błędów (w oknie wyświetla się komunikat „Życzymy sukcesów";
1 — wprowadzono niedopuszczalny nr kolumny lub nr wiersza (komunikat: „Ten numer musi być od 1 — 8");
2 — próba wykonania niedozwolonego ruchu (komunikat: „Tak nie porusza się konik");
3 — próba powtórnego wykorzystania pola (komunikat: „To pole już wykorzystano").

```
600 'Podprogram czyszczenia 2 okna i sprawdzania poprawnosci wprowadza
nych liczb
610 CLS £2:LOCATE £2,2,2
620 PRINT £2,"PODAJ NUMER"
630 LOCATE £2,2,4
640 INPUT £2,"KOLUMNY:",in
650 ob=1:nr=in:GOSUB 720
660 IF blad=1 GOTO 610
670 LOCATE £2,2,5
680 INPUT £2,"WIERSZA:",jn
690 nr=jn:GOSUB 720
700 IF blad=1 GOTO 610
710 RETURN
```

3. Zakończenie gry w sytuacji poprawnego rozwiązania zadania.

```
720 'Podprogram obslugi bledow
730 blad=0
740 IF ob=0 GOTO 1040
750 'Obsluga bledu 1 rodzaju
760 IF ob<>1 GOTO 820
770 IF (nr>0 AND nr<9) GOTO 1040
780 CLS £3:blad=1
790 LOCATE £3,3,2:PRINT £3,"TEN NUMER"
800 LOCATE £3,3,3:PRINT £3,"MUSI BYC"
810 LOCATE £3,3,4:PRINT £3,"OD 1 DO 8":GOTO 1030
820 'Obsluga bledu 2 rodzaju
830 IF ob<>2 GOTO 970
840 IF ruch=1 GOTO 1040
850 IF in=i+1 AND jn=j-2 GOTO 1040
860 IF in=i+2 AND jn=j-1 GOTO 1040
870 IF in=i+2 AND jn=j+1 GOTO 1040
880 IF in=i+1 AND jn=j+2 GOTO 1040
890 IF in=i-1 AND jn=j+2 GOTO 1040
900 IF in=i-2 AND jn=j+1 GOTO 1040
910 IF in=i-2 AND jn=j-1 GOTO 1040
920 IF in=i-1 AND jn=j-2 GOTO 1040
930 CLS £3:blad=2
940 LOCATE £3,4,2:PRINT £3,"TAK NIE"
950 LOCATE £3,2,3:PRINT £3,"PORUSZA SIE"
960 LOCATE £3,5,4:PRINT £3,"KONIK":GOTO 1030
970 'Obsluga bledu 3 rodzaju
980 IF s(in,jn)=0 GOTO 1040
990 CLS £3:blad=3
1000 LOCATE £3,4,2:PRINT £3,"TO POLE"
1010 LOCATE £3,4,3:PRINT £3,"JUZ WY-"
1020 LOCATE £3,3,4:PRINT £3,"KORZYSTANO"
1030 FOR T=1 TO 2000:NEXT
1040 CLS £3
1050 LOCATE £3,4,2:PRINT £3,"ZYCZYMY"
1060 LOCATE £3,4,4:PRINT £3,"SUKCESU"
1070 RETURN
1080 'Podprogram badania sukcesu
1090 FOR a=1 TO 8
1100 FOR b=1 TO 8
1110 IF s(a,b)=0 GOTO 1230
1120 NEXT b:NEXT a
1130 CLS £2:LOCATE £2,4,3:PRINT£2,"BRAWO!"
1140 LOCATE £2,2,4:PRINT£2,"ZWYCIĘZYLES"
1150 CLS £3:LOCATE £3,2,2:PRINT £3,"CZY CHCESZ"
1160 LOCATE £3,3,3:PRINT £3,"POWTORZYC"
1170 LOCATE £3,2,4:INPUT £3,"GRE (T/N):",a$
1180 suk=1
1190 IF a$="T" OR a$="t" GOTO 1230
1200 PAPER 2:CLS:PEN 0
1210 LOCATE 12,12:PRINT"KONIEC ZADANIA KONIKOWEGO"
1220 GOTO 1220
1230 RETURN
```

Życzymy wytrwałym Czytelnikom sukcesu w tej nieskomplikowanej grze konwersacyjnej.

Danuta KWASIŻUR
Mieczysław SKONIECZNY

„MASTERWORD"

·Masterword jest kolejnym edytorem tekstu przeznaczonym dla komputera Commodore — 64. W wersji opisanej poniżej (wersja 2.8J) napisany został przez Green Valey Publishing należący do Share Data Inc. w roku 1986.

Program ten zasadniczo różni się od opisywanego wcześniej edytora Geowrite. Może być wykorzystywany tylko przez osoby dysponujące stacją dysków.

Po uruchomieniu programu ukazuje się pierwsze menu o następującej zawartości:

1. MASTER WORD
2. BRIEF INSTRUCTIONS (krótka instrukcja obsługi)
3. QUIT — zakończenie pracy z edytorem

Po wybraniu „1" następuje załadowanie edytora do pamięci komputera. Program na samym początku przypomina nam o konieczności sformatowania dyskietki, na której zamierzamy przechowywać tworzone dokumenty. Do formatowania należy użyć opcji PREPARE DOCUMENT DISK zawartej w menu DISK UTILITIES o czym będzie mowa w dalszej części artykułu.

I. Menu główne edytora Masterword

Zarówno menu główne edytora, jak i poszczególne funkcje przedstawione są w Masterword za pomocą ikon. Główne menu składa się z następujących funkcji: PRINT, EDIT, FILE, LAYOUT, CUSTOMIZE, DISK UTILITIES. Odpowiednią funkcję wybieramy za pomocą klawisza kursora, oznaczając za pomocą przesuwającej się ramki tą, która nas aktualnie interesuje, co potwierdzamy naciskając RETURN. Powrót do stanu poprzedniego menu następuje zawsze po naciśnięciu klawisza funkcyjnego F1.

W pamięci komputera stale rezydują tylko funkcje EDIT i LAYOUT, stąd po wybraniu którejkolwiek z pozostałych należy chwilę odczekać na jej załadowanie.

II. Opis poszczególnych funkcji zawartych w menu głównym.

1. PRINT

Po wybraniu tej funkcji na ekranie po pewnej chwili ukazuje się menu w postaci ikon, zawierające:

TYPE OF PAPER — rodzaj papieru (pojedyncze kartki lub papier podawany w sposób ciągły).
TYPEWRITER — przejście do trybu pracy maszyny do pisania — dana linia jest drukowana po wciśnięciu RETURN.
PRINT DOCUMENT — drukowanie dokumentu.
SEE FINAL DOCUMENT — obejrzenie na ekranie gotowego dokumentu.
STARTING PAGE NUMBER — numer pierwszej strony. Numer ten drukowany jest na dole z napisem PAGE (strona).

2. EDIT

Funkcja ta posiada następujące menu:
ERASE TEXT — usunięcie tekstu.
INSERT EREASED TEXT — umieszczenie tekstu usuniętego.
MOVE TEKST — przesunięcie (przeniesienie) tekstu w inne miejsce dokumentu.
COPY TEXT — skopiowanie tekstu w inne miejsce dokumentu, bez usuwania go z miejsca poprzedniego.
FIND — znalezienie danego tekstu
FIND AND REPLACE — znalezienie tekstu i zastąpienie go innym. Tekst przeznaczony do usunięcia, przeniesienia lub skopiowania zaznaczamy przez „podświetlenie" kursorem.

3. FILE

Opcja FILE składa się z następujących funkcji:
INSERT DOCUMENT — umieszczenie dokumentu w pisanym tekście, w miejscu gdzie znajduje się aktualnie kursor.
GET DOCUMENT — wywołanie na ekran napisanego wcześniej dokumentu. Jeżeli na ekranie znajduje się tekst, to ulega on kasowaniu przez tekst wywoływany.
SAVE DOCUMENT — zapisanie dokumentu na dyskietkę.
BEGIN NEW DOCUMENT — rozpoczęcie pisania nowego dokumentu.

4. LAYOUT

Funkcja ta pozwala na nadanie pisanym dokumentom odpowiedniej formy. Jej menu składa się z:
SET SPACING — (SET TOP/BOTTOM MARGINS, SET LEFT/ RIGHT MARGINS, SET LINE SPACING) — ustalenie marginesów: dolnego, górnego, lewego i prawego oraz odstępu między wierszami.
START NEW PAGE — rozpoczęcie nowej strony.
ALIGNMENT — (ALIGN RIGHT, ALIGN LEFT, CENTER NEXT LINE) — wyrównanie tekstu do prawej lub lewej strony, ustawienie centralne następnego wiersza.
PRINT STYLE — (BOLDFACE TEXT, NORMAL TEXT, UNDERLINE TEXT) — rodzaj druku — tekst pogrubiony, normalny i podkreślony.
HEADINGS/FOOTINGS — DRUKOWANIE TEKSTU JAKO NAGŁÓWKA LUB STOPKI po lewej/prawej stronie lub centralnie.

5. CUSTOMIZE

Funkcja ta posiada następujące możliwości:
MAKE BACKUP DOCUMENTS — wykonywanie kopii dokumentu podczas zapisywania go na dyskietce.
CHANGE PRESET MARGINS — (SET TOP/BOTTOM MARGINS, SET LEFT/ /RIGHT MARGINS, SET LINE SPACING, SET TAB STOPS, SET PHYSICAL PAGE LENGTH, SET PHYSICAL PAGE WIDTH) — zmiana poprzednio ustalonych parametrów takich jak: marginesy górny, dolny, lewy, prawy, odstęp między wierszami, tabulacji, długości i szerokości strony (w calach).
SAVE CUSTOMIZE CHOICES — zapisanie na dysku ustalonych wyżej parametrów.
TYPE OF PRINTER — wybór typu drukarki.
ACCESSORIES — określenie, czy praca z jedną, czy z dwiema stacjami dysków.

6. DISK UTILITIES

Funkcja ta zawiera następujące możliwości:
PREPARE DOCUMENT DISK — sformatowanie dyskietki w celu zapisywania na niej dokumentów pisanych za pomocą programu Masterword.
ERASE DOCUMENT — wymazanie dokumentu z dyskietki.
CATALOG DISK — wyświetlenie katalogu dyskietki (katalogu dokumentów na dyskietce)
EXIT — wyjście z programu (uruchomienie innego programu).

III. Klawisze funkcyjne programu

W programie Masterword znajduje się plik o nazwie HELP.DOC, w którym opisano funkcje przypisane pewnym klawiszom, a użycie których jest bardzo przydatne podczas tworzenia dokumentów.
Do klawiszy tych należą:
D i INST DEL — usunięcie znaku na lewo od kursora.
CRSR prawy — przesunięcie kursora o 1 spację w prawo.
SHIFT CRSR prawy lub F7 — przesunięcie kursora jedno miejsce w lewo.
CLR HOME — przejście do początku dokumentu.
F3 — klawisz tabulacji.

Naciśnięcie klawisza CTRL i jednego z niżej wymienionych daje następujące możliwości:
W — przejście kursora do poprzedniego wyrazu
E — przejście kursora do następnego wyrazu
N — wymazanie następnego słowa
O — wyświetlenie zawartości poprzedniego ekranu
P — wyświetlenie zawartości następnego ekranu
T — przejście do początku dokumentu
B — przejście do końca dokumentu
F — przejście do następnej strony (strony wydruku)
G — przejście dó poprzedniej strony (strony wydruku)
C — zmiana trybu pisania (pisanie z rozsuwaniem istniejącego tekstu lub pisanie z zamazywaniem tekstu istniejącego).
V — wyświetlenie wartości ASCII danego znaku.
X — umieszczenie wartości znaku ASCII.

