

C E N T R A L N Y
O Ś R O D E K
K O N S T R U K C Y J N O -
B A D A W C Z Y
P R Z E M Y S Ł U
M O T O R Y Z A C Y J E G O
Z E S P O Ł D/S
I N F O R M A C J I
T E C H N I C Z N E J
I E K O N O M I C Z N E J
W A R S Z A W A
U L . S T A L I N G R A D Z K A 23



DO UŻYTKU
SŁUŻBOWEGO

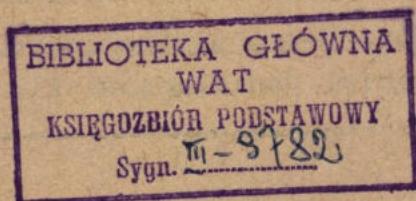
PROGRAMOWANIE „SAKO” na Mińsk - 22

EGZ. NR

WARSZAWA 1970 r.

PROGRAMOWANIE
"SAKO"

na Mińsk-22



Warszawa 1970

Opracowali:

Jan Szmelter

Henryk Zub

Katedra Mechaniki Teoretycznej i Wytrzymałości Materiałów
Wojskowej Akademii Technicznej

0-192/22

Druk COKEPMOT. W-wa, ul. Stalingradzka 23 tel. 11-42-31 w.
zam. 703, papier powielaczowy kl. VII, gr. 70, nakł. 50.

Spis treści	str
1. WSTĘP	3
1.1. Uwagi ogólne o rachunkach matematycznych	3
1.2. Uwagi ogólne o maszynie Mińsk-22	4
1.2.1. Urządzenie wejścia i wyjścia	4
1.2.2. Pamięć i inne urządzenia	7
1.3. Liczby	9
2. SAKO W PRZYKŁADACH	11
2.1. Wprowadzenie. Rozkazy wejścia i wyjścia	11
2.2. Obliczenie wartości wyrażenia algebraicznego	15
2.3. Funkcje standardowe	18
2.4. Funkcje definiowane	19
2.5. Definiowanie wielu funkcji	20
2.6. Funkcje definiowane wielu zmiennych	21
2.7. Skok bezwarunkowy	22
2.8. Skok warunkowy, napis	23
2.9. Powtórzenia	25
2.10. Wydawnictwo	26
2.11. Zmienne indeksowane	29
2.11.1. Zmienne o jednym wskaźniku	29
2.11.2. Rozkaz WARTOSC i komentarze	31
2.11.3. Zmienna indeksowana w podprogramie	32
2.11.4. Zmienna indeksowana o dwóch wskaźnikach. Rozkaz STRUKTURA. Cykl w cyklu	34
2.11.5. Zmienna indeksowana o wielu wskaźnikach	39
2.12. Rozkaz KLUCZ, wydruk kontrolny	40
2.13. Rozkaz PODSTAW	42
2.14. Podprogram wewnętrz programu	43
2.15. Współpraca z taśmą magnetyczną	44
2.16. Kontrola poprawności zapisu programu	45
2.17. Uruchamianie programów	47
3. LISTA ROZKAZÓW I SŁOWNIK WYRAZÓW	49
3.1. Lista rozkazów języka SAKO dla Mińsk-22	49
3.2. Słownik wyrazów	57
4. BIBLIOTEKA PODPROGRAMÓW	59

1. WSTĘP

Opracowanie niniejsze podaje opis w przykładach-konkretniej realizacji języka SAKO^{m/} dla maszyny MIŃSK-22. SAKO jest symbolicznym językiem, służącym do zapisu odpowiednich procesów obliczeniowych. Jego forma polega na użyciu do zapisu wyłącznie pewnych instrukcji, z których każda ma określoną budowę. Ustalona forma zapisu instrukcji pozwala na ich automatyczne przetłumaczenie na język wewnętrzny rozumiany przez maszynę.

1.1. Uwagi ogólne o rachunkach matematycznych

Obliczenia matematyczne polegają na podstawianiu wartości liczbowych do wzorów matematycznych, w celu otrzymania wyników. Np. na przebieg obliczenia pierwiastków równania kwadratowego:

$$ax^2 + bx + c = 0$$

składają się następujące etapy:

1/ ułożenie programu, to znaczy kolejnych wzorów, według których ma przebiegać obliczenie:

$$\Delta = b^2 - 4ac$$

$$x_1 = (-b - \sqrt{\Delta}) / 2a$$

$$x_2 = (-b + \sqrt{\Delta}) / 2a$$

2/ podstawienie danych liczbowych na miejsce zmiennych a, b, c np.:

$$a = 1$$

$$b = -3$$

$$c = 2$$

m/ W niniejszej publikacji wykorzystano idee zawarte w pracy L.Łukaszewicz, A.Masurkiewicz - "System automatycznego kodowania SAKO" - Wyd.PAN.Wrocław 1963 oraz odnośnie wejść i wyjścia informacje podane w opracowaniu H.Radzikowski - "Vademekum programisty MAT"-Wyd.Komisja Planowania przy Radzie Ministrów,Centrum Obliczeniowe,Warszawa 1968.Ponadto wzorowano się na opracowaniu: J.Szmelter,K.Deloff-"Programowanie SAKO dla początkujących"-Katedra Mechaniki Technicznej Politechniki Łódzkiej, Łódź 1962.

3/ wykonanie rachunków według ustalonego schematu i podanie wyników:

$$x_1 = 1$$

$$x_2 = 2$$

Gdy podamy rachmistrzowi powyższy schemat obliczeń i dane, to stosując się ściśle do nich otrzyma on poprawne wyniki, chociażby nie rozumiał zadania. Dlatego też można go zastąpić automatem, jakim jest maszyna matematyczna. Do wykonania tego zadania maszyna musi mieć następujące urządzenia:

1. wejście, to znaczy urządzenie, za pomocą którego przekazujemy program i dane,
2. wyjście, za pomocą którego otrzymujemy wyniki,
3. urządzenie pozwalające maszynie na wykonanie zadania; znajomość tego urządzenia nie jest konieczna dla korzystania z maszyny.

1.2. Ogólne wiadomości o maszynie Mińsk-22

1.2.1. Urządzenia wejścia i wyjścia

Treść zadania, a więc program i dane dla maszyny przygotowujemy na dalekopisie. Jest to urządzenie podobne do zwykłej maszyny do pisania, na którym piszemy przez naciskanie odpowiednich klawiszy. Znaki pisarskie dalekopisu stosowane w SAKO są następujące:

1/ Litery

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2/ Cyfry

0 1 2 3 4 5 6 7 8 9

3/ Inne

+ - . / () ', : = ? # spacja linia powrót

Spacja jest to odstęp między znakami; Linia jest to przesunięcie wałka maszyny o jedną linię; powrót – ustawienie karterki z wałkiem do pisania od początku wiersza.

po-
Na dalekopisie pisze się wierszami. Dlatego np. kolumny
liczb

321	528
125	45
4	7

piszemy w kolejności: 321 528 125 45 4 7.

Przy pisaniu normalnego tekstu, jednocześnie wychodzi z dalekopisu papierowa taśma perforowana, która jest kopią tego tekstu zapisaną układem dziurek. Jeżeli taką taśmę wsuniemy do dalekopisu, to zapisaną na niej treść wydrukuje on uruchamiając automatycznie odpowiednie klawisze. Zatem dalekopis może służyć do przygotowania taśmy, jak również do przetłumaczenia jej na normalne pismo. Sposób zapisu znaków dalekopisowych na taśmie perforowanej podaje tablica 1.

Teksty wprowadzane do maszyny liczącej są przygotowane na taśmie dalekopisowej, którą wsuwamy do czytnika. Taśma, przebiegając przez czytnik, wzbudza w maszynie impulsy elektryczne, odpowiadające dziurkom wydrukowanym na niej. Odpowiedzi maszyna wydaje za pośrednictwem drukarki. //

// Istnieją także możliwości wprowadzania danych z kart perforowanych i z klawiatury maszyny.

