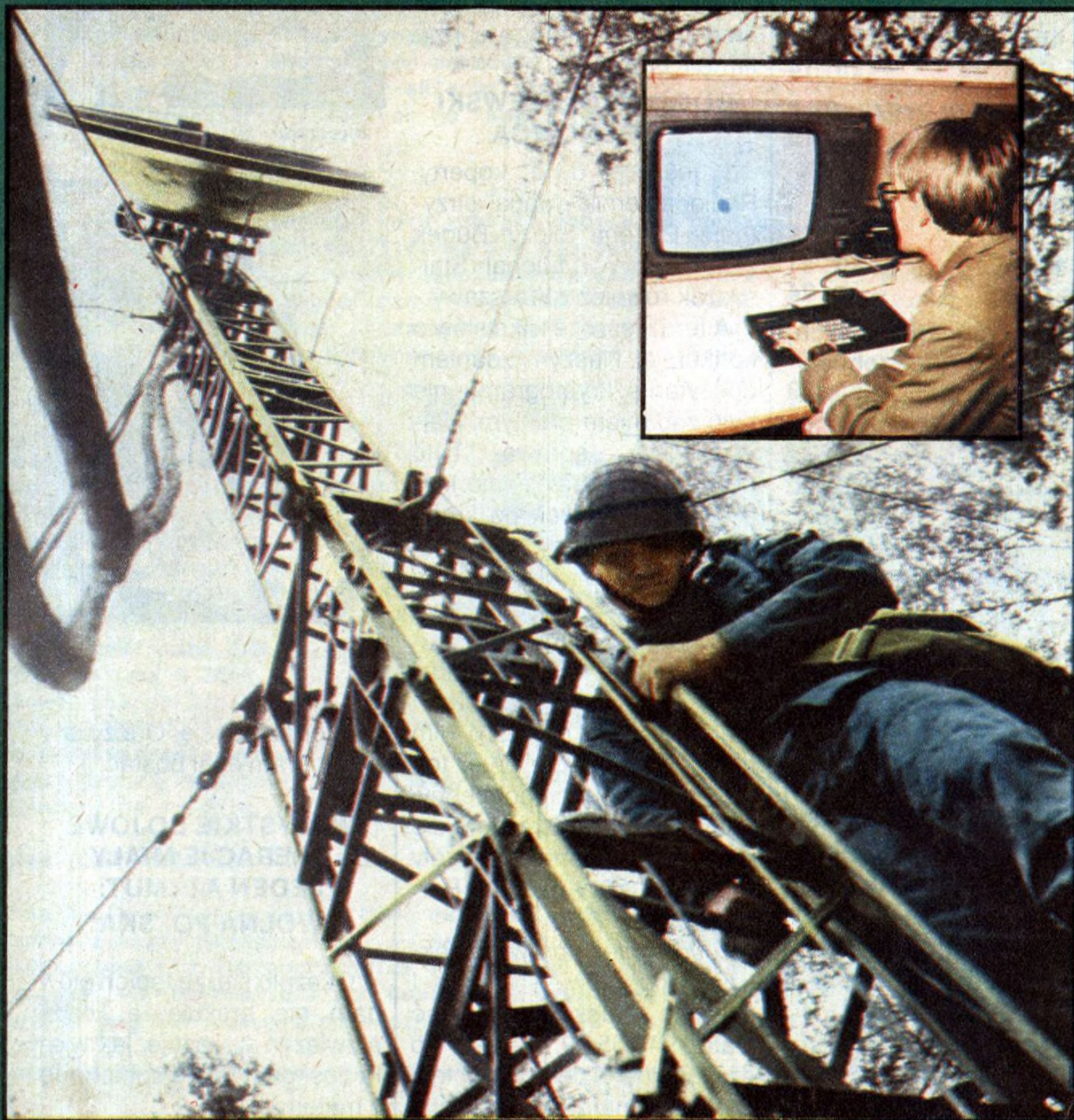


5
1987
(13)

INFORMATYKA
KOMPUTERY
SYSTEMY

Dodatek „Żołnierza Wolności” ISSN 0860-2794 Cena 50 zł



W KLUBIE ŁĄCZNOŚCIOWCÓW

str. 15—16

W numerze:

Czy komputer może uczyć?
— str. 4

W szponach Atari:

Kreślenie okręgów — str. 5

— **Supertekst** — str. 7

— **Reversi** — str. 9

Sharp:

Kto winien? — str. 10

Mikrokomputery IBM (2)
— str. 12

Infosystem'87 — str. 13

CPC 464 — Kreska (3)
— str. 14

W klubie, który będzie
— str. 15

Spectrum:

Sztuki i sztuczki (7)
— str. 18

Pascal (5) — str. 21

Basic (7) — str. 24

**Myślenie po godzinach,
czyli z czego żyją rzemieślnicy**
— str. 29

Liga Myślących — str. 30

UWAGA! PRZYPOMINAMY

Już wkrótce do kiosków Ruchu dotrze drugi numer specjalnego numeru „IKS-a” poświęconego wyłącznie programom. Wierymy, że wśród kilkunastu propozycji nasi Czytelnicy znajdą coś ciekawego dla siebie. Zatem uwaga użytkownicy Atari, Commodorów, Spectrum i Amstradów, pytajcie o programy „IKS-a” w kioskach.

Szyfr już złamany

Znamy już szczęśliwego laureata naszego konkursu. Spośród nadesłanych do redakcji kilkunastu tysięcy prawidłowych rozwiązań pięcioletnia Monika wylosowała zwyciężcę.

Uwaga. Niebieska koperta: Posiadaczem komputera **ZX Spectrum** +, magnetofonu, joysticka i zestawu programów ufundowanego przez firmę ICL został

BOGDAN OLSZEWSKI z SOSNOWCA

I jeszcze dwie koperty. Radioodbiorniki „Maria” otrzymują: Bożena Silezin-Budek z Warszawy i Michał Stajszczak również z Warszawy.

A teraz jeszcze kilka słów o konkursie. Naszym zdaniem odczytanie kryptogramu nie było zadaniem prostym. Najważniejszą sprawą było oczywiście odnalezienie klucza, a potem krok po kroku należało odczytać niezrozumiałą ciąg znaków, w którym znalazły się dodatkowe utrudnienia.

Zatem najważniejszy był klucz. W kolejnych numerach staraliśmy się podać o nim bliższe informacje. Ale ten trud okazał się niepotrzebny. Już wkrótce po ukazaniu się pierwszego numeru zaczęły napływać rozwiązania. Prawidłowe?

Klucz nie sprawił większych kłopotów — pisano. I słusznie, klucz miał postać Data:1945.05.09. — jest to szczególny dzień w historii naszego narodu i świata. Kiedy znamy klucz, reszta



Tak poznaliśmy zwyciężcę...



A to już wszyscy szczęśliwcy.

jest formalnością. Odczytany kryptogram miał postać:

**WSZYSTKIE BOJOWE
OPERACJE MIAŁY
JEDEN AZYMUT:
*WOLNA POLSKA***

Okazało się, że tęgich głów nam nie brakuje, a liczba rozwiązań wskazuje, jak wielu doskonale daje sobie radę z matematyką.

Zwycięzcom gratulujemy.

***** NOWINKI *****

***** STC POBIŁA REKORD ŚWIATA *****

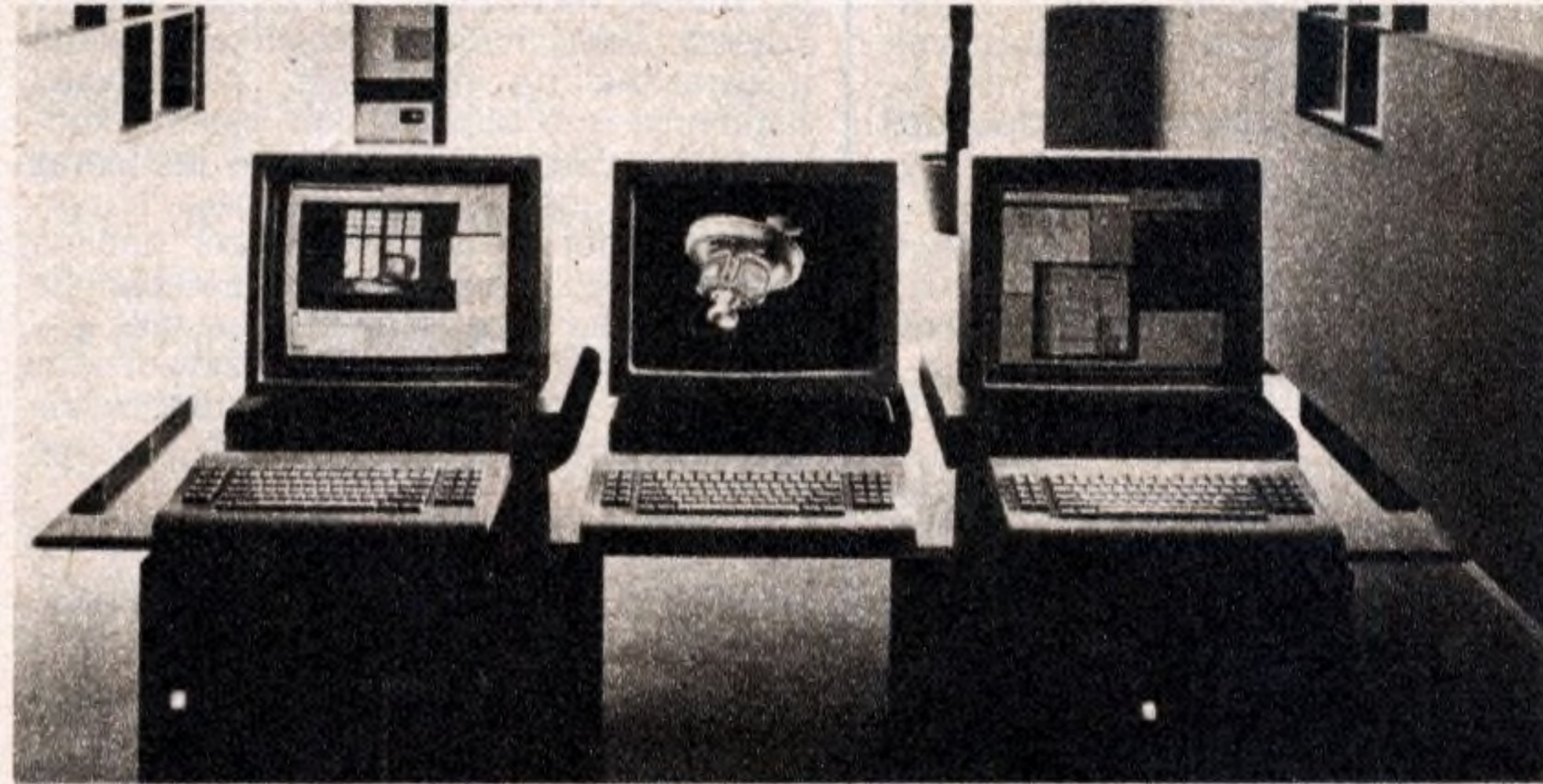
STC - Standard Telephones and Cables wygrała kontrakt wartości 3 milionów funtów na położenie kabla światłowodowego o długości 56 mil łączącego wyspę Man (Wielka Brytania) z lądem stałym. Kabel ten pozwoli na wprowadzenie cyfrowej telekomunikacji na wyspie. Po raz pierwszy kabel światłowodowy tej długości nie będzie potrzebował po drodze wzmacniaczy co jest swoistym rekordem świata.

***** TANI SUPERKOMPUTER *****

Ceny superkomputerów rozpoczynają się od kwoty 7.5 miliona dolarów, gdy tymczasem Uniwersytet w Southampton obiecuje już niedługo taki komputer za.....750 tysięcy dolarów. Superkomputer składać się będzie z 320-stu transputerów czyli kompletnych komputerów zawartych na "kości" wielkości ...znaczka pocztowego.

***** ICL I SUN *****

Jeszcze pod koniec ubiegłego roku zostało podpisane porozumienie pomiędzy ICL - International Computers Ltd. i firmą SUN Microsystems Ltd. specjalizującą się w systemach graficznych. W ramach porozumienia nastąpi sprzedaż systemów komputerowych ICL i jako stacji graficznych systemów SUN - wymagających maszyn o dużej mocy przetwarzania. Drugą płaszczyzną porozumienia jest wzajemna współpraca w zakresie projektowania i konstruowania urządzeń graficznych nowej generacji.



Na zdjęciu pozbawiony migotania 19 calowy monitor kolorowy o rozdzielczości 1152 * 900 punktów. Może wyświetlać jednocześnie 256 kolorów lub odcieni szarości z palety16.7! milionów. Stacje graficzne SUN wyposażone są w 32 - bitowy mikroprocesor MOTOROLA 68020, a ich

moc przetwarzania wynosi od 1.5 do 4 MIPS (milionów operacji wykonywanych w ciągu jednej sekundy). Stacje graficzne firmy SUN są obecnie uważane za najlepsze na świecie.

***** DWA NOWE PC APPLA *****

Firma Apple wylansowała w USA dwa nowe modele komputera osobistego McIntosh. Nazwane zostały MacIntosh II i MacIntosh SE. Posiadają znacznie większą moc obliczeniową niż ich poprzednicy i mogą być rozbudowywane. Oba mogą wiernie naśladować PC IBM-owskie i ich kopie-przy czym MacIntosh II proponuje bardziej zaawansowaną technologię od oryginalnego IBM PC przy znacznie niższej cenie.

(sch)

CZY KOMPUTER MOŻE UCZYĆ?

Kilka dni temu miałem okazję porozmawiać z Robertem — absolwentem — „skomputeryzowanej uczelni”. Zważywszy na bardzo ciekawą (i pouczającą) treść tej rozmowy przedstawiam ją w niewielkim skrócie.

Komputery w uczelni Roberta wykorzystywane są niemal we wszystkich dziedzinach życia szkoły: zarządzanie, administracja, prace naukowo-badawcze, wszelkiego typu banki informacji — a w szczególności informacja biblioteczna i naukowa. Jednak najciekawszą, moim zdaniem, sferą zastosowań jest nauczanie wspomagane przez komputer, tym bardziej że dotyczy ono przedmiotów humanistycznych.

Podczas nauki wspomaganą przez maszynę, uczący się ma do dyspozycji indywidualne stanowisko — miejsce odizolowane od pozostałej części sali o „wyciszających” ścianach (w kolorze pastelowym). Miejszczy się tam stolik wyposażony jest w mikrokomputer, który stanowi element sieci komputerowej uczelni. Dzięki połączeniom (sieci) z kilkoma liniami telefonicznymi — każde stanowisko ma łączność z wieloma innymi uczelniami, bibliotekami, a także z domem.

Rozmiar mikrokomputera, często nazywanego nanokomputerem, jest odwrotnie proporcjonalny do jego możliwości. Przede wszystkim, nanokomputer wyposażono w kolorowy monitor „z dostępem” do każdego z 1024 x 1024 pikseli. Oznacza to, iż każdy umowny punkt (piksel) ekranu może mieć inny kolor — zmieniany zależnie od potrzeb. W efekcie pozwala to uzyskać dobrej jakości postać tekstów oraz bardziej złożonych rycin na ekranie. Istotny udział w komforcie odbioru (czytelności) tej formy informacji ma fakt, iż obraz na ekranie „odświeżany”, jest 100 razy w czasie jednej sekundy — dwukrotnie częściej niż ma to miejsce w standardach TV (50—60 Hz). Rezultatem tego jest „papierowy” obraz, którego oglądanie, nawet przez kilka godzin, nie męczy i jest zdecydowanie mniej szkodliwe dla zdrowia oczu.

Mówiąc o ekranie, nie sposób nie wspomnieć o tym, że „wskazanie komputerowi” dowolnego punktu ekranu, polega na dotknięciu tego miejsca, na przykład po prostu palcem. Czynność tę można także wykonać „tradycyjną myszką”, piórem świetlnym lub tak zwanym digitizerem. Urządzenia te umożliwiają przemieszczanie na ekranie kursora (migającą plamkę).

Zatem podczas nauki geografii, a w zasadzie kontroli zdobywania wiadomości z tego przedmiotu — komputer wyświetla na ekranie mapę określonego kraju i żąda wskazania na niej miejsc usytuowania niektórych bogactw naturalnych. Jednocześnie na lewej części ekranu ukazują się małe okienka z obrazkami symbolizującymi węgiel, siarkę, miedź itp. Są to tak zwane ikony. Na przykład ikona siarki to po prostu rycina zapalek z symbolem Si. Uczący się „wybiera” dowolną ikonę przez jej dotknięcie (lub wskazanie). Potwierdzeniem, że komputer „zrozumiał” nasz wybór jest ukazanie się tej ikony w lewym górnym rogu ekranu. Wystarczy teraz wskazywać odpowiednie miejsca na mapie. Gdy znajomość geografii Polski jest przedmiotem kontroli — wskazanie wybrzeża jako miejsca bogatych pokładów węgla napotyka na zrozumiałe „zdziwienie” komputera, natomiast w poprawnie wskazanych punktach pojawiają się miniaturki ikon. Gdy węgiel został „rozdy-

ponowany”, uczący się dotyka ikonę siarki, następnie miedzi itp.

W trakcie tej pracy można oczywiście popełnić błąd, który został zauważony po pewnej chwili — jego poprawie i innym typowym czynnościom służy zestaw stałych ikon (niezależnie od przedmiotu) wyświetlanych w dolnej części ekranu. Symbolizują one takie czynności jak na przykład: kasowanie (mazanie, ścieranie) — najczęściej w postaci obrazka gumki, prośba o pomoc (ikona uniesionej do góry dłoni), polecenie zapisywania wyróżnionych informacji (związanych z treściami nauczania) do własnego (prywatnego) zbioru — dostępnego także w domu, powiększanie fragmentu ekranu (obrazek lupy, szkła powiększającego) itp.

Wydaje się, że bardzo atrakcyjną zdolnością komputera podczas zajęć z geografii jest powiększanie fragmentów obrazu aktualnie wyświetlanego na ekranie monitora. Na przykład oglądając mapę Polski prosimy o powiększenie jej centralnej części — w odpowiedzi komputer wyświetla fizyczną mapę województwa warszawskiego i części sąsiednich województw. Można żądać kolejnych powiększeń — ostatnim krokiem jest np.: plan miasta Łowicza.

Regulacji jasności i kontrastu obrazu zazwyczaj dokonuje sam komputer, bowiem uprzedza chwilę, w której użytkownik sam zdecydowałby się na zmianę tych parametrów ekranu. Wykonywane jest to w bardzo przemyślny sposób — komputer mierzy dokładnie czas naszych odpowiedzi na obrazy (treści) ukazywane na ekranie. Korzystając ze znanej (maszynie) zależności między czasem reakcji na informacje (komunikaty) wyświetlane na monitorze, a skutkami wspomnianych regulacji (im większy kontrast tym szybciej rozpoznawany jest obraz — niestety kosztem zdrowia oczu i pogorszenia samopoczucia), komputer dobiera odpowiednie parametry wyświetlanego obrazu. W praktyce ustawiona jest taka jasność i kontrast ekranu by pogodzić potrzeby (szybkość oraz poprawność) i możliwości (nasze zdrowie) odczytywanych z niego treści.

Mając na uwadze indywidualne wymagania uczących się, w podobny sposób jak jasność i kontrast, komputer dobiera wielkość i proporcje stosowanych znaków i symboli. Uwzględnia to między innymi stan zdrowia naszych oczu. Na początku współpracy z użytkownikiem maszyna stopniowo zwiększa wielkość, aż do chwili, gdy dalsze powiększenie nie daje widocznego skutku (szybkość i poprawność odczytywania informacji z ekranu nie zmienia się) lub po prostu nie jest fizycznie to już możliwe. Analogicznie przebiega proces odwrotny — kontrolowane zmniejszanie znaków. Procedura ta, łącznie z podobnie wykonywaną weryfikacją proporcji znaków, realizowana jest bez przerwy. Ma to na celu uwzględnianie postępującego zmęczenia uczących się.

Wracając do meritum sprawy — nauki wspomaganą przez komputer — rozpoczyna się ona od prezentacji — uczący wymienia swoje nazwisko, pisząc je na ekranie (za pośrednictwem klawiatury) lub ostatnio po prostu (wyraźnie) na papierze, który stanowi jeden ze śledzonych przez kamerę obiektów. Jest ona sprzężona z komputerem, który odpowiednio analizuje przekazywane mu obrazy. Również informacje można przekazać do maszyny za pośrednictwem głosu — rozpoznaje ona ponad siedem tysięcy słów. Czasami niektóre z nich trzeba powtarzać,

szczególnie podczas pierwszych godzin współpracy z komputerem, bowiem z czasem maszyna się „uczy” rozpoznawać nasz głos. Zatem przekazanie informacji do komputera jest czasem o wiele prostsze niż wykonanie tej samej czynności w rozmowie z kolegą — maszyna jest bardziej cierpliwa i dostarcza dodatkowe środki przekazywania informacji — pismo oraz ikony. W ogólności stanowi to efekt prac nad interfejsem użytkownika, często nazywanym „zyczliwością komputera” lub „łatwo w użyciu”.

Komputer wiedząc z kim ma do czynienia, sięga do odpowiednich informacji, które dotyczą użytkownika. Wśród nich jest życiorys uczącego się, dane o jego stanie zdrowia (w tym oczu), o cechach osobowości, znaczących doświadczeniach. Uzupełnieniem tych informacji są dane aktualne: ocena dotychczas zdobytej przez uczącego się wiedzy na temat będący przedmiotem, zajęć, a także aktualny stan zdrowia, który stanowi kolejny temat „rozmowy” po przedstawieniu się studenta. Mówiąc lub pisząc, można poinformować maszynę, że nasza obecność wynika tylko z obowiązku — stanowi to ceną wytyczną do doboru stosowanej przez komputer strategii nauczania — rzadziej zwraca uwagę, najczęściej chwali i nie wymusza dużej aktywności uczenia się. Jak ON to czyni — jest bardzo przebiegły, bo poza znajomością naszego życiorysu, samopoczucia, kontroluje naszą aktualną aktywność emocjonalną. Czyni to bardzo dyskretnie i bez przerwy — poprzez wspomniany precyzyjny pomiar czasów reakcji (odzwierciedlają one w dużej mierze aktywność emocjonalną człowieka) oraz gdy wyrazimy zgodę! — kontrolując szybkość przepływu krwi oraz EKG i GSR (galwaniczną reakcję skóry) — wymaga to założeń na lewy przegub całym eleganckiej „bransoletki”. Podłączenie to, początkowo budzi mieszane uczucia studentów i przez kilka pierwszych zajęć „bransoletka” wisi „na gwoździu”. Jednak przykład bardziej zaawansowanych w „bojach z komputerem” przekonuje, tym bardziej że to nie boli, a nauka przebiega o ponad 30% sprawniej (szybciej i „trwalej”).

Jak to się dzieje? Komputer dysponując pomiarami z „bransoletki” szybko określa naszą optymalną aktywność emocjonalną, to znaczy taką, przy której najszybciej uczymy się. Istnieje bowiem ścisła zależność naszej aktywności i efektów uczenia się. Gdy materiał jest zbyt trudny, aktywność nasza wzrasta ponad granice poprawnego funkcjonowania tych części mózgu, które odpowiadają za przyswajanie nowej wiedzy, często zachodzi proces odwrotny — nasza aktywność maleje (po prostu nudzimy się) — materiał jest zbyt trudny. Spadek aktywności emocjonalnej towarzyszy także prezentacji zbyt łatwego materiału — mechanizm i skutek jest ten sam — nudzenie się. Maszyna „zdając sobie z tego sprawę” dobiera odpowiednią trudność materiału, by efekty nauki były jak największe. Jeśli i to nie pomaga, komputer dysponuje setkami komend o doskonale znanych mu efektach działania na zmianę stanu aktywności emocjonalnej uczących się. Są to zazwyczaj pochwały, nagany, propozycje przerwy lub jeszcze bardziej przekonujące komentarze.

Temat ten stanowił początek naszej kolejnej rozmowy z Robertem — absolwentem „skomputeryzowanej uczelni”.

W. G.

Sposoby kreślenia okręgów

Podczas pisania programów wykorzystujących lub tworzących grafikę od czasu do czasu musimy wykreślić okrąg. Wtedy z zazdrością patrzemy na właścicieli ZX Spectrum, którzy mogą to zrobić za pomocą jednej instrukcji BASIC'a. Mamy nadzieję, że po przeczytaniu tego artykułu i wprowadzeniu zamieszczonych programów, również właściciele Atari będą mogli kreślić różnorodne okręgi różnymi metodami. Zaprezentujemy kilka procedur od najprostszej, w języku BASIC, która potrzebuje na kreślenie około 60 sekund, do procedury w języku maszynowym, która kreśli okrąg w ułamku sekundy.

Pierwszy z prezentowanych programów „punktuje” okrąg w ciągu około 60 sekund, bez względu na wielkość. Trwa do dosyć długo. Gdy musieliśmy wykreślić kilka różnych okręgów na ekranie i czekaliśmy na efekt końcowy, nasunęły się nam pewne refleksje. Okrąg jest symetryczny, dlaczego zatem nie wykorzystywać tej własności? Jeżeli znamy współrzędne jednego punktu, możemy zrobić jego „odbicie” względem osi x lub osi y. To znaczy, jeśli wiemy, że punkt (x,y) leży na okręgu, to punkt (x,-y) również leży na okręgu. To samo jest prawdą dla (-x, y) i (-x,-y). Musimy zatem wykonać tylko jedną czwartą obliczeń. Ale dlaczego nie iść dalej? Okręgi są także symetryczne względem linii y=x oraz y=-x. Jeśli wiemy, że (x,y) leży na okręgu, wtedy to samo dotyczy (x,y). Teraz musimy wyznaczyć tylko jedną ósmą punktów. Opisana metoda wykorzystana jest w drugim programie.