Możliwe jest także bezpośrednie wybieranie funkcji określonych przez ikony za pomocą równoczesnego naciśnięcia klawisza Commodore i następujących klawiszy:
A — umieszczenie dokumentu w pisanym tekście,
G — załadowanie dokumentu,
H — pokazanie na ekranie pliku HELP,
S — zapis dokumentu na dyskietce,
C — kopiowanie tekstu,
E — usunięcie tekstu,
M — umieszczenie usuniętego tekstu,
F — znalezienie tekstu,
R — znalezienie tekstu i zastąpienie go innym,
Y — przesunięcie tekstu,
B — drukowanie (litery) pogrubione,
D — początek nowej strony,
K — ustalenie prawego/lewego marginesu,
L — wyrównanie tekstu do lewej strony,
N — tekst normalny,
O — napisanie następnego wiersza centralnie,
Q — ustalenie górnego i dolnego marginesu,
T — nagłówek/stopka,
U — podkreślenie tekstu,
W — ustalenie odstępu między wierszami,
Z — wyrównanie tekstu do prawej strony.

Stosowanie wyżej wymienionych klawiszy przyspiesza znacznie pracę podczas pisania dokumentów, ponieważ nie wymaga poruszania się, przeglądania i wybierania poszczególnych ikon.

IV. Pisanie dokumentów za pomocą edytora Masterword.

Pracę z edytorem, jak już wcześniej wspomniano należy zacząć od przygotowania dyskietki na dokumenty. Należy w tym miejscu zaznaczyć, że dokumenty (pliki) umieszczone na takiej dyskietce, utworzone podczas pracy z Masterword, nie są widoczne przy próbie ich wylistowania za pomocą instrukcji LOAD „$", 8 i LIST. Chcąc utworzyć nowy dokument wybieramy z menu FILE opcję BEGIN NEW DOCUMENT.

Podczas pisania ekran podzielony jest na trzy okna. Okno główne przeznaczone jest do wpisywania tekstu. W oknie po lewej stronie na dole znajduje się informacja o numerze aktualnej strony, ilość wolnej pamięci komputera oraz ilość wolnego miejsca na dyskietce, na którą wpisywane są tworzone dokumenty (dwie ostatnie informacje w postaci jasnych pasków, zmniejszających swoją długość).

W oknie na dole po prawej stronie ekranu widoczny jest rzeczywisty układ tekstu na pisanej stronie (układ taki jaki zostanie wydrukowany).

Podczas pisania dłuższych tekstów należy co jakiś czas wpisać na dyskietkę tekst już napisany, aby w sytuacji awaryjnej (zanik napięcia) nie było konieczne zaczynanie całej pracy od nowa.

V. Ocena edytora

Masterword jest programem bardzo przyjaznym i łatwym w obsłudze. Zastosowane ikony ułatwiają bardzo pracę, szczególnie początkującym użytkownikom. Masterword w zasadzie spełnia wszystkie wymagania stawiane „poważnym" edytorom tekstu przeznaczonym na komputery osobiste.

Do jego wad należy zaliczyć jedynie dosyć ubogi wybór rodzajów pisma (zastosowany krój czcionki może się jednak podobać) oraz oczywiście brak polskiego alfabetu.

Tadeusz CISEK

Kompilator języka C firmy HISOFT

(Amstrad CPC-6128)

Język C z racji nieco skomplikowanej syntaktyki jest jeszcze mniej popularny od BASIC'a, czy PASCAL'a, lecz pomimo to istnieje wiele jego wersji mikrokomputerowych. Do opanowania podstawowych konstrukcji świetnie nadaje się kompilator wyprodukowany przez firmę HISOFT, która z konsekwencją lansuje własny styl. Dzięki temu, ten kto opanował obsługę kompilatora PASCAL na Spectrum, czy Amstradzie, nie powinien mieć w zasadzie kłopotów z obsługą C. Niniejszy opis dotyczy wersji Amstrada.

1. URUCHOMIENIE KOMPILATORA

Kompilator składa się z jednego zbioru (HISOFTC. BIN). Wystarczy wpisać

RUN"HISOFT

i włożyć dyskietkę lub uruchomić magnetofon z kasetą. Po prawidłowej operacji ładowania powinien ukazać się napis:

HISOFT — C Compiler v1.2
Copyright C 1984 HISOFT

oraz znak zachęty > będący sygnałem zgłoszenia się edytora. Obok pojawi się kursor. Ekran będzie zorganizowany w trybie 2.

2. EDYTOR

Komendy edytora są jednoliterowe i nie mogą mieć więcej niż 4 parametry. Pierwsze dwa są numeryczne (początkowo równe 10), a kolejne dwa tekstowe (początkowo puste). Nie wszystkie są zapamiętywane, co stanowi pewne odstępstwo od standardu HISOFT. Oto skrócony opis opcji edytora:

I m, n	—	wpisywanie linii programu od m co n;
L m, n	—	listowanie programu od linii m do n;
D m, n	—	kasowanie bloku linii od m do n; (kasowanie jednej linii D m);
R m, n	—	przenumerowanie linii od m co n;
W m, n	—	analogicznie jak L, lecz na drukarkę;
S, znak	—	zmiana separatora (początkowo przecinek);
V	—	opcja informacyjna podająca bieżące wartości separatora oraz 4 parametrów komend, poza tym wyświetlenie adresu początku i końca pamięci zawierającej program źródłowy;
F m, n, f, s	—	szukanie ciągu znakowego f w tekście między liniami m, n; po edycji linii można wybrać dwie kombinacje: CTRL f — dalsze szukanie; CTRL s — zamiana łańcucha f na s;
P m, n, s	—	zapis programu źródłowego od linii m do n pod nazwą s na dyskietkę lub kasetę;
G,, s	—	odczyt programu źródłowego o nazwie s;
E n	—	redagowanie linii n; opcje dostępne dla tej czynności są następujące:
RETURN	—	zakończenie edycji linii z uwzględnieniem poprawek;
ESC	—	zakończenie edycji linii bez uwzględniania poprawek;
prawo	—	kursor w prawo;
lewo	—	kursor w lewo
CTRL prawo	—	kursor na koniec linii;
CTRL lewo	—	kursor na początek linii;
COPY	—	kopiowanie drugim kursorem;
CTRL TAB	—	przełącznik trybu wkładania znaków w tekst na tryb nakładania znaków na tekst i vice versa;
CLR	—	kasowanie znaku pod kursorem;
DEL	—	kasowanie znaku na lewo od kursora;

Jak można zauważyć, opcje trybu redagowania linii są bardziej zbliżone do edytora BASIC'a w Amstradzie, niż do poprzednich rozwiązań firmy.

Kolejne linie programu wprowadza się przy wykorzystaniu opcji I lub bezpośrednio wczytując numer linii i instrukcje. Jeśli w pewnym momencie edytor zareaguje raportem:

ERROR 60
LIMIT: no more memory

co oznacza przepełnienie pamięci, to należy wpisać dyrektywę kompilatora #errór, która spowoduje odzyskanie pamięci zajętej przez komunikaty o błędach (ok. 2KB).

Powrót do BASIC'a następuje po naciśnięciu CTRL—SHIFT—ESC, przy czym niszczy to kompilator i „ciepły" restart nie jest możliwy.

3. KOMPILACJA PROGRAMU UTWORZONEGO W EDYTORZE

Kompilacja utworzonego pod edytorem programu zachodzi po wprowadzeniu opcji c (po znaku zachęty). Kompilator przedstawi się wówczas znaną już formułą. Teraz wystarczy podać dyrektywę #include, po której zachodzi kompilacja z jednoczesnym listowaniem programu źródłowego na ekranie. W przypadku używania magnetofonu proces może być przerywany napisem Press PLAY then any key, szczególnie gdy dołączamy zbiory biblioteczne.

Przejście do fazy uruchomienia programu nastąpi po naciśnięciu kombinacji CTRL-z. Wygodnie jest zasygnalizować ten moment w programie odpowiednim komentarzem. Oto typowy schemat przebiegu obsługi kompilacji w skrócie:

```
>c
        (formuła)
#include
        (kompilacja)
CTRL-z
        (type y to run)
y
```

4. KOMPILACJA „NA GORĄCO"

Jak wspomniałem wyżej, opcja c pozwala wejść do trybu kompilacji. Możliwe jest wtedy podawanie dyrektyw kompilatora oraz redagowanie programu (bez numerów linii), który zostanie natychmiast przetłumaczony po CTRL-z. Wygodne do testowania, gdyż kompilacja taka nie niszczy programu pod edytorem.

5. PRACA INTERPRETACYJNA

Kompilator C ma dość ciekawą własność. Jeśli znajdujemy się na poziomie kompilatora, to dyrektywa #direct+ spowoduje przejście do trybu pracy interpretacyjnej. Możliwe jest wpisywanie pojedynczych instrukcji, które zostaną natychmiast wykonane, podobnie jak w trybie bezpośrednim w BASIC'u. Wyłączenie tego trybu zajdzie po #direct—.

6. DYREKTYWA #INCLUDE

Poznana już dyrektywa #include jest podstawą dla procesu kompilacji programu znajdującego się w pamięci operacyjnej, a napisanego pod edytorem. Używa się jej również do innych celów:

a) kompilacji zbioru o podanej nazwie

#include nazwa. typ

b) włączeniu innego zbioru zawierającego tekst źródłowy do kompilacji tekstu głównego przez użycie dyrektywy #include nazwa. typ jako linii w programie głównym;

c) kompilacji warunkowej; jeśli użyjemy w programie linii:

#include? nazwa. typ?

to do kodu wynikowego zostaną włączone tylko te funkcje, które są używane w programie głównym;

Inne postacie tej dyrektywy w standardzie języka, a mianowicie:

#include <nazwa. typ>
#include „nazwa. typ"

są równoważne zapisowi #include nazwa. typ.

7. TWORZENIE SAMODZIELNEGO ZBIORU WYNIKOWEGO

Dyrektywa #translate nazwa. typ pozwala na uzyskanie niezależnego zbioru wynikowego, który nie będzie wymagał otoczenia kompilatora c w celu uruchomienia. Zbiór taki zostanie odłożony na dyskietce lub kasecie, a z poziomu AMSDOS'u może być uruchamiany przez:

RUN" nazwa. typ

8. INNE DYREKTYWY KOMPILATORA

#list+	włączenie wydruku podczas kompilacji;
#list-	wyłączenie wydruku podczas kompilacji;
#define	w kompilatorze HISOFT jest to opcja ograniczona w zasadzie do przedefiniowywania nazw stałych;

W porównaniu do standardu języka brakuje implementacji dyrektyw #undef, #line oraz #if i pochodnych.

Krzysztof MAMCARZ

Jak oszukać, czyli jak rozszerzyć edytor ZX...

Niestety, edytor ZX Spectrum nie jest zbyt udaną konstrukcją. Poprawianie linii programu za jego pomocą, to żmudna praca, zwłaszcza gdy są one długie i zawierają błędy składniowe. Aby dostać się do ostatniego znaku, kursor musi przejść przez całą linię i bardzo często trwa to tyle czasu, że można sobie bez pośpiechu zrobić nawet herbatę.

Proponuję zwiększyć możliwości edytora, trochę go oszukując. Edytor wykonuje polecenia użytkownika przez kontrolę klawiatury, a ściślej: sprawdza wartości zmiennych systemowych LAST-K, FLAGX, FLAGS, FLAS2, MODE. Te zmienne systemowe informują m.in., czy i jaki klawisz został wciśnięty, w jakim trybie znajduje się kursor, itp. Są one jakby „oknem na świat" edytora. O początku i końcu edytowanej linii informują zmienne E-LINE i WORKSP, a o położeniu kursora zmienna K-KUR. O wszystkich tych zmiennych systemowych była już mowa w poprzednich numerach „IKS-a" i dlatego nie będę ich omawiał.