Wyniki można otrzymać także za pośrednictwem perforatora w postaci taśmy perforowanej, lub na drukarce wąskiej. Z urządzeń tych nie korzystają użytkownicy języka SAKO.

ścieście początku taśmy

Po znaku "CYFRY"		Po znaku "LITERY"	
kod	znak dalekopisu	kod	znak dalekopisu
10	8	•	puste
0	0 0	•	5 T
1	0 1	• o	karetka
2	0 2	• o	9 0
3	0 3	• oo	spacja
4	0 4	• o	10 H
5	0 5	• oo o	,
6	0 6	• oo	N M
7	0 7	• oo o	linia
8	1 0	o •	,
9	1 1	o . o	4 R
10	1 2	o . o	1 G
11	1 3	o . oo	8 I
12	1 4	o . o	0 P
13	1 5	o . o o	:
14	1 6	o . oo	= V
15	1 7	o . oo o	3 E
16	2 0	o .	+
17	2 1	o . o	# Z
18	2 2	o . . o	,
19	2 3	o . oo	D S
20	2 4	o . o	,
21	2 5	o . o o	6 Y
22	2 6	o . oo	L F
23	2 7	o . oo o	7 X
24	3 0	oo .	- A
25	3 1	oo . o	2 W
26	3 2	oo . o	:
27	3 3	oo . oo	cyfry U
28	3 4	oo . o	7 Q
29	3 5	oo . o o	C K
30	3 6	oo . oo	litery
31	3 7	oo . oo o	

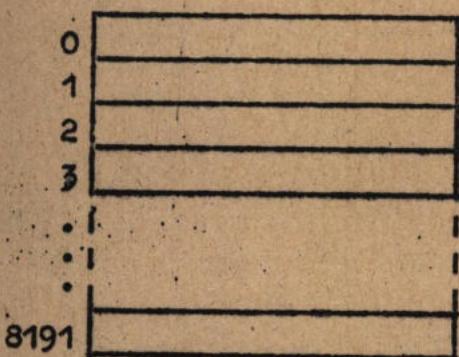
Taśma

ścieście końca taśmy

Każdy układ dziurek na taśmie ma dwojakie znaczenie. Wszystkie znaki na taśmie, po znaku LITERY, traktowane są jako litery, aż do pojawienia się znaku CYFRY. Po tym znaku, wszystkie znaki na taśmie traktowane są jako cyfry, aż do pojawienia się znaku LITERY.

1.2.2. Pamięć i inne urządzenia

Maszyna musi mieć możliwość przechowywania danych, wzorów, rozkazów, wyników działań pośrednich i końcowych. Organem wykonującym to zadanie jest pamięć operacyjna. Można ją przedstawić jako 8192 komórek ponumerowanych. Numer komórki nazywamy jej adresem:

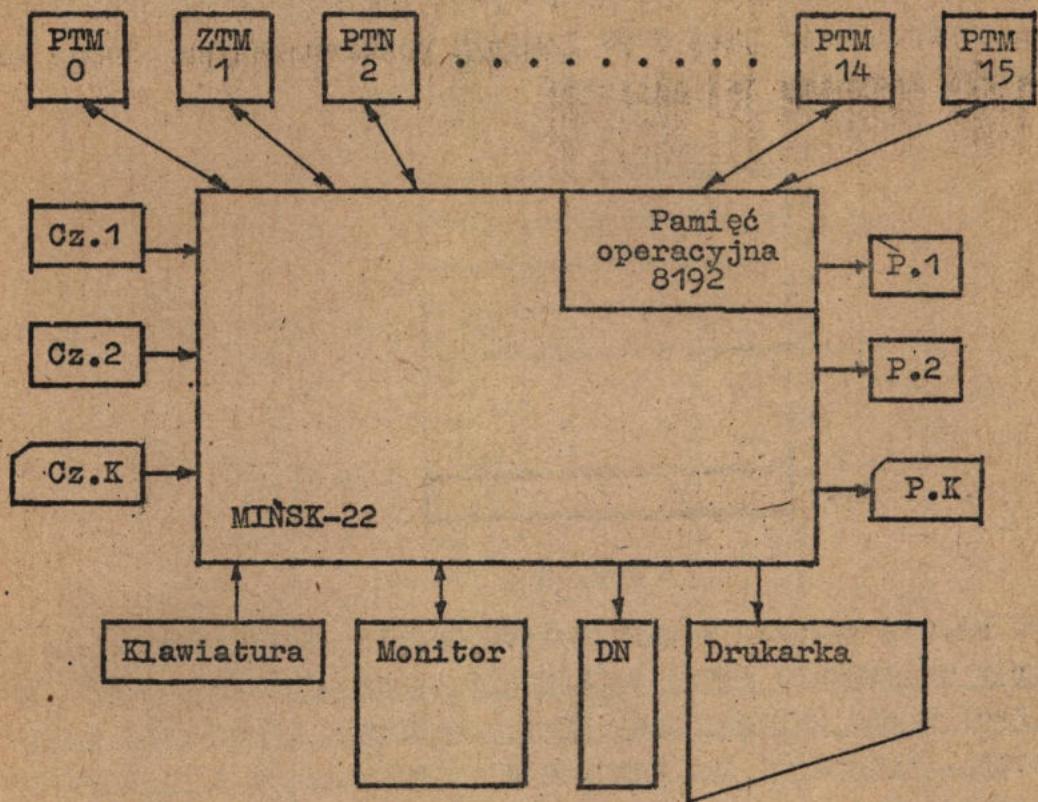


Rys. 1

Do każdej komórki możemy wpisać dowolną informację. Przy wpisywaniu nowej informacji, poprzednia zawartość komórki ulega zniszczeniu. Przy pobieraniu informacji z komórki, zawartość jej pozostaje niezmieniona, a pobieramy jej kopię.

Oprócz pamięci operacyjnej maszyna ma pamięć zewnętrzna, którą stanowi taśma magnetyczna. Działań nie można dokonywać bezpośrednio na liczbach zapisanych na taśmach magnetycznych. Przedtem trzeba je przesłać do pamięci operacyjnej. Obliczenia przeprowadzone przez maszynę wykonywane są przez arytmometr. Przebieg obliczeń jest sterowany urządzeniami sterowania, działającymi zgodnie z instrukcjami programu.

Zestaw EMC Mińsk-22



- PTM - przewijacz taśmy magnetycznej
- Cz. - czytnik taśmy perforowanej
- P. - perforator taśmy
- Cz.K - czytnik kart
- P.K - perforator kart
- DN - drukarka numeryczna /wąska/

Rys. 2

1.3. Liczby

W obliczeniach posługujemy się liczbami całkowitymi i rzeczywistymi.^{w/} W zapisie różnią się one tym, że liczba rzeczywista ma przecinek pozycyjny, a liczba całkowita nie. Liczba może być poprzedzona znakiem "+" lub "-".

A oto przykłady zapisu liczb:

a/ liczby całkowite:

+2 -328 0 002 1970

b/ liczby rzeczywiste:

+2,0 3,14 003,140 -0,3

Miedzy poszczególnymi znakami liczby nie może występować spacja. Np. liczba zapisana w postaci +u2 będzie traktowana jako osobny znak "+" oraz liczba całkowita 2, a zapis 10u000 traktowany będzie jako zapis dwóch liczb 10 i 0, a nie jako 10000.