Niestety, nawet wykonując tylko

jedną ósmą obliczeń, wciąż potrzebujemy ponad 10 sekund na wykreślenie okręgu. Być może istnieje lepszy sposób? Zamiast używania funkcji trygonometrycznych sinus i cosinus wykorzystajmy następujące równanie:

$$x \cdot x + y \cdot y = r \cdot r$$

Nie jest ono zbyt przydatne, lecz po przekształceniu otrzymamy:

$$y = \text{SQRT}(r \cdot r - x \cdot x)$$

Zatem wszystko co musimy zrobić, to wyznaczyć y dla wartości x, zmieniających się od -r do r. Ponieważ funkcja pierwiastka kwadratowego daje tylko wartości dodatnie, musimy pamiętać o wykreśleniu ujemnych wartości pierwiastka. Program 3 jest przykładem, jak to zrobić. Metoda ta jest szybsza od wykorzystującej funkcje trygonometryczne, lecz wciąż wymaga ponad 16 sekund. Użyjmy zatem sposobu wypróbowanego w programie 2. W efekcie otrzymujemy program 4, który zużywa tylko pięć sekund na kreślenie dużych okręgów i nieco mniej dla mniejszych.

I to jest minimalny czas kreślenia okręgu, możliwy do uzyskania w języku BASIC za pomocą znanych wzorów. Dalsze sukcesy możemy uzyskać, zmieniając albo sposób podejścia, albo język programowania, albo jedno i drugie.

Zacznijmy od pierwszego. Wyobraźmy sobie ekran jako siatkę, której oczka reprezentowane są przez elementarne punkty graficzne (pixele). Każdy punkt adresowany jest za pomocą dwóch współrzędnych (x,y), które są za-

wsze liczbami całkowitymi. Wprowadzić możemy napisać PLOT 0.5,0.5, lecz adresy są zaokrąglane do wartości całkowitych. Dzięki temu, jeśli jesteśmy w pewnym punkcie na okręgu i próbujemy określić, gdzie będzie następny punkt, możemy przenieść się w jednym z ośmiu kierunków. Jeśli podzielimy okrąg na ćwiartki, wtedy poprawne są tylko trzy z tych kierunków. Jeżeli jednak podzielimy okrąg na osiem części, wtedy możemy wybrać tylko jeden z dwóch kierunków. Na przykład, jeśli jesteśmy na okręgu w punkcie (r,0), następnym punktem może być albo (r,1), albo (r-1,1). Ponieważ ekran nie może kreślić punktów innych niż o adresach całkowitych, istnieje mały błąd, który nie zawsze równy jest zero. Zastosujemy sposób sprawdzony w programie 2. Zajmuje to tylko trzy sekundy i nigdy nie kreślimy linii o dowolnej długości. W ten sposób działa program 5.

Zauważmy także, że uzyskujemy żądany efekt, wykorzystując tylko dodawanie i odejmowanie. Takie programy mogą być łatwo przetłumaczone na język maszynowy. Program 7 jest przykładem procedury kreślenia okręgów w języku maszynowym. Program 6 służy do wywołania tego ostatniego i zużywa mniej niż 0,2 sekundy na kreślenie okręgu. Wywołanie odbywa się za pomocą funkcji USR. Przekazywanymi parametrami są kolejno: adres kodu procedury; współrzędne środka okręgu x i y, promień okręgu i tryb kreślenia.

Poszczególne wartości trybu kreślenia oznaczają: 0 — zgaszenie punktu; 1 — zapalenie punktu; 2 — odwrócenie punktu (tj. zapalenie lub zgaszenie, w zależności od stanu, w jakim się znajduje).

Jedyną trudnością w używaniu programu w języku maszynowym jest niekontrolowanie czy okrąg nie wyszedł poza ekran.



```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
KV 120 REM PROGRAM #1
QS 130 REM
MW 140 REM CZAS KRESLENIA OKOLO 6
0 SEK.
MF 200 DEG
ZT 210 GRAPHICS 8:COLOR 1
PD 230 SETCOLOR 2,0,0
BF 240 A=160:B=80:R=50
OZ 300 FOR ALFA=0 TO 360
JA 310 X1=INT(R*COS(ALFA)+0.5)
LJ 320 Y1=INT(R*SIN(ALFA)+0.5)
GD 330 PLOT A+X1,B+Y1
VY 340 NEXT ALFA

```

```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
LN 120 REM PROGRAM #2
QS 130 REM
HC 140 REM CZAS KRESLENIA OKOLO 1
0 SEK.
MF 200 DEG
ZT 210 GRAPHICS 8:COLOR 1
EI 230 SETCOLOR 2,0,0:A=160
IH 250 B=80:R=50:PLOT A+R,B
CE 300 FOR ALFA=0 TO 45
JA 310 X1=INT(R*COS(ALFA)+0.5)
LJ 320 Y1=INT(R*SIN(ALFA)+0.5)
GD 330 PLOT A+X1,B+Y1
HB 340 PLOT A-X1,B+Y1
HN 350 PLOT A+X1,B-Y1
IL 360 PLOT A-X1,B-Y1
GG 370 PLOT A+Y1,B+X1
HE 380 PLOT A-Y1,B+X1
HQ 390 PLOT A+Y1,B-X1
HV 400 PLOT A-Y1,B-X1
VT 410 NEXT ALFA

```

```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
MF 120 REM PROGRAM #3
QS 130 REM
PL 140 REM CZAS KRESLENIA OKOLO 1
7 SEK.
QW 150 REM
ZT 210 GRAPHICS 8:COLOR 1
EI 230 SETCOLOR 2,0,0:A=160
QZ 250 B=80:R=50
HT 270 X0=-R:Y0=0
SQ 300 FOR X1=-R TO R
WA 310 Y1=INT(0.5+SQR(R*R-X1*X1))
EJ 320 PLOT A+X0,B+Y0:DRAWTO A+X1
,B+Y1
IF 330 PLOT A+X0,B-Y0:DRAWTO A+X1
,B-Y1
HS 340 X0=X1:Y0=Y1
GQ 350 NEXT X1

```

```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
MX 120 REM PROGRAM #4
QS 130 REM
UT 140 REM CZAS KRESLENIA OKOLO 5
SEK.
QW 150 REM
ZT 210 GRAPHICS 8:COLOR 1
EI 230 SETCOLOR 2,0,0:A=160
QZ 250 B=80:R=50
FP 270 X0=-R:Y0=0:X1=-R
WP 290 Y1=INT(0.5+SQR(R*R-X1*X1))
EF 300 PLOT A+X0,B+Y0:DRAWTO A+X1
,B+Y1
HH 310 PLOT A-X0,B+Y0:DRAWTO A-X1
,B+Y1
ID 320 PLOT A+X0,B-Y0:DRAWTO A+X1
,B-Y1
LF 330 PLOT A-X0,B-Y0:DRAWTO A-X1
,B-Y1
ED 340 PLOT A+Y0,B+X0:DRAWTO A+Y1
,B+X1
HF 350 PLOT A-Y0,B+X0:DRAWTO A-Y1
,B+X1
IB 360 PLOT A+Y0,B-X0:DRAWTO A+Y1
,B-X1
LD 370 PLOT A-Y0,B-X0:DRAWTO A-Y1
,B-X1
IA 380 X0=X1:Y0=Y1
YU 390 IF -X1>=Y1 THEN X1=X1+1:GO
TO 290

```

```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
NP 120 REM PROGRAM #5
QS 130 REM
SL 140 REM CZAS KRESLENIA OKOLO 3
SEK.
QW 150 REM
ZT 210 GRAPHICS 8:COLOR 1
PD 230 SETCOLOR 2,0,0
BF 240 A=160:B=80:R=50
YD 270 FI=0:Y1=0:X1=R
BY 300 FIY=FI+Y1+Y1+1
KA 310 FIXY=FIY-X1-X1+1
GB 320 PLOT A+X1,B+Y1
GZ 330 PLOT A-X1,B+Y1
HL 340 PLOT A+X1,B-Y1
IJ 350 PLOT A-X1,B-Y1
GE 360 PLOT A+Y1,B+X1
HC 370 PLOT A-Y1,B+X1
HO 380 PLOT A+Y1,B-X1
IM 390 PLOT A-Y1,B-X1
UL 400 FI=FIY:Y1=Y1+1
LX 420 IF ABS(FIXY)<ABS(FIY) THEN
FI=FIXY:X1=X1-1
TP 430 IF X1>=Y1 THEN 300

```

```

KZ 100 REM KRESLENIE OKREGU
Q0 110 REM
QH 120 REM PROGRAM #6
QS 130 REM
EA 140 REM CZAS KRESLENIA OKOLO
1833 SEK.
ZT 210 GRAPHICS 8:COLOR 1
PD 230 SETCOLOR 2,0,0
BF 240 A=160:B=80:R=50
DB 270 P=7*16*16*16
VJ 300 I=USR(P,A,B,R,1)
KA 310 FIXY=FIY-X1-X1+1
GB 320 PLOT A+X1,B+Y1
GZ 330 PLOT A-X1,B+Y1
HL 340 PLOT A+X1,B-Y1
IJ 350 PLOT A-X1,B-Y1
GE 360 PLOT A+Y1,B+X1
HC 370 PLOT A-Y1,B+X1
HO 380 PLOT A+Y1,B-X1
IM 390 PLOT A-Y1,B-X1
UL 400 FI=FIY:Y1=Y1+1
LX 420 IF ABS(FIXY)<ABS(FIY) THEN
FI=FIXY:X1=X1-1
TP 430 IF X1>=Y1 THEN 300

```

```

BW 10 REM PODPROGRAM KRESLENIA KO
LA-28000
PU 20 GOSUB 28000
YR 30 END
BJ 28000 FOR I=0 TO 758:READ A:PO
KE 28672+I,A:NEXT I
OU 28004 RESTORE 29500
MT 28005 FOR I=1577 TO 1584:READ
A:POKE I,A:NEXT I
DS 28010 RETURN
GW 28672 DATA 104,104,141,5,6,104
EG 28678 DATA 141,4,6,104,141,7
CY 28684 DATA 6,104,141,6,6,104
DA 28690 DATA 141,9,6,141,12,6
DT 28696 DATA 104,141,8,6,141,11
YR 28702 DATA 6,104,104,141,10,6
UB 28708 DATA 201,3,144,1,96,169
YD 28714 DATA 0,141,13,6,141,14
DX 28720 DATA 6,141,15,6,141,16
KB 28726 DATA 6,24,173,4,6,109
HL 28732 DATA 11,6,141,25,6,173
EH 28738 DATA 5,6,109,12,6,141
JG 28744 DATA 26,6,24,173,4,6
JS 28750 DATA 109,13,6,141,29,6
JG 28756 DATA 173,5,6,109,14,6
IZ 28762 DATA 141,30,6,56,173,4
IX 28768 DATA 6,237,11,6,141,27
IX 28774 DATA 6,173,5,6,237,12
QH 28780 DATA 6,141,28,6,56,173
GF 28786 DATA 4,6,237,13,6,141
FX 28792 DATA 31,6,173,5,6,141
EG 28798 DATA 14,6,141,32,6,24
FY 28804 DATA 173,6,6,109,11,6
IM 28810 DATA 141,33,6,173,7,6
FR 28816 DATA 109,12,6,141,34,6
IL 28822 DATA 24,173,6,6,109,13
NE 28828 DATA 6,141,37,6,173,7
LB 28834 DATA 6,109,14,6,141,38
PP 28840 DATA 6,56,173,6,6,237
JH 28846 DATA 11,6,141,35,6,173
FR 28852 DATA 7,6,237,12,6,141
PY 28858 DATA 36,6,56,173,6,6
MP 28864 DATA 237,13,6,141,39,6

```

```

LG 28870 DATA 173,7,6,237,14,6
JE 28876 DATA 141,40,6,173,25,6
GV 28882 DATA 141,0,6,173,26,6
KM 28888 DATA 141,1,6,173,37,6
LL 28894 DATA 141,2,6,173,38,6
CX 28900 DATA 141,3,6,32,106,114
DD 28906 DATA 173,27,6,141,0,6
ED 28912 DATA 173,28,6,141,1,6
QN 28918 DATA 32,106,114,173,25,6
GK 28924 DATA 141,0,6,173,26,6
KF 28930 DATA 141,1,6,173,39,6
EG 28936 DATA 141,2,6,173,40,6
DX 28942 DATA 141,3,6,32,106,114
ED 28948 DATA 173,27,6,141,0,6
FD 28954 DATA 173,28,6,141,1,6
UB 28960 DATA 32,106,114,173,29,6
CO 28966 DATA 141,0,6,173,30,6
FL 28972 DATA 141,1,6,173,33,6
IE 28978 DATA 141,2,6,173,34,6
EX 28984 DATA 141,3,6,32,106,114
AB 28990 DATA 173,31,6,141,0,6
CV 28996 DATA 173,32,6,141,1,6
SO 29002 DATA 32,106,114,173,29,6
BB 29008 DATA 141,0,6,173,30,6
FW 29014 DATA 141,1,6,173,35,6
GV 29020 DATA 141,2,6,173,36,6
DK 29026 DATA 141,3,6,32,106,114
YO 29032 DATA 173,31,6,141,0,6
BI 29038 DATA 173,32,6,141,1,6
ND 29044 DATA 32,106,114,173,14,6
DO 29050 DATA 205,12,6,240,3,144
LQ 29056 DATA 10,96,173,13,6,205
ML 29062 DATA 11,6,144,1,96,173
CW 29068 DATA 11,6,133,4,173,12
FX 29074 DATA 6,133,5,173,13,6
UE 29080 DATA 133,205,173,14,6,13
3
MZ 29086 DATA 206,6,4,38,5,6
NT 29092 DATA 205,38,206,56,165,2
05
JM 29098 DATA 109,15,6,141,17,6
XF 29104 DATA 165,206,109,16,6,14
1
JV 29110 DATA 18,6,24,173,17,6
QS 29116 DATA 229,4,141,19,6,173
EB 29122 DATA 18,6,229,5,141,20
NC 29128 DATA 6,173,18,6,16,27
SG 29134 DATA 73,255,141,22,6,173
MW 29140 DATA 17,6,73,255,24,105
BQ 29146 DATA 1,141,21,6,173,22
XF 29152 DATA 6,105,0,141,22,6
GR 29158 DATA 24,144,9,141,22,6
EX 29164 DATA 173,17,6,141,21,6
LD 29170 DATA 173,20,6,16,27,73
WV 29176 DATA 255,141,24,6,173,19
ER 29182 DATA 6,73,255,24,105,1
IS 29188 DATA 141,23,6,173,24,6
AV 29194 DATA 105,0,141,24,6,24
MW 29200 DATA 144,9,141,24,6,173
JO 29206 DATA 19,6,141,23,6,173
IV 29212 DATA 17,6,141,15,6,173
GH 29218 DATA 18,6,141,16,6,24
BR 29224 DATA 173,13,6,105,1,141
EI 29230 DATA 13,6,173,14,6,105
CQ 29236 DATA 0,141,14,6,173,22
MY 29242 DATA 6,205,24,6,144,39
LV 29248 DATA 208,8,173,21,6,205
UQ 29254 DATA 23,6,144,29,173,19
DH 29260 DATA 6,141,15,6,173,20
OM 29266 DATA 6,141,16,6,56,173
UO 29272 DATA 11,6,233,1,141,11
BE 29278 DATA 6,173,12,6,233,0
LJ 29284 DATA 141,12,6,76,55,112
WE 29290 DATA 173,2,6,133,205,169
ML 29296 DATA 0,133,206,6,205,38
ND 29302 DATA 206,6,205,38,206,6
TA 29308 DATA 205,38,206,165,205,
133
HO 29314 DATA 4,165,206,133,5,6
UP 29320 DATA 205,38,206,6,205,38
PG 29326 DATA 206,24,165,205,101,
4
YO 29332 DATA 133,205,165,206,101
,5
LU 29338 DATA 133,206,173,0,6,133
DX 29344 DATA 4,173,1,6,133,5
AU 29350 DATA 70,5,102,4,70,5
WN 29356 DATA 102,4,70,5,102,4
OU 29362 DATA 24,165,205,101,4,13
3
AC 29368 DATA 205,165,206,101,5,1
33

```

LI 29374 DATA 206,24,165,205,101,
88
WZ 29380 DATA 133,205,165,206,101,
89
GI 29386 DATA 133,206,173,0,6,41
CF 29392 DATA 7,170,160,0,173,10
NN 29398 DATA 6,208,10,189,41,6
MW 29404 DATA 73,255,49,205,145,2
05
VS 29410 DATA 96,201,1,208,8,189
PS 29416 DATA 41,6,17,205,145,205
TU 29422 DATA 96,189,41,6,81,205
XK 29428 DATA 145,205,96,0,0,0
MN 29500 DATA 128,64,32,16,8,4,2,
1

Supertekst

Każdy z właścicieli komputera Atari wcześniej czy później zetknie się z dosyć dużą trudnością, jaką jest umieszczenie na ekranie jednocześnie rysunku i tekstu. Część problemów rozwiązuje okno tekstowe, lecz nie wszystkie. Chciałoby się mieć znacznie większe możliwości, np. literatury o różnej wysokości, w różnych kolorach, w dowolnym miejscu, itp. Podprogram opisany w 5 numerze „IKS-a” z ubiegłego roku umożliwiał umieszczanie tekstów na ekranie w trybie graficznym 8. Miał jednak duże ograniczenia: był bardzo wolny (napisany w języku BASIC), pozwalał wykorzystać tylko wielkie litery i cyfry oraz tylko w jednym trybie graficznym.

Poniżej prezentujemy procedurę w języku maszynowym, która umożliwi kreślenie obrazów znaków w dowolnym trybie tekstowym lub graficznym. Ma ona następujące możliwości:

- 1) Kreśli cały zestaw znaków ATASCII, włączając wielkie i małe litery, znaki graficzne, symbole specjalne i negatywy każdego z tych znaków w dowolnym trybie graficznym lub tekstowym. Mogą być również kreślone opracowane przez nas znaki. W tym celu należy zmienić zawartość zmiennej systemowej CHBAS (o adresie 756) dla wskazania położenia własnego zestawu znaków.
- 2) Umożliwia podanie ciągu kreślonych znaków. Długość ciągu ograniczona jest wyłącznie

ilością miejsca przeznaczonego na jego wyświetlenie.

- 3) Umożliwia określenie pozycji początkowej kreślonego ciągu, mogą to być dowolne współrzędne (x,y) na ekranie.
- 4) Pozwala na zakrywanie istniejących już rysunków lub łączenie kreślonych znaków z grafiką.
- 5) Wybieranie rejestrów kolorów używanych dla tła i kreślonych znaków.
- 6) Skalowanie każdego ciągu znaków niezależnie, zarówno w poziomie, jak i w pionie, przez określenie liczby wierszy i kolumn dla każdego znaku. Aktualna wielkość każdego znaku zależy od rozmiaru elementarnego punktu graficznego wybranego trybu.
- 7) Ciągi znaków mogą być kreślone w jednym z czterech kierunków.

Opisane możliwości stają się dostępne przez wywołanie procedury w języku maszynowym za pomocą instrukcji USR. Jej składnia jest następująca:

A = USR(ADR(ASM\$), ADR(S\$), LEN(S\$), XS, YS, ORIENT, IWIER, IKOL, RKZ, RKT, PRIOR)

Używane parametry mają następujące znaczenie:

ADR(ASM\$) — Początkowy adres procedury „Supertekst”.

ADR(S\$) — Adres kreślonego ciągu, a dokładniej adres pierwszego znaku tego ciągu. Poprawny będzie dowolny adres, może on również wskazywać podciąg zawarty w długim ciągu.

LEN(S\$) — Parametr ten określa liczbę kreślonych znaków. Funkcja LEN dostarcza właściwej wartości w wypadku kreślenia całego ciągu.

XS, YS — Współrzędne (x,y) początku kreślonego tekstu (lewy dolny róg pierwszego znaku). Wskazany punkt używany jest również jako oś obrotu podczas obracania ciągu. (xs, ys) muszą określać punkt wewnątrz granic aktualnego trybu graficznego.

ORIENT — Parametr ten określa kierunek kreślonego ciągu znaków. Tekst obracany jest w kierunku przeciwnym do ruchu wskazówek zegara od poziomej półosi +x wokół punktu (xs, ys). Wartość parametru powinna być całkowita i interpretowana jest następująco: ORIENT = 0 — obrót o 0 stopni, ORIENT = 1 — obrót o 90 stopni, ORIENT = 2 — 180 stopni, ORIENT = 3 — obrót o 270 stopni. Wartość ORIENT interpretowana jest modulo 3, dlatego ORIENT = 4 oznacza to samo, co ORIENT = 0.

IWIER — Określa, ile wierszy na znak będzie kreślonych, definiuje zatem wysokość każdego znaku. Zwykle IWIER jest większe lub równe osiem, poprawne są także wartości dodatnie mniejsze niż osiem, lecz w efekcie kreślone są znaki z „opuszczonymi” wierszami. Maksymalną wartością jest IWIER = 255.

IKOL — Liczba kolumn na znak, definiuje zatem szerokość każdego znaku.

RKZ — Określa rejestr koloru, który będzie używany do kreślenia znaków. Pośrednio określa kolory znaków. W trybach tekstowych (0—2) powinien być określony jako kod ATASCII.

RKT — Określa rejestr koloru tła dla każdego ciągu znaków. Jeśli wartość parametru PRIOR jest zerowa, to RKT nie ma wpływu na kreślone znaki.

PRIOR — Określa priorytet tła kreślonego ciągu znaków. Jeśli PRIOR = 0, tło znaków nie jest kreślone i istniejąca na ekranie grafika nie będzie naruszona. Jeśli PRIOR ma wartość dodatnią, kolor pośrednio określony przez RKT jest kreślony dla tła.

Podczas określania wartości parametrów należy uwzględnić następujące uwagi:

1. Procedura „Supertekst” nie sprawdza, czy kreślone znaki mieszczą się w obszarze wyświetlania.
2. Negatywy znaków mogą być tworzone dwoma sposobami:



a) podanie negatywów znaków w ciągu, który będzie kreślony,

b) podanie normalnych znaków w ciągu i zamiana wartości RKZ i RKT.

3. Jeśli parametrom RKZ i RKT przydzielona jest ta sama wartość (i prior jest dodatnie), ciąg znaków zostanie wykreślony, lecz pojawi się jako ciągłe kolorowe bloki.

4. Jeśli podana jest niewłaściwa ilość parametrów w instrukcji USR, „Supertekst” nie wykona żadnej akcji.

5. Wartość A zwracana przez funkcję USR nie ma żadnego znaczenia.

Przedstawiony poniżej program zawiera zarówno opisaną procedurę, jak i programy demonstracyjne pokazujące sposoby jej używania.

„Supertekst” rozpoczyna się od linii o numerze 8000. Początkowe instrukcje służą do wprowadzania właściwej procedury w języku wewnętrznym komputera. Proszę zwrócić uwagę na nietypowe instrukcje DATA oraz rozbudowane instrukcje ładowania kodu maszynowego. Spowodowane to jest specyficzną postacią procedury, która umożliwia umieszczenie jej w dowolnym obszarze pamięci, a w szczególności w zmiennej tekstowej. Taką postać nazywamy przemieszczalną. Wszystkie adresy w instrukcjach są względne, wyznaczone względem początku procedury. Właśnie te adresy są oznaczone jako liczby ujemne w instrukcjach DATA.