Władzę nad edytorem uzyskamy, kontrolując i odpowiednio zmieniając wartości tych zmiennych. W tym celu skorzystamy z drugiego typu przerwań (IM 2), co oznacza, że co 1/50 s. oprócz kontroli klawiatury zostanie uruchomiony nasz program. Opiszę sposób oszukania edytora na przykładzie: tryb specjalny oznaczony kursorem X uzyskujemy naciskając jednocześnie Symbol Shift i Spację (SS+Sp). W rzeczywistości jest inaczej: w czasie przerwania nasz program sprawdza czy jednocześnie wciśnięto SS+Sp, jeśli tak, to ustawia zmienną LAST.K, tak jakby wciśnięto CAPS LOOK (do LAST-K wpisuje wartość 6). Dodatkowo do zmiennej MODE wpisuje 138. Tymczasem edytor na podstawie LAST-K ustawia tryb C kursora, ale ponieważ zamiast MODE=0 jest MODE=138, zostaje wyświetlony kursor X. Dodatkowo program zapamiętuje aktualny tryb pracy, aby po ponownym wciśnięciu SS+Sp przywrócić dawny stan kursora.

W trybie X uzyskujemy następujące możliwości:
kursory — pozwalają poruszać się w odpowiednich kierunkach po edytowanej linii.
P — ustawia kursor na początku edytowanej linii

```
 100 CLEAR 65014
1000 RESTORE 9000: FOR k=0 TO 25
1010 LET s1=0: LET s2=0: LET z=1
1030 FOR l=0 TO 19: READ a: LET
s1=s1+a: LET s2=s2+z*a: LET z=-z
: POKE 65015+20*k+l,a: NEXT l: R
EAD a1: READ a2: IF s1<>a1 OR a2
<>s2 THEN PRINT "blad w linii :"
;9000+10*k: STOP
1040 NEXT k
1050 READ a: POKE 65535,a: IF a<
>134 THEN PRINT "blad w linii 92
60": STOP
2000>RANDOMIZE USR 65015: REM wl
aczenie rozszerzenia edytora

9000 DATA 62,254,237,71,237,94,1
75,50,208,255,201,205,56,0,245,5
8,113,92,203,111,2927,547
9010 DATA 194,211,254,197,213,22
9,58,8,92,254,32,194,80,254,205,
142,2,62,24,186,2891,-583
9020 DATA 194,80,254,58,208,255,
203,127,202,56,254,203,191,50,20
8,255,58,209,255,50,3370,684
9030 DATA 65,92,195,72,254,203,2
55,50,208,255,58,65,92,50,209,25
5,62,138,50,65,2693,203
9040 DATA 92,62,6,50,8,92,195,20
8,254,58,208,255,203,127,202,208
,254,237,86,58,2863,153
9050 DATA 8,92,254,8,202,206,254
,254,10,202,213,254,254,11,202,3
3,255,254,9,202,3177,145
9060 DATA 206,254,254,245,202,99
,255,254,112,202,99,255,254,80,2
02,99,255,254,240,202,4023,135
9070 DATA 113,255,254,107,202,11
3,255,254,75,202,113,255,254,233
,202,135,255,254,100,202,3833,-1
87
9080 DATA 135,255,254,68,202,135
,255,254,255,202,120,255,254,122
,202,120,255,254,90,202,3889,155
9090 DATA 120,255,254,238,202,16
8,255,254,105,202,168,255,254,73
,202,168,255,254,7,202,3891,-247
9100 DATA 148,255,254,13,202,166
,255,58,59,92,203,175,50,59,92,2
37,94,225,209,193,3039,93
9110 DATA 241,201,6,33,42,91,92,
126,254,13,202,113,255,254,165,2
10,244,254,254,32,3082,428
9120 DATA 210,235,254,4,35,34,91
,92,16,230,195,105,255,205,7,255
,218,105,255,195,2996,76
9130 DATA 1,255,255,195,2,254,42
,91,92,195,234,254,33,128,255,22
,0,203,63,245,2819,-865
9140 DATA 95,25,241,126,218,26,2
55,15,15,15,15,230,15,79,120,145
,71,201,6,32,1945,157
9150 DATA 237,91,91,92,26,254,16
5,210,86,255,254,32,210,56,255,2
54,13,202,56,255,3094,-308
9160 DATA 4,27,42,89,92,175,237,
82,210,99,255,237,83,91,92,16,21
9,26,254,32,2362,614
9170 DATA 218,55,255,19,237,83,9
1,92,195,105,255,205,7,255,218,1
05,255,237,91,91,3069,575
9180 DATA 92,195,55,255,42,89,92
,34,91,92,62,8,50,8,92,195,206,2
54,42,97,2051,-403
9190 DATA 92,43,195,102,255,237,
91,91,92,42,97,92,43,43,205,229,
25,195,113,255,2537,-121
9200 DATA 237,91,89,92,42,91,92,
205,229,25,195,99,255,58,110,92,
50,73,92,58,2275,507
9210 DATA 111,92,50,74,92,205,17
9,255,195,206,254,237,94,205,179
,255,62,13,50,8,2818,-284
9220 DATA 92,195,208,254,175,50,
208,255,58,209,255,50,65,92,58,1
06,92,203,95,202,2922,-310
9230 DATA 202,255,203,159,195,20
4,255,203,223,50,106,92,201,0,0,
3,98,54,133,52,2688,544
9240 DATA 85,68,68,68,68,52,68,6
8,83,69,84,68,82,34,102,70,133,1
34,119,135,1656,126
9250 DATA 134,133,119,137,101,13
5,102,105,88,165,85,120,118,101,
118,103,101,107,69,103,2244,-174
9260 DATA 134
```

```
00012  ; program wykorzystuje przerwa-
00013  ; nia maskowalne, ustawia tryb 2
00014  ; przerwan.
00015  ; program nie jest relokowalny.
00016  ;
00017  rej.i  equ  0feh;wartosc rej. i
00018  ; kody rozkazow
00019  EDIT   equ  7
00020  DIM    equ  233
00021  COPY   equ  255
00022  LIST   equ  240
00023  INPUT  equ  238
00024  ENTER  equ  13
00025  PRINT  equ  245
00026  kur.X  equ  138 ;kursor X
```

```
00027  kur.C  equ  8    ;kursor C
00028  kur.lf equ  8    ;kursor w lewo
00029  kur.rt equ  9    ;kursor w prawo
00030  kur.dn equ  10   ;kursor w gore
00031  kur.up equ  11   ;kursor w dol
00032  ; adresy procedur ROM-u
00033  recmin equ  19e5h
00034  keyb   equ  38h
00035  keybsc equ  28eh
00036  ; adresy zmie nnych systemowych
00037  flags  equ  5c3bh
00038  flags2 equ  5c6ah
00039  flagx  equ  5c71h
00040  e_ppc  equ  5c49h
00041  oldppc equ  5c6eh
00042  e_line equ  5c59h
00043  mode   equ  5c41h
00044  k_kur  equ  5c5bh
00045  worksp equ  5c61h
00046  last_k equ  5c08h
00047  ;
00048          org  65015
00049  ; inicjalizacja trybu 2
00050  ustim2 ld   a,rej.i
00051         ld   i,a
00052         im   2
00053         xor  a
00054         ld   (flag.p),a
00055         ret
00056  ;
00057  ; rozszerzenie przerwania
00058  ; program wykonywany w czasie
00059  ; przerwania
00060  przerw call keyb
00061         push af
00062         ld   a,(flagx)
00063         bit  5,a
00064         jp   nz,tryb.i ;tryb inp.
00065         push bc
00066         push de
00067         push hl
00068         ld   a,(last_k)
00069         cp   32
00070         jp   nz,etyk2
00071         call keybsc
00072         ld   a,18h
00073         cp   d
00074         jp   nz,etyk2
00075  ;zmiana trybu
00076         ld   a,(flag.p)
00077         bit  7,a
00078         jp   z,ustaw7
00079         res  7,a ;wyl. trybu X
00080         ld   (flag.p),a
00081         ld   a,(modez)
00082         ld   (mode),a
00083         jp   etyk1
00084  ustaw7 set  7,a ;wlacz. trybu X
00085         ld   (flag.p),a
00086         ld   a,(mode)
00087         ld   (modez),a
00088         ld   a,kur.X
00089         ld   (mode),a
00090  etyk1  ld   a,kur.C
00091         ld   (last_k),a
00092         jp   konprw
00093  etyk2  ld   a,(flag.p)
00094         bit  7,a
00095         jp   z,konprw
00096         im   1
00097  ; program obslugujacy rozkazy
00098  ; trybu X
00099         ld   a,(last_k)
00100         cp   kur.lf
00101         jp   z,koniec
00102         cp   kur.dn
00103         jp   z,kurDN
00104         cp   kur.up
00105         jp   z,kurUP
00106         cp   kur.rt
00107         jp   z,koniec
00108         cp   PRINT
00109         jp   z,obsl.p
00110         cp   'P'
00111         jp   z,obsl.p
00112         cp   'P'
00113         jp   z,obsl.p
00114         cp   LIST
00115         jp   z,obsl.k
00116         cp   'K'
00117         jp   z,obsl.k
00118         cp   'K'
00119         jp   z,obsl.k
00120         cp   DIM
00121         jp   z,obsl.d
00122         cp   'd'
00123         jp   z,obsl.d
00124         cp   'D'
00125         jp   z,obsl.d
00126         cp   COPY
00127         jp   z,obsl.z
00128         cp   'Z'
00129         jp   z,obsl.z
00130         cp   'Z'
00131         jp   z,obsl.z
00132         cp   INPUT
00133         jp   z,obsl.i
00134         cp   'i'
00135         jp   z,obsl.i
00136         cp   'I'
00137         jp   z,obsl.i
00138         cp   EDIT
00139         jp   z,obslED
00140         cp   ENTER
00141         jp   z,obslER
00142  ; zablokowanie pozostalych
00143  ; klawiszy
00144         ld   a,(flags)
00145         res  5,a
00146         ld   (flags),a
00147  koniec im   2
00148  konprw pop  hl
00149         pop  de
00150         pop  bc
00151  tryb.i pop  af
00152         ret
00153  ;
00154  ; procedury obslugujace
00155  ;
00156  ; procedura: kursor w dol
00157  kurDN  ld   b,33
00158  naprz  ld   hl,(k_kur)
00159         ld   a,(hl)
00160         cp   ENTER
00161         jp   z,obsl.k
00162         cp   165
00163         jp   nc,rozk
00164         cp   32
00165         jp   nc,znak
00166  znaknd inc  b
00167  znak   inc  hl
00168         ld   (k_kur),hl
00169         djnz naprz
00170         jp   lastLF
00171  rozk   call rozkwk
00172         jp   c,lastLF
00173         jp   most
00174  ;
00175         org  256*rej.i+255
00176  ; adres obslugi przerwania trybu
00177  ; im 2
00178         dw   przerw
```

```
00179  ;
00180  most   ld   hl,(k_kur)
00181         jp   znaknd
00182  ;wewn. proc. przeskakiw. rozk.
00183  rozkwk ld   hl,tabl-164/2
00184         ld   d,0
00185         srl  a
00186         push af
00187         ld   e,a
00188         add  hl,de
00189         pop  af
00190         ld   a,(hl)
00191         jp   c,nieprs
00192         rrca
00193         rrca
00194         rrca
00195         rrca
00196  nieprs and  00001111b
00197         ld   c,a
00198         ld   a,b
00199         sub  c
00200         ld   b,a
00201         ret
00202  ;
00203  ; procedura: kursor w gore
00204  kurUP  ld   b,32
00205  cofaj  ld   de,(k_kur)
00206         ld   a,(de)
00207         cp   165
00208         jp   nc,rozk1
00209         cp   32
00210         jp   nc,znak1
00211         cp   ENTER
00212         jp   z,znak1
00213  znakn1 inc  b
00214  znak1  dec  de
00215         ld   hl,(e_line)
00216         xor  a
00217         sbc  hl,de
00218         jp   nc,obsl.p
00219         ld   (k_kur),de
00220         djnz cofaj
00221         ld   a,(de)
00222         cp   32
00223         jp   c,znakn1
00224         inc  de
00225         ld   (k_kur),de
00226         jp   lastLF
00227  rozk1  call rozkwk
00228         jp   c,lastLF
00229         ld   de,(k_kur)
00230         jp   znakn1
00231  ;
00232  ; procedura: kursor na poczatek
00233  obsl.p ld   hl,(e_line)
00234  krs.lf ld   (k_kur),hl
00235  lastLF ld   a,kur.lf
00236         ld   (last_k),a
00237         jp   koniec
00238  ;
00239  ; procedura: kursor na koniec
00240  obsl.K ld   hl,(worksp)
00241         dec  hl
00242         jp   krs.lf
00243  ;
00244  ; procedura: kasuj za kursorem
00245  obsl.z ld   de,(k_kur)
00246         ld   hl,(worksp)
00247         dec  hl
00248         dec  hl
00249         call recmin
00250         jp   obsl.K
00251  ;
00252  ; procedura: kasuj do kursora
00253  obsl.d ld   de,(e_line)
00254         ld   hl,(k_kur)
00255         call recmin
00256         jp   obsl.p
00257  ;
00258  ; procedura: EDIT
00259  obslED ld   a,(oldppc)
00260         ld   (e_ppc),a
00261         ld   a,(oldppc+1)
00262         ld   (e_ppc+1),a
00263         call norm.k
00264         jp   koniec
00265  ;
00266  ; procedura: ENTER
00267  ; procedura: ENTER+tryb 1
00268  obslER im   2
00269  obsl.i call norm.k
00270         ld   a,ENTER
00271         ld   (last_k),a
00272         jp   konprw
00273  norm.k xor  a
00274         ld   (flag.p),a
00275         ld   a,(modez)
00276         ld   (mode),a
00277         ld   a,(flags2)
00278         bit  3,a
00279         jp   z,ustaw3
00280         res  3,a
00281         jp   etyk4
00282  ustaw3 set  3,a
00283  etyk4  ld   (flags2),a
00284         ret
00285  ; zmienne systemowe procedury
00286  flag.P ds   1
00287  modez  ds   1
00288  ;
00289  ; tablica ilosci liter rozkazow
00290  ; od RND (kod 165) do COPY (kod
00291  ; 255). Kazdy bajt zawiera dlu-
00292  ; gosci dwoch kolejnych rozkazow
00293  ; zapisanych joko liczby BCD
00000  ;
00294  tabl   ;      45  67  89  01  23
00295  t16... db     03h,62h,36h,85h,34h
00296  t17... db     55h,44h,44h,44h,44h
00297  t18... db     34h,44h,44h,53h,45h
00298  t19... db     54h,44h,52h,22h,66h
00299  t20... db     46h,85h,86h,77h,87h
00300  t21... db     86h,85h,77h,89h,65h
00301  t22... db     87h,66h,69h,58h,0a5h
00302  t23... db     55h,78h,76h,65h,76h
00303  t24... db     67h,65h,6bh,45h,67h
00304  t25... db     86h
00305         end
```