Największa wartość bezwzględna liczby dziesiętnej wynosi:

68700000000

^{w/} Dla celów specjalnych stosuje się czasami liczby oktalne, które w zapisie poprzedzone są znakiem "#", przed którym w zapisie może występować znak "+" lub "-", np.:

+ # 0074
- # 0000325
127

Największa wartość bezwzględna liczby oktalnej wynosi

#777777777777

-tych mewtoñkowym na oznaczeniu przed znakiem F.

- 2. SAKOW W PRZAKŁADACH

2.1. Wprowadzenie, rozkazy wejścia, wyjścia i sterujące

Przykład F1.

Napisać program, w którym polecimy maszynie wczytać trzy liczby i wydrukować.

Rozwiązanie: Program ten ma postać następującą:

F1
WEJSCIE 3
DRUKARKA
CZLICZBE A
CZLICZBE B
CZLICZBE C
LINIA 1
DRLICZBE(3:2):A:B:C
LINIA 1
STOP
START F1

Program F1

8	7,2	-3,14
---	-----	-------

Dane

8,00	7,20	-3,14
------	------	-------

Wyniki

Objaśnienie

Pierwszy wiersz zawiera nazwę programu F1. Każda nazwa programu zaczyna się od litery F. Dalsze znaki mogą być literami lub cyframi. W każdej nazwie maszyna rozpoznaje pierwszych 6 znaków. Wszystkie inne są pomijane. Np. nazwy:

FPRZYKŁAD1

FPRZYKŁAD2

są traktowane jednakowo jako

FPRZYK

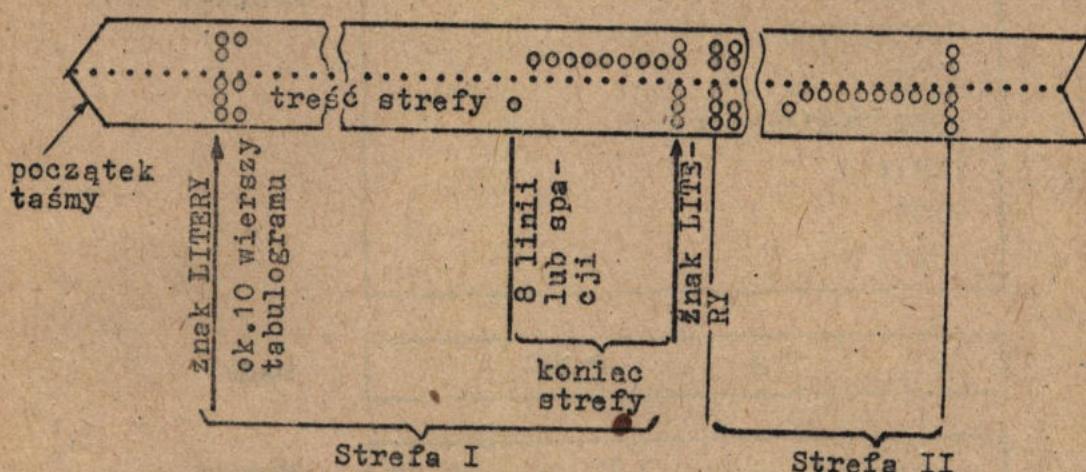
W drugiej linijce napisano:

WEJSCIE 3

Oznacza to, że dane będą wczytywane za pośrednictwem czytnika 1 sposobem strefowym^{x/}. Jest to zalecany sposób wczytywania danych. Inne wejścia podano na liście rozkazów.

x/ Czytanie strefowe wymaga specjalnego przygotowania taśmy. Na początku i końcu taśmy zostawia się pewną ilość pustej taśmy, co ułatwia zakładanie taśmy pod czytnik. Każda strefa rozpoczyna znak LITERY. Jeżeli treść strefy rozpoczyna się od cyfr, to znak CYFRY umieszcza się jako drugi. Strefa może zawierać do 800 znaków dalekopisowych, to jest około 10 wierszy tabulogramu. Wewnątrz strefy nie mogą występować znaki PUSTA TASMA. Osiem lub więcej znaków linii lub spacji i znak LITERY kończy strefę.

Nową strefę można poprzedzić znakami PUSTA TASMA.



Rys.4 Odcinek taśmy perforowanej z zapisem strefowym

Kolejność umieszczania danych na taśmie określona jest w zapisie programu. Liczby rzeczywiste można podawać bez przecinka pozycyjnego, natomiast liczby całkowite nie można pisać jako rzeczywiste. Do zatrzymania maszyny w trakcie czytania danych /rozkazem CZLICZBE/ np. celem założenia kolejnego odcinka z danymi można w danych umieścić znak). Po starcie czytanie taśmy jest kontynuowane dalej. Tablica 1 podaje kod M2/CCIT.

W trzeciej linijce napisaliśmy

DRUKARKA

Oznacza to, że wydruk wyników będzie odbywał się na drukarce. Jest to zalecane wyjście. Pozostałe wyjścia podano w liście rozkazów.

W linijce trzeciej napisaliśmy rozkaz

CŁLICZBE A

Oznacza to, że maszyna ma przeczytać pierwszą daną i wpisać do komórki pamięci nazwanej A. A jest tu nazwa pojedyńczej komórki pamięci, czyli jest tak zwana nazwa zmiennej.

Rozróżniamy zmienne całkowite i rzeczywiste. Zmienne całkowite to takie, które mogą przyjmować tylko wartość liczb całkowitych. Nazwy zmiennych całkowitych muszą jako pierwszą literę mieć jedną z następujących:

G M I J K L M N O P Q R

a nazwy zmiennych rzeczywistych zaczynają się od liter

A B C D E S T U V W X Y Z

w siódmej linijce napisano

LINIA 2

Oznacza to, że wydruk ma się rozpocząć od nowej drugiej linii /jedna linia będzie pusta/. Wydruk jest podobny do wydruków uzyskiwanych na zwykłej maszynie do pisania, przy czym drukować można słowo za słowem, wiersz za wierszem i nie można cofać drukarki.

W ósmej linijce napisano

DRLICZBE(3:2) :A:B:C

Jest to rozkaz wydrukowania zawartości komórek pamięci o nazwach A, B, C. Wydruk przebiega tak jak na zwykłej maszynie do pisania. Dla każdej z podanych liczb zarezerwowało 3 znaki przed przecinkiem oraz 2 znaki po przecinku. Ponadto każda liczba ujemna zaopatrzona będzie w znak "minus", a liczba dodatnia będzie poprzedzona spacją.

Jeżeli na początku liczby wystąpią zera przed przecinkiem, to zostaną one zastąpione spacjami. W naszym przykładzie jest $A=8$, $B=7,2$, $C=-3,14$ co zostanie wydrukowane w następującej postaci:

 8,00 7,20 -3,14

Parametry (3:2) rozkazu DRLICZBE muszą być zapisane jako liczby całkowite^{w/}.