SUPER TEKST ATARI

```

HW 1000 REM
ZC 1010 REM -----INICJACJA-----
GT 1030 DIM S$(40):DEG :? "PROSZE
    CZEKAC 45 SEK.":GOSUB 8000
II 1040 REM
CR 1050 REM -----DEMO #1-----
IO 1060 REM
XM 1070 GRAPHICS 7+16
VY 1080 S$="INFORMATYKA":XS=24:YS
=24:ORIENT=0:NROWS=24:NCOLS=8:
FCR=3:BCR=0:PRIOR=0:GOSUB 8000
HL 1090 S$="KOMPUTERY":XS=24:YS=4
4:ORIENT=0:NROWS=8:NCOLS=8:FCR
=1:BCR=0:PRIOR=0:GOSUB 8000
UT 1100 S$="SYSTEMY":XS=24:YS=95:
ORIENT=0:NROWS=32:NCOLS=16:FCR
=2:BCR=0:PRIOR=0:GOSUB 8000
ZP 1110 S$="ATARI ":NROWS=8:NCO
LS=8:FCR=3:BCR=1:PRIOR=1
YI 1120 XS=7:YS=64:ORIENT=1:PRIOR
=1:GOSUB 8000
PB 1130 XS=151:YS=2:ORIENT=3:GOSU
B 8000
ZR 1140 GOSUB 7000
IN 1150 REM
BM 1800 REM -----DEMO #2-----
-
IP 1810 REM
EX 1820 GRAPHICS 7+16:COLOR 3
WH 1840 PLOT 34,2:DRAWTO 126,2:DR
AWTO 126,94:DRAWTO 34,94:DRAWTO
0 34,2
QP 1850 PLOT 51,19:DRAWTO 109,19:
DRAWTO 109,77:DRAWTO 51,77:DR
AWTO 51,19
TS 1860 PLOT 60,28:DRAWTO 100,28:
DRAWTO 100,68:DRAWTO 60,68:DR
AWTO 60,28
CT 1870 PLOT 34,2:DRAWTO 60,28
HX 1880 PLOT 126,2:DRAWTO 100,28
DF 1890 PLOT 126,94:DRAWTO 100,68
SZ 1900 PLOT 34,94:DRAWTO 60,68
DL 1910 S$="ATARI":FCR=2:BCR=0:NR
OWS=8:NCOLS=8:PRIOR=0
MM 1920 XS=60:YS=27:ORIENT=0:GOSU
B 8000
WS 1930 XS=101:YS=28:ORIENT=3:GOS
UB 8000
WH 1940 XS=98:YS=69:ORIENT=2:GOSU
B 8000
SV 1950 XS=58:YS=67:ORIENT=1:GOSU
B 8000
CS 1960 NROWS=16:NCOLS=11:FCR=1
NX 1970 XS=53:YS=18:ORIENT=0:GOSU
B 8000
SR 1980 XS=110:YS=21:ORIENT=3:GOS
UB 8000
CA 1990 XS=107:YS=78:ORIENT=2:GOS
UB 8000
NW 2000 XS=50:YS=75:ORIENT=1:GOSU
B 8000
VP 2010 XS=61:YS=67:ORIENT=0:FCR=
3:BCR=2:NROWS=39:NCOLS=39:PRIO
R=1
CY 2020 FOR I=1 TO 10
MP 2030 S$="IKSIKSIKX":S$=S$(I,I
):GOSUB 8000
FC 2040 NEXT I
ZT 2050 GOSUB 7000
QU 2060 GOTO 1070
IC 7000 REM
VS 7010 REM -----ZMIANA KOLORU-----
-
II 7020 REM
CT 7030 FOR I=1 TO 50:FOR J=0 TO
2:SETCOLOR J,RND(0)*16,RND(0)*
16:FOR W=1 TO 5:NEXT W:NEXT J:
NEXT I:RETURN
NG 8000 IF ASMLD=1 THEN A=USR(ADR
(ASM$),ADR(S$),LEN(S$),XS,YS,0
RIENT,NROWS,NCOLS,FCR,BCR,PRIO
R):RETURN
EZ 8010 ASMLD=1
SA 8020 DIM ASM$(725)
OO 8030 FOR I=ADR(ASM$) TO ADR(AS
M$)+724
QG 8040 READ A
BJ 8050 ON (SGN(A)+2) GOSUB 8080,
8160,8220
FO 8060 NEXT I
QP 8070 GOTO 8000
RD 8080 READ B
QX 8090 ADDR=ABS(A)+256*ABS(B)+AD
R(ASM$)
KS 8100 ADDRHI=INT(ADDR/256)
EE 8110 ADDRLO=ADDR-256*ADDRHI
WQ 8120 POKE I,ADDRLO
SQ 8130 POKE I+1,ADDRHI
KY 8140 I=I+1
AX 8150 RETURN
QZ 8160 READ B
EI 8170 IF B<0 THEN 8090
AQ 8180 POKE I,A
CB 8190 POKE I+1,B
KO 8200 I=I+1
AN 8210 RETURN
AA 8220 POKE I,A
AT 8230 RETURN
FS 8240 DATA 104,141,-255,0,10,17
0,240,8
OD 8250 DATA 104,157,-255,0,202,7
6,-6,0
QO 8260 DATA 173,-255,0,201,10,24
0,1,96
BK 8270 DATA 173,-16,-1,208,1,96,
173,-8
EA 8280 DATA -1,208,1,96,173,-6,-
1,208,1
XK 8290 DATA 96,173,-10,-1,41,3,1
41,-10
ZX 8300 DATA -1,173,-18,-1,133,20
3,173
OR 8310 DATA -19,-1,133,204,169,0
,141
RY 8320 DATA -17,-1,238,-17,-1,56
,173
AT 8330 DATA -16,-1,237,-17,-1,16
,3,76
VC 8340 DATA -254,0,32,-32,-1,173
,-6,-1
HN 8350 DATA 141,-7,-1,206,-7,-1,
174,-7
TM 8360 DATA -1,224,255,208,3,76,
-251,0
NK 8370 DATA 172,-6,-1,32,-138,-1
,140,-3
AH 8380 DATA -1,173,-6,-1,174,-17
,-1,172
FO 8390 DATA -7,-1,32,-195,-1,140
,-20,-1
HM 8400 DATA 142,-21,-1,169,255,1
41,-9
PB 8410 DATA -1,238,-9,-1,174,-9,
-1,236
IR 8420 DATA -8,-1,208,3,76,-248,
0,172
QS 8430 DATA -8,-1,32,-138,-1,140
,-5,-1
XJ 8440 DATA 172,-5,-1,177,205,17
2,-3,-1
ZU 8450 DATA 57,-24,-1,240,2,169,
1,141
QV 8460 DATA -1,-1,173,-135,-1,24
0,9,56
BL 8470 DATA 169,1,237,-1,-1,141,
-1,-1
ZM 8480 DATA 173,-1,-1,208,5,173,
0,-1
WM 8490 DATA 240,46,174,-2,-1,173
,-1,-1
LH 8500 DATA 240,3,174,-4,-1,142,
-212,-2
QY 8510 DATA 173,-8,-1,162,1,172,
-9,-1
KU 8520 DATA 32,-195,-1,140,-22,-
1,142

```

WU 8530 DATA -23,-1,32,-16,-2,173,-184	OM 8750 DATA 169,0,141,-192,-1,162,3,24	CE 8970 DATA -23,-1,141,-186,-2,96,202
OH 8540 DATA -2,174,-183,-2,172,-185,-2	TY 8760 DATA 46,-191,-1,46,-192,-1,202	XO 8980 DATA 208,39,56,173,-14,-1,237
AT 8550 DATA 32,-187,-2,76,-134,0,76,-89	LJ 8770 DATA 208,246,160,255,200,56,173	IC 8990 DATA -22,-1,141,-183,-2,173,-15
YT 8560 DATA 0,76,-65,0,96,0,0,0,0,0,0,0	ZV 8780 DATA -191,-1,237,-194,-1,141	MV 9000 DATA -1,237,-23,-1,141,-184,-2
UH 8570 DATA 0,0,0,0,0,0,0,0,0,0,0,0	RS 8790 DATA -191,-1,173,-192,-1,233,0	VP 9010 DATA 56,173,-12,-1,237,-20,-1
GA 8580 DATA 0,0,0,0,1,2,4,8,16,32,64	LX 8800 DATA 141,-192,-1,16,235,142,-191	MC 9020 DATA 141,-185,-2,173,-13,-1,237
CS 8590 DATA 128,169,0,141,-135,-1,172	KI 8810 DATA -1,96,0,0,0,0,141,-12,-2	AB 9030 DATA -21,-1,141,-186,-2,96,202
NG 8600 DATA -17,-1,136,177,203,141,-136	OT 8820 DATA 140,-13,-2,169,0,141,-14,-2	WL 9040 DATA 208,39,56,173,-14,-1,237
GI 8610 DATA -1,16,13,169,1,141,-135,-1	XE 8830 DATA 141,-15,-2,24,173,-12,-2	FZ 9050 DATA -20,-1,141,-183,-2,173,-15
IY 8620 DATA 173,-136,-1,41,127,141,-136	WV 8840 DATA 109,-14,-2,141,-14,-2,169,0	LZ 9060 DATA -1,237,-21,-1,141,-184,-2
HF 8630 DATA -1,56,173,-136,-1,233,32,16	JL 8850 DATA 109,-15,-2,141,-15,-2,202	UP 9070 DATA 24,173,-12,-1,109,-22,-1
PA 8640 DATA 12,24,173,-136,-1,105,64	FW 8860 DATA 208,235,56,173,-14,-2,233,1	KF 9080 DATA 141,-185,-2,173,-13,-1,109
AE 8650 DATA 141,-137,-1,76,-107,-1,56	PW 8870 DATA 141,-14,-2,173,-15,-2,233,0	RR 9090 DATA -23,-1,141,-186,-2,96,24
WO 8660 DATA 173,-136,-1,233,96,16,12,56	EL 8880 DATA 141,-15,-2,56,173,-14,-2	KG 9100 DATA 173,-14,-1,109,-22,-1,141
IK 8670 DATA 173,-136,-1,233,32,141,-137	WZ 8890 DATA 237,-13,-2,141,-14,-2,168	GA 9110 DATA -183,-2,173,-15,-1,109,-23
RC 8680 DATA -1,76,-107,-1,173,-136,-1	MT 8900 DATA 173,-15,-2,233,0,141,-15,-2	UF 9120 DATA -1,141,-186,-2,24,173,-12
EY 8690 DATA 141,-137,-1,169,0,133,206	DO 8910 DATA 170,96,0,0,0,0,174,-10,-1	LD 9130 DATA -1,109,-20,-1,141,-185,-2
BG 8700 DATA 173,-137,-1,133,205,162,3	QX 8920 DATA 208,39,24,173,-14,-1,109	IZ 9140 DATA 173,-13,-1,109,-21,-1,141
AR 8710 DATA 24,38,205,38,206,202,208	GK 8930 DATA -20,-1,141,-183,-2,173,-15	ZO 9150 DATA -186,-2,96,0,0,0,0,134,85
CW 8720 DATA 248,24,165,206,109,244,2	LJ 8940 DATA -1,109,-21,-1,141,-184,-2	NN 9160 DATA 133,86,132,84,162,96,169,11
BO 8730 DATA 133,206,96,0,0,0,142,-193	ZC 8950 DATA 56,173,-12,-1,237,-22,-1	GI 9170 DATA 157,66,3,169,0,157,72,3,173
CC 8740 DATA -1,140,-194,-1,142,-191,-1	NF 8960 DATA 141,-185,-2,173,-13,-1,237	EX 9180 DATA -212,-2,32,86,228,96,1

REVERSI

Jedną z ciekawszych gier, jakie mamy dla komputera Atari jest REVERSI. Nie odznacza się oształmianą grafiką ani fascynującym dźwiękiem. Nie przenosi nas również w inny, nieznaną świat. Jednak każda rozgrywka jest interesująca, tym bardziej że jej przebieg zależy od nas. Gra nie wymaga biegłości w posługiwaniu się manipulatorem ani dobrego refleksu. Wymaga za to logicznego myślenia i przewidywania skutków posunięć.

Program jest komputerową wersją tradycyjnej gry, znanej od końca XVIII w., szczególnie chętnie uprawianej w drugiej połowie XIX wieku w całej Europie.

Do gry potrzebna jest plansza o 64 polach (jak zwykła szachownica, lecz jednobarwna) oraz 64 specjalne pionki — każdy z nich

jest z jednej strony biały, a z drugiej czarny, może więc być użyty jako biały lub czarny, w zależności od położenia na planszy.

Celem gry jest ustawienie, a dokładniej uzyskanie na planszy maksymalnej liczby własnych pionków.

Rozpoczynającego grę wyznacza się przez losowanie. Obydwaj gracze, na zmianę, kolejno ustawiają swoje pionki na planszy. Pionki raz postawione nie wykonuje żadnych ruchów, kolejne posunięcia polegają na dostawianiu na planszy coraz to nowych pionków.

Obowiązkiem graczy jest w pierwszych czterech ruchach ustawić pionki na czterech centralnych polach planszy (D4, D5, E4, E5). W następnych ruchach gracze ustawiają swoje pionki już na dowolnie wybranych polach, przy czym w wyniku każdego posunięcia co najmniej jeden pion prze-

ciwnika musi zostać zdobyty. Jeśli gracz, na którego przypada kolejność wykonania ruchu, nie jest w stanie zrobić tego, tak by zdobyć przynajmniej jeden pion przeciwnika, musi zrezygnować z wykonania ruchu.

Bicie pionków odbywa się przez oskrzydlenie z dwóch stron pól przeciwnika — pionowo, poziomo lub po przekątnej. Pionki przeciwnika, znajdujące się pomiędzy naszymi, zostają zdobyte; odwrócone na drugą stronę stają się naszymi.

Jeden i ten sam pion, stojący na planszy, może więc w toku rozgrywki wiele razy zmieniać przynależność do jednego lub drugiego gracza. Często znaczna przewaga może być zniwelowana przez jeden strategicznie umieszczony pion.

W komputerowej wersji gry odpowiednia plansza pojawia się na ekranie z czterema pionkami usta-

wionymi w centralnych polach planszy. Jeden gracz może grać przeciwko drugiemu lub komputerowi na trzech różnych poziomach, za pomocą manipulatora.

Na najłatwiejszym poziomie komputer zawsze stara się zabrać jak najwięcej pionów w jednym ruchu (podobnie grają początkujący gracze) i daje się łatwo ograć. Za to na najwyższym poziomie rozważa swoje ruchy, nasze odpowiedzi i

swój następny ruch; bierze również pod uwagę każde pole, ponieważ niektóre z nich są traktowane jako bardziej wartościowe niż inne. Inna opcja oferuje niezwykłą własność — możemy zobaczyć, w jaki sposób komputer wyznacza swój następny ruch (dostępne na najwyższym poziomie). Wyświetla wtedy liczbę pól, które będą uzyskane dla dowolnego ruchu i wartość każdego kwadratu (pola) po roz-

ważeniu innych możliwych ruchów.

Komputer jest wymagającym przeciwnikiem, zmuszającym nas do intensywnego myślenia. Umożliwia nam również, po nabraniu wprawy, rozegranie partii z prawdziwym ludzkim przeciwnikiem. Dostarcza nam planszę i pionki, przestrzega zasad gry oraz na bieżąco podaje wynik.

Ludwik PIELA

Tomasz MROWIEC

PROGRAM

SHARP

KTO WINIEN?

Prezentowany program jest komputerową... opowieścią kryminalną. Ma ona 2295 odmian, które zapewnia funkcja RANDOM.

A oto fabuła opowieści. Popelniono przestępstwo. Nie perfidne morderstwo, lecz „zwyczajną” kradzież. Jaką? Poda komputer! „Zaprezentuje” on ośmiu zatrzymanych. Należy zdecydować, *kto winien?*

Prześlanką są zaprezentowane „ustalenia”, metodą zaś — dwa podstawowe pytania kryminalistyki:

1. — *KTO zostawił ślady?*
2. — *KTO skorzystał z kradzieży?*

„Śledztwo” jest dwuetapowe: najpierw uzyskujemy część danych i wstępnie typujemy winnych.

W drugim etapie dostajemy pełną dokumentację i podejmujemy ostateczne decyzje.

Czy były one słuszne? Odpowiedź... komputer.

Program wywołuje etykieta „X” (RUN 100).

Uwaga! W programie wykorzystano „japońskie delikatesy”: 120:Restore 10 — oznacza

```

10: DATA "ADAM", "JAN", "EDWARD", "HENRYK", "ZENON", "MARIAN", "PAWEL", "ROMAN"
20: DATA " ZYJE UBOGO", " ZYJE SKROMNIE", " ZYJE SREDNIO"
30: DATA " CZESTE LIBACJE", " BUDUJE GARAZ", " STAWIA DOM"
40: DATA " WLAMANIE DO", " OGRABIENIE", " WYNIESIENIE Z"
50: DATA " PAWILON U", " MAGAZYNU", " KASY"
60: DATA " SLADY PALCOW", " NITKI UBRANIA", " SL

```

„czytaj dane (Data) TYLKO linii 10”. 280:ON A GOSUB 300, 320, 330 — „gdy A = 1, adresuj 300; gdy A = 2, to 320; gdy A = 3, to 330”.

Życzymy osiągnięcia absolutnej bieguności...

Janusz MILLER

```

ADY OBUWIA"
70: DATA " KILKA W LOSOW", " NIEDOPALEK", " BRAK SLADOW"
80: A=INT Y:Y=(Y-A)*10:B=INT Y:Y=(Y-B)*10
90: C=INT Y:Y=(Y-C)*10:D=Y:RETURN
100: "X":CLEAR:RANDOM:USING:WAIT 0:PRINT "S L E D Z T W O"
110: DIM X$(8),K$(6),Z$(6),S$(6),X(8),Y(8)
120: RESTORE 10:FOR K=1 TO 8:READ X$(K):NEXT K
130: RESTORE 20:FOR

```

```

K=1TO 3:READ K
$(K):NEXT K
140:RESTORE 30:FOR
K=4TO 6:READ K
$(K):NEXT K
150:RESTORE 40:FOR
K=1TO 3:READ Z
$(K):NEXT K
160:RESTORE 50:FOR
K=4TO 6:READ Z
$(K):NEXT K
170:RESTORE 60:FOR
K=1TO 3:READ S
$(K):NEXT K
180:RESTORE 70:FOR
K=4TO 6:READ S
$(K):NEXT K
190:G=RND 8:FOR K=
1TO G
200:R=RND 8:IF X(R
)>0GOTO 200
210:X(R)=3:NEXT K:
WAIT
230:FOR K=1TO 8:IF
X(K)>0GOTO 260
240:R=RND 2:IF R=2
LET X(K)=1:
GOTO 260
250:X(K)=2
260:NEXT K:R=RND 3
:S=(RND 3)+3:
PRINT Z$(R);Z$(
S)
270:FOR K=1TO 8:A=
INT X(K)
280:ON AGOSUB 300,
320, 330

```

```

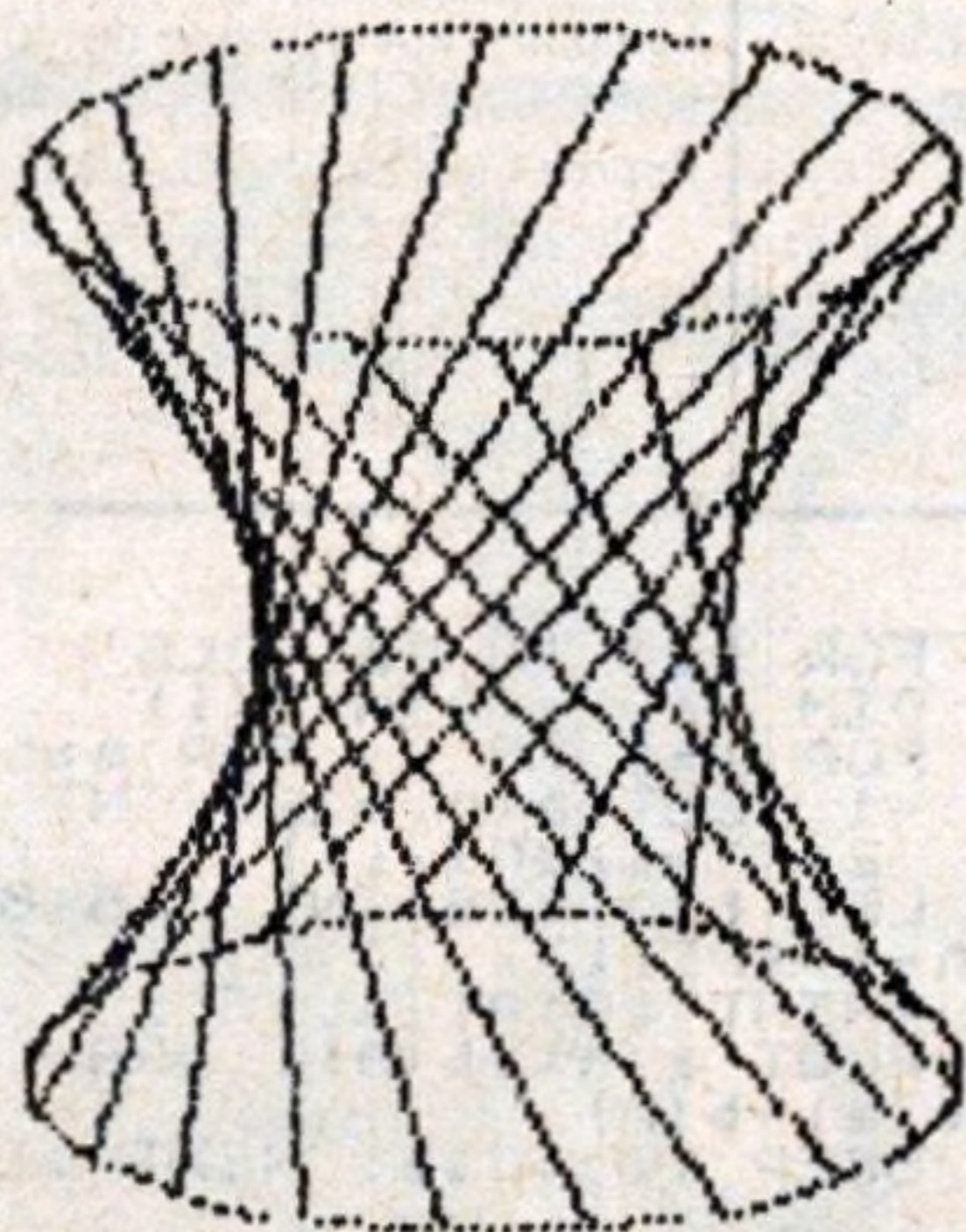
290:NEXT K:PRINT "
Z A T R Z Y M
A N I":GOTO 36
0
300:X(K)=X(K)+0.6
310:R=RND 6:X(K)=X
(K)+R/100+R/1E
3:RETURN
320:R=RND 5:X(K)=X
(K)+R/10:GOTO
310
330:R=RND 5:S=RND
3:T=(RND 3)+3
340:X(K)=X(K)+R/10
+S/100+T/1E3:
RETURN
350:R=RND 3:S=(RND
3)+3:PRINT Z$(
R);Z$(S):Q=0
360:U=0:W=RND 8:
FOR K=1TO 8
370:T=W+K:IF T>8
LET T=T-8
380:IF (Q=1)AND (Y
(T)<3)GOTO 460
390:Y=X(T):GOSUB 8
0:U=U+1
400:PRINT U;" IMIE
:";X$(T)
410:PRINT "DOWODY:
";S$(B)
420:PRINT "ZWYKLE:
";K$(C):IF Q=
0GOTO 440
430:PRINT "OSTATNI
0:";K$(D)

```

```

440:INPUT "WINIEN
? T=1 N=0 ";E:
IF E=1LET Y(T)
=3
450:IF E=0LET Y(T)
=1
460:NEXT K:IF Q=1
THEN PRINT "W
Y N I K.I":
GOTO 480
470:PRINT "ETAP DR
UGI":Q=1:GOTO
360
480:WAIT 0:U$=" UW
AGA ! ";W$=" W
INNY ";N$=" NI
EWINNY ";D$="
ZGODA: "
490:FOR K=1TO 8:A=
INT X(K):B=Y(K
):C$=X$(K):
PRINT STR$ K;:
WAIT
500:IF (A<3)AND (B
<3)THEN PRINT
D$;C$;N$
510:IF (A=3)AND (B
=3)THEN PRINT
D$;C$;W$
520:IF (A=3)AND (B
<3)THEN PRINT
U$;C$;W$
530:IF (A<3)AND (B
=3)THEN PRINT
U$;C$;N$
540:WAIT 0:PRINT "
":NEXT K:END

```



ATARI

```

5 REM * GRAFIKA ATARI *
*HIPERBOLOIDA JEDNOOBRO
TOWA *
10 GRAPHICS 8:COLOR 1:P
I=3.14:N=-10
20 FOR A=-PI TO PI STEP
0.01*PI:AA=A+2/3*PI
30 X=INT(COS(A)*45):XX=
INT(COS(AA)*45):Y=INT(S
IN(A)*17):YY=INT(SIN(AA
)*17)
40 IF A=N*PI/10 THEN N=
N+1:PLOT 160+X,30+Y:DRA
WTO 160+XX,120+YY
50 PLOT 160+X,30+Y:PLOT
150+XX,120+YY:NEXT A

```

JANUSZ W. JANIEC

MIKROKOMPUTERY IBM PC (2)

System mikroprocesorowy, tworzący mikrokomputer IBM PC/XT, można zaliczyć do systemów wieloprocusorowych z dzieloną magistralą lokalną, a uściślając: przypisać go do systemów wielomikroprocesorowych silnie powiązanych, gdyż komunikacja między mikroprocesorami odbywa się przez wspólną pamięć operacyjną.

Z punktu widzenia funkcjonalnego w systemie można wyróżnić cztery zasadnicze podsystemy: podsystem mikroprocesora i współpracujących z nim elementów, podsystem pamięci stałej ROM, podsystem pamięci zapisywalnej RAM, podsystem wejścia — wyjścia.

Z punktu widzenia konstrukcyjnego system można podzielić na pakiety między innymi: pakiet systemowy i pakiety dodatkowe (rozszerzające). Na rysunku 3 przedstawiono uproszczony schemat blokowy mikrokomputera IBM PC/XT.

W systemie IBM PC/XT jako jednostkę centralną wykorzystano INTEL 8088, będący wersją 16-bitowego mikroprocesora INTEL 8086, wersją z 8-bitową zewnętrzną magistralą danych. Mikroprocesor 8088, zastosowany w omawianym mikrokomputerze, pracuje z częstotliwością zegara 4,77 MHz. Sygnał zegarowy dla mikroprocesora dostarcza układ 8284, generator sygnału zegarowego. Źródłem częstotliwości sterującej pracą układu 8284 jest zewnętrzny rezonator kwarcowy o częstotliwości 14,31818 MHz, która to częstotliwość dzielona jest przez 3.

Częstotliwość 4,77 MHz pozwala na pracę mikroprocesora 8088 z cyklem magistrali trwającym cztery takty zegara po 0,210 μ s lub 0,840 μ s.

Układy wejścia — wyjścia pracują z cyklem pięciu taktów zegara po 0,210 μ s lub 1,05 μ s.

INTEL 8088 ma 20-bitową magistralę adresową ($2^{20} = 1 \text{ MB} = 1024 \text{ KB}$, przy czym 1 KB = 1024 bajty), a więc może adresować przestrzeń 1 048 576 bajtów. Z ośmioma mniej znaczącymi bitami adresu multipleksowane są bity danych. Bity adresowe i bity danych przekazywane są do układów współpracujących z mikroprocesorem za pomocą buforów adresowych i buforów danych.

Mikroprocesor 8088 steruje pracą pozostałych elementów systemu przez kontroler magistrali sterującej, układ 8288, nazywany również generatorem sygnałów magistrali sterującej w konfiguracji maksymalnej (mikroprocesora 8086 lub 8088).

Koprocesor INTEL 8087 jest procesorem numerycznym i przeznaczony jest do wykonywania operacji arytmetycznych. Pracuje równoległe z jednostką centralną (zwaną mikroprocesorem nadrzędnym), pobiera ten sam ciąg rozkazów, ale wykonuje rozkazy przeznaczone tylko dla niego.

Programowalny kontroler przerw 8259 A przeznaczony jest do nadzorowania przerw (zadań obsługi) przekazywanych

do mikroprocesora przez układy współpracujące. Układ 8259 A umożliwia realizację ośmiopozomowych przerw priorytetowych, tzw. przerw maskowalnych INTR. Sześć przerw przekazywanych jest do pakietu systemowego z pakietów dodatkowych, dołączonych do systemu za pomocą złączek (łączówek). Wśród sygnałów przerw, zgłaszanych przez pakiety dodatkowe są sygnały z dysku stałego, dysku elastycznego, drukarki. Dwa kolejne przerwania zgłaszają układy z pakietu systemowego: moduł czasowy 8253 i klawiatura. W omawianym systemie realizowane jest również przerwanie niemaskowalne, po wykryciu błędu parzystości w pamięci operacyjnej.