K — ustawia kursor na końcu edytowanej linii
D — kasuje znaki edytowanej linii od początku linii do kursora
Z — kasuje znaki edytowanej linii za kursorem
ENTER — działa, jak klasyczne ENTER, ale wyłącza tryb X
I — tak samo jak ENTER, ale zostaje ustawiony tryb pierwszy przerwań
EDIT — edytuje linię wskazywaną przez CONTINUE
Uwaga: pozostałe klawisze w trybie X są ignorowane
SS+Sp — włącza i wyłącza tryb X

Ponieważ program jest przedstawiony w zapisie mnemonicznym, to ze względu na czytelność nie mógł być zbyt obszerny (stąd niewiele dodatkowych funkcji). Należy go więc traktować jako przykład, a nie jak ostateczne rozwiązanie.

Proponuję natomiast czytelnikom wtajemniczonym w programowanie w assemblerze uzupełnienie programu o nowe funkcje. Myślę, że wspólnymi siłami ulepszymy edytor ZX Spectrum. Wraz z redakcją „IKS-a" czekam na propozycje.

Dla tych, którzy chcą skorzystać z prezentowanego programu, zamieszczam program ładujący kod maszynowy.

Tajemnica żółtego światła

Włączył magnetofon. Muzyka wypełniła pokój. Erta Kitt śpiewała C'est si bon. Ulubioną piosenkę jego młodości. Słuchał jej zawsze, zanim wyruszył w trasę, zajadając sandwicza i popijając sok grejpfrutowy. Chwycił przewieszoną przez poręcz krzesła cienką kurtkę z impregnowanego papieru, z której uśmiechało się słońce i przyciągał wzrok tęczowy napis Miami Beatch. Zbiegł schodami do holu, podszedł do wiszącego na ścianie „wiecznego" kalendarza i ustawił pokrętłami datę. Był 28 sierpnia 1979 roku.

Sprawnie wspiął się do umieszczonej wysoko nad ziemią szoferki swej potężnej ciężarówki, włączył radio i w dostojnym rytmie jakby najważniejszej w życiu pracy uruchomił silnik. Skontrolował dokładnie wskaźniki rozbłysłe kolorowymi światełkami na pulpicie. Niedbale wrzucił do skrytki obok kierownicy potężnego colta kalibru 32.

Uśmiechnął się do swych myśli: — Z bronią nie rozstawaj się nigdy, chłopcze — przestrzegał szef. Tak jakby na swojej nowoczesnej, niezwykłej, trzydziestometrowej ciężarówce z naczepą przewoził złoto, a nie artykuły żywnościowe na lokalnej linii transportowej pomiędzy Winnchester a Friedrickburgiem. — Z bronią bezpieczniej i pewniej — tłumaczył szef. Tak jakby on, 28-letni kierowca zawodowy, Hary Joe Turner, musiał się czegoś lub kogoś bać. On, barczyście zbudowany i wysportowany, młody przecież jeszcze człowiek, były żołnierz amerykańskiej piechoty morskiej, były zawodowy bokser — miałby mieć stracha? Niedorzeczność...

Nastawił głośniej radio. Powietrze, czyste i świeże, jakby naraz drgnęło. Pierwsze promienie ukośnie padły na ziemię i zajrzały do wnętrza szoferki. Na niebie, aż po horyzont, rozsypały się białe, drobniutkie obłoczki. Słońce pomarańczową kulą wygramoliło się zza widnokręgu. Koła nawijały kolejne mile asfaltowej szosy. Silnik pracował z nużącą monotonią. I nagle... Co za licho? Zamilkło radio. Cisza i zupełna pustka. Stało się to tak niespodziewanie, że niemal poczuł się nieswojo. Spojrzał w boczne lusterko. Wypełniało je żółte światło. Zbliżało się z dużą prędkością do samochodu. Jakiś „pirat" drogowy pędzi na złamanie karku — pomyślał Turner i zjechał na prawą stronę drogi. Po chwili jednak żółty świetlisty snop objął ze wszystkich stron ciężarówkę.

Joemu nieznośnie załopotało serce. Mieszanina stężonego niepokoju i podniecenia wypełniła mózg i ciało. Żółtawa plama światła wpełzła do środka szoferki, poruszała się po ścianach, wspięła aż po dach, potem opadła i rozpełzła się po wszystkich zakamarkach. Lepki żar wypełnił wnętrze wozu. Poczuł, że zabrakło mu powietrza. Kleiły się oczy, rwała głowa. Coraz trudniej było zwalczać obezwładniające zmęczenie. Do tego jeszcze ten dziwny powiew ciepła, przygniatający ciężar w piersiach i dźwięk — wysoki, nieznośny. Uległ sennemu bezwładowi.

Dopiero teraz uświadomił sobie, że czuje lęk. Strach przed nieznanym niebezpieczeństwem. Niczego nie mógł dojrzeć przed samochodem. Nie widział nawet swoich rąk, zaciśniętych kurczowo na kierownicy. Tylko to mocne, jaskrawe światło z fioletowym odcieniem, sprawiające wrażenie płynu. Tak mu się przynajmniej wydawało — to światło było lepkie.

Samochód przestał go słuchać. Joe popadł w stan ni to snu, ni to odrętwienia. Nie widział już niczego, o niczym nie potrafił myśleć, wszystko zatraciło się jakby w gęstej, przytłaczającej żółtej mgle.

Nie pamięta, jak długo to trwało. Wie tylko, że w jakiejś chwili wróciła świadomość otaczającej rzeczywistości. Znowu wszystko widział i słyszał, czuł, że trawi go gorączka, męczą dreszcze. Z przerażeniem stwierdził, że znajduje się we wnętrzu obszernego pomieszczenia, jakby kopulastej hali o przymglonych i słabo widocznych ścianach. W przypływie rozpaczy próbował uruchomić silnik — bezskutecznie. I wtedy usłyszał rozedrgany szept, jakby ktoś przewijał od tyłu, w przyspieszonym tempie, taśmę magnetofonową, a potem, w refleksie dziwnej poświaty, pojawiły się zwaliste, jakby z grubej bryły wyciosane, istoty. Podobne były do ludzi wzrostem i wyglądem. Także ich twarze — nieruchome, zamknięte i mroczne — przypominały ludzkie.

Turner wyszarpnął ze skrytki pistolet, odbezpieczył i prawie nie celując, wypalił osiem razy — przez szybę — w kierunku dziwnych postaci, ubranych w obcisłe kombinezony. I wtedy poczuł, że jakaś tajemnicza siła nakazuje mu opuścić broń, wyjść z kabiny. Nie mógł się oprzeć tym telepatycznym rozkazom. Posłusznie je wykonał.

Czuł się, jakby śnił jakiś makabryczny, nierealny sen. Czyż to mogła być jawa? A przecież... Czemu oni tak dokładnie oglądali samochód. Brrr... Jakaż ich skóra była zimna w dotyku. I te dziwne błony między palcami, jakby minipłetwy. I ta szybka, śpiewna, niezrozumiała mowa... Czyż to mógł być sen?

Kto to, gdzie ja jestem? — zadawał sobie pytania, umierając z przerażenia. Czegoż oni mogą ode mnie chcieć? Po co ta istota o kamiennej twarzy wchodzi pod samochód... Ależ to są chyba... Tak, to... roboty. Przynajmniej tych trzech. Na pewno. Temu pod ciężarówką, kiedy się pochylił, spadł z głowy kaptur. Na skórze, nad oczami miał wytatuowany ciąg cyfr albo liter, jakiegoś dziwnego, hieroglificznego pisma.

Ten czwarty — to szef. Daje jakieś znaki rękami i tamci trzej skrupulatnie wykonują polecenia. Może to tylko gestykulacja? Może porozumiewają się także telepatycznie? Tak jak z nim. Tak, z pewnością. Nie można przecież kilkoma gestami przekazać tak szczegółowo złożonych poleceń...

Czuł straszny ból głowy. Jakby ktoś napompował czaszkę do granic wytrzymałości powietrzem. Humanoidalne istoty nakazały mu podejść do mglistej ściany kopuły. Przedziwne to wszystko. Jakby wyglądał zza najczystszej, niewidzialnej niemal szyby. Ależ tam rozciąga się... kosmos. Ujrzał, że ich olbrzymi pojazd znajduje się nad jakąś dziwną planetą. Z góry widać było bardzo dokładnie „nibymiasta", nakryte czymś w rodzaju gigantycznych, przezroczystych kopuł.