Gdybyśmy chcieli wydrukować wartość zmiennej całkowitej, to podawalibyśmy tylko jeden parametr, określający liczbę cyfr. Rozkaz taki miałby np. postać

DRLICZBE(5):I

co dla $I=-100$ spowodowałby wydruk w postaci

 -100

Rozkaz DRLICZBE może dotyczyć wydruku wartości jednej, jak również wielu zmiennych. Można więc napisać

DRLICZBE(5):I:G:R

W dziesiątej linijce napisano rozkaz

STOP

Jest to rozkaz zatrzymania pracy maszyny. Maszyna przejdzie do wykonywania następnego rozkazu po naciśnięciu przycisku START. Jako zasadę należy przyjąć poprzedzanie rozkazu STOP rozkazem LINIA 1

w/ Jeżeli liczba pozycji całkowitych w rozkazie DRLICZBE okaże się za mała wówczas nastąpi wydruk liczby w formacie wykładniczym z zastąpieniem liczby 10 przez / np. rozkazem DRLICZBE(1:5) liczba 3584,052=0,3584052•10⁴ zostanie wydrukowana w postaci ,358405200/4

W ostatniej linijce zapisane są dane do tego programu.
Kolejne liczby w zapisie muszą być oddzielone co najmniej
jednym symbolem spacja, linia lub powrót^{m/}

2.2. Obliczenie wartości wyrażenia algebraicznego

Przykład F2:

Napisać program, w którym maszyna ma wczytać pięć liczb
 A, B, C, D, X i wydrukować wynik obliczenia wartości

$$Y = \frac{AX^2+BX+C}{X-D+2}$$

Rozwiązanie:

F2

```
WEJSCIE 3 DRUKARKA
CZLICZBE A CZLICZBE B CZLICZBE C
CZLICZBE D CZLICZBE X
Y=(A.X.X+B.X+C)/(X-D+2)
LINIA 5 DRLICZBE(5:5):Y
LINIA 1 STOP
START F2
```

m/ Na taśmie perforowanej dane muszą zaczynać się od nowej strefy. /Patrz odnośnik na str. 12 /

Objaśnienie:

W tym przykładzie widzimy, że w jednym wierszu można napisać po kilka rozkazów. Będą one wykonywane w kolejności napisania^{w/}.

W piątej linijce podano wzór na obliczenie wartości Y. Wzór ten stanowi zapis formuły przy zastosowaniu znaków:

- + dla dodawania
- dla odejmowania
- / dla dzielenia
- * dla mnożenia

Kolejność działań wskazują nawiasy okrągłe (). Najpierw wykonuje się obliczenia wartości wyrażeń zawartych w największej liczbie nawiasów. W obrębie jednego nawiasu najpierw wykonuje się mnożenie liczb, potem dzielenie, a następnie dodawanie i odejmowanie, w kolejności napisania.

Aby działania zostały wykonane poprawnie, wszystkie wartości po prawej stronie znaku równości muszą być określone wcześniej. Oznacza to np., że najpierw trzeba wczytać wartości A,B,C,D,X i potem obliczać wartość Y. Po lewej stronie znaku równości napisana jest nazwa tylko jednej komórki Y, do której należy przesyłać wynik obliczenia wartości wyrażenia algebraicznego stojącego po prawej stronie znaku równości.

Znak równości = w języku SAKO ma nieco inne znaczenie niż w zwykłej algebrze. Oznacza on przesłanie wartości wyrażenia występującego po prawej stronie znaku równości na zmienną występującą po lewej stronie znaku równości - stąd też wszystkie rozkazy algebraiczne mogą mieć postać

zmienna = wyrażenie

Wzór na obliczenie wartości Y można było zaprogramować także w inny sposób, np.:

w/ Zbędne znaki: spacja, linia, powrót karetki są pomijane. Natomiast nie wolno na taśmie zostawiać znaków "pustej taśmy".

$$Y = ((A \cdot X + B) \cdot X - C) / (X - D + 2)$$

Jak widać, nawiasy można stosować wielokrotnie. Można także rozbić obliczenie wartości Y na kilka działań pośrednich, np.:

$$U = A \cdot X \cdot X + B \cdot X + C$$

$$V = X - D + 2$$

$$Y = U / V$$

Program F2 można zapisać także w postaci:

F2A WEJSCIE 3 DRUKARKA LINIA
CZLICZBE A CZLICZBE B CZLICZBE C
CZLICZBE D CZLICZBE X
LINIA 5
DRLICZBE(4:5):(A \cdot X \cdot X + B \cdot X + C) / (X - D + 2)
LINIA 1 STOP
START F2A

W tym wariantie wartość Y nie była przesyłana do żadnej komórki, ale od razu poleciliśmy ją wydrukować. Widac, że argumentem rozkazu DRLICZBE może być także wyrażenie algebraiczne.

Przykład F3

Wczytaj liczby A i B oraz wydrukuj osobno część całkowitą i część ułamkową otrzymaną w wyniku dzielenia A/B

F3 WEJSCIE 3 DRUKARKA
CZLICZBE A CZLICZBE B
Y=A/B I=Y C=Y-I
LINIA 1
DRLICZBE(5):I DRLICZBE(5:4):C
LINIA 1 STOP
START F3

Objaśnienia wymaga trzeci wiersz programu. Najpierw obliczamy wartość ilorazu A/B i przesyłamy ją do komórki Y, której nazwa wskazuje, że jest to zmienna rzeczywista. Następnie wartość Y przesyłamy do komórki I, która jest zmienną całkowitą. Przy przesyłaniu liczb niecałkowitych do zmiennych całkowitych, liczby te ulegną zaokrągleniu do najbliższej wartości całkowitej i ta wartość zostanie zapamiętana. Dlatego w komórce C w następnym działaniu

znajdzie się część ułamkowa liczby.

Uwaga: zaokrąglenie do liczby całkowitej jest nieokreślone w przypadku liczb kończących się na połówkę. Np. nie można z góry przewidzieć czy 2,5 zostanie zaokrąglone do wartości 2 czy wartości 3.

2.3. Funkcje standardowe

Przykład F4

Ułożyć program w którym maszyna wczyta a i α i wydrukuje wynik obliczenia wartości wyrażenia

$$y = a \sin (\alpha + \pi)$$

przy czym $\pi = 3,141592$

Rozwiążanie

F4
WEJSCIE 3 DRUKARKA
CZLICZBE ALFA CZLICZBE A
 $Y=A \cdot \text{SIN}(ALFA+3,141592)$
LINIA 1 DRLICZBE(2:4) :Y
LINIA 1 STOP
START F4

Zmienną α występującą we wzorze zastąpiliśmy nazwą ALFA, gdyż wśród liter SAKO nie ma liter alfabetu greckiego.

Pojawia się także funkcja standardowa SIN, której argument ALFA+3,141592 ujęty w nawiasy pisze się po nazwie funkcji.

Pozostałymi funkcjami standardowymi języka SAKO są

SIN(X)	
COS(X)	X - w radianach
ARCTG(X)	
LOG(X)	oznacza $\ln x$ dla $x > 0$
EXP(X)	oznacza e^x
ABS(X)	oznacza $ X $ wartość bezwzględna X/
PWK(X)	oznacza \sqrt{x}

Wartości wszystkich funkcji standardowych traktowane są jako rzeczywiste.

Argumenty funkcji pojedynczej zmiennej mogą zawierać w sobie także tę funkcję. Np. dopuszczalne są zapisy

$\text{SIN}(\text{SIN}(X))$

$X+\text{SIN}(2 \cdot X+\text{COS}(2 \cdot X))$

esłone
można
artości

2.4. Funkcje definiowane

Przykład F5

Ułożyć program na obliczenie wartości wyrażenia

$$Y = \sin(x - \varphi) \cdot \operatorname{sh}(x - \varphi) + A e^x \operatorname{sh}(2 \cdot x)$$

dla $x=0,2$ $\varphi=0,5$ $A=2.$

Rozwiązańe

rukuje

```
FSH:X  
FSH=(EXP(X)-EXP(-X))/2  
WROC
```

Podprogram funkcji
definiowanej

```
F5  
WEJSCIE 3 DRUKARKA  
CZLICZBE X CZLICZBE AFI CZLICZBE A  
Y=SIN(X-AFI).FSH(X-AFI)+A.EXP(X).FSH(2.X).  
LINIA 1 DRLICZBE(3:3):Y  
LINIA 1 STOP  
START F5
```

Program
główny

0,2	0,5	2,0
-----	-----	-----

Dane

Objaśnienie:

Funkcja sh /sinus hiperboliczny/ występująca w programie nie występuje wśród funkcji standardowych. W związku z tym sposób jej obliczenia należy zdefiniować w oddzielnym podprogramie zaczynającym się od nazwy funkcji FSH, po której występuje dwukropki i nazwa argumentu funkcji. Nazwa ta używana jest w samym podprogramie, w którym podaje się sposób obliczenia funkcji. Wszystkie nazwy występujące w podprogramie zachowują znaczenie wewnętrz tego podprogramu i nie kolidują z nazwami stosowanymi w pozostałych partiach programu. Bezpośrednio po rozkazie podającym sposób obliczania funkcji musi wystąpić rozkaz WROC.