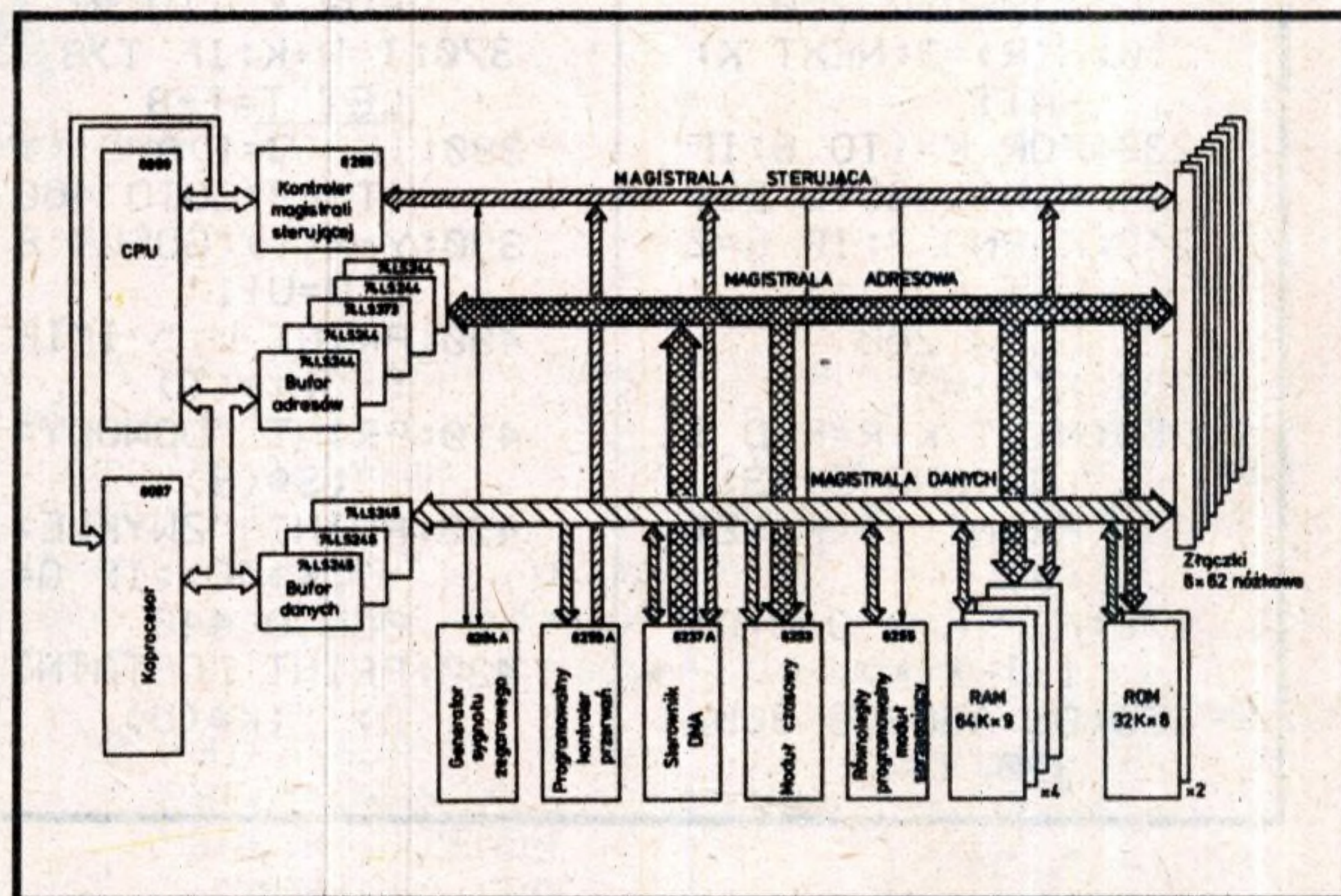
Sterownik DMA (Direct Memory Access) 8237 A umożliwia prowadzenie szybkich transmisji między modułami zewnętrznymi mikrokomputera a pamięcią operacyjną, z

komputera, przez równoległy programowany moduł sprzęgający 8255.

Poprzez porty (bramy wejścia — wyjścia) układu 8255 informacje z / do magistrali danych przekazywane są do / z różnych układów pakietu systemowego, m.in. do głośnika.

W systemie znajduje się 256 KB pamięci operacyjnej RAM, w czterech blokach 64 KB \times 9, dziewiąty bit to kontrolny bit parzystości. W jednym bloku znajduje się dziewięć układów pamięci dynamicznej, o organizacji 64 K \times 1 bit i czasie dostępu 0,200 μ s. Pamięć operacyjną można zwiększyć do 640 KB po dołączeniu do pakietu systemowego, poprzez jedną ze złączek pakietu dodatkowego, zawierającego również układy organizujące pracę: monitora monochromatycznego i drukarki matrycowej, mechanizmów dysków elastycznych, monitora kolorowego itp. Kolejne złączki umożliwiają dołączanie innych pakietów.

Ostatnim z omawianych elementów jest pamięć stała. Mogą to być pamięci ROM lub EPROM. W pakiecie znajdują się dwa ukła-



pominięciem mikroprocesora. Steruje on transmisją danych odczytanych (zapisywanych) na dysku do/z pamięci operacyjnej.

Moduł czasowy (Timer/Cunter) 8253 wykorzystywany jest do generowania przebiegów wyjściowych o zadanej częstotliwości, przebiegów sterujących głośnikiem mikro-

dy o organizacji 32 K \times 8 bitów. Mikrokomputer wykorzystuje do 40 KB pamięci stałej, w której zapisany jest ROM-BIOS, interpreter języka BASIC, diagnostyka.

Dalszy opis mikrokomputera w następnym numerze „IKS-a”.

Jacek WOJTALA

Spectrum Kryształ

(J. Rubinowicz)

```

10 FOR i=3 TO 20: CLS
20 DIM a(i): DIM b(i)
30 FOR n=1 TO i: FOR m=n TO 1
STEP -1: LET a(n)=128+INT (.5+86
*SIN (2*(n-1)*PI/i))
40 LET b(n)=88+INT (.5+86*COS
(2*(n-1)*PI/i))
50 PRINT #0; AT 1,15; i
60 PLOT a(n), b(n): DRAW a(m)-a
(n), b(m)-b(n)
70 BEEP .01,3*M: NEXT m: NEXT
n
80 PAUSE 50: NEXT i: GO TO 10
90 SAVE "*KRYSTAL*" LINE 10
    
```



INFOSYSTEM '87




```

4000 REM*****
4010 REM      KRESLENIE TROJKATA
4020 REM*****
4025 CLS #2:PRINT #2," "+CHR$(244)+" (R)

```

```

OWNORAMIENNY (D)OWOLNY";
4030 PRINT #2," (M)ENU KRESLENIE (F)I
GURY";
4040 IF INKEY(38)=0 THEN 560
4050 IF INKEY(53)=0 THEN 1990
4060 IF INKEY(50)=0 THEN w$="R": GOTO 41
00
4080 IF INKEY(61)=0 THEN w$="D": GOTO 41
00
4090 GOTO 4040
4100 CLS #2
4110 PRINT #2," WYZNACZ PIERWSZY WIERZCH
OLEK ";
4120 LOCATE #2,1,2:PRINT #2,"UZYWAJ :"+C
HR$(240)+CHR$(242)+CHR$(243)+CHR$(241)+"
COPY LUB JOYSTICK"
4130 GOSUB 1720:GOSUB 1070:GOSUB 1210
4140 GOSUB 1010:GOSUB 1070:GOSUB 1120
4150 GOSUB 1010
4160 IF INKEY(FX)=0 THEN 4180
4170 GOTO 4130
4180 x1%=XPOS:y1%=YPOS:LOCATE #2,1,1:PRI
NT #2," WYZNACZ DRUGI WIERZCHOLEK ";
4190 q%=0
4200 GOSUB 1720:GOSUB 1070:GOSUB 1210

4210 GOSUB 1010:GOSUB 1070:GOSUB 1120
4230 GOSUB 1010
4240 IF q%=0 THEN PLOT x1%,y1%,1:MOVE x,
y
4250 IF INKEY(FX)=0 THEN 4270
4260 GOTO 4200
4270 x2%=XPOS:y2%=YPOS
4275 IF w$="R" THEN LOCATE #2,1,1:PRINT
#2," WYZNACZ WYSOKOSC ";
4280 IF w$="D" THEN LOCATE #2,1,1:PRINT
#2," WYZNACZ TRZECI WIERZCHOLEK ";
4300 q%=0
4310 IF w$="R" THEN x4%=x1%+(x2%-x1%)/2:
y4%=y1%+(y2%-y1%)/2:MOVE x4%,y4%:x%=XPOS
:y%=YPOS
4320 GOSUB 1720:GOSUB 1070:GOSUB 1210

4330 GOSUB 1010:GOSUB 1070:GOSUB 1120

4340 GOSUB 1010
4345 IF q%=0 AND w$="D" THEN PLOT x2%,y2
%,1:MOVE x%,y%
4350 IF INKEY(FX)=0 THEN 4370
4360 GOTO 4320
4370 x3%=XPOS:y3%=YPOS
4380 MOVE x1%,y1%:DRAW x2%,y2%,kolor:DRA
W x3%,y3%,kolor:DRAW x1%,y1%,kolor
4390 GOTO 4000

```



W KLUBIE, KTÓRY BĘDZIE

Pierwszy mikrokomputer Spectrum Plus, pojawił się w Wyższej Szkole Oficerskiej Wojsk Łączności im. płk. B. Kowalskiego w październiku 1985 roku. Od siedmiu lat istniał już wówczas Ośrodek Obliczeniowy, który prócz usług informatycznych prowadził prace naukowo-badawcze dla potrzeb uczelni. Dzięki komputerowi „Odra 1305” utworzono całą sieć abonencką, z końcówkami, drukarkami w komendzie i cyklach nauczania. Na podstawie systemu opracowane były różne programy specjalistyczne i uniwersalne.

W miarę przybywania sprzętu, czyli mikrokomputerów, pracownikom ośrodka oraz podchorążym zamarzyło się powstanie klubu mikrokomputerowego. Pozostaje on nadal w sferze marzeń, choć już bardziej realnych. Klubu jako takiego nie ma, gdyż brak dla niego na razie pomieszczenia. Najprawdopodobniej już w tym roku znajdzie swoje lokum w Klubie Podchorążego. Tymczasem trwają poszukiwania nazwy dla klubu, który w świadomości sporej grupy zapaleńców już istnieje. Przy ośrodku powstało Koło Naukowe Informatyków, które wraz z funkcjonującym przy jednym z cykliów Kołem Naukowym Podchorążych Teleinformatyków skupia przyszłych członków klubu.

— Jest ich około czterdziestu — wyjaśnia kierownik ośrodka i

zarazem opiekun komputerowej grupy, mjr **Tadeusz Szuszkiewicz**. — Połowa rekrutuje się spośród słuchaczy szkoły oficerskiej, druga połowa to podchorążowie SPR. Mamy możliwość korzystania z kilkunastu mikrokomputerów Spectrum, Unipolbrit 2086 i Amstrad 6128 M. Kilka spośród nich zawsze jest w ośrodku. Do tego dochodzi jeszcze kilka prywatnych mikrokomputerów, podchorążych, którzy nie potrafili rozstać się ze swoim sprzętem i zabrali go do szkoły.

Komputerowa „mania” nie okazała się słomianym ogniem, lecz zjawiskiem znacznie trwalszym. Od lat ma swoich wiernych zwolenników. Tu też wzbudza ogromne zainteresowanie. Podchorążowie podkreślają, że trudno dzisiaj oddzielić nowoczesną technikę od komputeryzacji. W programie zegrzyńskiej łącznościówki jest nauka postugi-



wania się mikrokomputerem i podstawowego programowania. Tę wiedzę można wzbogacić na klubowych spotkaniach.

Ważną rolę odgrywają tam słuchacze Szkoły Podchorążych Rezerwy. Są oni absolwentami renomowanych uczelni technicznych. Wielu z nich to informatycy lub cybernetycy. Oni organizują kursy języków, uczą zasad programowania i licznych sposobów wykorzystania mikrokomputera. Chętnych na te zajęcia jest tak dużo, że nie jest ich w stanie pomieścić sporych rozmiarów sala.

Podchorążowie SPR korzystają ze sprzętu zgromadzonego w Ośrodku Obliczeniowym i fachowej rady jego pracowników. Współpraca ta zresztą przeciąga się na dłuższy okres, trwa nawet już po przejściu podchorążych do cywila. Dobrymi tego przykładami są kontakty z informatykami Akademii Górniczo-Hutniczej czy Politechniki Gliwickiej. Do WOSWŁ dociera w ten sposób trudna do zdobycia literatura specjalistyczna, interesujące programy, a także sprawozdania z prowadzonych aktualnie prac naukowo-badawczych.

Etap zafascynowania grami komputerowymi szybko minął. To naturalna kolej rzeczy, że zaczyna się od rzeczy najprostszych, choć efektywnych. Teraz podchorążowie rozwiązują już konkretne problemy, wyczytane w literaturze lub wynikiem z programu studiów. Za swoje największe osiągnięcie przyszli klubowicze uważają opracowanie egzaminatora z dowolnego przedmiotu. Za ten program zostali wyróżnieni przez ministra obrony narodowej. Poza tym większość czasu poświęcają na przygotowanie programów do zastosowania w wojskach łączności.

Sierż. pchor. WSO **Krzysztof Fąferek** jest właścicielem mikrokom-

putera Atari 800 XL. Mówi, że to jego wielkie hobby, któremu poświęca większość swego wolnego czasu. Jest w tej korzystnej sytuacji, że nie jest ograniczony czasem i po zamknięciu ośrodka ma swój sprzęt, który zresztą udostępnia kolegom.

— Najczęściej pracujemy nad konkretnymi problemami — stwierdza kpr. pchor. WSO, **Tomasz Szafranowski** — lecz w bardzo szerokim pojęciu. Rozwiązujemy bowiem skomplikowane kwestie techniczne, ale budujemy też pomoce naukowe do zdawania egzaminów, poczynając od reguaminów, a kończąc na taktyce. Oczywiście staramy się tworzyć coś nowego, indywidualnego, może nie mającego tak powszechnego, praktycznego zastosowania, ale przynoszącego za to najwięcej satysfakcji i nowych doświadczeń.

Absolwent Politechniki Warszawskiej, plut. pchor. SPR, **Jacek Kuśmierczyk** chwali sobie, że właśnie tutaj trafił do wojska. Jest informatykiem, ma więc szansę uzupełniać swoje wiadomości. Podczas studiów wiele się nauczył, został doskonale przygotowany teoretycznie, ale jeszcze dwa lata temu Politechnika nie dysponowała odpowiednim zapleczem sprzętowym. Tutaj więc weryfikuje swe umiejętności. Uczy kolegów, a przy okazji poznaje nowe typy mikrokomputerów.

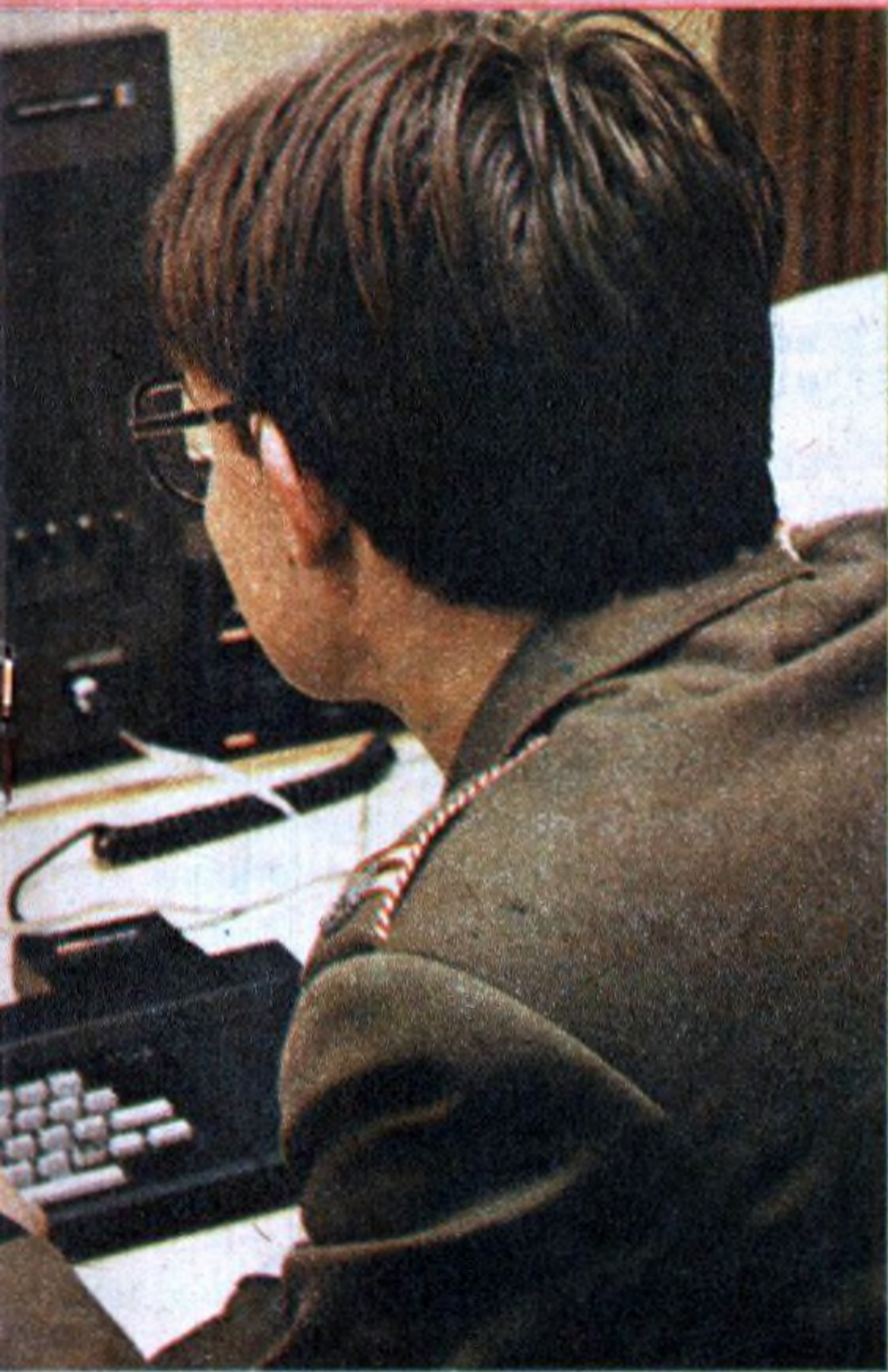
Widać, że ten nie istniejący jeszcze klub już działa. Ma ambicje oddziaływania nie tylko na słuchaczy szkoły, ale i na miejscową młodzież. Żeby tak było, niezbędny jest lokal i więcej sprzętu. Pomieszczenie wkrótce będzie, może znajdzie się i mecenas, który zwiększy stan posiadania o kilka mikrokomputerów.

J. RAJCH

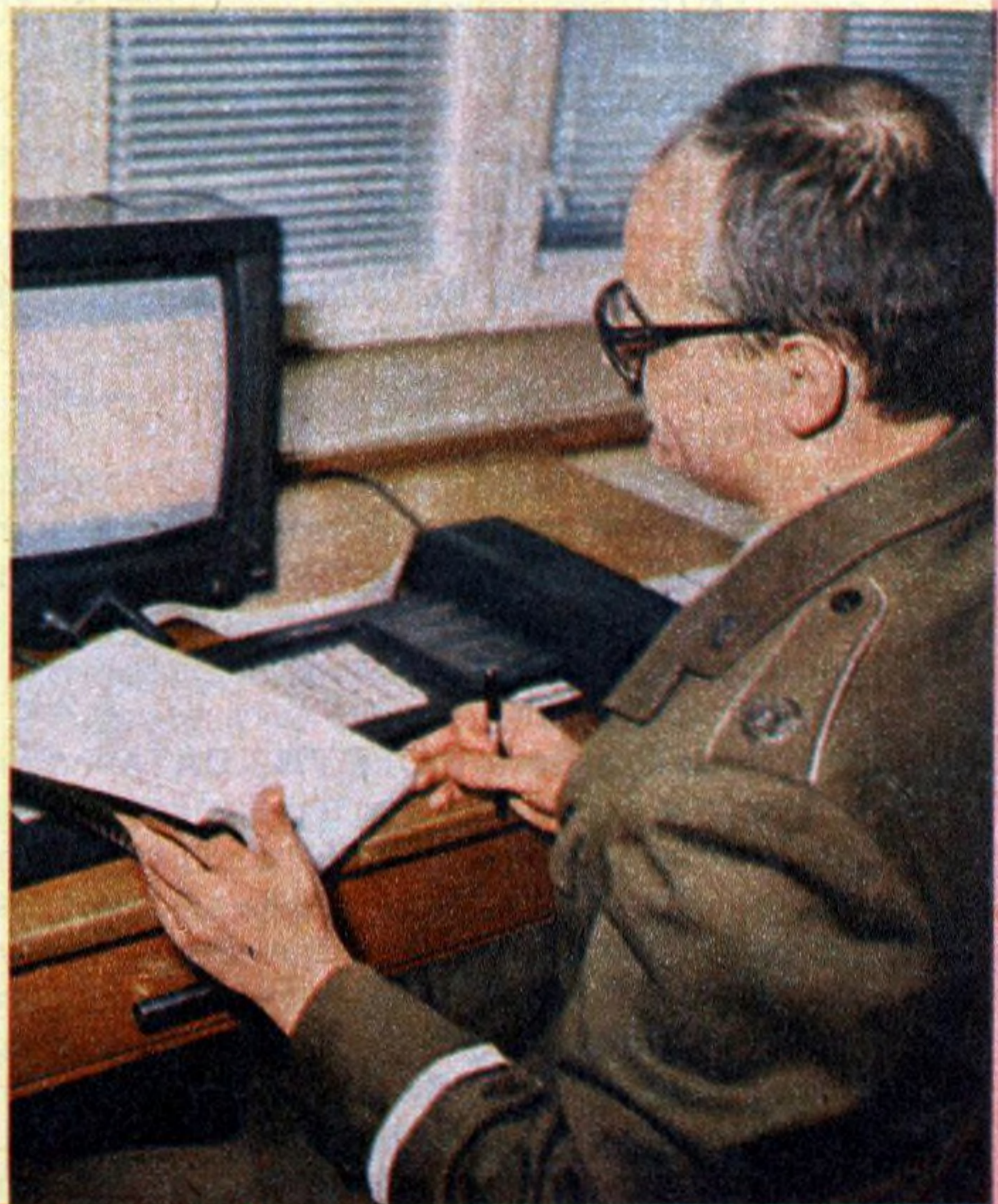
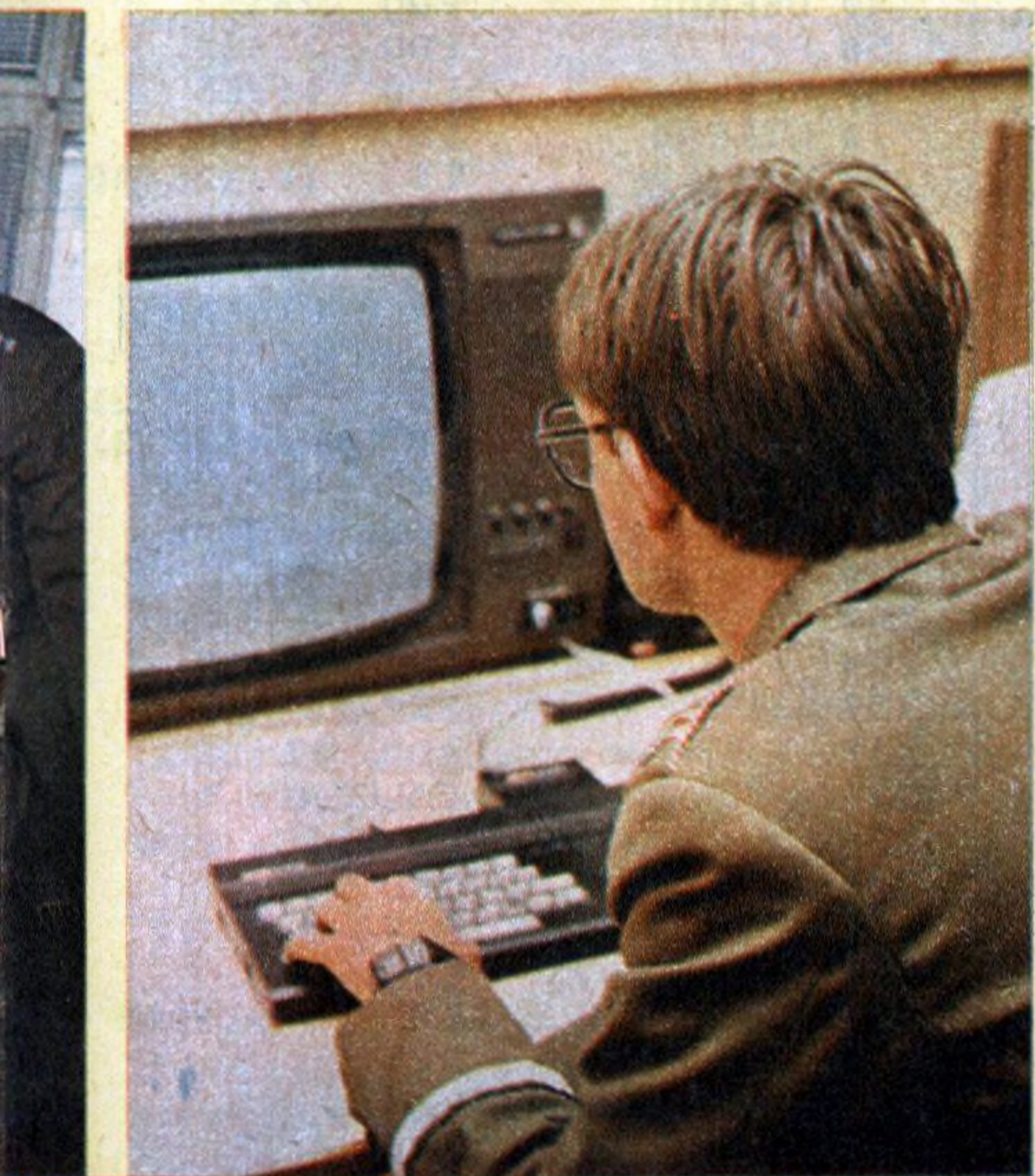


W KLUBIE





BIE, KTÓRY BĘDZIE



Zmienne systemowe ROM Spectrum

Wskaźniki obszarów programu i zmiennych BASICA

LISTSP	23563/4
S—TOP	23660/1
E—PPC	23625/6
DEFADD	23563/4
DATADD	23639/40
DEST	23629/30
STRLEN	23666/7

Obszar, w którym przechowywana jest lista instrukcji do interpretacji, nazywa się obszarem programu. Ograniczają go wskaźniki **PROG** i **VAR**. W odpowiednim formacie może on być przekazany na ekran (lub drukarkę) po komendzie **LIST** (LLIST). Ciekawym zjawiskiem w Spectrum jest listing automatyczny. Instrukcje pojawiają się na ekranie po naciśnięciu ENTER bez potrzeby wczytania LIST. S—TOP trzyma tzw. automatyczny numer linii. E—PPC — numer linii podlegającej edycji po naciśnięciu EDIT. Jeśli E—PPC jest większe od S—TOP, to linia z kurorem programu wystąpi na górze ekranu, w przeciwnym razie na dole. LISTSP przechowuje na czas procesu wskaźnik stosu. Bezpośrednia zmiana zawartości E—PPC np.:

POKE 23625,100

POKE 23626,0

wywoła podobny efekt, co **LIST** 100, lecz oczywiście ekran nie zareaguje przewijaniem w celu pokazania linii z kurorem. Listing automatyczny, choć wygodny dla raczkujących, wydaje się jednak zbędny, czego potwierdzeniem jest nieobecność tego zjawiska w innych mikrokomputerach.