Chwycił się oburącz za głowę. Trudno powiedzieć, czy w natłoku wrażeń, czy też dlatego, że ten sam spokojny, przenikliwy głos wtłaczał mu do świadomości wiadomość, której tak naprawdę, nie rozumiał: — Znajdujemy się o dwa i pół roku świetlnego od gwiazdy Alfa Centaurii... Co to jest ta Alfa Centaurii? Lata świetlne? Co to wszystko ma znaczyć? Toż to jakieś majaki w malignie...

Ależ nie... Za przezroczystą kopułą wehikułu przesuwała się monotonna czerń, rozjaśniona od czasu do czasu skąpym światłem różnej wielkości punktów — jakby patrzył na gwieździste niebo w bezchmurną noc. Przez głowę przelatywały mu strzępy myśli... Gorączkowo próbował obmyśleć jakiś plan działania, ale nic mu nie przychodziło do głowy. Na plecach czuł wzrok tych przedziwnych istot. Śledzili czujnie każdy jego ruch. Tego był pewny i to w jakiś dziwny sposób nie pozwalało mu na wykonanie żadnego ruchu.

I znowu nachodziły go uparcie monotonne fale snu. Starał się walczyć z tym koszmarnym obezwładnieniem. Na próżno. Musiał zrezygnować z tej męczącej walki. Zdawało mu się, że czas mijał niemiłosiernie wolno. Wszystko dokoła zaczęło wirować w coraz szybszym tempie. Stracił świadomość.

Ocknął się na parkingu w Friedricksburgu. Jak się tutaj znalazł? Przecież... Rozejrzał się po szoferce. Rzeczy osobiste oraz pieniądze, które wiózł ze sobą, narzędzia, leżały bezładnie porozrzucane na podłodze. Wzrok ślizgał się przez chwilę po tym bałaganie... Nie, niczego nie brakowało. Zegar na ścianie domu dworcowego, przed którym zawsze parkował, przywożąc towar, wskazywał godzinę trzecią po południu. Miał mętlik w głowie, czuł się źle.

Samochód, co z samochodem? — pomyślał po chwili, kiedy nareszcie doszedł do siebie. Niezdarnie wygramolił się z kabiny. Na plac wjeżdżała jakaś ciężarówka. Może ktoś ze znajomych? Musi mu opowiedzieć o swojej historii, potrzebuje przecież pomocy.

Obcy kierowca jechał wprost na niego, nie zmniejszając prędkości. Był już tylko 20 metrów, 10, 5... Turner osłupiał z wrażenia. Tamten zatrzymał się zaledwie metr od niego. Przecież jeszcze sekunda i... Podniósł z wysiłkiem rękę, gestem człowieka, który pilnie potrzebuje pomocy. Obcy kierowca nie reagował. Wyszedł niespiesznie z wozu i skierował się w stronę marketu. Roztrzęsiony powlókł się do wozu. Przybity przeżyciami ostatnich godzin oparł głowę na kierownicy i zapadł w głęboki sen.

Poczuł silne szarpnięcie. Przez chwilę wydawało mu się, że walczy z niewidzialnym wrogiem. Kiedy nareszcie otworzył oczy, nie bardzo wiedział, gdzie jest. Trwał przez chwilę w nieruchomym odrętwieniu. Powieki miał jak z kamienia. Wreszcie doszło do niego: pochyla się nad nim, mówiąc coś bez przerwy kierowca, który przed chwilą o mało go nie rozjechał. Przed chwilą..? Spojrzał na zegar... To już siódma rano. Spał więc prawie dobę.

Czego on chce? Acha, żebym mu pozwolić obejrzeć ciężarówkę. Nie widział jeszcze czegoś takiego. Ktoś niedawno też ją oglądał... A ten tu, w ostatniej chwili przed nim zatrzymał wóz.

— Słuchaj przyjacielu, dlaczego chciałeś mnie wczoraj przejechać — wypowiedziane słowa wydały mu się jakieś dziwne, jakby nie swoje. — Przecież wczoraj, o trzeciej, mało mnie nie rozwaliłeś. Tak, tak, tutaj stałem, w tym właśnie miejscu.

Spoglądał w wytrzeszczone ze zdziwienia oczy kolegi po fachu. Tłumaczył mu przez chwilę raz jeszcze, jak to było, ale cierpliwość tamtego, z każdym jego słowem zmieniała się w złość. Popukał się wymownie palcem w czoło i mamrocząc coś o pomyleńcach — odszedł pospiesznie.

Dlatego uznał za stosowne zgłosić się do szpitala? To przez te paskudne dolegliwości: mdłości, potworny ból głowy, ogólne rozbicie. Opowiedział dokładnie to, co pamiętał. Nic więcej naprawdę nie wie. Została, gdzieś w pamięci, jedynie ta garść faktów, które z niemałym trudem starał się teraz wyłuskać i ułożyć w jakąś logiczną całość. O niczym więcej nie wie... Niczego już więcej nie potrafi sobie przypomnieć.

— I co pan o tym myśli, dyrektorze? — w głosie ordynatora, który asystował przy kolejnej próbie wydobywania z niezwykłego pacjenta, jakim był Hary Joe Turner, informacji na temat zadziwiających historii, zabrzmiała jakby nutka niepokoju. Co dalej robić z tym przypadkiem?

— Myślę, że już nic więcej nam nie powie. Może pan doktorze obudzć Joego z hipnozy. Wygląda to tak, jakby ktoś zablokował mu świadomość... Ale dlaczego do licha, wypuściliście go tak wcześnie ze szpitala — znany futurolog, dyrektor stanowy Północno-Zachodniej Wirginii, Fred Whiting, który prowadził dochodzenie w tej sprawie, nie ukrywał irytacji.

— Żądał tego — tłumaczył ordynator. — Był zdenerwowany, opowiadał, że oni jeszcze czegoś od niego chcą. Że nie znajdzie spokoju, póki nie spełni ich żądania. Nie muszę chyba panu tłumaczyć, iż niektórzy nasi lekarze patrzyli na niego z niechęcią, uważali to wszystko jedynie za pozę, szpanowanie.

— Tak, rozumiem. — Dyrektor wziął do ręki arkusz z ostatnimi „zeznaniami" Turnera. — Wszystko się powtórzyło — mówił nie odrywając wzroku od gęsto zapisanych kartek. Znów to żółte światło dopadło go w tym samym miejscu — 18 mil do Winnchesteru. Znalazł się ponownie w kopulastej hali. Teraz jednak mniej już się bojąc, sam wysiadł z samochodu i zapytał ze zdziwieniem o co chodzi: Czy niemało tych kłopotów, których narobili.

Twierdzi, że otrzymał telepatyczną odpowiedź, że istoty, które go porwały, bardzo przepraszają, ale zwracając samochód zapomniano, go wysterylizować. Na jego oczach jakaś niewidzialna siła odwróciła wóz do góry kołami. Słoje z ketchupem i musztardą, które przewoził, wymieszały się dokładnie. Samochód został zlany jakąś dziwną cieczą... Po pewnym czasie znalazł się znów na szosie, dokładnie w tym samym miejscu, skąd został porwany po raz pierwszy. Na siedemnastej mili...

— I co pan na to, dyrektorze? — powtórzył pytanie ordynator. Ale teraz w jego głosie było zaciekawienie.

— Można by — dyrektor mówił wolno, cedząc każde słowo — uznać to za wierutne brednie, wymysł chorej wyobraźni, gdyby nie pewne fakty, niepodważalne, choć trudne do wytłumaczenia. Przeprowadzono bardzo dokładną ekspertyzę samochodu, zaraz po tym, jak powiadomiliście nas o swym dziwnym pacjencie. Już podczas wstępnych badań odkryto niezwykłości, dla których trudno znaleźć jakieś logiczne wytłumaczenie. Wnętrze kabiny, wszystkie metalowe i szklane powierzchnie pokryte zostały warstewką spalenizny. Jakby w szoferce eksplodował ładunek elektryczny, np. piorun kulisty. Ale wtedy Turner by już nie żył.

Taki sam nalot znaleziono na skórze kierowcy. Antena radiowa na zewnątrz samochodu została częściowo stopiona. Licznik wskazywał, że samochód przejechał jedynie 17 mil z 80 dzielących go od miasta. W jaki sposób więc się tam znalazł? Mam też meldunek od patrolu policyjnego, który nadjechał akurat wtedy, kiedy Turner wysiadał z samochodu, aby obejrzeć go po drugiej przygodzie. Policjanci piszą o swym zdziwieniu. Zarówno olbrzymia ciężarówka, trzydziestotonowy ładunek, wnętrze szoferki oraz sam Turner ociekali jakąś szybko parującą, eterycznie pachnącą cieczą.

Wie pan dobrze, że teren, przez który jechał samochód, jest pustynny, nigdzie w pobliżu nie znaleziono żadnego zbiornika wodnego. Żeby tak zmoczyć wóz, trzeba by tej cieczy ogromną ilość. W ciężarówce nie znaleziono także żadnego pojemnika.

— Cóż to więc było, dyrektorze? — nie dawał za wygraną ordynator.

— Nie wyjaśnione zjawisko. UFO. Tak to trzeba zakwalifikować.

W holu czekają dziennikarze „Evening Star" z Winnchester i „Times Dispatch" z Richmond. Będzie pan z nimi rozmawiał?

— Oczywiście. Chodźcie i wy, panowie.

Anna KALETA

Oto gra dla dwóch osób. Każdy z graczy prowadzi swoją linię czterema klawiszami (w prawo, w lewo, w dół i w górę). Przegrywa ten gracz, który pierwszy przetnie swoją linią, linię przeciwnika lub linię pola, po którym poruszają się obaj gracze. W wypadku gdy gracze spotkają się — obaj przegrywają. Gra ta posiada 9 stopni trudności.

K. POŹNIAK

```
  10 REM GRA W DWIE LINIE
  15 INK 7: PAPER 0: BORDER 0: C
LS
  20 PRINT AT 3,6;"'GRA W DWIE L
INIE'"
  21 PRINT AT 7,4;"W GRE GRA DWO
CH GRACZY"
  22 PRINT TAB 0;"  TEN GRACZ PR
ZEGRYWA, KTORY       PIERWSZY PR
ZETNIE LINIE"
  23 PRINT AT 13,4;"GRACZ 1
   GRACZ 2"
  24 PRINT TAB 3;"lewo - 'w'
 lewo - 'o'"
  25 PRINT TAB 2;"prawo - 'e'
 prawo - 'p'"
  26 PRINT TAB 3;"gora - 'q'
 gora - 'i'"
  27 PRINT TAB 4;"dol - 'a'
 dol - 'k'"
  30 INPUT "  STOPIEN TRUDNOSCI
<1,9> ? ";t: IF t<0 OR t>9 OR t<
>INT t THEN GO TO 30
  35 CLS
  36 LET stop1=0: LET stop2=0
  40 LET gx=INT (255/t)*t
  41 LET gy=INT (175/t)*t
  50 PLOT 0,0: DRAW gx,0: DRAW 0
,gy: DRAW -gx,0: DRAW 0,-gy
  60 LET x1=t: LET y1=t
  61 LET x2=gx-t: LET y2=gy-t
  65 LET kx1=t: LET ky1=0
  66 LET kx2=-t: LET ky2=0
 100 IF IN 64510=190 THEN LET kx
1=0: LET ky1=t: GO TO 150
 110 IF IN 64510=189 THEN LET kx
1=-t: LET ky1=0: GO TO 150
 120 IF IN 64510=187 THEN LET kx
1=t: LET ky1=0: GO TO 150
 130 IF IN 65022=190 THEN LET kx
1=0: LET ky1=-t: GO TO 150
 150 IF IN 57342=187 THEN LET kx
2=0: LET ky2=t: GO TO 200
 160 IF IN 57342=189 THEN LET kx
2=-t: LET ky2=0: GO TO 200
 170 IF IN 57342=190 THEN LET kx
2=t: LET ky2=0: GO TO 200
 180 IF IN 49150=187 THEN LET kx
2=0: LET ky2=-t: GO TO 200
 200 LET nx1=x1+kx1: LET ny1=y1+
ky1
 201 LET nx2=x2+kx2: LET ny2=y2+
ky2
 210 IF POINT (nx1,ny1) THEN LET
 stop1=1
 211 IF POINT (nx2,ny2) THEN LET
 stop2=1
 212 IF nx1=nx2 AND ny1=ny2 THEN
 LET stop1=1: LET stop2=1
 220 PLOT x1,y1: DRAW kx1,ky1: L
ET x1=nx1: LET y1=ny1
 221 PLOT x2,y2: DRAW kx2,ky2: L
ET x2=nx2: LET y2=ny2
 230 IF NOT stop1 AND NOT stop2
THEN GO TO 100
 240 FLASH 1: PRINT AT 8,6;
 250 IF stop1 AND stop2 THEN PRI
NT "OBAJ GRACZE PRZEGRALI"
 260 IF stop1 AND NOT stop2 THEN
 PRINT "WYGRAL GRACZ DRUGI"
 270 IF stop2 AND NOT stop1 THEN
 PRINT "WYGRAL GRACZ PIERWSZY"
 280 FLASH 0: BRIGHT 1: INVERSE
1: PRINT AT 12,9;"WCISNIJ 'ENTER
'": BRIGHT 0: INVERSE 0
 290 IF IN 49150=190 THEN RUN
 300 GO TO 290
```