Uwaga 1: Nazwa funkcji zaczyna się zawsze od litery F.

Uwaga 2: Jeśli chcemy korzystać z funkcji definiowanej, to

jej podprogram musi być napisany wcześniej niż odwołanie do tej funkcji.

Uwaga 3: Podprogram można było napisać w postaci kilku rozkazów np.:

```
FSH:X  
A=EXP(X)    B=1/A  
FSH=(A-B)/2  
WROC
```

2.5. Definiowanie wielu funkcji

Mожет zdarzyć się, że w jednym programie trzeba zdefiniować kilka funkcji. Wtedy każdą z nich określamy oddzielnym podprogramem.

Przykład F7

Ułożyć program na obliczenie wartości wyrażenia

$y = \operatorname{tg} x - 2\operatorname{tgh} 2x + \sinh 3x$
dla wczytanej wartości x

Rozwiązanie

```
FSH:X  
FSH=(EXP(X) -EXP(-X))/2  
WROC
```

```
FCH:X  
FCH=(EXP(X) +EXP(-X))/2  
WROC
```

```
FTG:X  
FTG=FSH(X)/FCH(X)  
WROC
```

```
FTG:X  
FTG=SIN(X) /COS(X)  
WROC
```

```
F7  
WEJSCIE 3 DRUKARKA  
CZLICZBE X  
Y=FTG(X) -2.TGH(2.X)+FSH(3.X)  
DRLICZBE(2:6):Y LINIA 1  
STOP  
START F7
```

Objaśnienie

W naszych podprogramach zdefiniowaliśmy funkcje hiperboliczne:

$$\text{sh } x = \frac{e^x - e^{-x}}{2}$$

sinus hiperboliczny

$$\text{ch } x = \frac{e^x + e^{-x}}{2}$$

cesinus hiperboliczny

$$\text{th } x = \frac{\text{sh } (x)}{\text{ch}(x)}$$

tangens hiperboliczny

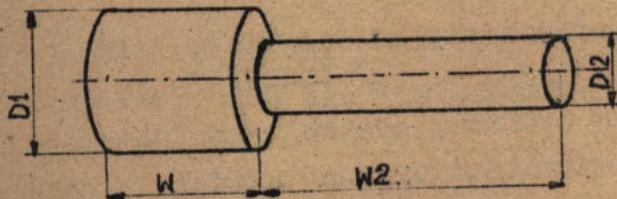
W podprogramie tangensa hiperbolicznego korzystaliśmy ze zdefiniowanych innymi podprogramami funkcji $\text{sh } x$ i $\text{ch } x$. Podprogramy FSH i FCH muszą być napisane przed podprogramem FTGH.

Podprogramy wywoływane w programie głównym nazywamy podprogramami I stopnia, wywołane zaś w podprogramie I stopnia - programami II stopnia itd. O stopniu podprogramu decyduje więc jego wywołanie. Np. podprogram FSH w naszym przykładzie był raz podprogramem I stopnia, a przy wywołaniu go w podprogramie FTG - podprogramem II stopnia. Stopień podprogramu nie może przekraczać 8.

2.6. Funkcje definiowane wielu zmiennych

Przykład F8

Obliczyć objętość V bryły składającej się z dwóch walców o średnicach D_1 i D_2 oraz o wysokościach W i W_2 , przy czym $W_2=2 \cdot W$.



Rozwiążanie

Rys. 5

FOBJETOSC:D:W
FOBJETOSC=3,14159.D.D.W/4
WROC

Pedprogram

F8

```
WEJSCIE 3 DRUKARKA  
CZLICZBE D1 CZLICZBE D2 CZLICZBE W  
V=FOBJETOSC(D1:W)+FOBJETOSC(D2:W+W)  
LINIA 1 DRLICZBE(3:4):V  
LINIA 1 STOP  
START F8
```

Program
główny

W podprogramie podaliśmy sposób obliczania objętości walca.

W nagłówku podprogramu FOBJETOSC:D:W po nazwie podprogramu

FOBJETOSC podajemy po dwukropku listę argumentów podprogramu, oddzielonych pomiędzy sobą dwukropkami.

Przy obliczaniu objętości V w wywołaniu podprogramu FOBJETOSC D1:W na parametry D i W zostaną podstawione, zgodnie z kolejnością zapisu, wartości zmiennych D1 i W, a w wywołaniu FOBJETOSC(D2:W+W) wartość zmiennej D2 i wartość wyrażenia (W+W)

2.7. Skok bezwarunkowy

Przykład F9

Ułożyć program, który po każdym naciśnięciu przycisku START drukuje w nowym wierszu liczbę 5.

```
F9  
DRUKARKA  
ET) DRLICZBE(1) :5  
LINIA 1 STOP  
SKOCZ DO ET  
START F9
```

Objaśnienie:

Trzeci wiersz zaczyna się od etykiety ET) Etykietą może być dowolna nazwa, zaczynająca się od litery różnej od F, zakończona nawiasem zamykającym. Do piątej linijki maszyna będzie wykonywać program zgodnie z kolejnością zapisu. Następnie rozkaz

SKOCZ DO ET

spowoduje przejście do rozkazu poprzedzonego etykietą ET.

2.8. Skok warunkowy, napis

Przykład F10

Ułożyć program dla rozwiązań równania kwadratowego

$$ax^2 + bx + c = 0$$

Rozwiązanie

```

F10 WEJSCIE 3 DRUKARKA
POCZ)
CZLICZBE A CZLICZBE B CZLICZBE C
DELTA=B.B-4.A.C
LINIA 1
GDY DELTA:0 SKOCZ DO UJEMNA:ZEROWA:DODATNIA
UJEMNA) NAPIS,BRAK PIERWIASTKOW RZECZYWISTYCH #
SKOCZ DO KONIEC
ZEROWA) NAPIS X0=X1=# DRLICZBE(5:5):-B/(2.A)
SKOCZ DO KONIEC
DODATNIA) NAPIS X0=# DRLICZBE(5:5):(B+PWK(DELTA))/(2.A)
NAPIS X1=# DRLICZBE(5:5):(B-PWK(DELTA))/(2.A)
KONIEC)
LINIA 1 STOP
SKOCZ DO POCZ
START F10

```

W szóstym wierszu napisaliśmy nowy rozkaz GDY DELTA:0
SKOCZ DO UJEMNA:ZEROWA:DODATNIA

Słowa UJEMNA, ZEROWA, DODATNIA są tutaj etykietami. Instrukcja ta poleca porównać dwie wartości wyrażeń stojących po słowie GDY i rozdzielonych dwukropkiem. Jeżeli wartość wyrażenia stojącego po lewej stronie dwukropka jest mniejsza od wartości wyrażenia po prawej stronie, należy przejść do wykonywania rozkazu poprzedzonego etykietą wypisaną po rozkazie SKOCZ DO. W przypadku równości wartości wyrażeń porównywanych, należy wykonywać instrukcje poprzedzone etykietą środkową i w przypadku, gdy wartość prawej strony jest mniejsza od lewej - skok do ostatniej etykiety.

W naszym przypadku:

gdzie $\text{DELTA} < 0$ skok do etykiety UJEMNA

$\text{DELTA} = 0$ skok do etykiety ZEROWA

$\text{DELTA} > 0$ skok do etykiety DODATNIA

Jeżeli w miejsce którejkolwiek etykiety umieścimy 0 (zero)

- oznaczać to będzie skok do następnego rozkazu.