Dwie funkcje **Basic'a** **DEF FN** i **DATA** możemy umieścić w dowolnym miejscu programu, pozwalając interpreterowi na samodzielne ich odszukanie. Lepiej więc lokować je na początku, co przyspiesza akcję. Są one związane ze zmiennymi systemowymi **DATADD** i **DEFADD**. W pierwszej z nich znajdziemy adres wskazujący miejsce, od którego będzie czytać dane instrukcja **READ**. Zwykle jest to adres przecinka separującego kolejne elementy w instrukcji **DATA** lub końca linii. Poniższy program zlicza dowolne dane umieszczone po słowie **DATA**:

```
100 DEF FN p(a)=PEEK a +256*PEEK (a+1)
110 LET elem=0
120 RESTORE 200: REM numer linii
130 LET ad= FN p(23639)+6
140 IF PEEK ad<>34 THEN GOTO 160
150 LET ad=ad+1: IF PEEK ad<>34 THEN G
OTO 150
160 LET ad=ad+1+5*( PEEK ad=14)
170 LET elem=elem+( PEEK ad=13 OR PEEK
ad=44)
180 IF PEEK ad=13 THEN PRINT elem: STOP
190 GOTO 140
200 DATA SIN PI,"S",-3.14e-9, INT 1.1,
"", "",PEEK 0,"koniec"
```

Niewątpliwym mankamentem Spectrum-Basic jest brak instrukcji pomocniczych np. do przenumerowywania linii, automatycznego ich generowania, niszczenia bloków programu itp., które użytkownikom innych komputerów ułatwiają życie. Niezbyt fortunate rozwiązanie klawiszy doprowadziło do ograniczenia liczby słów kluczowych, a co za tym idzie, do rezygnacji z wielu prostych w gruncie rzeczy funkcji.

Dla przykładu program kasowania bloków linii zajmuje tylko 45 bajtów. Wykorzystamy tu własność zmiennej **DEFADD** wskazującej na identyfikator pierwszego parametru wyrażenia **DEF FN** w momencie obliczania wartości funkcji. Określmy funkcję:

```
10 DEF FN d(p,o) = USR 62000
```

gdzie p jest pierwszą linią, zaś o ostatnią w bloku przeznaczonym do fizycznego zniszczenia. Oba parametry muszą być liczbami naturalnymi.

```
10 ; ** DELETE '85
20 ; GENS3M2 assembler
30          ORG 62000
40 DEFADD  EQU 23563
50 PPC      EQU 23621
60 RECL1    EQU #19E5
70 LINEAD   EQU #196E
80          LD IX,(DEFADD)
90          LD L,(IX+4)
100         LD H,(IX+5)
110         CALL LINEAD
120         PUSH HL
130         LD L,(IX+12)
140         LD H,(IX+13)
150         INC HL
160         CALL LINEAD
170         POP DE
180         AND A
190         SBC HL,DE
200         RET C
210         ADD HL,DE
220         CALL RECL1
230         LD BC,(PPC)
240         INC BC
250         BIT 7,B
260         RET Z
270         LD BC,10000
280         RET
```

Procedura jest dostosowana do współpracy z komendą **GO TO**, a więc kasowanie bloku powinno odbyć się po instrukcji np.

GO TO FN d (3000, 5000),

którą można użyć jako linię w programie, pozwalając nawet, aby skasowała samą siebie.

Opis tekstu źródłowego dla asemblera:

* linie 80—170 — rejestr IX przechwytyje wskazanie

DEFADD: numery linii podawane są do procedury **LINEAD**, która zwraca w **HL** ich adresy;

* linie 180—200 — sprawdzenie, czy pierwsza linia nie jest większa od ostatniej;

* linia 220 — fizyczne zniszczenie obszaru programu między adresami zawartymi w **HL** i **DE**;

* linie 230—260 — powrót, jeśli instrukcja odwołania jest linią w programie;

* linie 270—280 — powrót dla instrukcji wczytanej w trybie bezpośrednim (przyjme ona postać **GO TO 10000**).

Kod jest relokowalny

Następny problem dotyczy zapisu na nośniku tylko obszaru zmiennych BASIC. Najlepiej będzie pozbyć się obszaru programu, wykorzystując w tym celu określoną wcześniej funkcję FN d (p.o) lub krótszy kod:

10	ORG	62500
20	PROG	EQU 23635
30	VARs	EQU 23641
20	LD	HL, (VARs)
30	LD	DE, (PROG)
40	JP	6629

Po uruchomieniu tych 10 bajtów można zapisać na taśmie (dyskiecie) wszystkie użyte w programie zmienne Basic'a przez:

SAVE „zmienne”,

a w odpowiednim momencie dołączyć je do programu (choćby w celu wymuszenia innych warunków jego działania przez: **MERGE** „zmienne”).

Spectrum od strony implementacji ma tylko dwa typy zmiennych: numeryczne rzeczywiste i tekstowe. Tablica liczb naturalnych (do 65535), która zajmowałaby przykładowo w pamięci 5KB jest w większości (ok. 3KB) nie wykorzystana. Brakuje też rozkazu, który usuwałby ją z pamięci, tworząc miejsce dla innej. Tablicę np. m (10, 100, 20, 10) można wyzerować przez ponowną jej deklarację, zredukować do minimalnych rozmiarów przez DIM m(1), ale fizycznie wyrugować jej z pamięci nie można. Chyba że nauczy my komputer...

Najpierw jednak musimy się dowiedzieć od niego, jak przechowuje zmienne w pamięci. Szczegółowe dane zawiera podręcznik firmowy, ale brakuje tu kodowania identyfikatora zmiennej.

bit	7 6 5	4 3 2 1 0
znaczenie	typ	litera

Typów szczegółowych jest sześć, przy czym typ 1 nie istnieje:

- 2) zmienna tekstowa (string);
- 3) zmienna numeryczna 0 1-literowym identyfikatorze;
- 4) tablice numeryczne (numeric array);

5) zmienna numeryczna o wieloliterowym identyfikatorze;

6) tablica tekstowa (string array);

7) zmienna numeryczna 1-literowa sterująca pętlą FOR-NEXT;

Teraz możemy przedstawić program ERASE, który służy do niszczenia tablic, ale po pewnych przeróbkach „zajmie się” każdym typem zmiennej.

```
10 ; ** ERASE '86
20 ;GENS3M2  assembler
30          ORG 63000
40 DEFADD  EQU 23563
50 VARS    EQU 23627
60 RECL    EQU $19E5
70 PARERR  EQU $288B
80 VARERR  EQU $1C2E
90 NEXT    EQU $19B8
100 START  LD  HL, PARERR
110        PUSH HL
120        LD  IX, (DEFADD)
130        LD  L, (IX+4)
140        LD  H, (IX+5)
150        LD  C, (HL)
160        SET 7,C
170        RES 5,C
180        RES 6,C
190        LD  A, (IX+7)
200        AND A
210        RET NZ
220        LD  A, (IX+6)
230        DEC A
240        JR  Z, ERA
250        DEC A
260        RET NZ
270        INC HL
280        LD  A, (HL)
290        CP  "x"
300        RET NZ
310        SET 6,C
320 ERA    POP  HL
330        LD  HL, (VARs)
340 LOOP   PUSH BC
350        CALL NEXT
360        POP  BC
370        LD  A, (HL)
380        CP  128
390        JP  Z, VARERR
400        CP  C
410        EX  DE, HL
420        JP  Z, RECL
430        JR  LOOP
```

Opis:

* linie 100—110 — umieszczenie na stosie adresu procedury błędu w przypadkach niewłaściwych danych w wyrażeniu **DEF FN**;

* linie 120—320 — sprawdzenie poprawności podanych parametrów wejściowych oraz przygotowanie identyfikatora tablicy w rejestrze C (uwaga! w linii 290 powinien być w cudzysłowach znak dolara);

* linie 330—430 — przeszukanie obszaru zmiennych; w przypadku znalezienia identyfikatora zniszczenie tablicy, w przypadku przeciwnym wydruk raportu Variable not found;

Kod jest również relokowalny. Do korzystania z niego potrzebne jest zdefiniowanie funkcji:

10 **DEF FN** e\$(o\$) = (**USR** 63000)



Zwróćmy uwagę, że będziemy przekazywać do procedury **ERASE** parametr w postaci zmiennej znakowej, czyli nazwę tablicy. Oto przykłady niszczenia tablic g oraz g\$:

```
20 IF VAL (FN e$ („g")) THEN
30 IF VAL (FN e$ (gs")) THEN
```

DEST i **STRLEN** dotyczą zmiennych **BASIC**. Pierwsza wskazuje adres nazwy i wartości zmiennej w postaci zmiennoprzecinkowej (5 bajtów poprzedzonych znakiem kontrolnym 14), przy czym jeśli zmienna jest dopiero oznaczana, adres odnosi się do obszaru programu, jeśli zaś jej wartość jest modyfikowana, adres dotyczy obszaru zmiennych **BASICa**. **STRLEN** trzyma długość łańcucha tekstowego, jeśli w użyciu jest zmienna tekstowa. Zarówno **DEST**, jak **STRLEN** uczestniczą w tworzeniu +— bajtowej reprezentacji stringu, której używa kalkulator do obróbki tekstów w obszarze roboczym (**WORKSP**). W trakcie interpretacji **STRLEN** jest modyfikowane i na wyjściu oba bajty trzymają w zasadzie niezależne informacje. Pod 23666 znajdziemy pierwszą literę użytej zmiennej numerycznej lub nowo utworzonej zmiennej tekstowej. W przypadku przekształcania tej drugiej bajt zawiera liczbę znaków przed modyfikacją, co potwierdzi przykład:

```
10 LET a$="stringpierwotny"
20 PRINT LEN (a$)
30 LET a$=a$( TO 6)
40 PRINT PEEK 23666
```

Bajt 23667 podaje „utrudnioną” postać pierwszej litery wprowadzonej zmiennej, przy czym inny jest kod po komendzie **LET**, a inny po **INPUT** i **READ**. W wypadku wycinania tekstu (slicing) bajt ten zawiera zero. Za miesiąc o kanałach i strumieniach, ale bez wody...

Krzysztof MAMCARZ

W naszym komputerlandzie

Szaleństwo wybierania najpiękniejszej ogarnęło w końcu i komputerland. Sam Spektreus zapalił się najpóźniej, ale za to najgoręcej i z własnej pensji zafundował nagrodę, a nawet stanął na czele jury. Szarfę dla przyszłej Miss komputerlandu zrobiono z taśmy perforowanej.

Jury złożone z kompetentnych czynników postanowiło dopuścić do konkursu wszystkie komputerzyce, wychodząc z założenia, że wszystkie są urodziwe. Kiedy jednak do konkurencji zgłosiły się rzeczywiście wszystkie, jury znalazło się w niemałym kłopotcie, bo wybrać trzeba było jedną.

Wśród plejady gwiazd uwagę zwróciły dwie piękności — niemłoda już, lecz nieźle utrzymana Odra i młodzianka, nieśmiała Mazovia.

Po prezentacji jury udało się na naradę, jak na prawdziwy kolektyw przystało i ustaliło kryteria oceny.

Po pierwsze: Miss komputerlandu powinna reprezentować wysokie walory osobiste, przy czym wymagane są wymiary 96—84—96.

Po drugie — Miss powinna być inteligentna, lecz na tyle, by nie przytłaczała członków jury, aby nie popadli w kompleksy.

Po trzecie: Miss powinna się umieć ubrać, żeby miała się z czego rozbierać.

Wreszcie po czwarte — Miss nie musi za dobrze władać polskim, angielskim, ani żadnym innym ludzkim językiem, bo jury nie lubi gadatliwych komputerzyc, szczególnie, gdy się przechwalają — która ładniejsza.

Niestety po długiej naradzie jury nie przyznało tytułu Miss komputerlandu, gdyż żadna z kandydatek nie spełniała czwartego warunku. Żadna nie chciała poddać się zakneblowaniu na czas pozostawiania Miss...

Podglądał:
Eugeniusz MLECZAK

Spektrum

```
30 > BORDER 0: LET s=0: LET w=0
40 FOR i=0 TO 7: FOR p=0 TO i
50 BEEP .05,2*i: PRINT AT w+2,
s;i;"+";p; BRIGHT 1; INK i; PAPE
R p;"UUUUUUUUUUUUUU"
60 LET w=w+1: IF w=16 THEN LET
w=0: LET s=16
70 NEXT p: NEXT i
80 PAUSE 300: CLS : GO TO 30
90 SAVE "36 kolorow" LINE 10:
PRINT "Verify!": VERIFY "": GO
TO 10
```

J. Rubinowicz

PASCAL (5)

Jedną z podstawowych dziedzin zastosowań komputerów jest przetwarzanie danych. Wiąże się ono z gromadzeniem i przechowywaniem informacji o wielu obiektach, z których każdy może być opisywany wieloma cechami. W mniejszej części omówiono podstawowe właściwości tych konstrukcji języka PASCAL, które umożliwiają opis obiektów charakteryzowanych przez wiele cech, gromadzenie informacji o wielu obiektach (ich ilość nie musi być z góry określona) i przechowywanie jej w pamięciach zewnętrznych.

1. Rekordy

Rekordy umożliwiają opisywanie obiektów, które można scharakteryzować wieloma różnymi cechami. Przykładem takiego obiektu może być opis mikrokomputera dostępnego na rynku krajowym. Przyjmując dla uproszczenia, że zawiera on 20-znakową nazwę mikrokomputera, 12-znakową nazwę firmy produkującej mikrokomputery oraz minimalną cenę mikrokomputera (w tys. zł), „rynkowy” mikrokomputer można opisać następująco:

TYPE *opmikrok* = **RECORD**

```
    naznik: string [20];
    firma: string [12];
    cenamin: integer
END;
```

gdzie:

opmikrok — nazwa definiowanego rekordu,

naznik, *firma*, *cenamin* — nazwy składowych rekordu zwanych polami

RECORD—END — para słów kluczowych, oznaczająca definicję rekordu,

string — typ standardowy, umożliwiający definiowanie łańcuchów znaków (stałe typu „string” są już Czytelnikowi znane z instrukcji **WRITE**).

Powyższa definicja umożliwia deklarowanie zmiennych np. według formuły

VAR r1, r2,...rn: *opmikrok*;
z których każda opisuje inny mikrokomputer.

Sposoby odwoływania się do pól zmiennej rekordowej przez tzw. nazwy kwalifikowane wyjaśnia przykład 14, w którym poszczególnym polom nadaje się odpowiednie wartości.

Przykład 14

```
PROGRAM prog14;
  (zakładanie zbioru KOMPUTER.DTA
   na dyskietce)
TYPE
  opmikrok=RECORD (opis mikrokomputera )
    nazmk:string[20];
    firma:string[12];
    cenamin:integer
  END;
  zbopmk=FILE OF opmikrok;
    (definicja pliku)
VAR
  zbior:zbopmk; (deklaracja pliku)
  rek:opmikrok;
    (zmienna opisująca mikrokomputer)
  ilop, (liczba wprowadzanych opisów)
  i: integer;
BEGIN
  READLN(ilop);
  ASSIGN(zbior,'komputer.dta');
    (dowiązanie fizycznej reprezentacji
     pliku-zbioru o nazwie KOMPUTER.DTA)
  REWRITE(zbior);
    (przygotowanie pliku do zapisu)
  FOR i:=1 TO ilop DO
    WITH rek DO
      BEGIN
        READLN(nazmk,firma,cenamin);
          (wczytanie opisu mikrokomputera
           z monitora)
        WRITE(zbior,rek)
          (zapisanie opisu mikrokomputera
           do pliku)
      END;
  CLOSE(zbior) (zamknięcie pliku)
END.
```

Po wykonaniu przedstawionej sekwencji wykonanie instrukcji

X:=rek.firma;

gdzie „X” jest typu „string [12]” spowoduje przypisanie zmiennej X wartości „SINCLAIR”. Zauważmy, że łańcuch znaków o długości krótszej niż podana w definicji jest uzupełniany spacjami z prawej strony.

Przedstawiony sposób odwoływania się do pól zmiennej rekordowej może być uciążliwy (zwłaszcza, gdy jest ich dużo lub są to pola typu strukturalnego), gdyż każde pole musi być poprzedzone nazwą zmiennej.

Wygodniejszy dostęp do pól zmiennej rekordowej umożliwia instrukcja

WITH, zwana instrukcją wiążącą lub instrukcją dostępu. Pozwala ona na odwoływanie się pól zmiennej rekordowej bezpośrednio przez nazwy pól, przy jednokrotnym podaniu nazwy zmiennej rekordowej. Wykorzystując instrukcję WITH fragment programu z przykładu 14 można zmodyfikować następująco:

Przykład 15

2. Pliki

Przechowywanie programów w pamięci zewnętrznej pozwala na ich wielokrotne wykorzystanie. Również dane, które podlegają przetwarzaniu co pewien czas (nieraz przez różne programy), muszą być przechowywane w pamięci zewnętrznej. Przykładem takich danych może być zarówno wykaz odbiorców energii elektrycznej, dla których nalicza się okresowo wysokość opłat oraz program źródłowy, przetwarzany w zależności od potrzeb przez edytor lub kompilator.

W PASCALu istnieje możliwość korzystania z pamięci zewnętrznej przez tzw. pliki. Plik, podobnie jak tablica jest ciągiem elementów tego samego typu.

Od tablicy różni się metodą dostępu do elementów (w danej chwili może być dostępny tylko jeden element pliku oraz typu) i tym, że liczba elementów nie musi być z góry określona i może się zmieniać w trakcie wykonywania programu. Po zakończeniu działania programu informacje zgromadzone w pliku pozostają w pamięci zewnętrznej i mogą być wykorzystane przez ten sam lub inne programy. Z punktu widzenia fizycznej reprezentacji plik jest wyróżnionym obszarem pamięci zewnętrznej — zbiorem o zadanej nazwie.

Sposób zakładania na dyskietce zbioru podanych wcześniej opisów mikrokomputerowych ilustruje przykład 16.

Przykład 16

```
PROCEDURE wydruk(VAR zb:zbopmk;
  naz:nazwz);
  (procedura wyprowadzająca na
   monitor zawartosc zbioru
   o zadanej nazwie)
VAR
  rek:opmikrok;
BEGIN
  ASSIGN(zb,naz);
  RESET(zb);
    (przygotowanie pliku do
     odczytu)
  WHILE NOT EOF(zb) DO
    BEGIN
      READ(zb,rek);
        (wczytanie kolejnego opisu
         mikrokomputera z pliku)
      WITH rek DO
        WRITELN(' ',nazmk,' ',firma,' ',
          cenamin)
    END;
  CLOSE(zb)
END;
```



W przedstawionym programie wykorzystano procedury standardowe **ASSIGN**, **REWRITE**, **WRITE** i **CLOSE**.

ASSIGN dowiązuje do pliku fizyczny zbiór o podanej nazwie. Po wyniku procedury **ASSIGN** wszystkie operacje na pliku są równoważne działaniu na jego fizycznej reprezentacji. Nazwa zbioru jest łańcuchem znaków o postaci przyjętej dla zbiorów roboczych edytora.

REWRITE tworzy na dyskietce fizyczny zbiór o nazwie podanej w procedurze **ASSIGN** i udostępnia do zapisu pierwszy element pliku. Jeżeli zbiór o podanej nazwie już istnieje, to jest on kasowany. Po wykonaniu **REWRITE** plik jest pusty, tzn. nie zawiera żadnych elementów.

WRITE — dopisuje do pliku element umieszczony w zmiennej, podanej jako parametr procedury **WRITE**. Po zapisaniu danego elementu udostępniany jest kolejny element pliku.

CLOSE — zamyka plik. Procedura ta powinna być wykonana po zakończeniu działań na pliku.

Wprowadzając dane do programu należy pamiętać, że łańcuchy znaków muszą być łańcuchami o długości podanej w opisie rekordu (spacja też jest znakiem) oraz że kolejne łańcuchy nie są oddzielone od siebie żadnym separatorem. Program wykonano dla następujących danych:

```
ZX_Spectrum_Plus.....SINCLAIR.....110
ZX_Spectrum_48kB.....SINCLAIR.....75
130XE.....ATARI.....130
C-128.....COMMODORE....285
gdzie: _ — oznacza spację.
```

Zawartość założonego zbioru można wyprowadzić na monitor wywołując procedurę „wydruk” podaną w przykładzie 17. W wywołaniu procedury parametr „” należy zastąpić łańcuchem znaków komputer dta:

Przykład 17

```
PROGRAM prog15;
  (wyprowadzenie zawartosci zbioru
   KOMPUTER.DTA na monitor)
TYPE
  opmikrok=RECORD
    nazak:string[20];
    firma:string[12];
    cenamin:integer
  END;
  zbopak=FILE OF opmikrok;
  nazwazb:string[12];
VAR
  zbior:zbopak;
  nazwa:nazwazb;
  (nazwa wyprowadzanego zbioru)
PROCEDURE wydruk(VAR zb:zbopak;
  naz:nazwazb);
  (procedura wyprowadzajaca na
   monitor zawartosc zbioru
   o zadanej nazwie)
VAR
  rek:opmikrok;
BEGIN
  .
  .
  .
END;
PROCEDURE podkresl;
BEGIN
  WRITELN('-----',
  '-----');
END;
BEGIN
  READ(nazwa);
  CLASCR; (wykasowanie zawartosci ekranu)
  WRITELN;
  podkresl;
  WRITELN(' :4, 'NAZWA KOMPUTERA', ' :6,
  ' FIRMA', ' :7, 'CENA$');
  podkresl;
  wydruk(zbior, nazwa);
  podkresl;
  WRITELN(' $ cena minimalna w tys zl')
END.
```

W procedurze „wydruk” poza już omówionymi procedurami standardowymi, wykorzystano procedury **RESET** i **READ**.

RESET udostępnia do przetwarzania (odczyt lub zapis) pierwszy element istniejącego pliku. Jeżeli brak jest fizycznej reprezentacji pliku, to wystąpi błąd wykonania.

READ odczytuje element pliku i umieszcza go w podanej zmiennej określonego typu. Po odczytaniu danego elementu, do przetwarzania zostaje udostępniony kolejny element pliku. Sprawdzenie czy wystąpił koniec pliku wykonuje funkcja standardowa EOF.

Program wyprowadzania zawartości założonego zbioru zamieszczono w przykładzie 17.

Przykład 17

Po wykonaniu programu „prog15” uzyskano następujące wyniki:

NAZWA KOMPUTERA	FIRMA	CENA\$
ZX Spectrum Plus	SINCLAIR	110
ZX Spectrum 48kB	SINCLAIR	75
130XE	ATARI	130
C-128	COMMODORE	285

\$ cena minimalna w tys zl

Zawartość zbioru może być zmieniana m.in. przez dopisanie nowych elementów lub skasowanie już istniejących, wybranych elementów.

Program dopisywania nowych elementów na koniec założonego zbioru (tylko takie jest dopuszczalne) zamieszczono w przykładzie 18.

Przykład 18

```
PROGRAM prog16;
  (dopisanie rekordow na koniec
   zbioru KOMPUTER.DTA)
TYPE
  opmikrok=RECORD
    nazak:string[20];
    firma:string[12];
    cenamin:integer
  END;
  zbopak=FILE OF opmikrok;
VAR
  zbior:zbopak;
  rek:opmikrok;
  ilop,
  i: integer;
BEGIN
  READLN(ilop);
  ASSIGN(zbior, 'komputer.dta');
  RESET(zbior);
  SEEK(zbior, FILESIZE(zbior));
  (przewinięcie pliku na koniec)
  FOR i:=1 TO ilop DO
  WITH rek DO
  BEGIN
    READLN(nazak, firma, cenamin);
    WRITE(zbior, rek)
  END;
  CLOSE(zbior)
END.
```

Przewinięcie pliku na koniec umożliwia procedura standardowa **SEEK** (parametr **FILESIZE** jest funkcją standardową obliczającą ilość elementów istniejącego pliku).