Na komputerowym rynku

Systemy na sprzedaż

W świecie informatyki, komputerów i oprogramowania obowiązują takie same prawa i zasady, jak w każdej innej, mogącej przynieść zyski, dziedzinie. Po nasyceniu rynków wewnętrznych najnowszymi osiągnięciami myśli naukowo-technicznej, firmy — zarówno najpotężniejsze, jak i te całkiem małe — zaczynają szukać możliwości rozszerzenia pola działania, co w tym wypadku oznacza zdobycie nowych rynków zbytu. Coraz częściej są to przedsięwzięcia na dużą skalę, przynoszące realne korzyści obu uczestnikom transakcji.

Do takich właśnie udanych przedsięwzięć można z pewnością zaliczyć zawartą właśnie w Moskwie po czterech miesiącach negocjacji umowę, o zastosowaniu brytyjskiego systemu expert jako standardu przy opracowywaniu tego typu programów w Związku Radzieckim. Opiewający na ponad 3 miliony funtów kontrakt przewiduje zacieśnienie współpracy między brytyjskimi i radzieckimi programistami w ciągu najbliższych trzech lat. Zgodnie z osiągniętym porozumieniem, ZSRR będzie stosował program Xi Plus firmy Expertech z Berkshire, jako podstawę przy wszelkiego typu pracach związanych z tym systemem. Umowa ma także dalsze konsekwencje — wszystkie firmy zagraniczne, zainteresowane współpracą ze Związkiem Radzieckim, będą w pewnym sensie zmuszone do przechowywania danych i przetwarzania informacji właśnie w systemie Xi Plus.

Jeszcze w tym roku, radzieccy specjaliści do oprogramowania przybędą do Wielkiej Brytanii w celu doskonalenia umiejętności odpowiedniego wykorzystania systemu Xi. Do jego zapisu używa się jeszcze obecnie dwóch języków — C i Prolog, ale eksperci Expertechu opracowują nową wersję, w całości zakodowaną przy użyciu C.

Podpisane w Moskwie porozumienie jest jednym z największych transferów brytyjskiego software do Związku Radzieckiego i największym radzieckim zakupem importowym w dziedzinie expert system. Firma Expertech zajmuje się także opracowywaniem programów dla joint-ventures, w którego działalność zaangażowany jest Algorytm — przedsiębiorstwo zajmujące się doskonaleniem radzieckiej techniki software, działające pod auspicjami Państwowego Komitetu ds. Informatyki i Komputeryzacji, utworzonego w 1986 roku.

Brytyjczycy nie ukrywają zadowolenia z zawartej umowy. Wiążą z jej podpisaniem olbrzymie nadzieje. Chodzi przecież o naprawdę dużą rzecz — rozpowszechnienie w całym, tak dużym, kraju zakupionego u nich systemu. Korzyści można upatrywać także z innych powodów. Kontrakt otwiera szeroko drzwi do radzieckiego know-how w takich dziedzinach, jak medycyna, rolnictwo czy oświata, których zasady mogą docierać do Wielkiej Brytanii i innych państw europejskich, właśnie w postaci zapisu w systemie Xi Plus. W firmie Expertech trwają właśnie badania nad zastosowaniem zautomatyzowanego tłumacza, który w znacznym stopniu uprości przechodzenie z angielskiej na rosyjską wersję systemu, z możliwością ponownego przetłumaczenia na wersję angielską.

Radziecko-brytyjskie przedsięwzięcie, oprócz wymienionych już korzyści dla obu stron, jest także potwierdzeniem faktu, że osiągnięcia najnowszej techniki mogą służyć nawiązywaniu bliskich kontaktów w dziedzinie ekonomiczno-gospodarczej.

Małgorzata NOWOTNY

ZX Spectrum

Warto przypomnieć Czytelnikom ten mało znany program z roku 1983 pokazujący w górnym prawym rogu ekranu numer aktualnie wykonywanego wiersza.

Program uruchamia się przez RANDOMIZE USR 65250, a zatrzymuje przez RANDOMIZE USR 65257.

Jan RUBINOWICZ

```
  1 DATA 1841,62,254,237,71,237
,94,201,237,86,62,63,237
  2 DATA 1546,71,201,255,245,22
9,42,69,92,36,40,69,197
  3 DATA 1347,213,37,175,24,2,2
40,254,71,79,17,232,3
  4 DATA 1743,237,62,60,48,251,
25,61,229,205,65,255,225
  5 DATA 1070,17,100,0,175,237,
62,60,48,251,25,61,14
  6 DATA 1299;1,229,205,65,255,
225,17,10,0,125,147,20
  7 DATA 1447,48,252,21,131,103
,229,14,2,122,205,65,255
  8 DATA 1885,241,14,3,205,65,2
55,209,193,225,241,201,33
  9 DATA 1051,28,64,9,135,135,1
35,235,79,33,128,61,9
  10 DATA 786,6,8,126,47,18,35,2
0,16,249,201,0,60
  15 RESTORE : CLEAR 65249: LET
adr=65250
  16 FOR n=1 TO 10: READ suma: L
ET s=0: FOR m=1 TO 12:: READ bt:
 POKE adr,bt: LET s=s+bt: LET ad
r=adr+1: NEXT m: IF s<>suma THEN
.PRINT "Blad w linii ";n: LIST n
: STOP
  17 NEXT n: RESTORE : RANDOMIZE
 USR 65250
  30 REM  Program demonstracyjny
  40 GO SUB 9800: PAUSE 50: GO T
O 300
  50 GO SUB 9800: PAUSE 50: GO T
O 5550
  70 GO SUB 9800: PAUSE 50: GO T
O 40
 300 GO SUB 9800: PAUSE 50: GO T
O 9000
5550 GO SUB 9800: PAUSE 50: GO T
O 70
9000 GO SUB 9800: PAUSE 50: GO T
O 50
9800 PAUSE 20: IF INKEY$<>"" THE
N RANDOMIZE USR 65257: PRINT "cz
y wlaczyc program ? (T/N)": PAUS
E 0: IF INKEY$="t" THEN RANDOMIZ
E USR 65250
9801 LET b=INT (RND*7): BORDER b
: BEEP .05,b: CLS : RETURN
9997 REM   Zapis na tasme
9998 SAVE "TRASAcode"CODE 65250,
118: STOP
9999 SAVE "TRASA" LINE 1:
```

INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

W poprzednim numerze rozpoczęliśmy drukować hasła zaczynające się na literę „F", niestety i tu namieszał złośliwy Chochlik — przez nieuwagę opuściliśmy pierwsze hasła tej litery. Dziś powtarzamy.

F

FABRIC - budowa, struktura,
FABRICATION - wyrób, fabrykacja,
FACE - 1. druk, nadruk, 2. powierzchnia czołowa, czoło, lico, 3. licować, okładać,
FACE BONDING - montaż czołowy,
FACE-CHANGE CHARACTER - znak zmiany rodzaju druku,
FACE DOWN - nadrukiem do dołu,
FACEDOWN FEED - podawanie (kart) prawą stroną do dołu,
FACE OF A CARD - prawa (w znaczeniu: właściwa) strona karty,
FACE OF THE DISC - górna powierzchnia dysku,
FACE UP - prawą stroną do góry,
FACEUP FEED - podawanie (kart) prawą stroną do góry,

FACILITATE - ułatwiać,
FACILITY - 1. urządzenie (pomocne, ułatwiające), aparatura, 2. sposoby, możliwości, środki (łączności), 3. połączenie, łączność, 4. udogodnienie, ułatwienie,

FACSIMILE - faksymile (dokładna reprodukcja, podobizna, kopia), także: symiliografia,
FACSIMILE TELEGRAPHY - telegrafia kopiowa, symilografia,
FACSIMILE TRANSMISSION - nadawanie obrazów nieruchomych,
FACSIMILE TRANSMITTER - aparat telekopiowy, symilograf,

FACTOR - czynnik, współczynnik,
FACTORABLE - rozkład na czynniki,
FACTOR GROUP - grupa ilorazowa,
FACTORIAL - silnia,
FACTORING - rozkładanie na czynniki,
FACTORIZATION - patrz: FACTORING,
FACTOR OF AN INTEGER - dzielnik liczby całkowitej,

FADE - zanikać, także: tracić zabarwienie, blaknąć,
FADE-DOWN - obniżanie się poziomu mocy wyjściowej,
FADE-IN - wzmacnianie sygnału, także: rozjaśnianie obrazu,
FADE-OUT - zanikanie sygnału, także: ściemnianie obrazu,
FADING - zanik sygnału, także: ściemnianie lub rozjaśnianie obrazu,

FAIL - ulegać awarii, także: załamać się, zawieść,
FAILPROOF - niezawodny, także: zabezpieczony przed uszkodzeniami,
FAIL-SAFE - odporny na uszkodzenia, bezpieczny w razie uszkodzenia,
FAIL-SOFT - częściowo odporny na uszkodzenia (pozwalający na działanie systemu w ograniczonym zakresie, w razie uszkodzenia niektórych elementów),
FAIL-SOFT SYSTEM - system częściowo odporny na uszkodzenia,
FAIL TO COMPLY WITH ... - nie spełniać (warunków), nie odpowiadać (wymaganiom),
FAIL-TO-EXHAUSTION - błędy do usunięcia,

FAILURE - uszkodzenie, defekt, niezadziałanie,
FAILURE FLOW - strumień uszkodzeń,
FAILURE-FREE - bezawaryjny, bezusterkowy,
FAILURE RATE - wskaźnik awaryjności,
FAIRLY CLEAR CARBON - dość dobra kopia (odbitka),
FALL - spadek, zmniejszenie się, obniżenie, także: spadać, opadać, obniżać się, zmniejszać się,

FALLBACK - rezerwa awaryjna, przejście na awaryjny tryb pracy, zmiana trybu pracy lub konfiguracji sprzętu w celu uniknięcia niesprawności,
FALLOUT - znikanie, zanikanie, przepadanie,

FALSE - fałszywy, także: fałsz (pojęcie używane w logice),
FALSE ACCEPTANCE - przyjęcie fałszywej hipotezy,
FALSEHOOD - przekłamanie, fałszywy wniosek, fałszywa decyzja,
FALSE REJECTION - odrzucenie hipotezy fałszywej,
FALSE RETRIEVAL - uzyskanie niepotrzebnej informacji w automatycznym systemie wyszukiwania,
FALSE STATEMENT - twierdzenie fałszywe,