Rozkaz

NAPISU

powoduje wydruk tekstu napisanego po spacji za słowem NAP bez znaku # kończącego tekst. Początkowe spacje tekstu są pomijane. Po wykonaniu rozkazów poprzedzonych etykietą UJEMNA zachodzi potrzeba ominięcia kilku rozkazów. Dokon się tego za pomocą rozkazu skoku bezwarunkowego SKOCZ DO KONIEC

Przykład F11

Obliczyć wartość wyrażenia $\pi/4$ jaka sumę szeregu

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots$$

z dokładnością do 1/100.

Rozwiązanie możnaby napisać w postaci:

$$Y = 1 - 1/3 + 1/5 - 1/7 + 1/9$$

dałoby to wynik z dokładnością do 1/9. Gdybyśmy chcieli uzyskać wyższą dokładność, to należałoby wypisać bardzo dużo wyrazów. Zamiast tego wygodniej będzie napisać następujący program:

```
Y=0  
Y=Y+1  
Y=Y-1/3  
Y=Y+1/5  
Y=Y-1/7  
Y=Y+1/9
```

Tutaj użyliśmy wielokrotnie tej samej nazwy Y pamiętając, że w rozkazie algebraicznym znak równości = oznacza przesunięcie wartości wyrażenia stojącego po prawej stronie na zmieniąną występującą po lewej stronie, więc rozkaz

```
Y=Y+1
```

oznacza, że nowa wartość Y równa się poprzedniej wartości Y plus 1. Zmienna występująca po obu stronach znaku = oznacza różne wielkości /zapisanego równania nie można uprościć przez Y/.

Ponieważ liczba dodawana po prawej stronie zmienia się w sposób regularny, zapiszemy program w nowej postaci wykorzystu-

jąc tą regularność

$Y=0$	$L=1$	$M=1$
$Y=Y+L/M$	$L=-L$	$M=M+2$

Tutaj w komórkach X, L i M zapisuje się coraz to nowe wartości. Ponieważ ostatnie rozkazy powtarzają się możemy to wykorzystać zapisując program w następującej postaci:

$Y=0$ $L=1$ $M=1$
LICZ) $Y=Y+L/M$ $L=-L$ $M=M+2$
GDY M:100 SKOCZ DO LICZ:0:0

Rozkazy zapisane w drugiej linijce będą wykonywane tak dugo, aż osiągnie się dokładność 1/99 /dla M=101 powtórzenie już nie nastąpi/. Pełny program może mieć postać

F11 DRUKARKA
 $Y=0$ $L=1$ $M=1$
LICZ) $Y=Y+L/M$ $L=-L$ $M=M+2$
GDY M:100 SKOCZ DO LICZ:0:0
DRLICZBE(2:3) :Y
LINIA 1 STOP
START F11

Uwaga: Jeżeli argumenty rozkazu GDY przyjmują wartości rzeczywiste, to skok do odpowiedniej etykiety może nie być jednoznacznie określony. Wynika to z zaokrąglenia rachunków. Np. okazuje się, że równość $0,1 + 0,1 + 0,1 = 0,3$ w układzie dwójkowym, w którym maszyna liczy nie jest ścisłe spełniona.

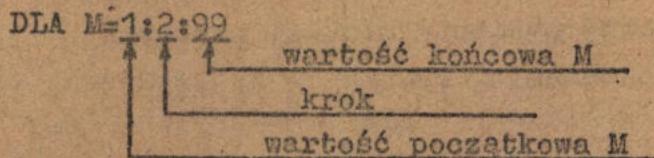
2.9. Powtórzenia

Dalsze uproszczenia omawianego wyżej programu obliczania wartości $\pi/4$ polega na wprowadzeniu specjalnych rozkazów powtarzania:

$Y=0$ $L=1$

DLA M:1:2:99
 $Y=Y-L/M$ $L=-L$
POWTORZ M

Rozkazy DLA i POWTORZ są ze sobą związane. Po nich następuje nazwa tej samej zmiennej M. Odcinek programu między tymi rozkazami będzie wielokrotnie powtarzany, przy czym wartość zmiennej M będzie zmieniała się tak jak to określają parametry występujące po znaku = w rozkazie



Pierwsza wartość M=1; następnie będzie ona wzrastać za każdym powtórzeniem o 2. Powtarzanie skończy się, gdy wartość M będzie różniła się od 99 mniej niż o pół kroku równego 2, a więc dla M=99. Z powyższego wynika, że część programu objęta rozkazami DLA i POWTORZ tworzy jedną złożoną instrukcję zwaną cyklem.

Ostateczny program ma postać:

```
F12POWTORZENIA
WEJSCIE 3 DRUKARKA LINIA 1
ET) Y=0 L=1
    DLA M=1:2:99
    uY=Y+L/M      L=-L
    POWTORZ M
    DRLICZBE (2:3) :Y   LINIA 1
    STOP SKOCZ DO ET
    START F12POWTORZENIA
```

Uwaga 1:

W rozkazie DLA wartość początkowa, krok i wartość końcowa zmiennej M można określać również przy pomocy wyrażeń algebraicznych. Wtedy wartość tych wyrażeń nie może ulegać zmianie w trakcie wykonywania rozkazu DLA.

2.10. Wydawnictwo

Przykład F13:

Obliczyć wartość funkcji $y = a \cdot x + b$ dla wczytanych współczynników a i b. Wyniki wydrukować w postaci tablicy podanej na rysunku 6.

PRZYKŁAD F13 WYDAWNICTWO
TABLICE FUNKCJI Y=A.X+B

LP.	X	Y
1	-DDDD, DDD	-DDDD, DDD
2	=====	=====
3	=====	=====
4	=====	=====
⋮	⋮	⋮
p	=====	=====

Rys. 6

Litery D oznaczają ilość zarezerwowanych pozycji dla drukowanych liczb.

A oto program:

```
F13:C(30)
WEJSCIE 3      DRUKARKA
E) STRONA 12
RAMKA 3
KOPIA=-#621161040003
CZTEKST C      LINIA 2
CZLICZBE A      CZLICZBE B      CZLICZBE N
DRTEKST C
NAPIS   TABLICE FUNKCJI Y=#
DRLICZBE(4:3):A
NAPIS X+A,DRLICZBE(4:3):B
LINIA 2
NAPIS LP. I  X      Y
-----
#          I=1:1:N
DRLICZBE(2):I
NAPIS I - #
CZLICZBE X  DRLICZBE(4:3):X  SPACJA 2
DRLICZBE(4:3):(A.X+B)
LINIA 1
POWTORZ I
DOKONCZ
STOP SKOCZ DO E
START F13
```

i dane:

"PRZYKŁAD F13 WYDAWNICTWO #

500, 11 20, 222 10

1, 111 2, 22 3, 3 4 5 6 7 8 9 10

Za nazwą programu F13 po dwukropku napisaliśmy C(30)

spowoduje to zarezerwowanie bloku 30 komórek o nazwie C.
Rozkaz

CZTEKST C

spowoduje przeczytanie tekstu występującego na taśmie z danymi. Tekst ten kończy się znakiem #. W naszym przykładzie danych tekst jest następujący:

"PRZYKŁAD F13 WYDAWNICTWO #.

Tekst ten zapisze się do komórek bloku C w następujący sposób /po sześć znaków w komórce/:

C₀ = [] PRZY

C₁ = KLADE#

C₂ = F13 WY

C₃ = DAWNICT

C₄ = TWOI# []

C₅ = [] [] []

.....

a pozostałe komórki bloku C nie będą zmienione.

Blok C musi być zadeklarowany w nagłówku programu.