Po dopisaniu dla założonego zbioru elementu

```
800 XL.....ATARI...70
zawartość zbioru była następująca
(ponownie wykonano program „prog
15”)
```

NAZWA KOMPUTERA	FIRMA	CENA\$
ZX Spectrum Plus	SINCLAIR	110
ZX Spectrum 48kB	SINCLAIR	75
130XE	ATARI	130
C-128	COMMODORE	285
800XL	ATARI	70

\$ cena minimalna w tys zl
Prosty program kasowania wybranych elementów założonego zbioru zamieszczono w przykładzie 19.

Przykład 19

```
PROGRAM prog17;
  {kasowanie rekordow ze zbioru
  KOMPUTER.DTA}
TYPE
  opmikrok=RECORD
    nazmk:string[20];
    firma:string[12];
    cenamin:integer
  END;
  zbopmk=FILE OF opmikrok;
VAR
  zbior:zbopmk;
  nazwamk:string[20];
  ilopkas,
  i: integer;
PROCEDURE kasrek;
VAR
  zbpom:zbopmk;
  rek:opmikrok;
BEGIN
  ASSIGN(zbior,'komputer.dta');
  RESET(zbior);
  ASSIGN(zbpom,'pom.dta');
  REWRITE(zbpom);
  WHILE NOT EOF(zbior) DO
  BEGIN
    READ(zbior,rek);
    IF rek.nazmk<>nazwamk THEN
      WRITE(zbpom,rek)
  END;
  CLOSE(zbior);
  CLOSE(zbpom);
  ERASE(zbior);
  {skasowanie zbioru KOMPUTER.DTA}
  RENAME(zbpom,'komputer.dta')
  {zmiana nazwy zbioru POM.DTA na
  KOMPUTER.DTA}
END;
BEGIN
  READLN(ilopkas);
  FOR i:=1 TO ilopkas DO
  BEGIN
    READLN(nazwamk);
    kasrek
  END
END.
```

Przyjęto tutaj, że skasowanie nawet jednego elementu z pliku wymaga przejścia całego pliku. Ponieważ nie jest możliwe skasowanie elementu „ze środka” pliku, wykorzystuje się plik pomocniczy (pom, dla).

Elementy danego pliku są czytane kolejno i te z nich, które nie podlegają kasowaniu są zapisywane do pliku pomocniczego. Po przeglądnięciu całego pliku jest on kasowany (ERASE) a plik pomocniczy zostaje przemianowany na poprzednią nazwę pliku (RENAME). Identyfikatorem określającym, który element pliku ma być kasowany jest tutaj nazwa mikrokomputera.

Po skasowaniu opisu mikrokomputera 130 XE zawartość zbioru komputer.dta była następująca:

NAZWA KOMPUTERA	FIRMA	CENA*
ZX Spectrum Plus	SINCLAIR	110
ZX Spectrum 48kB	SINCLAIR	75
C-128	COMMODORE	285
800XL	ATARI	70

* cena minimalna w tys zł

3. Zadania

Zadanie 12

Napisać program zakładania zbioru opisów pozycji domowej biblioteczki.

Zadanie 13

Napisać program porządkowania założonego zbioru alfabetycznie według nazwiska autora.

Uwaga: porządkowanie wykonać w pamięci operacyjnej. Można zmodyfikować poznaną już procedurę porządkowania ciągu.

Zadanie 14

Napisać program umożliwiający aktualizację zbioru opisów pozycji domowej biblioteczki uwzględniający dopisywanie lub kasowanie opisów.

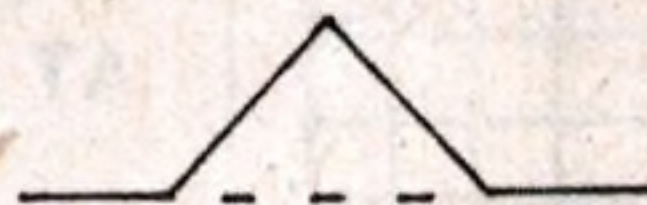
Zadanie 15

Napisz program udostępniający informacje o danej pozycji domowej biblioteczki.

S. ROZMUS

PŁATEK KOCH'a

Każdy bok trójkąta równobocznego jest podzielony na trzy równe części i zamieniany na następującą łamaną:



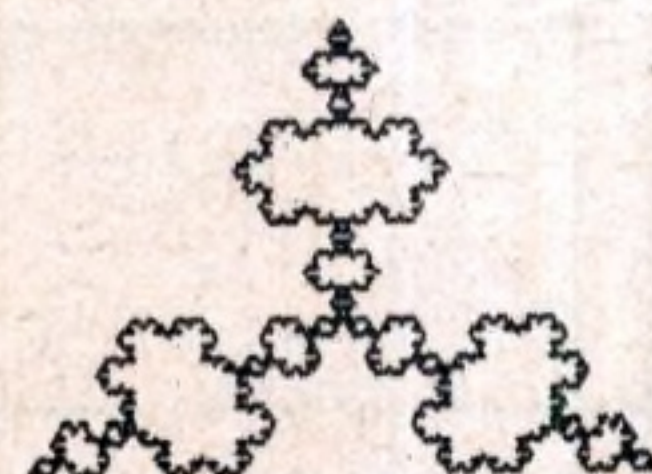
Każdy z odcinków łamanej ulega takiemu samemu przekształceniu. Operacji dokonuje procedura ZAMIANA metodą rekurencyjną. Proces kończy się w momencie,

gdy długość boku jest odpowiednio mała (stała MINBOK).

Proponujemy również modyfikację procedury polegającą na wprowadzeniu zmiennego kąta. Zastąp parametr aktualny przy wywołaniu procedur LEFT i RIGHT zmienną typu REAL (zastanów się dlaczego REAL, a nie INTEGER), którą zadeklaruj i dopisz instrukcje jej wprowadzania.

Płatek Koch'a uzyskujemy dla kąta = 60 stopni, a modyfikacja pozwala otrzymać inne niezwykle wzory.

```
10 PROGRAM platek;
20
30 {*****}
40 (** PROGRAM rysuje platek **)
50 (** sniegu Kocha **)
60 (** **)
70 (** NAUKOWE KOLO PODCHORAZYCH **)
80 (** WSOWL 1987 **)
90 {*****}
100
110 CONST minbok=5;
120 VAR
130 bok:real;
140 i:integer;
150
160 {*****}
170 (** tu nalezy wstawic pakiet **)
180 (** procedur grafiki zolwia **)
190 {*****}
200
210 PROCEDURE zamiana (bok:real);
220 BEGIN
230 IF bok>minbok THEN
240 BEGIN
250 zamiana(bok/3);
260 left(60);
270 zamiana(bok/3);
280 right(120);
290 zamiana(bok/3);
300 left(60);
310 zamiana(bok/3);
320 END
330 ELSE fwd(bok)
340 END; (* zamiana *)
350
360 BEGIN
370 write('bok=');
380 read(bok);
390 page;
400 turtle;
410 left(60);
420 xcor:=320-bok*sqrt(3)/3;
430 ycor:=200-bok*sqrt(3)/6;
440 FOR i:=1 TO 3 DO
450 BEGIN
460 zamiana(bok);
470 right(120);
480 END;
490 END.
```



Podstawy programowania

Kontynuujemy wykład z programowania w języku BASIC na mikrokomputer ZX Spectrum. W wykładzie tym podajemy podstawowe wiadomości o grafice i kolorach wraz z omówieniem odpowiednich instrukcji BASICa dla ZX Spectrum.

Wykład siódmy:

grafiki ZX Spectrum; instrukcje PRINT AT n, l; PRINT TAB 1;
grafika: instrukcje PLOT, DRAW, CIRCLE

W wykładzie pierwszym omawialiśmy ekran telewizyjny (gdy ZX Spectrum podłączone jest do telewizora) lub ekran monitora (gdy podłączone jest do monitora). Przyjmijmy (jak w poprzednich wykładach) nazwę ekran telewizyjny. Powtórzmy to, co było powiedziane w pierwszym wykładzie. Ekran telewizyjny zawiera 24 linie po 32 znaki i jest podzielony na dwie części w poziomie i w pionie. Linie i znaki numerowane są od zera. Górna część ekranu zawiera 22 linie (0—21), wyświetlany jest na niej program albo wyniki i wykorzystywana jest w grafice. Dolna część ekranu jest wykorzystywana do wprowadzania rozkazów, wierszy programu i danych wejściowych, a także do wyświetlania komunikatów (zawiera ona dwie linie o numerach 22 i 23).

Narysujmy teraz schemat naszego ekranu w postaci:

	0	1	2	29	30	31
0							
1							
2							
:							
:							
:							
:							
:							
:							
20							
21							
22							
23							

Położenie każdego znaku na ekranie określamy przez podanie numeru linii (wiersza) i kolumny (innymi słowy współrzędne znaku).

Zajmiemy się teraz rysowaniem obrazków, wykorzystując symbole graficzne („grafiki”) ZX Spectrum. **Symbole te wprowadzamy z klawiatury po uprzednim przejściu na tryb G** (kursor G — otrzymujemy naciskając jednocześnie klawisze CAPS SHIFT i 9). Naciśnięcie (gdy jest kursor G) klawiszy 1—8 spowoduje wypisanie na ekranie odpowiedniego symbolu graficznego w takiej postaci, w jakiej są narysowane na klawiszach, a z klawiszem SYMBOL SHIFT (sh) — odwrotnie tzn. kolor czarny stanie się biały, a kolor biały czarny. Poniżej podajemy szesnaście symboli graficznych (litera G — oznacza kursor G, sh — SYMBOL SHIFT):

symbol	sposób użycia	symbol	sposób użycia
□	<u>G</u> 8	■	<u>G</u> sh 8
◻	<u>G</u> 1	◼	<u>G</u> sh 1
◻	<u>G</u> 2	◼	<u>G</u> sh 2
◻	<u>G</u> 3	◼	<u>G</u> sh 3
◻	<u>G</u> 4	◼	<u>G</u> sh 4
◻	<u>G</u> 5	◼	<u>G</u> sh 5
◻	<u>G</u> 6	◼	<u>G</u> sh 6
◻	<u>G</u> 7	◼	<u>G</u> sh 7

Każdy znak zajmuje jedną pozycję na ekranie. Do wpisania takiego znaku w dowolne miejsce ekranu używana jest najczęściej instrukcja **PRINT** połączona z **AT**, postaci:

PRINT AT n, l; „znak”

gdzie: *n* — numer linii (0÷21)
l — numer kolumny (÷31)
znak — mogą to być znaki, teksty, a bez cudzysłowu nazwy zmiennych prostych, wyrażenia arytmetyczne itp.

AT n, l — przesuwa pozycję wydruku na wskazane miejsce ekranu (o współrzędnych (*n, l*)).

Po **AT n, l**; można pisać wszystko to, co było już omawiane w PRINT (średnik jest obowiązkowy).

Przykład 6.1.

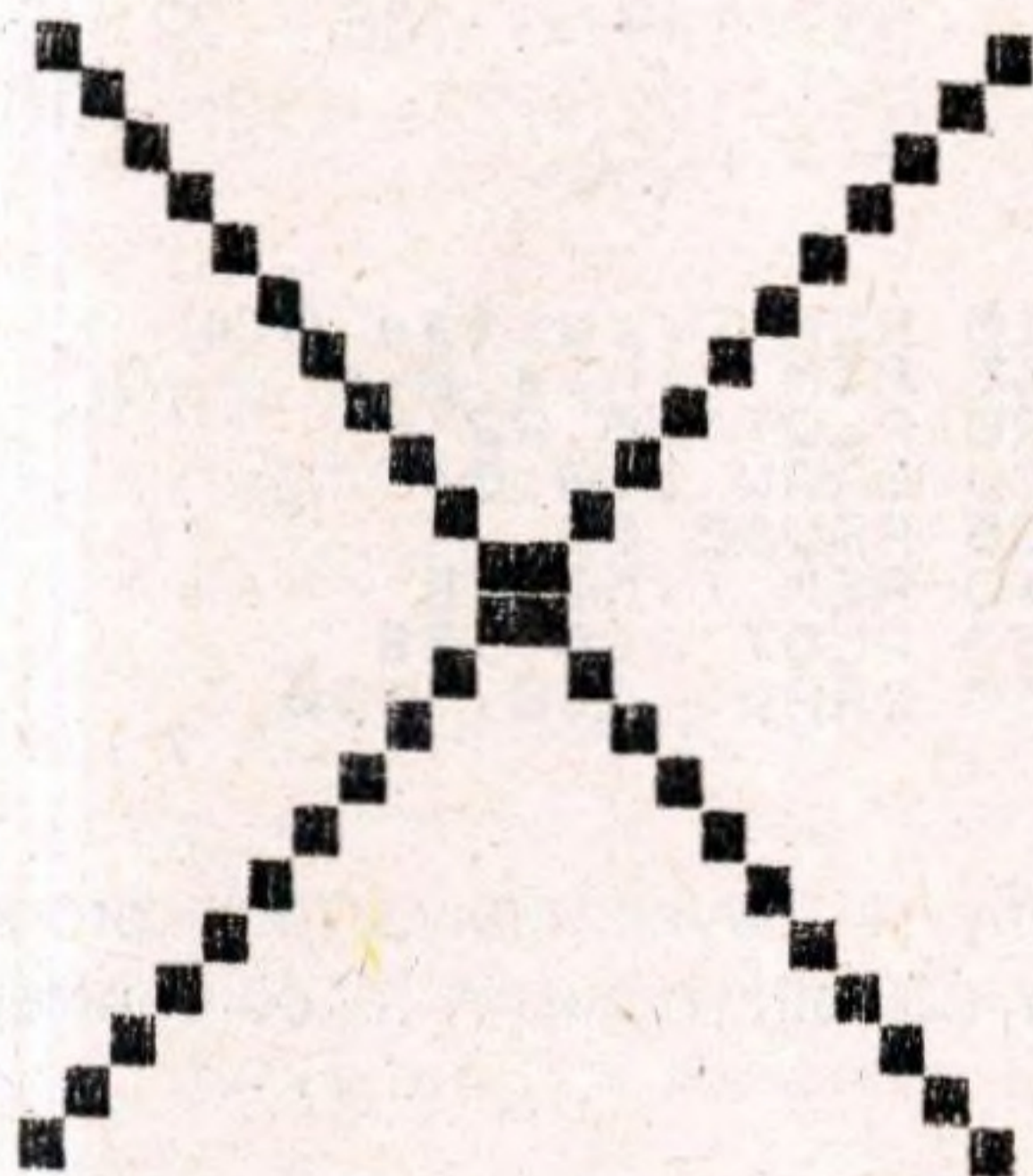
W kwadracie o boku równym 22 znaki narysować przekątne używając symbolu „■”.

```

10 REM Przykład 6.1
20 FOR i=0 TO 21
30 FOR j=0 TO 21
40 IF i=j THEN PRINT AT i,j;"■"
45 IF i=21-j THEN PRINT AT i,j;"■"
50 NEXT j
60 NEXT i

```

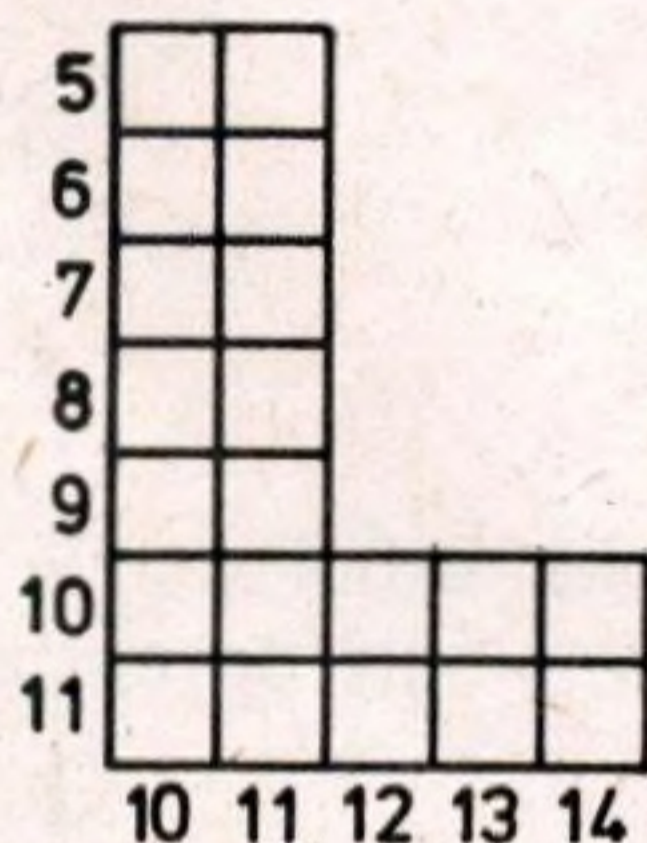
Wprowadźmy i wykonajmy ten program. Otrzymane wyniki na ekranie są postaci:



Przykład 6.2.

Zaczynając od 5 wiersza i 10 kolumny wypisać dużą literę L, używając symbolu „■”.

a) w pierwszym kroku „planujemy” obraz:



b) program:

```

10 REM Przykład 6.2
20 REM rysowanie w liniach 5-9
30 FOR i=5 TO 9
40 PRINT AT i,10;"■"
50 NEXT i
60 REM rysowanie w liniach 10,
11
70 FOR j=10 TO 14
80 PRINT AT 10,j;"■"
90 PRINT AT 11,j;"■"
100 NEXT j

```

Wprowadźmy i wykonajmy ten program. Otrzymane wyniki na ekranie są postaci:



Przykład 6.3.

Dany jest program:

```

10 REM Przykład 6.3
20 PRINT AT 7,15;"■"
30 PRINT AT 8,15;"■"
40 PRINT AT 9,15;"■"
50 PRINT AT 10,15;"■"
60 PRINT AT 11,15;"■"
70 PRINT AT 12,15;"■"

```

Wprowadzając ten program, używamy symboli graficznych: □, ■, ▣. Wprowadźmy i wykonajmy ten program. Otrzymane wyniki na ekranie są postaci:



Nauczyliśmy się przesuwać wydruk we wskazane miejsce ekranu, używając AT n, l. Za pomocą AT n, l umiemy np. pisać w dolnej części ekranu, a potem w środku albo w górnej części ekranu, nawet w miejscach uprzednio zapisanych.

W instrukcji **PRINT** można zastosować **TAB /** (/ — nr kolumny) **TAB /** — dopisuje spacje, tak aby przesunąć miejsce wydruku do wymienionej kolumny, zostając w tej samej linii. Zdarza się, że trzeba przesunąć się do następnej linii i także do wskazanej kolumny. Komputer zmniejsza numer kolumny „modulo 32” (dzieli przez 32 i bierze resztę), więc TAB 33 oznacza to samo co TAB 1 (w następnej linii). Np. wprowadźmy i wykonajmy program ilustrujący działanie **TAB /**:

```

10 FOR n = 0 TO 20
20 PRINT TAB 6 * n, n;
30 NEXT n

```

Uwaga: używając w instrukcji PRINT: AT n, l lub TAB l zawsze piszemy średnik, tzn. PRINT AT n, l; lub PRINT TAB 1;



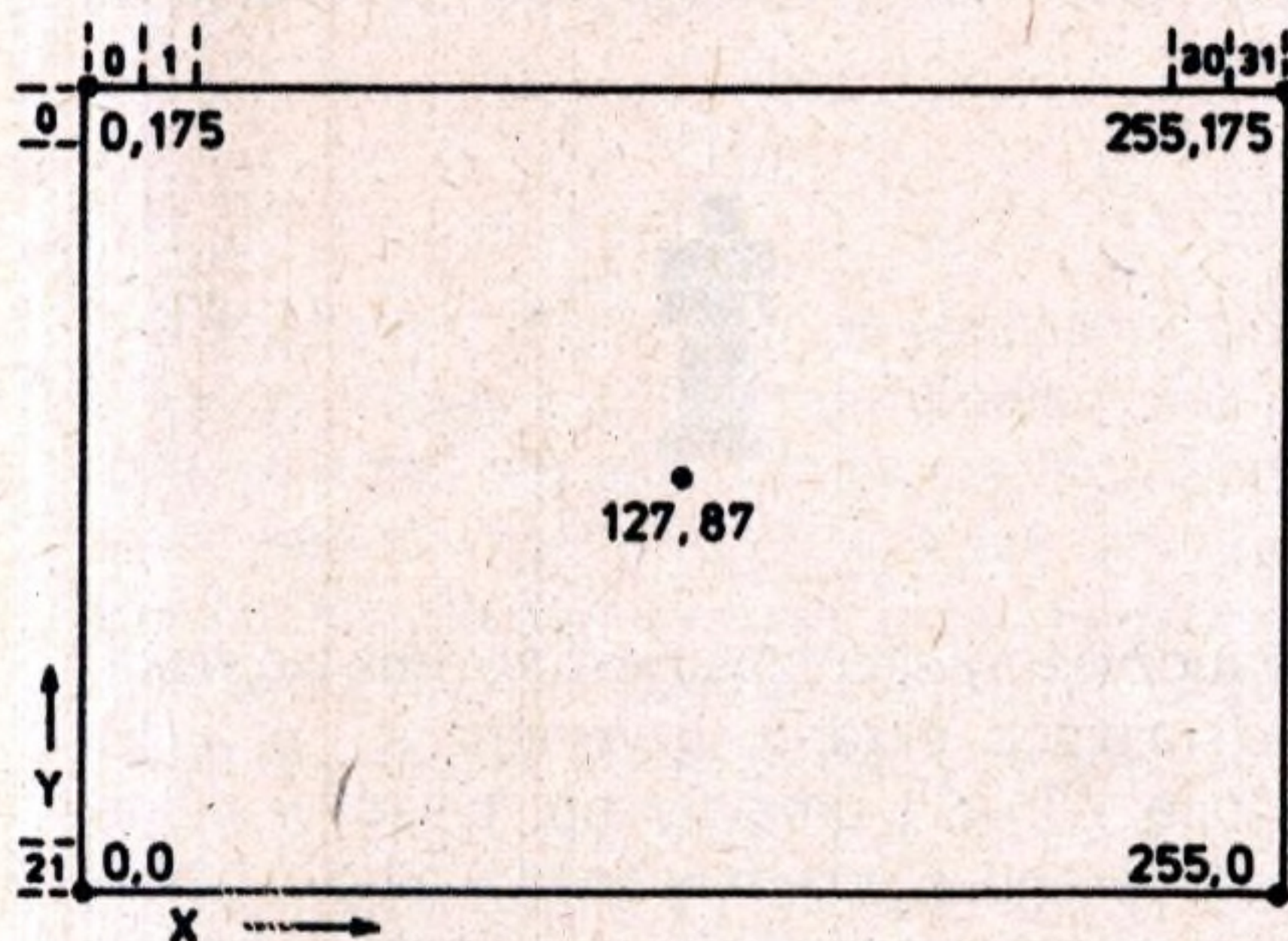
► Sądzę, że podane zasady i przykłady powinny wystarczyć do samodzielnej zabawy w „grafikę” ZX Spectrum.

Teraz zajmiemy się rysowaniem obrazków na ZX Spectrum.

Część ekranu, którą możemy używać do tego celu, zawiera 22 linie i 32 kolumny, czyli $22 \times 32 = 704$ pozycje znakowe. Każda pozycja znakowa utworzona jest przez kwadracik 8 na 8 punktów. Punkty te nazwano pixelami (mamy $256 \times 176 = 45\,056$ pixeli). Pixel jest określony przez dwie współrzędne:

- pierwszą x, która określa odległość pixela od lewego dolnego rogu ekranu (w prawo),
- drugą y, która określa wysokość pixela nad linią 22.

Na rysunku poniżej podane są współrzędne rogów i środka ekranu.



Współrzędne pixeli są inne niż linii i kolumn w AT. Punkty na ekranie rysujemy instrukcją postaci:

PLOT x, y $(0 \leq x \leq 255, 0 \leq y \leq 175)$

Punkty rysowane są w kolorze atramentu.

Linie proste na ekranie rysujemy instrukcją postaci:

DRAW x, y $(-255 \leq x \leq 255, -175 \leq y \leq 175)$

Linia rysowana przez instrukcję **DRAW** zaczyna się w miejscu, gdzie ostatnio skończyło się rysowanie instrukcji **PLOT**, **DRAW** lub **CIRCLE** (zwane też pozycją PLOT-u), kończy się w miejscu odległym o x pixeli w prawo i y pixeli w górę. Każda instrukcja rysowania przesuwają pozycję PLOT-u w miejsce, gdzie skończyło się rysowanie. Instrukcja **DRAW** nie określa punktu początkowego, tylko kierunek i długość.

Przykład 6.4.