FAMILY - rodzina,
FAMILY OF CHARACTERISTICS - rodzina charakterystyk,
FAMILY OF CURVES - rodzina krzywych,
FANFOLD - druk ciągły składany w "harmonię",
FANFOLD FORM - formularze ciągle złożone w postaci harmonii,

FAN-IN - obciążalność wejściowa (bramki logicznej),
FAN-OUT - obciążalność wyjściowa (bramki logicznej),
FAR CALL - wywołanie długie, wywołanie odległe,
FAR END CROSS TALK - przesłuch skrośny,
FAR PLANE - daleki plan (w grafice komputerowej określenie płaszczyzny ograniczającej przestrzeń zobrazowania na ekranie),
FAR-REACHING - dalekosiężny,

FAST - szybki, szybko,
FAST-ACCESS STORAGE - pamięć szybka, pamięć o dostępie szybkim,
FAST-ACTING - szybki, szybko działający,
FAST LINE PRINTER - szybka drukarka wierszowa,
FAST MEMORY - szybka pamięć (operacyjna),
FAT - patrz: FILE ALLOCATION TABLE,

FATAL ERROR - błąd uniemożliwiający kontynuację programu,
FATHER FILE - początkowa (wyjściowa) wersja pliku (zbioru),
FAULT - 1. błąd, 2. uszkodzenie, zakłócenie, niesprawność, 3. wada (fizyczna wada sprzętu),
FAULT CLEARING - likwidacja zakłóceń, usuwanie uszkodzeń,
FAULT DETECTOR - wskaźnik uszkodzeń,
FAULT FINDING - wykrywanie uszkodzeń, także: usuwanie zakłóceń,
FAULTLESS - bezusterkowy, bezbłędny,
FAULTLESS TAPE - taśma sprawdzona, taśma bezbłędna,
FAULT LOCATION - lokalizacja uszkodzeń,
FAULT SIMULATOR - symulator uszkodzeń, symulator zakłóceń,
FAULT TIME - czas przestoju z powodu uszkodzenia,
FAULT-TOLERANT SYSTEM - system odporny na uszkodzenia,
FAULT TRACING - wyszukiwanie uszkodzeń,
FAULT TREE GATE - element drzewa niesprawności (usterek),

FAULTY - 1. błędny, 2. uszkpdzony, zakłócony, niesprawny, 3. wadliwy,
FAULTY OPERATION - działanie wadliwe, działanie błędne,
FAX - patrz: FACSIMILE,
FAY - przylegać, dokładnie pasować,
FC - patrz: FONT-CHANGE CHARACTER,
FCB - patrz: FILE CONTROL BLOCK,
FCFS - patrz: FIRST-COME FIRST-SERVED,
FCP - patrz: FILE CONTROL PROCESSOR,

FCS - patrz: FRAME CHECK SEQUENCE,
FD - patrz: FILE DESCRIPTION,
FDB - patrz: FILE DESCRIPTION BLOCK,
FDM - patrz: FREQUENCY DIVISION MULTIPLEXING,
FEASIBLE - możliwy (do wykonania), wykonalny, realny, dopuszczalny,
FEASIBLE SOLUTION - rozwiązanie dopuszczalne,
FEATURE - cecha, właściwość, także: możliwość, możność,
FEED - 1. magazynek podawczy kart, 2. zasilanie maszyny (np. kartami, drukami, papierem), 3. dostarczać, wprowadzać (np.dane do komputera),
FEEDBACK - sprzężenie zwrotne,
FEEDBACK CONTROL - sterowanie ze sprzężeniem zwrotnym, sterowanie w układzie zamkniętym,
FEED BACK ELEMENT - element sprzęgający,
FEEDBACK FACTOR - współczynnik sprzężenia zwrotnego,
FEEDBACK FILTER - filtr sprzężenia zwrotnego,
FEEDER - 1. zasilacz, 2. podajnik, 3. dozownik, 4. przewód zasilający,
FEEDFORWARD - uprzedzanie, poprzedzanie, zapobieganie, wyprzedzenie,
FEED HOLE - dziurka prowadząca, dziurka przesuwu,
FEED HOPPER - kaseta podająca,
FEEDING - zasilanie, podawanie, wprowadzanie,
FEEDING INFORMATION - dostarczanie danych, wprowadzanie danych,
FEED INTO A COMPUTER - wprowadzić do (pamięci) komputera,
FEED PITCH - odstęp linii,
FEED RACK - gąsienica przesuwu, traktor przesuwu,

FEED SPOOL - szpula, z której taśma się rozwija,
FEEDTHROUGH - bezpośrednie połączenie, bezpośrednie łącze, połączenie skrośne, łącze przelotowe,
FEED TRACK - ścieżka prowadząca,
FEELOW - członek towarzystwa naukowego,
FEP - patrz: FRONT-END PROCESSOR,
FERRITE - ferryt, także: ferrytowy,
FERRITE CORE - rdzeń ferrytowy,
FERRITE-CORE STORE - pamięć ferrytowa,
FERRITE SWITCHING CORE - rdzeń ferrytowy przełączający,
FERROELECTRIC - ferroelektryczny,
FERROGRAPHY - ferrografia,
FERROMAGNETIC - ferromagnetyczny,
FERROUS COATED TAPE - taśma z warstwą tlenkową,
FET - patrz: FIELD EFFECT TRANSISTOR,
FETCH - ładować do pamięci, pobierać, także: cykl maszynowy odczytu kodu rozkazu,
FETCH CYCLE - cykl maszynowy odczytu kodu rozkazu,
FETCHING - patrz: FETCH,
FETCH TIME - czas ładowania (wczytania) do pamięci,
F/F - patrz: FLIP-FLOP,
FF - patrz: FIRST FIT ALGORITHM,
FF - patrz: FORM FEED,
FFD - patrz: FIRST FIT DECREASING ALGORITHM,
FIB - patrz: FILE IDENTIFICATION BLOCK,
FIBER - włókno szklane, także: swiatłowód,
FIBER-GLASS FILAMENT - włókno szklane,
FIBONACCI NUMBERS - liczby Fibonacciego,
FIBRE - patrz: FIBER,
FIBRE OPTICS - technika swiatłowodowa,
FICHE - mikrofilm, mikrofotokopia,
FIDELITY - dokładność, wierność (odtwarzania),
FIELD - pole, obszar w zapisie (rekordzie),
FIELD EFFECT TRANSISTOR - tranzystor polowy,

FIELD EXTRACTOR - czynność wyboru pola zapisu (rekordu),
FIELD MARK - cecha pola, znacznik pola,
FIELD OF KNOWLEDGE - dziedzina wiedzy,
FIELD-PROGRAMMABLE - programowany w warunkach eksploatacji,
FIELD-PROGRAMMABLE LOGIC ARRAY - tablica logiczna programowalna przez użytkownika,
FIELD-PROGRAMMABLE ROM - pamięć stała reprogramowalna polem,
FIELD SELECTION - wybór (selekcja) pola zapisu (rekordu),
FIFO - patrz: FIRST-IN FIRST-OUT,
FIFO LIST - kolejka typu FIFO,
FIFO PROCESSING - przetwarzanie (obsługa) według kolejności typu FIFO,
FIFO QUEUE - patrz: FIFO LIST,
FIFTH-GENERATION COMPUTER - komputer piątej generacji,
FIGURATIVE - obrazowy,
FIGURATIVE CONSTANT - stała symboliczna,
FIGURE - figura, także: cyfra,
FIGURE OUT (UP) - obliczać, wyliczać,
FIGURES KEY - klawisz cyfrowy,
FIGURES SHIFT - przełączanie na cyfry,
FIGURE STAMP - znacznik cyfrowy,
FIGURING - pismo cyfr, także: cyfry skali,
FILE - plik, zbiór,
FILE ADDRESSING - metoda dostępu,
FILE ALLOCATION - rozmieszczenie zapisów (rekordów) pliku (zbioru),
FILE ALLOCATION ROUTINES - programy rezerwacji pliku (zbioru),
FILE ALLOCATION TABLE - tablica rezerwacji plików (zbiorów),
FILE AREA - obszar pliku (zbioru),
FILE ATTRIBUTE - cecha pliku (zbioru),
FILE BACKUP - składowanie zbioru,
FILE CONTROL BLOCK - blok kontrolny pliku (zbioru),
FILE CONTROL PROCESSOR - program powtarzalny do otwierania i zamykania zbiorów danych,
FILE CREATION - zakładanie pliku (zbioru),
FILE DEFINITION AREA - opis pliku (zbioru),
FILE DESCRIPTION - opis pliku (zbioru),
FILE DESCRIPTION BLOCK - blok opisu pliku (zbioru),
FILE DEVICE - urządzenie plikowe, urządzenie o strukturze pliku,
FILE DIRECTORY - katalog plików (zbiorów), kartoteka plików (zbiorów),
FILE EVENT - zwrócenie się do pliku (zbioru),
FILE EXCHANGE UTILITY - program kopiowania plików (zbiorów) z możliwością zmiany struktury,
FILE FORMAT - struktura pliku (zbioru),
FILE HANDLER - manipulator zbiorów,
FILE HEADER LABEL - etykieta początku zbioru danych,
FILEID - identyfikator pliku (zbioru),
FILE IDENTIFICATION BLOCK - blok określający (identyfikujący) plik (zbiór),
FILE LABEL - etykieta (znacznik) pliku (zbioru),
FILE LAYOUT - rozplanowanie układu pliku (zbioru),
FILE LOCKING - zajęcie pliku (zbioru), zablokowanie pliku (zbioru),
FILE MAINTENANCE - aktualizacja zbioru danych,
FILE MANAGER - określenie części systemu operacyjnego odpowiedzialnego za wykonywanie operacji na plikach (zbiorach),

FILEMARK - znak ostatniego zapisu (rekordu) w pliku (zbiorze), także: znacznik sekcji,
FILE MERGING - scalanie pliku (zbioru),
FILE NAME - nazwa zbioru (pliku) danych,
FILE ORGANIZATION - organizacja pliku (zbioru),
FILE ORGANIZATION CODE - symbol organizacji pliku (zbioru),
FILE-ORIENTED - przeznaczony do pracy ze zbiorami,
FILE OVERFLOW AREA - obszar przepełnienia pliku (zbioru),
FILE-PROCESSING - przetwarzanie pliku (zbioru),
FILE PROTECTION - zabezpieczenie pliku (zbioru) danych,
FILE-PROTECTION RING - pierścień zezwalający na zapis, pierścień zapisu,
FILE QUALIFIER - kwalifikator pliku (zbioru), klucz pliku (zbioru),
FILER - patrz: FILE MANAGER,
FILE SECURITY - zabezpieczenie programowe pliku (zbioru), także: ograniczenie dostępu do pliku (zbioru),
FILE SEPARATOR - patrz: FILE SEPARATOR CHARACTER,
FILE SEPARATOR CHARACTER - znak rozdzielający pliki (zbiory), znak oddzielający pliki (zbiory),
FILE SEQUENCE - porządek kartoteki,
FILE SHARING - rozdział zasobów pliku (zbioru),
FILE SORTING - sortowanie pliku (zbioru),
FILE SPECIFIER - wyróżnik pliku (nazwa pliku i opcjonalnie oznacznik urządzenia),
FILESTORE - pamięć zbiorów systemu (np. w SO GEORGE-3), także: magazyn, skład plików (zbiorów),
FILE-STRUCTURED - o organizacji plikowej, o strukturze plikowej,
FILE SYSTEM - patrz: FILE MANAGER,
FILE TRAILER IDENTIFIER - etykieta końcowa,
FILETYPE - identyfikator typu pliku (zbioru),
FILE UNIT - zewnętrzna pamięć masowa,
FILE UPDATING - aktualizacja pliku (zbioru),
FILING - kartotekowanie, segregowanie dokumentu, ewidencjonowanie, także: zapis do pliku (zbioru),
FILL - dowolny symbol wypełniający lub uzupełniający,
FILL-AREA ATTRIBUTE - atrybut zamazywania, atrybut maskowania (w grafice sposób wydzielenia obszaru obrazu),
FILL CHARACTER - znak wypełniający, znak pusty,
FILLER - znak wypełniający, znak pusty, wypełniacz,
FILL THE FORM - wypełnić formularz,
FILL UP - dopełnić, uzupełnić,
FILO - patrz: FIRST-IN LAST-OUT ,
FILTER - filtr, także: filtrować,
FINAL - końcowy,
FINAL CONTROL ELEMENT - człon wykonawczy,
FINAL SEGMENT OF SET - reszta zbioru,
FIND - znaleźć, odkryć, wykryć,
FINDABLE - wykrywalny,
FIND APPLICATION - znajdować zastosowanie,
FINDER - szukacz, także: wyznacznik,
FINDING - wnioskowanie (w systemach ekspertowych),
FINDINGS - 1. wyniki poszukiwań, wyniki badań, 3. dane uzyskane, 4. wnioski,
FIND OUT - dowiedzieć się, wykryć,
FIND USE - znajdować zastosowanie,
FINE INDEX - indeks szczegółowy, indeks wtórny,
FINISH - kończyć, skończyć,
FINITE - określony, skończony,