Kolejny rozkaz

DRTEKST C

oznacza polecenie wydruku tekstu, wczytanego poprzednio rozkazem CZTEKST C. Symbol # kończący tekst nie zostanie wydrukowany.

Uwaga: W danych zawierających tekst, przed tekstem nie może tego być komentarza ujętego w znaki = # gdyż zostałby on wczytany jako tekst.

Za etykietą E) umieściliśmy trzy rozkazy dotyczące arkusza wydawniczego.

Strona 12

jest polecienniem druku 12 wierszy na jednej stronie.

RAMKA=3

jest polecienniem pozostawienia 3 wolnych wierszy przed treścią i po treści strony.

KOPIA=-#621161040003

powoduje podzielenie strony wyników na dwie części, przy czym treść części pierwszej, umieszczonej na pozycjach od 0 do 59, zostanie przedrukowana na pozycjach od 68 do 127. Rozkaz KOPIA=0 oznacza likwidację wydruku kopii.

Treść nagłówka napisaliśmy posługując się rozkazem NAPIS, przy czym do utworzenia linii poziomej użyliśmy znaku - a do utworzenia linii pionowej - znaku litery I, gdyż nie dysponujemy znakiem | pionowej kreski.

Rozkaz

SPACJA 2

oznacza wydruk dwóch spacji po ostatnio wydrukowanym znaku.
Rozkaz

DOKONCZ

oznacza dokonczenie treści drukowanej strony wolnymi wierszami.

2.11. Zmienne indeksowane

2.11.1. Zmienne indeksowane o jednym wskaźniku

Przykład F14:

Dane są następujące liczby:

22,55 22,88 22,32 23,05 24,12

Należy obliczyć ich wartość średnią.

Rozwiązanie

Zespół tych liczb wprowadzimy do bloku komórek pamięci maszyny, któremu nadamy jedną nazwę np. A. Dowolny element tego bloku oznaczmy nazwą A, za którą w nawiasach () podamy indeks określający miejsce komórki w bloku. Np. A(0) - będzie oznaczać komórkę zerową, A(I) - będzie I-tą komórką

w bloku A.

Indeks może przyjmować tylko wartości całkowite i dodatnie. Utworzone w ten sposób zmienne indeksowane mogą być zmiennymi całkowitymi lub rzeczywistymi - decyduje o tym pierwsza litera nazwy /patrz 2.1/.

Wczytanie podanych liczb do kolejnych komórek bloku A oraz obliczenie sumy tych liczb możemy zapisać w postaci programu:

SUMA=0

CZLICZBE A(1)

SUMA=SUMA+A(1)

CZLICZBE A(2)

SUMA=SUMA+A(2)

CZLICZBE A(5)

SUMA=SUMA+A(5)

Przebieg tych obliczeń dla N wyrazów realizuje w pętli powtarzającą się następujący program:

```
F14:A(20)
WEJSCIE 3 DRUKARKA
E)CZLICZBE N
SUMA=0
DLA I=1:1:N
CZLICZBE A(I)
SUMA=SUMA+A(I)
POWTORZ I
SREDNIA=SUMA/N
LINIA 1 DR LICZBE(4:5):SREDNIA
LINIA 1 STOP SKOCZ DO E
START F14
```

Dane:

5 22,55 22,88 22,32 23,05 24,12

Za nazwą programu F14, po dwukropku napisaliśmy A(20).

Deklaracja ta spowoduje zarezerwowanie w tym programie dla bloku A - 20 kolejnych komórek;

A(0), A(1), A(2), ..., A(18), A(19) .

W przypadku bloków zmiennych konieczna jest deklaracja wielkości tych bloków.

Dane do programu zgodnie z jego zapisem, zostały poprzedzo-

ne liczbą całkowitą określającą liczbę składników dla obliczenia średniej.

2.11.2. Rozkaz WARTOSC, KOMENTARZE

Przykład F15:

Obliczyć i wydrukować wartość wielomianu

$$y = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3$$

dla wczytanej wartości x. Współczynniki wielomianu są równe:

$$a_0 = 2,0 \quad a_1 = -1,0 \quad a_2 = 3,87 \quad a_3 = 5,21$$

Rozwiązańe

Podany wielomian możemy przedstawić w postaci:

$$y = a_0 + x \cdot (a_1 + x \cdot (a_2 + x \cdot (a_3)))$$

Podstawiając do komórki Y kolejno obliczone wartości wyrażeń zawartych w największej liczbie nawiasów otrzymujemy

$$\begin{aligned} Y &= 0 \\ Y &= A(3) + X \cdot Y \\ Y &= A(2) + X \cdot Y \\ Y &= A(1) + X \cdot Y \\ Y &= A(0) + X \cdot Y \end{aligned}$$

Powyższy program można zapisać krótko:

```
Y=0
→ DLA I=3:-1:0
  Y=A(I)+X.Y
POWTORZ I
```

Program rozwiązujący zadanie będzie mieć postać:

```
F15:A(4)
WEJSCIE 3 DRUKARKA LINIA 1
'Nadanie wartości zmiennym bloku A'
WARTOSC A=2,0:-1,0:3,87:5,21
E)CZLICZBE X
'obliczenie wartości wielomianu według
schematu Hornera'
Y=0
→ DLA I=3:-1:0
  Y=A I +X.A(I)
POWTORZ I
DRLICZBE(5:5):Y
LINIA 1 STOP
SKOCZ DO E
START F15
```

Dane:

```
=SCHEMAT HORNERA
#
27,3 14,8
```

Rozkaz

WARTOSC A=2,0:-1,0:3,87:5,25

powoduje nadanie kolejnym zmiennym bloku A A(0), A(1), A(2) kolejnych wartości zapisanych po znaku = i rozdzielonych znakiem : jest to równoważne rozkazom A(0)=2,0 A(1)=-1,0 itd.

W wierszu trzecim i szóstym umieściliśmy komentarze programu ujęte w apostrofy. W komentarz może wchodzić ciąg dowolnych znaków, nie zawierający apostrofu. Komentarze nie zmają sensu programu i przy jego tłumaczeniu przez translator są całkowicie pomijane.

W danych umieściliśmy komentarz danych

```
=SCHEMAT HORNERA
#
```

Przy czytaniu danych /liczb/ nastąpi przedruk tekstu bez obejmujących go znaków = i #. Oczywiście wewnątrz tekstu nie może wystąpić znak # oznaczający koniec komentarza. Podany tekst zostanie wydrukowany w postaci

SCHEMAT HORNERA

Dalszy druk nastąpi od miejsca oznaczonego znakiem # - to jest z nowej linii.

2.11.3. Zmienna indeksowana w podprogramie

Wczytać współrzędne wektorów \vec{U} , \vec{V} , \vec{W} i znaleźć ich wypadkową $\vec{Z} = \vec{U} + \vec{V} + \vec{W}$

Jak wiadomo wypadkowa C dwóch wektorów \vec{A} i \vec{B} ma następujące składowe:

$$\begin{aligned}C(0) &= A(0) + B(0) \\C(1) &= A(1) + B(1) \\C(2) &= A(2) + B(2)\end{aligned}$$

FSUMA
DLA I=
G(I)=
POWTOR
WROC

W nagłówku
występuje
rezerwowa
Po pod-
cy zadania

F
WI
?V
D?
C
C
C
P
?

F
E
?

Dane

Rozkazy
oznacza
 $\vec{Z} = \vec{U} + \vec{V} + \vec{W}$
I tu
że na
nazwa

Podprogramu znajdowania tej sumy będzie miał postać:

```
FSUMA:A#:B#:C#
DLA I=1:1:2
C(I)=A(I)+B(I)
POWTORZ I
WROC
```

W nagłówku tego podprogramu za nazwami bloków A, B, C występują znaki #. Oznacza to, że w tym podprogramie nie rezerwuje się miejsc dla tych bloków.