Narysować dwie linie:

linię **A**: punkt początkowy o współrzędnych (56, 22)

punkt końcowy o współrzędnych (205, 106)

linię **B**: punkt początkowy o współrzędnych (226, 152)

punkt końcowy o współrzędnych (34, 22)

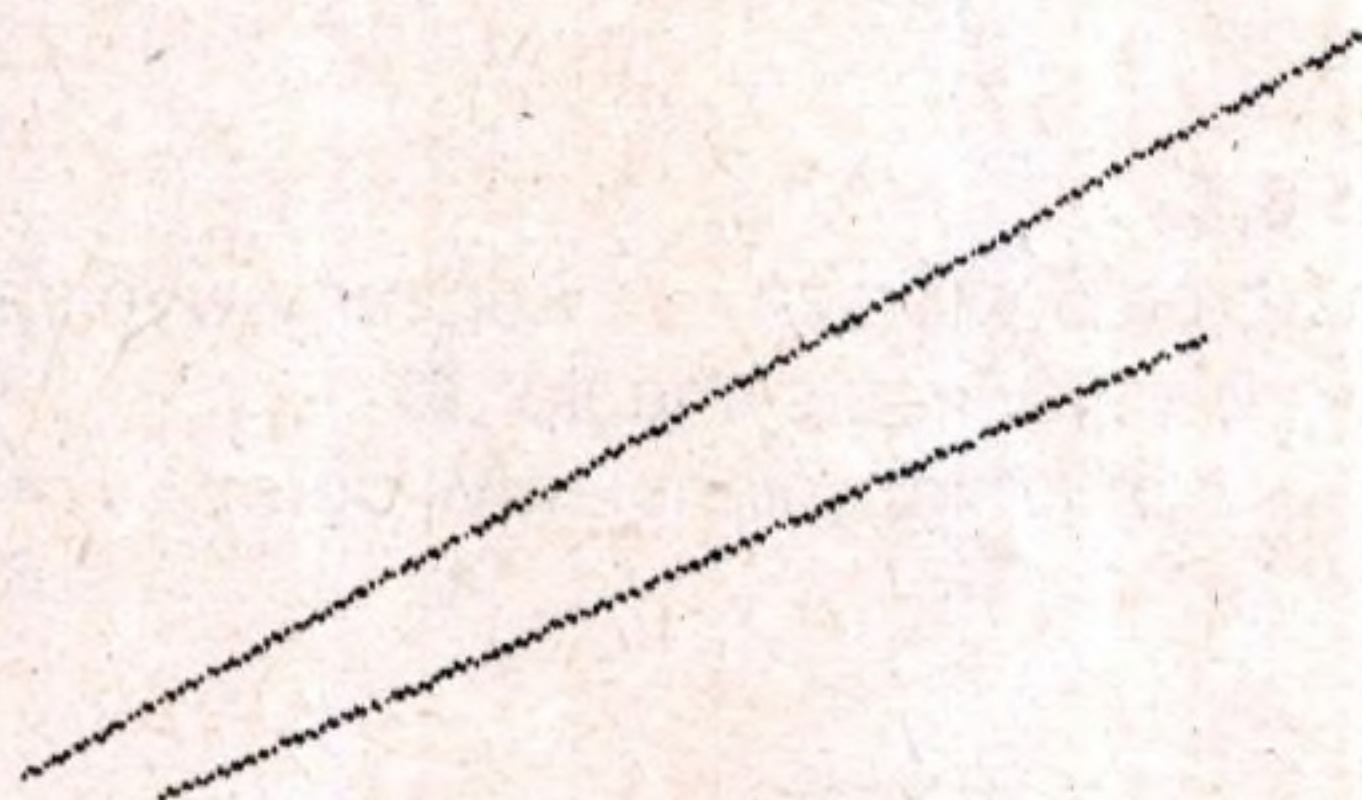
obliczamy współrzędne instrukcji **DRAW**:

linia **A**: $205 - 56 = 149$ (x) $106 - 22 = 84$ (y)

linia **B**: $34 - 226 = -192$ (x) $22 - 152 = -130$ (y)

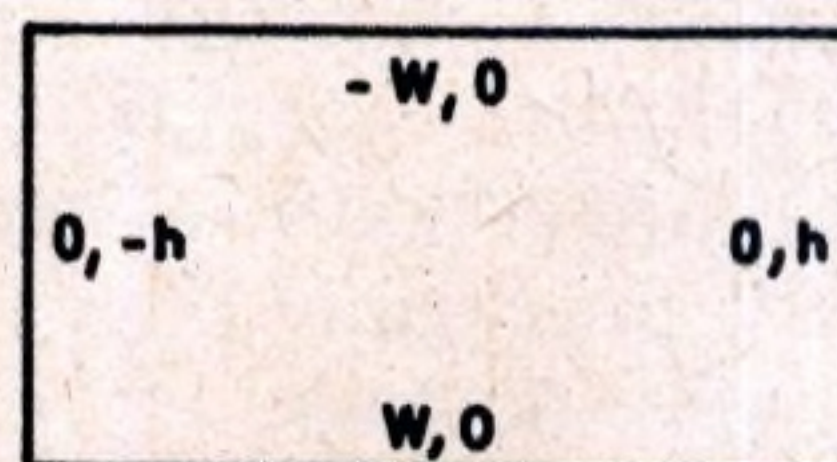
```
10 REM Przykład 6.4
15 REM Linia A
20 PLOT 56,22
30 DRAW 149,84
35 PAUSE 50
40 REM Linia B
50 PLOT 226,152
60 DRAW -192,-130
```

Wprowadźmy i wykonajmy ten program. Sprawdźmy, czy otrzymaliśmy to, co poniżej:



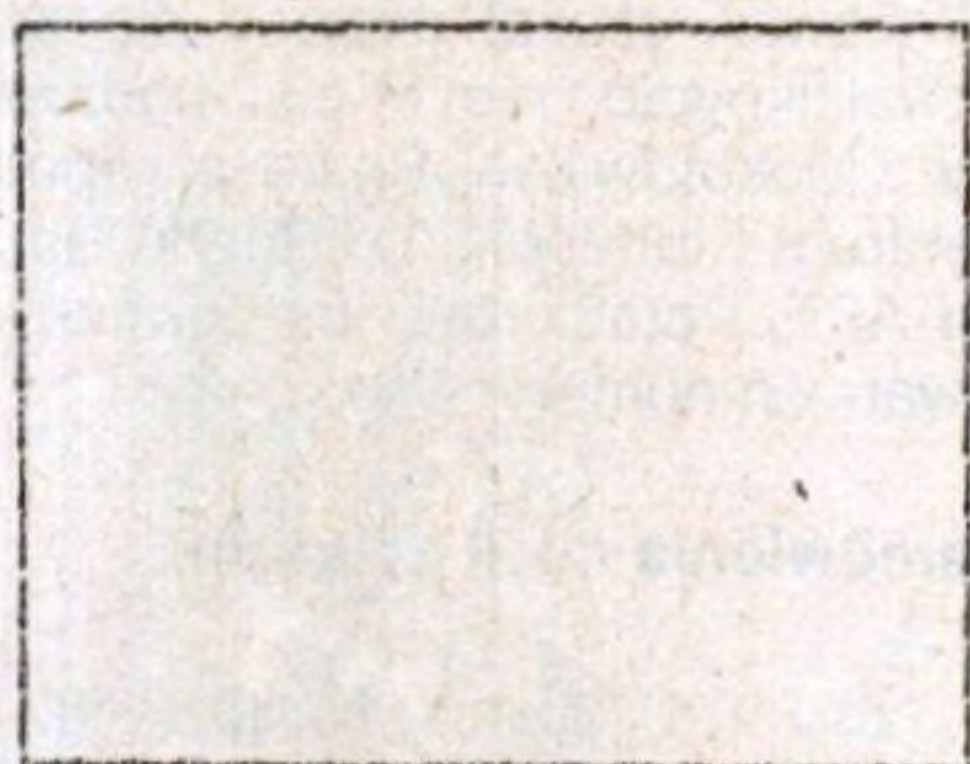
Przykład 6.5.

Narysować prostokąt o bokach: w (równoległy do osi x) i h (równoległy do osi y) w środku ekranu:



```
10 REM Przykład 6.5
15 REM PROSTOKAT
20 INPUT "podaj wartosc w (1<=
w<=255)";w
30 INPUT "podaj wartosc h (1<=
h<=175)";h
35 PLOT (255-w)/2,(175-h)/2
40 DRAW w,0
50 DRAW 0,h
60 DRAW -w,0
70 DRAW 0,-h
```

Wprowadźmy i wykonajmy ten program. Otrzymany rysunek na ekranie dla $w = 150$ i $n = 100$ jest postaci:



Instrukcję **DRAW** można użyć do rysowania fragmentów okręgów zamiast odcinków linii prostej. Postać tej instrukcji jest następująca:

DRAW x, y, a

x i y określają punkt końcowy jak normalnie, a określa liczbę radianów, w jaką należy się obrócić w trakcie rysowania, jeżeli $a > 0$ to obrót jest w lewo, jeżeli $a < 0$ to obrót jest w prawo (cały okrąg ma 2π radianów, π jest na klawiszu M).

Jeżeli $a = \pi$, to niezależnie od wartości x i y zostanie narysowana połówka okręgu.

Wykonanie ciągu instrukcji:

10 PLOT 100, 100

20 DRAW 50, 50, π

spowoduje narysowanie połowy okręgu.

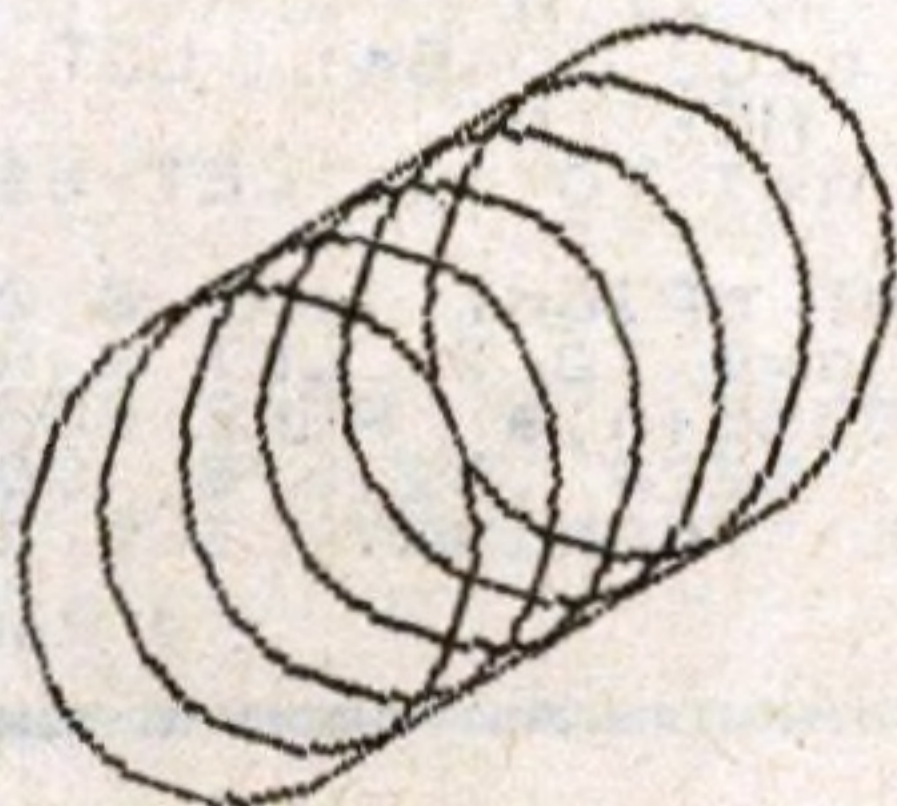
Przykład 6.6.

Narysować ciąg 6 okręgów za pomocą instrukcji DRAW x, y, a , zaczynając od punktu początkowego (40, 40).

Przyjmijmy np. $x = y = 50$

```
10 REM PRZYKŁAD 6.6
20 FOR i=10 TO 60 STEP 10
30 PLOT 40+i, 40+i
40 DRAW 50, 50, PI
50 DRAW -50, -50, PI
60 NEXT i
```

Wprowadźmy i wykonajmy ten program. Otrzymany rysunek na ekranie jest postaci:



Cały okrąg rysujemy stosując zwykle instrukcję **CIRCLE**, postaci:

CIRCLE x, y, r

x, y — współrzędna środka okręgu

r — promień okręgu.

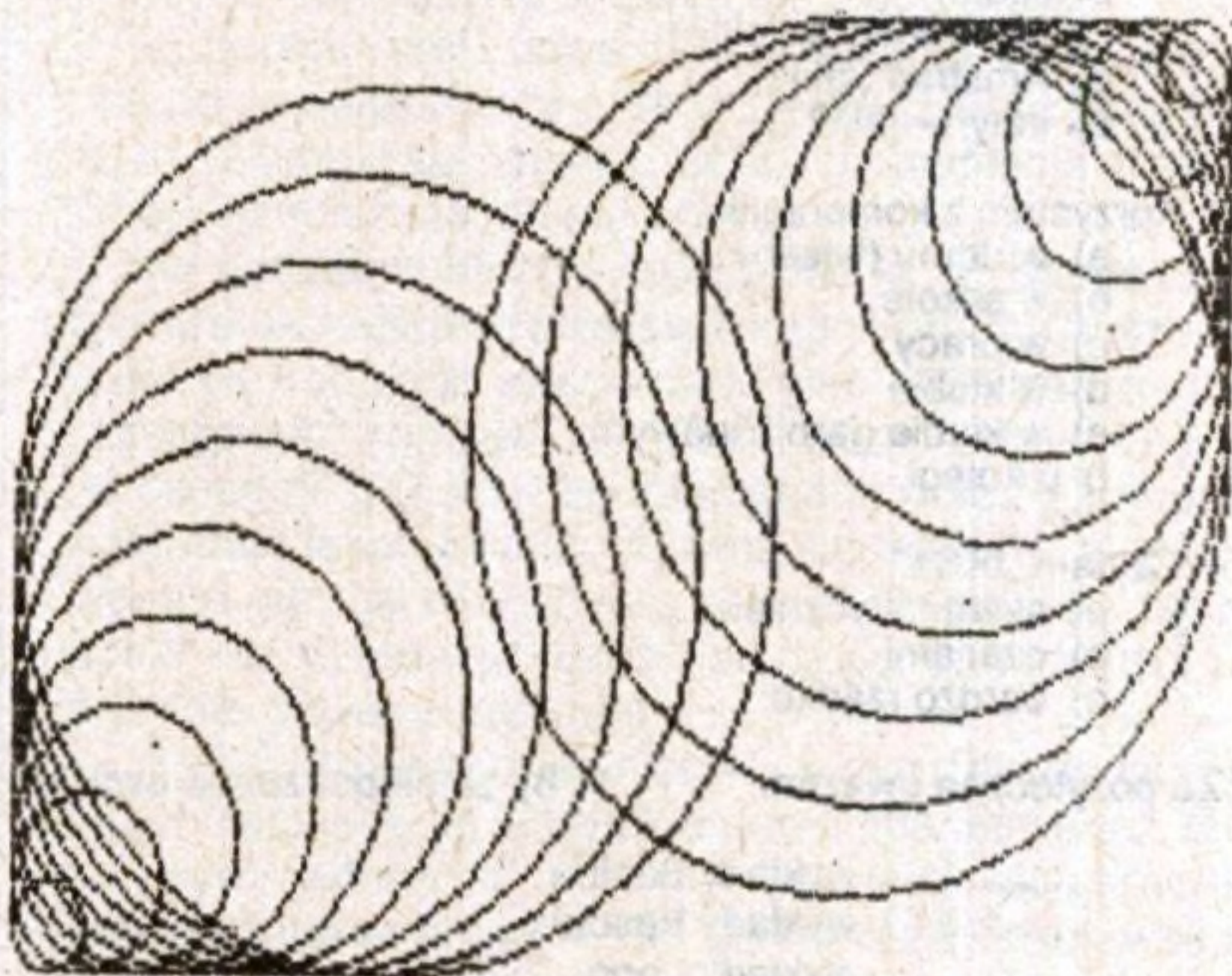
Przykład 6.7.

Narysować ciągi okręgów „wychodzące” z lewego dolnego i prawego górnego rogu ekranu, o współrzędnych środka i promieni danych wzorem:

$$x_i = x_0 + ih \quad (\text{dla } i = 0, 8, \dots, 80, h = 8, x_0 = 0)$$

```
10 REM PRZYKŁAD 6.7
20 FOR x=0 TO 80 STEP 8
30 REM prawy górny róg ekranu
30 CIRCLE 255-x, 175-x, x
40 REM lewy dolny róg ekranu
40 CIRCLE x, x, x
50 NEXT x
```


Wprowadźmy i wykonajmy ten program. Otrzymany rysunek na ekranie jest postaci:



Przykład 6.8.

Narysować kilka wykresów funkcji $i * \sin x$ oraz $i * \cos x$, dla $0 \leq x \leq 4$ oraz $i = 10, 20, \dots, 80$

```
10 REM PRZYKŁAD 6.8
20 REM rysowanie osi x
30 PLOT 0, 88: DRAW 255, 0
40 FOR i=10 TO 80 STEP 10
50 FOR x=0 TO 255
60 PLOT x, 88+i*SIN (x/64*PI)
70 PLOT x, 88+i*COS (x/64*PI)
80 NEXT x
90 NEXT i
```

ZW. 

Ankieta

Minął już rok od chwili ukazania się pierwszego numeru naszego pisma. W każdym kolejnym staramy się zaspokoić informacyjne potrzeby naszych Czytelników. Znamy je z listów nadchodzących do redakcji, ze spotkań. Tym razem, podobnie jak przed rokiem, prosimy o odpowiedź na poniższą ankietę. Z pewnością Wasze odpowiedzi pomogą nam w redagowaniu „IKSa”. Wśród tych, którzy nadeślą pod naszym adresem swoje uwagi rozlosujemy nagrody.

A teraz odpowiedzcie.

Kto nas czyta?

- 1) Mam lat....
- 2) Mieszkam w mieście, które ma:
 - a) więcej niż 150 tys. mieszkańców
 - b) 50—150 tys. mieszkańców
 - c) mniej niż 50 tys. mieszkańców
 - d) mieszkam na wsi.
- 3) Jestem
 - a) uczniem szkoły podstawowej
 - b) uczniem szkoły średniej
 - c) studentem uczelni cywilnej
 - d) studentem uczelni wojskowej
 - e) pracuję.
- 4) Posiadam lub mam dostęp do
 - a) ZX Spectrum
 - b) Atari
 - c) Commodore (jaki?)
 - d) Amstrad (jaki?)
 - e) inny — jaki?
- 5) Korzystam z komputera
 - a) w domu (własny)
 - b) w szkole
 - c) w pracy
 - d) w klubie
 - e) w klubie garnizonowym
 - f) u kolegi.
- 6) Czytam „IKSa”
 - a) systematycznie
 - b) czasami
 - c) bardzo rzadko
- 7) Za pożyteczne uważam:

a)	wykłady Basica
b)	wykłady Pascala
c)	wykłady Logo
d)	publikacje dot. systemów mikrokomputerowych
e)	programy gier
f)	programy użytkowe
g)	programy graficzne
h)	informacje o nowościach tech.
i)	zadania Ligi Myślących
j)	porady techniczne
k)	Twoje propozycje
.....
.....
- 8) za niepotrzebne uważam:

Nasz adres:
 Redakcja „IKS”
 W-wa 00-950
 ul. Grzybowska 77

Początkowo można było pracować. Byli klienci, były zlecenia, a w konsekwencji — pieniądze. Jeśli pracowało się szybko i skutecznie, zarobki przewyższały nie tylko stypendia, ale i średnią krajową płacę. Można było też mieć satysfakcję z wdrożeń swoich opracowań. Zmieniło się to nagle. Zmiana zarządu Studenckiej Spółdzielni spowodowała zmianę atmosfery i przepisów.

Włodzimierz Witkowski szukał miejsca dla siebie. Nie chciał się przekwalifikować, nie chciał podzielić losów dawnych kolegów z Mokotowskiej. Przed laty spotkali się w warszawskim Centrum Studenckiego Ruchu Naukowego. Na kursie języka APT. Potem robili oprogramowania dla obrabiarek sterowanych numerycznie.

Zamówienia mieli poważne

— ze Świdnika, „Nowotki”, Mielca, Kalisza. Na początku dekady wszystko się rozpadło. Przyszła bessa dla rodzimych informatyków. Część wyjechała, za granicę i tam pracuje w zawodzie. Z tych co zostali jeden jest taksówkarzem, inny produkuje sztuczną biżuterię, a reszta jakoś przetrwała. Na dodatek nastąpiła rewolucja. Minikomputery wyparty „Odry” i RIADY. Wiernych informatyce pozostała garstka.

Entuzjaści — Witkowski tak określa ich, siebie.

Czy na entuzjazmie można zarobić? Okazało się, że tak. Witkowski trafił do nie znanego wtedy nikomu Towarzystwa Wynalazców „RATIO”.

Mimo szumnej nazwy firma to w istocie niewielka. Całe biuro mieści się w skromnym pomieszczeniu na warszawskim Bródnie przy ul. Ogińskiego 1. Kapitał zakładowy wszystkiego pół miliona. Tyle co na opłaty lokalowe i rachunki telefoniczne przez kilka miesięcy. Dwa i pół etatu przeznaczono na cały personel. Prezes, urzędujący wiceprezes i te pół dla głównego księgowego. Pensja: czterdzieści tysięcy. Ale żeby ją dostać trzeba było przyjść z ewentualnymi klientami towarzystwa. Z klientami, bo spółka od pewnego czasu praktycznie nie funkcjonowała.

Jedynym atutem jest koncesja na działalność. I to nawet rozległą. Prawie taką jaką prowadzi instytut naukowy i jeszcze kilka innych placówek różnego typu — wdrażanie projektów i opracowywanie systemów informatycznych, prace konstrukcyjne i ekspertyzy techniczne, ekonomiczne, prawne, consulting i marketing, analiza wyników badań i nadzór autorski. Słowem: dla każdego coś miłego, jeśli tylko pragnie wydać parę groszy za myślenie.

z administracji żyją.

Co prawda głównie tej gospodarce.

Klient musi zobaczyć za co ma płacić...

Witkowski wrzuca Amstrada na tylne siedzenie fiata i jedzie do jakiegoś zakładu. Tam rozstawia wszystko w sali konferencyjnej, wczytuje program i...

Spektrum

```

1 REM PROGRAM TKANINA
10 LET a$="": LET b$=""
20 FOR n=1 TO 32: LET a$=a$+CHR$ 144: NEXT n
30 FOR n=33 TO 64: LET a$=a$+CHR$ 145: NEXT n
40 FOR n=1 TO 11: LET b$=b$+a$
: NEXT n
50 PRINT BRIGHT 1; AT 0,0; b$
60 FOR n=0 TO 7: LET a=RND*255
: POKE 65368+n,a: POKE 65383-n,a
: NEXT n: BEEP .02,a/5: GO TO 50
99 SAVE "TKANINA" LINE 10
  
```

J. Rubinowicz

LIGA MYŚLĄCYCH

Zadanie 1

Ułamek $\frac{1}{2}$ można przedstawić w postaci ułamka, np. $\frac{6729}{13458}$. W ułamku tym występują wszystkie cyfry (z wyjątkiem zera). Korzystając z tej zasady należy podać składające się z dziewięciu cyfr ułamki, o wartościach:

$\frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}$.

przy czym żadna z dziewięciu cyfr nie może się powtarzać.

Zadanie 2

Jaki rok zeszłego stulecia zwiększył się $4\frac{1}{2}$ -krotnie, jeśli popatrzymy na odbicie w lustrze cyfr zapisanych na kalkulatorze, tworzących tę datę? (Trzeba poeksperymentować).

Zadanie 3

Pewien wujek przyjechał w odwiedziny do swoich dwóch siostrzeńców i trzech siostrzenic. Najpierw wyszedł mu na spotkanie mały Krzysiek z siostrzyczką Basią i od razu poinformował wujka, że jest dwa razy starszy od swej siostrzyczki. Następnie przybiegła Ela. Ojciec jej wyjaśnił gościowi, że suma lat obu dziewczynek jest dwa razy większa od wieku chłopca. Gdy powrócił ze szkoły Michał, ojciec powiedział, że suma lat obu chłopców jest dwa razy większa od sumy lat dziewczynek. Wreszcie zjawiała się Krysia i ujrawszy gościa krzyknęła radośnie: Wujku przyjechałeś do nas akurat w moje urodziny, właśnie dzisiaj kończę dwadzieścia jeden lat. Ojciec jej zaś dodał: właśnie zauważyłem, że w chwili obecnej suma lat wszystkich mych córek jest dwa razy większa od sumy lat mych synów. Ile lat ma każda córka i każdy syn?

Zadanie 4

Przyrost produkcji w fabryce w porównaniu z rokiem poprzednim wynosi za pierwszy rok $p\%$, za drugi $g\%$. Jaki powinien być procent przyrostu produkcji w 3 roku, aby średni przyrost roczny produkcji za trzy lata był równy $r\%$.

Zadanie 5

Winda ma dwa systemy hamowania różniące się między sobą i niezależnie pracujące. W razie zerwania się liny pierwszy system zatrzymuje windę średnio w 995 przypadkach na 1000, a drugi system w 998 przypadkach na 1000. Obliczyć, jakie jest prawdopodobieństwo P , że w razie zerwania się liny winda zostanie zahamowana?

Nasz adres:
Redakcja „IKS”
W-wa 00-950
ul. Grzybowska 77

Pełne rozwiązania zadań prosimy przysyłać pod adresem redakcji do końca lipca br., z dopiskiem „Liga Myślących”. Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi” czekają dodatkowe cenne nagrody — niespodzianki.

Poczta „IKS-a”

„IKS” kończy PIERWSZY ROK. W ciągu tego (niezbyt długiego) czasu otrzymaliśmy wiele, bardzo wiele listów: pocztówek, starannie zaklejonych kopert, a nawet pracownie sporządzonych maszynopisów.

Każdy, nawet najkrótszy, był uważnie przeczytany, a zawarte w nim myśli natychmiast wykorzystane, gdyż... „IKS” JEST DLA WAS!

Odpowiedzi było mało.

Mogliśmy bowiem albo odpowiadać, albo... ROBIĆ „IKSa”!

A nas było (i nadal jest) bardzo, bardzo mało.

Jan MŁYNARZ

Tomasz Nabożny:

„...Jestem czytelnikiem waszego czasopisma. Zbieram prospekty różnych firm, a także interesuję się informatyką. I właśnie pomyślałem sobie, czy nie moglibyście przysłać mi adresów firm komputerowych, takich jak Sinclair, Atari, Commodore, Sharp itd. ...Gdzie można zakupić drogą wysyłkową książki do nauki informatyki dla początkujących na ZX Spectrum +...”

Wydawnictwa Komunikacji i Łączności niedawno wydały książkę PRZEWODNIK PO ZX SPECTRUM. Nakład duży, cena wysoka (490 zł). Co się zaś tyczy „zbieractwa”, to prawda jest taka: duże firmy są zainteresowane MASOWĄ SPRZEDAŻĄ swoich wyrobów i niechętnie (lub wcale) nie interesują się kolekcjonerami. Nasza rada: dużo się ucz (nie tylko angielskiego), a za parę lat powinny pojawić się polskie minikomputery i polskie prospekty tych minikomputerów.

Wojciech Zarzyka z Sanoka:

„...Interesuję się informatyką, chciałbym się dowiedzieć, ile kosztuje ZX SPECTRUM i gdzie go można kupić. A oto mój program...”

Dziękujemy za program. Komputery (za złotówki) można kupić trojako:

- „z ogłoszenia” (polecamy, gdyż znany jest sprzedawca),
- w komisie (sprzedawca pobiera dodatkową marżę),
- na giełdzie (bardziej ryzykowne — jest to typowe kupowanie kota w worku).