FINITE-DIFFERENCE METHOD - metoda różnic skończonych,
FINITE-ELEMENT METHOD - metoda projekcyjna, także: metoda różnicowo-projekcyjna,
FINITELY - skończenie,
FINITE NUMBER - liczba skończona,
FINITE SET - zbiór skończony,
FINITE-STATE AUTOMATON - automat skończony, automat ze skończoną ilością stanów,
FINITE-STATE GRAMMAR - gramatyka o skończonej liczbie stanów, także: gramatyka automatów,
FINITE-STATE MACHINE - patrz: FINITE-STATE AUTOMATON,
FINUFO - patrz: FIRST-IN-NOT-USED-FIRST-OUT,
FIRM - firma, także: zwarty, zwięzły, mocny,
FIRMWARE - oprogramowanie układowe, oprogramowanie sprzętowe, oprogramowanie mikrokomputerowe, mikrooprogramowanie,
FIRST BUCKET IN FILE AREA MARKER - wskaźnik pierwszego bloku w obszarze pliku (zbioru),

LIGA MYŚLĄCYCH

ZADANIE 1

Pewien obywatel podejmuje z końcem każdego roku ze swego wkładu złożonego w kasie oszczędności na 3%, 90.— zł z każdego tysiąca wkładu. Po jakim czasie podejmie on cały wkład?

ZADANIE 2

Piłka gumowa odbija się na $^2/_3$ wysokości, z której była puszczona. Piłkę puszczono z wysokości 8,1 metra. Po ilu odbiciach piłka wzniesie się na wysokość tylko 1,6 metra?

ZADANIE 3

Przed jednym z bardziej okazałych budynków miasta znajdują się schody. Gdyby liczbę stopni schodów o wysokości 3,5 metra zwiększyć o 10, to wysokość każdego stopnia zmniejszyłaby się o 4 cm. Ile stopni mają schody?

ZADANIE 4

Znaleźć liczbę dwucyfrową, jeżeli iloraz z podzielenia tej liczby przez iloczyn jej cyfr wynosi $2^2/_3$, a różnica między szukaną liczbą i liczbą o tych samych cyfrach, lecz napisanych w porządku odwrotnym, równa się 18.

ZADANIE 5

Pociąg A o prędkości V km/h wyjeżdża ze stacji po odejściu pociągu B, jadącego z prędkością V_1 km/h. Godzina odejścia pociągu A była tak dobrana, aby oba pociągi przyjechały jednocześnie do stacji końcowej. Pociąg B przejechawszy $^2/_3$ drogi zmuszony był zmniejszyć prędkość o połowę i wskutek tego pociągi spotkały się a kilometrów przed stacją końcową. Wyznaczyć odległość między tymi stacjami.

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

Odwracanka *ZX Spectrum*

W „IKS-ie" (2/86) opublikowano program „odwracanka". Jest to gra na mikrokomputer Atari. Na podstawie podanego opisu opracowałem uproszczoną wersję programu na ZX Spectrum (wydruk 1). W zeszycie „IKS" — programy 2 zamieszczona została inna wersja tej gry („Rubik") na ZX Spectrum. Ta wersja jest wygodniejsza ze względu na inny sposób wyboru odwracanego pola. Prezentowany program w minimalnym stopniu wykorzystuje komputer. Wszelkie decyzje podjął programista. W czasie pracowitego wpisywania linii programu. Można ten program skrócić w sposób dość znaczny. Wystarczy pola planszy oznaczyć kolejnymi literami alfabetu (nie pomijając V). Współrzędne i kształt „Krzyża" może wyznaczyć komputer (wydruk 2).

Wiesław RYCHLICKI

```
  0>REM W.RYCHLICKI 1987
  5 BORDER 0: PAPER 0: INK 7
 10 FLASH 0: BRIGHT 0: OVER 1
 15 LET a$="                "
 20 CLS : GO SUB 1000
 25 POKE 23658,8
 30 LET w$=INKEY$
 40 IF w$="0" THEN GO TO 20
 50 IF w$<"A" OR w$>"Y" THEN
    GO TO 30
 60 LET x=CODE w$-65
 70 LET y=4*INT (x/5)
 80 LET x=4*x-5*y
 90 GO SUB 2000: BEEP .2,15
100 GO TO 30
110
1000 FOR i=0 TO 5
1010 PLOT i*32,175: DRAW 0,-160:
     PLOT 0,i*32+15: DRAW 160,0
1020 NEXT i
1030 PRINT #0;AT 0,0;
     "0  START GRY OD POCZATKU"
1040 FOR i=0 TO 4: FOR j=0 TO 4
1050 PRINT AT 4*i+1,4*j+1;
     CHR$ (65+5*i+j)
1060 NEXT j: NEXT i
1070 RETURN
1080
2000 LET a=y: LET b=x-(x>0)*4
2010 LET c=8+(x>0 AND x<16)*4
2020 GO SUB 2100
2030 LET b=x: LET c=4
2040 IF y>0 THEN LET a=y-4:
     GO SUB 2100
2050 IF y<16 THEN LET a=y+4:
     GO SUB 2100
2060 RETURN
2070
2100 FOR i=0 TO 3
2110 PRINT AT a+i,b;a$( TO c)
2120 NEXT i: RETURN
```

```
    0>REM WIESLAW RYCHLICKI 1986
   10 CLS : PRINT AT 0,5;" O D W
R A C A N I E "
   20 PRINT : PRINT "NA 25 POLOWE
J PLANSZY ZNAJDUJA  SIE KWADRATY
 Z JEDNEJ STRONY    BIALE A Z DR
UGIEJ CZARNE"
   30 PRINT "PODAJAC WSPOLRZEDNE
KWADRATU     ODWRACAMY GO NA DRUG
A STRONE     WRAZ Z ODWRACANYM KW
ADRATEM      ODWRACAJA SIE KWADRA
TY SASIEDNIE"
   40 PRINT "NALEZY ODWROCIC WSZY
STKIE        KWADRATY"
   50 LET A=12: LET B=12
   60 LET PKT=0: FOR I=1 TO 5: PR
INT AT A+I,B;I;"      ";AT A,B+I;
I: NEXT I
   70 INPUT "ODWRACANY KWADRAT";"
WIERSZ "; LINE A$,"KOLUMNA "; LI
NE B$
   80 IF LEN A$<>1 OR LEN B$<>1 T
HEN GO TO 70
   90 IF A$<"1" OR A$>"5" OR B$<"
1" OR B$>"5" THEN GO TO 70
  100 LET X=VAL A$: LET Y=VAL B$
  110 PRINT AT A+X,B+Y; OVER 1; B
RIGHT 1;"■"
  120 IF X>1 THEN PRINT AT A+X-1,
B+Y; OVER 1; BRIGHT 1;"■"
  130 IF X<5 THEN PRINT AT A+X+1,
B+Y; OVER 1; BRIGHT 1;"■"
  140 IF Y<5 THEN PRINT AT A+X,B+
Y+1; OVER 1; BRIGHT 1;"■"
  150 IF Y>1 THEN PRINT AT A+X,B+
Y-1; OVER 1; BRIGHT 1;"■"
  160 LET PKT=PKT+1: PRINT AT 19,
0;"LICZBA RUCHOW : ";PKT
  170 GO TO 70
```

LOGOGRYF

1						
2	I	N	E	R	C	J	A
3						
4						
5						
6	L	O	T	E	R	I	A

Litery, które znajdą się w oznaczonych polach, czytane rzędami poziomymi, utworzą rozwiązanie zadania.
1) głos podobny do gwizdu, 2) bezwładność, 3) domena transportu, 4) roślina z rodziny jaskrowatych o żółtych kwiatach, 5) pochlebca, 6) z losami.

„MIREK"

1	S	A	W	A	N	N	A
2	N	A	D	A	W	C	A
3						
4						
5	K	A	S	K	H	D	A

Litery, które znajdują się w oznaczonych polach, czytane rzędami poziomymi, utworzą rozwiązanie zadania.
1) zbiorowisko roślinne typowe dla Afryki, 2) o kimś, kto wysyła coś za pośrednictwem poczty, 3) gruczoł wydzielania wewnętrznego, 4) dodatkowa porcja płynu, 5) niewielki wodospad.

„MIREK"

Magiczna krzyżówka dwuliterowa

ZNACZENIE WYRAZÓW: 1) artystka malarka, wyspecjalizowana w technice akwarelowej, 2) puszysta tkanina wełniana lub bawełniana, 3) pozostałość pochodząca z dawnych czasów ● naturalna lub zbożowa, 4) może być holownicza ● drąg używany przez linoskoczków, 5) gładziarka, 6) dom zajezdny na Wschodzie, miejsce postoju karawany.

„MIREK"

	1	2	3	4	5	6
1						
2					■	
3				■	KA	WA
4	LI	NA	■			
5		■				
6						

Pocztowa Giełda

Rozwiązanie krzyżówki z nr. 3/88 „IKS-a"
Nagrody książkowe otrzymują: Sławomir Bukowski — Pruszków, Barbara Kozina — Świdnica, Igor Zgoła — Gdynia, Wanda Roguzińska — Pisz, Jacek Pawlik — Pyskowice, Jan Mamajko — Kraków, Andrzej Kmiecicki — Gdynia, Irena Pietruszewicz — Poznań, Piotr Kaczmarek — Leszno, Leszek Knapczyk — Nowy Targ.

Rozwiązanie krzyżówki z nr. 4/88 „IKS-a"
Nagrody książkowe otrzymują: Sewer Nyckowski — Gostynin, Dorota Baranowska — Oleśnica, Dawid Płusa — Jelenia Góra, Joanna Hryniewicka — Białystok, Andrzej Herla — Głogówek, Andrzej Wawrzyński — Gdynia, Piotr Sinielewicz — Lublin, Piotr Rynkowski — Warszawa, Mariola Klimek — Gdańsk-Oliwa, Anna Cembrzyńska — Gorlice.

Rozwiązania zadań (przynajmniej jednego) prosimy nadsyłać pod adresem redakcji do końca września (naklejając kupon „IKS-a"). Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.

„IKS" — dodatek „Żołnierza Wolności". Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarra, Romuald Głąb, Włodzimierz Gogołek, Janusz Janiec, Henryk Krasuski, Ireneusz Miernik, Ludwik Piela, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe", Warszawa ul. Grzybowska 77. Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 1658 Nr ind. 361682.
U-10