



Dobry Polak poci się przy drugiej cyfrze dziesiątnej, przy piątej dostaje gorączki, przy siódmej zabija go apopleksja.

Bolesław Prus „Lalka”

Dziś, przy końcu dwudziestego wieku, sytuacja nie jest aż tak tragiczna; mamy do dyspozycji komputery. Są to automatyczne urządzenia liczące, które „potrafią” wykonywać setki tysięcy działań w czasie jednej sekundy.

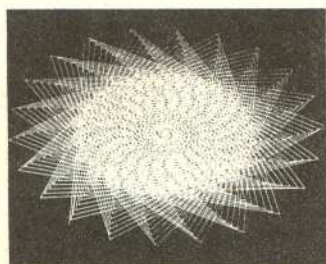
Nowy kłopot natomiast polega na tym, iż poleceń komputerowi nie wydajemy głosem, a w formie pisanej. Takie napisane polecenie to *program*. Oczywiście program musi być tak napisany, aby mógł być „zrozumiany” przez komputer. Czynność pisania programów nazywamy programowaniem. Napisany tekst (program) jest kodowany na taśmie perforowanej (dziurkowanej) lub kartach perforowanych i jest wprowadzany przez tzw. urządzenie wejściowe do pamięci komputera.

Popatrzmy jak przebiega proces rozwiązania prostego zadania z pomocą komputera.

Przypuśćmy, że chcemy dodać dwie macierze o n wierszach i m kolumnach. Co powinniśmy w tym celu zrobić?

Zastanówmy się, co mamy otrzymać w wyniku? Oczywiście macierz, o tej samej liczbie wierszy i kolumn co macierze dodawane. Każdy element macierzy — wyniku jest sumą odpowiednich macierzy — składników.

Możemy to zapisać w postaci tzw. sieci działań:

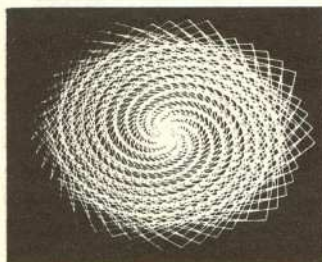


Dowolny prostokątny (kwadratowy) układ liczb nazywamy macierzą; np.

$$\begin{pmatrix} 0 & 3 & 2 \\ 1 & 5 & 7 \end{pmatrix}$$

jest macierzą o dwóch wierszach i trzech kolumnach.

Jeżeli mamy dwie macierze o tej samej liczbie wierszy i kolumn, to możemy je dodać i w wyniku otrzymamy trzecią macierz, której każdy element jest sumą elementów danych macierzy, stojących na odpowiadających sobie miejscach.

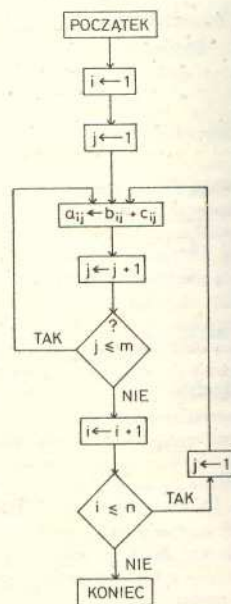


Liczby całkowite i oraz j będą nam służyły do liczenia odpowiednio wierszy oraz kolumn.

Dodajemy elementy macierzy zgodnie z określeniem sumy macierzy. Numer zwiększamy o jeden...

i pytamy: „Czy jest spełniony warunek $j \leq m$?” jeżeli tak, to dodajemy kolejne elementy, jeżeli nie, to zwiększamy numer wiersza o 1...

i pytamy: „Czy $i \leq n$?”, jeżeli tak, to numer kolumny ustalamy na 1, jeżeli nie, to skończyliśmy pracę.




```

dodmac: macro
jpt, 7.
n: .
m: .
a: .
b: .
c: .
i: .
j: .
r: .
lot, 1,1.
st, 1(i).
pow: lot, 1,1.
st, 1(j).
pow 1: lo, 1(i).
su, 1(1).
ex (m) mplw. 1.
od, 1(j).
su, 1(1).
ex (3) mplw. 1.
st, 1(r).
mod (r).-
log, 1(b).
mod (r).
addf (c).
mod (r).-
stg, 1(a).
lo, 1(j).
adt, 1,1.
st, 1(j)-
co, 1(m).
jptl (pow 1) .
jpte (pow 1).
lo, 1(1).
adt, 1,1.-
st, 1(i).
co, 1(n).
jptl (pow).
jpte (pow).
jp (dodmac).
fin macro .

```

Taki rysunek stanowi udogodnienie dla programisty, niestety nie może być bezpośrednio wprowadzony do komputera. Patrząc na rysunek piszemy program w języku komputera, którego używamy do pracy. Popatrzmy (obok) jak wygląda program, realizujący powyższy schemat w języku ASSK-4.

Ten tekst jest dla nas niezrozumiały, ale zrozumiały dla polskiego minikomputera K-202 i tylko dla niego, jest to bowiem tzw. język wewnętrzny tego urządzenia. Gdybyśmy mieli do dyspozycji inną maszynę, musielibyśmy nauczyć się jej wewnętrznego języka. Języki wewnętrzne — to jakby języki narodowe komputerów.

Pisanie w ten sposób programów jest bardzo uciążliwe i nużące.

Aby ułatwić pracę programistów opracowano języki zewnętrzne albo autokody, tzw. języki wyższego rzędu. Języki te są niezależne od typu komputera, aby jednak efektywnie zrealizować program zapisany w języku wyższego rzędu, komputer musi najpierw przetłumaczyć ten program na swój język wewnętrzny, a następnie dopiero wykonać obliczenia.

Przykładem języka wyższego rzędu może być międzynarodowy, uniwersalny język algorytmiczny ALGOL 60 (zainteresowanych odsyłam do książki R. Zubera „Metody numeryczne i programowanie”). Przytoczony obok program przepisemy w języku ALGOL 60.

```

procedure dodmac n,m,a,b,c ;
value n,m;
integer n,m;
array a,b;
begin integer i,j;
for i:= 1 step 1 until n do
for j:= 1 step 1 until m do
a [i,j] :=b [i,j] +c [i,j]
end

```

Napis to znacznie krótszy i przy znajomości kilkudziesięciu słów angielskich może być łatwo zrozumiały dla Czytelnika. I co bardzo ważne — ten tekst jest również zrozumiały dla większości komputerów zainstalowanych w Polsce.

Niektórzy uważają jeszcze i ten tekst (w ALGOLu) za zbyt skomplikowany. Tym możemy polecić „pakiet programów” MATLAN (od angielskich słów MATRIX LANGUAGE).

Omawiany przez nas przykład da się w nim zaprogramować w sposób następujący:

```

MAIN
READ B,C.      „czytaj macierze B, C”
ADD B,C,A      „dodaj B i C, wynikiem będzie A”
WRITE A        „pisz macierz A”
END

```

Opracowano oczywiście i inne pakiety (czasem mówimy: języki problemowo-zorientowane), np. do obliczeń statystycznych, optymalizacyjnych itd. To tylko drobna wzmianka o językach komputerowych, jeżeli przy okazji czytania tego tekstu pojawiły się u Was jakiegokolwiek wątpliwości — prosimy PYTAJCIE. Czy warto w ogóle było pisać programy dla tak nieskomplikowanego zadania? O ile chcielibyśmy dodać tylko dwie macierze, prawdopodobnie układanie programów nie miałoby sensu. Ale tutaj, zauważmy, napisany przez nas program może być używany wielokrotnie i do obliczeń znacznie bardziej skomplikowanych.