Sprzedaż komputerów prowadzi również Centralna Składnica Harcerska (Warszawa, Mokotowska), ale popyt dotychczas kilkakrotnie przewyższa podaż.

UWAGA!

Zrzeszenie Studentów Polskich
M I K R O K O M P
93-590 Łódź
Al. Politechniki 3

organizuje w dniach 17—18 października br. konferencję użytkowników SHARP

telefony: ZSP 36-84-89

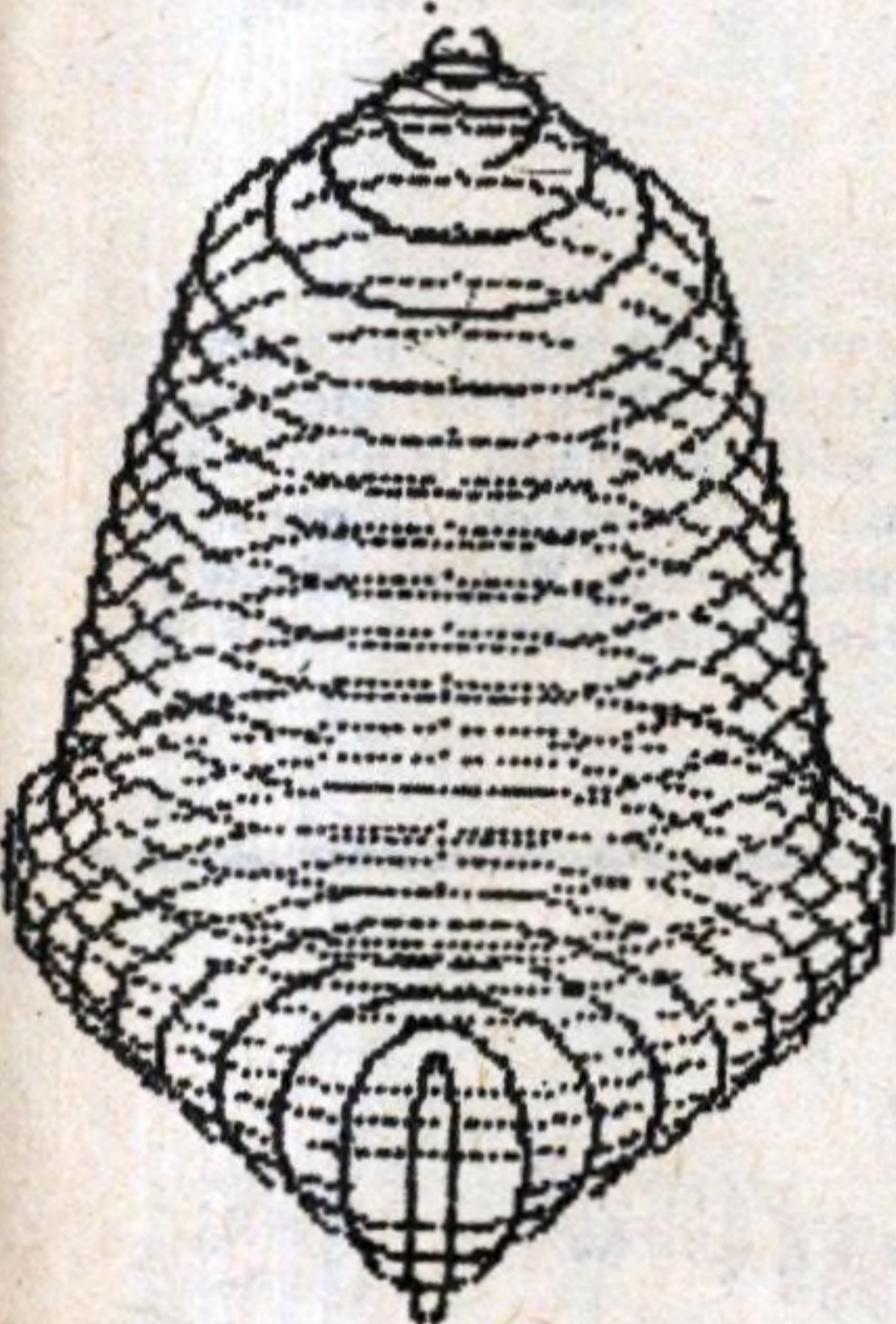
Prezes 32-57-83

Pocztowa giełda

Rozwiązanie krzyżówki z nr. 2/87 „IKS-a”
Hasło brzmi:

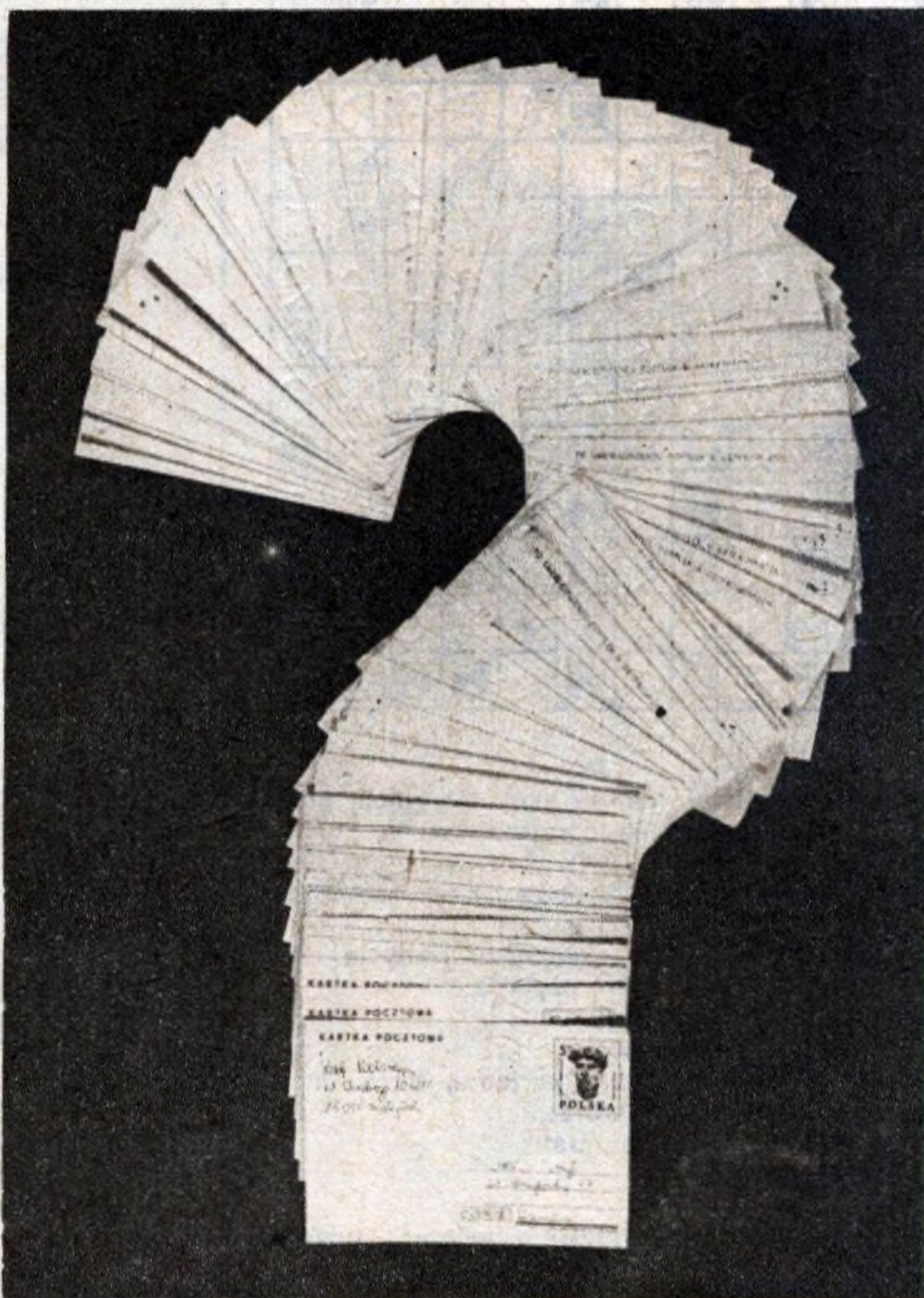
„ELWRO to komputer dla Ciebie”
Bony pieniężne (1000 zł) wylosowali:
Barbara Sułkowska — Wyszaków, Jacek
Golis — Zgierz, Stanisław Mączka — Le-
żajsk, Barbara Rejman — Kluczbork,
Marcin Brzeziński — Lublin.

Nagrody książkowe otrzymują: Marcin
Szalkowski — Skierniewice, Tomasz Sta-
ruszkiewicz — Przemyśl, Maciej Parzy-
szek — Warszawa-Anin, Ireneusz Nowak
— Tarnowskie Góry, Halina Wojturska —
Rzeszów, Jadwiga Zasada — Szczecin,
Ewa Bonder — Świętochłowice, Zbign-
iew Słowik — Włoszczowa, Janusz Do-
las — Częstochowa, Zdzisław Skiba —
Szczecin.



```
5 REM GRAFIKA ATARI
10 GRAPHICS 8:COLOR
1:BY=16:BC=1:DEG
20 FOR B=31 TO 76 ST
EP 3:BB=COS(3*B):CC=
COS(1.1*(B-18)):IF B
>64 THEN BC=BC-0.2
30 FOR A=-180 TO 180
STEP 2:X=INT(COS(A)
*30):Y=INT(SIN(A)*12
):IF B>64 THEN 70
40 PLOT 160+X*BB*1.1
,10+Y+B:PLOT 160+X*C
C,32+BY+Y+B
50 NEXT A:BY=BY+1.5:
NEXT B
70 PLOT 160+X*CC*BC,
32+Y*BC+B+BY:GOTO 50
```

Jeszcze raz konkurs



Oto plik kilkudziesięciu poprawnych oczywiście rozwiązań. Na odwrocie są różni nadawcy, ale jeden autor na pocztowe kartki na-
klejał powielone rozwiązanie. W losowaniu szczęścia niestety nie
miał, ale należy mu się piątka z rachunku prawdopodobieństwa.

Spektrum

```
10 REM Program "KILIM 3"
20 BORDER 0: LET n=1: LET a$="
FABCDEJIHGFABCDEJIHGFABCDEJIHGF
TONMLKPORSTONMLKPORSTONMLKPORSTO
": LET b$="": FOR m=1 TO 11: LET
b$=b$+a$: NEXT m: CLS
25 PRINT AT 8,1: BRIGHT 1:"OBL
ICZANIE TABELI ODWRACAJACEJ"
30 FOR s=0 TO 255
40 POKE 22432,s
50 LET t=POINT (0,0)+2*POINT (
1,0)+4*POINT (2,0)+8*POINT (3,0)
+16*POINT (4,0)+32*POINT (5,0)+6
4*POINT (6,0)+128*POINT (7,0)
60 POKE 60000+s,t: NEXT s
70 FOR z=0 TO 39
80 LET a=INT (RND*255): LET b=
PEEK (60000+a)
85 POKE 65368+z,a
90 POKE 65408+z,b
95 POKE 65487-z,a
100 POKE 65527-z,b
110 NEXT z
120 PRINT #0: AT 1,15:n: LET n=n
+1: BEEP .02,a/8
130 PRINT AT 0,0;b$: PAUSE 140:
BEEP .02,12+a/8: PRINT AT 0,0:
INVERSE 1;b$: GO TO 70
200 SAVE "KILIM 3" LINE 10
```

J. Rubinowicz

Krzyżówka nr 5

	A	B	C	D	E	F	G	H	I	J	K	L	Ł
1	M	I	K	R	O	P	R	O	C	E	S	O	R
2	A	⊗	O	⊗	P	⊗	Y	⊗	Z	⊗	Z	⊗	E
3	B	⊗	D	⊗	R	⊗	Z	G	A	⊗	P	⊗	S
4	S	⊗	O	⊗	W	⊗	O	⊗	Y	⊗	R	E	A
5	T	⊗	P	⊗	G	E	K		N	⊗	D	⊗	S
6	E	⊗	I	⊗	R	⊗	O	⊗	O	⊗	E	⊗	T
7	R	⊗	S	⊗	H	⊗	A	⊗	R	⊗	S	⊗	K
8	⊗	P	⊗	⊗	M	⊗	⊗	⊗	K	⊗	S	⊗	⊗
9	T	⊗	E	⊗	R	⊗	M	⊗	O	⊗	K	⊗	O
10	⊗	R	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
11	N	⊗	A	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
12	⊗	T	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
13	R	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
14	⊗	R	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗

POZIOMO: 1. układ scalony wielkiej skali integracji, podstawowy element budowy mikrokomputerów,

8. witka, łoża,
9. ciąg bitów mieszczący się w pojedynczej komórce pamięci,
10. pisarz angielski (1814—84) rozwijający tradycje dickensowskie (skojarz z rozkazem „czytaj”),
11. owadożerna niewielka jaszczurka,
12. włosy z ogona i grzywy końskiej,
14. wielkość określona jednoznacznie jedną liczbą rzeczywistą (np. długość, temperatura),
17. urządzenie peryferyjne do termografii,
18. np. informatyka, cybernetyka,
19. latacz, kliwer, sztaksel,
20. utracenie twardości pod wpływem wilgoci, rozmoknięcie:

PIONOWO: 1. fachowiec, mistrz,

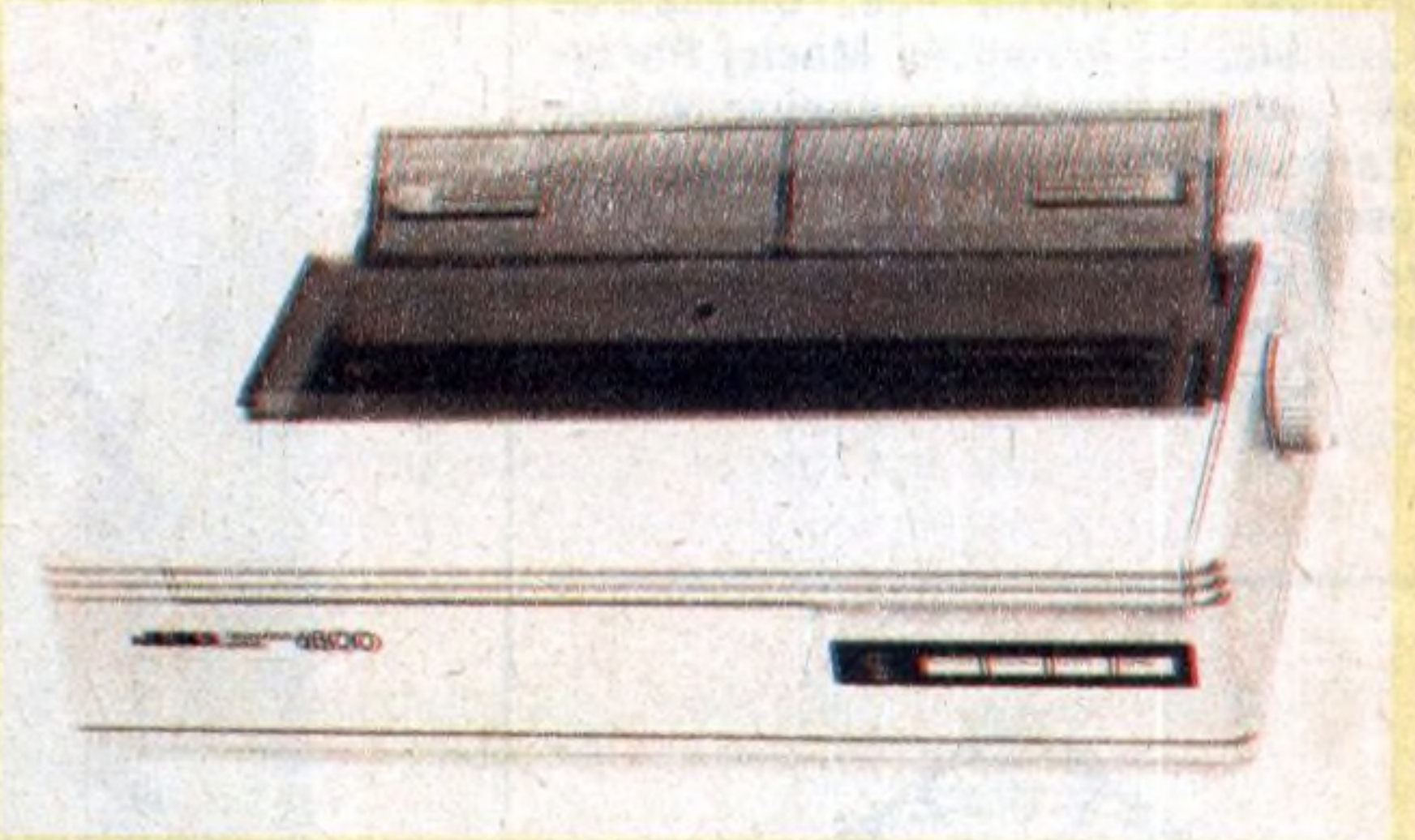
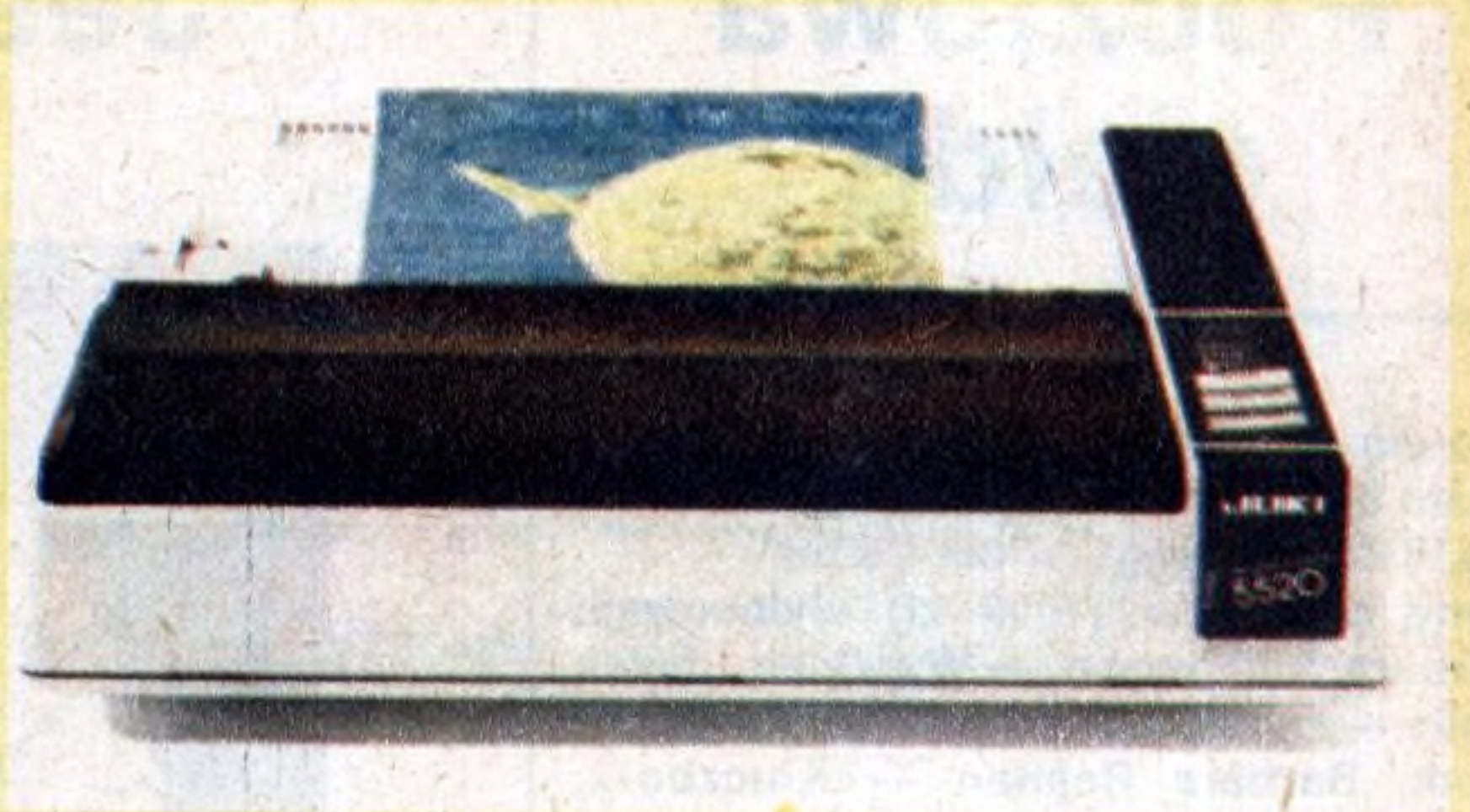
2. urządzenie klawiaturowe łączące funkcjonalne własności dalekopisu, elektrycznej maszyny do pisania i dziurkarki taśmy,
3. software,
4. przedsięwzięcie o niepewnym wyniku,
5. mag, czarodziej,
6. łopata, rydel,
7. urządzenie zapamiętujące pojedynczą wartość danej lub wyniku,
13. pracownik wyspecjalizowany w obsłudze komputera,
15. np. nikotyna, kofeina, strychnina,
16. element pamięci zawierający rozkaz lub dane i posiadający ściśle określony adres.

Rozwiązanie stanowi hasło utworzone z liter przeniesionych z następujących pól:

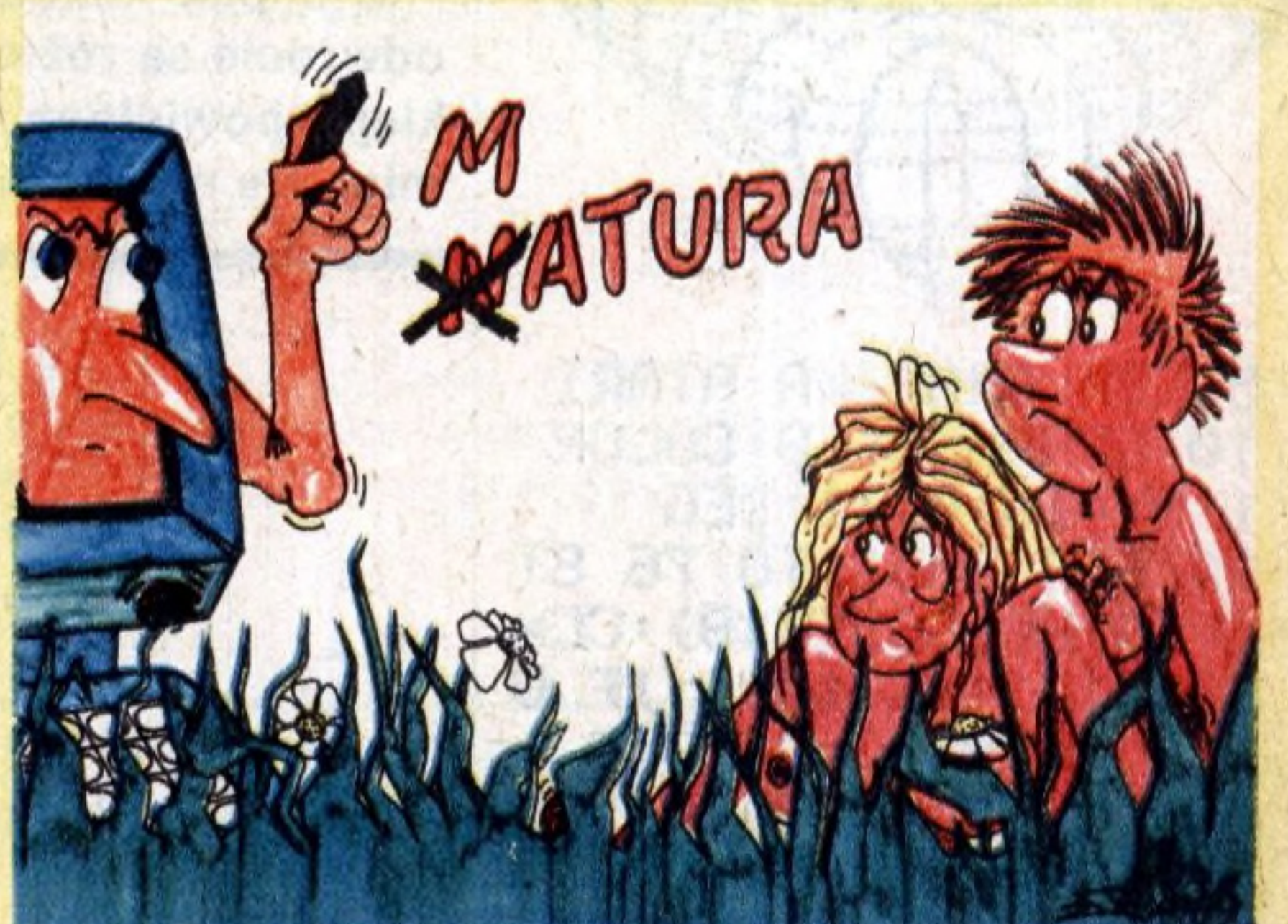
1B 1C 1K 5C 6E 6G 2E 4K 11K 9J 12B 12L
 9C 2L 13D 4D 6C 5F 5K 3G 4G 13I 6L 11L
 13K 7D 11A 13E 8G 9J 10G 13L 9F 12G 7B
 7I 1H 9D 8B 11C 5A 4J 7F 2C 10E 4L 3A

Rozwiązanie (tylko hasło) należy przesłać pod adresem redakcji do końca lipca br. Wśród czytelników rozlosujemy bonny pieniężne i nagrody książkowe.

Mikrociekawostki



Juki 5520 i 6500 ta sama firma, ale istotne różnice w cenie i możliwościach. Obie drukarki mają spory repertuar znaków, krojów i wielkości pisma. Istotnymi zaletami są przede wszystkim szybkość druku (ponad sto znaków na minutę) i cicha praca. Wada — niestety, z uwagi na cenę są to urządzenia dla amatorów w Polsce niedostępne.



Rys. Michał Przybyłowicz

„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogon. Stali współpracownicy: Włodzimierz Gogolek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, 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 rotograwiurony — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 8814. Nr ind. 361682.