Po podprogramie możemy napisać właściwy program rozwiązuający zadanie:

```
F16:U(3):V(3):W(3):Z(3)
WEJSCIE 3 DRUKARKA
'WCZYTANIE DANYCH'
DLA I=0:1:2
CZLICZBE U(I)
CZLICZBE V(I)
CZLICZBE W(I)
POWTORZ I
'SUMOWANIE'
FSUMA(U#:V#:Z#)
FSUMA(Z#:W#:Z#)
'WYDRUK'
DRLICZBE(5:5):Z(0):Z(2)
LINIA 1 STOP
START F16
```

Dane:

1	1	1,001
2	2	2,002
3	3	3,003

Rozkaz

FSUMA(U#:V#:Z#)

oznacza wywołanie podprogramu FSUMA, to jest obliczenie
 $Z=U+V$

I tutaj po nazwach U, V, Z występuje symbol #. Oznacza to, że w podprogramie FSUMA, zgodnie z kolejnością zapisu, na nazwę A podstawiamy nazwę U#, na nazwę B# podstawiamy nazwę V#, i na nazwę C# podstawiamy nazwę Z#. Wobec tego

działania określone na nazwach A#, B#, C# zostaną wykonane na podstawionych nazwach U#, V#, Z#.

Rozkaz

FSUMA (Z#:W#:Z#)
pozwala wykonać działania

Z=Z+W

przy zastosowaniu tego samego podprogramu, odniesionego do innych bloków, o innych niż poprzednio nazwach.

Wszystkie argumenty podprogramu, na które podstawa się wartości obliczone w tym podprogramie muszą być w nagłówku podprogramu oznaczone symbolem #.

Uwaga: Charakter nazw występujących w odwołaniu się do podprogramu musi być taki sam jak w nagłówku podprogramu.

Np. dla podprogramu FSUMA błędne byłyby wywołania:

FSUMA(U:V:Z) - brak znaków #

FSUMA(J#:Y#:Z#) - J# jest nazwą bloku całkowitego, podczas gdy A# jest nazwą bloku rzeczywistych

FSUMA(Y#:V#:Z#) - Y# jest nazwą niezadeklarowanego bloku

2.11.4. Zmienna indeksowana o dwóch wskaźnikach

Rozkaz STRUKTURA, Cykl w cyklu

Przykład F17A:

Wczytać elementy zawarte w tablicy

a ₀₀	a ₀₁	a ₀₂	a ₀₃
a ₁₀	a ₁₁	a ₁₂	a ₁₃
a ₂₀	a ₂₂	a ₂₃	a ₂₃

gdzie a₀₀, a₀₁, ..., a₂₃ są liczbami danymi.

Oznaczmy element tablicy przez A(I:K), jest to zmienna indeksowana o dwóch wskaźnikach, gdzie indeks I - odpowiada numerowi wiersza, a indeks K - numerowi kolumny. Wartość indeksu I zmienia się od zera do dwóch, wartość indeksu K od zera do trzech.

Wczytanie pierwszego wiersza równoważne jest wykonaniu rozkazów:

CZLICZBE A(0:0)
CZLICZBE A(0:1)
CZLICZBE A(0:2)
CZLICZBE A(0:3)

Można to zapisać w postaci cyklu

→ DLA K=0:1:3
 └ CZLICZBE A(0:K)
 └ POWTORZ K

lub biorąc pod uwagę że pierwszy indeks I=0 przedstawić w ogólnej postaci:

I=0
→ DLA K=0:1:3
 └ CZLICZBE A(I:K)
 └ POWTORZ K

Podobnie dla wczytania drugiego wiersza

I=1
→ DLA K=0:1:3
 └ CZLICZBE A(I:K)
 └ POWTORZ K

i dla trzeciego wiersza

I=2
→ DLA K=0:1:3
 └ CZLICZBE A(I:K)
 └ POWTORZ K

W tych trzech fragmentach powtarzają się cykle

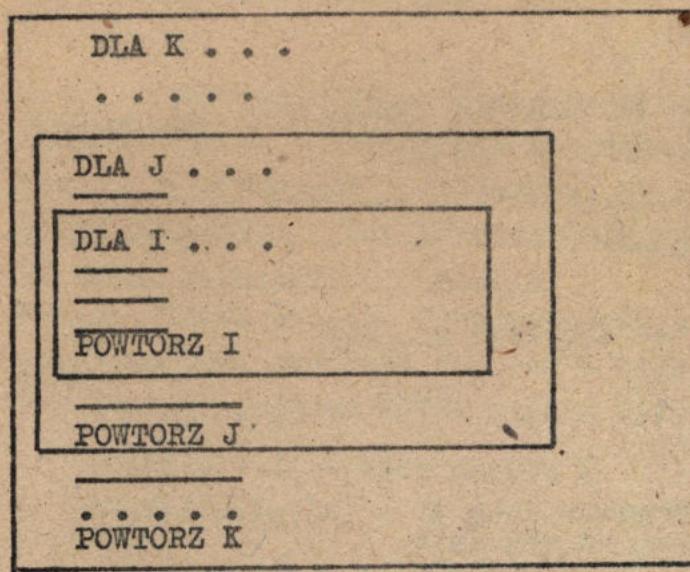
→ DLA J=0:1:2
 └ CZLICZBE A[I:J]
 └ POWTORZ J

dla indeksu I przyjmującego kolejno wartości 0, 1, 2.

Zatem te trzy kolejne fragmenty programu możemy zastąpić następującym:

→ DLA I=0:1:2
 └ DLA J=0:1:2
 └ CZLICZBE A(I:J)
 └ POWTORZ J
 └ POWTORZ I

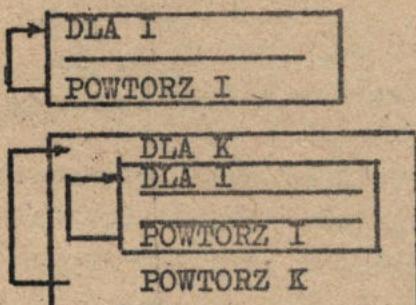
We fragmencie tym cykl DLA J..... POWTORZ J wchodzi w skład cyklu DLA I POWTORZ I. Taki zapis jest poprawny. Cykl może wchodzić kolejno w skład co najwyżej dziewięciu cykli:



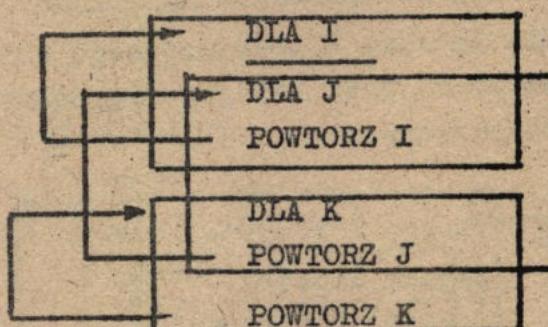
W podanych przykładach narysowano ramki obejmujące każdy cykl. Jeżeli w programie występuje kilka cykli, to odpowiadające im ramki mogą się obejmować lub występować jedna za drugą, nie mogą natomiast przecinać się.

A oto przykłady dobrego i błędного zapisu programu:

Dobrze

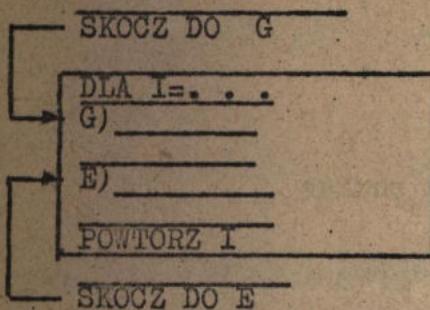


Źle



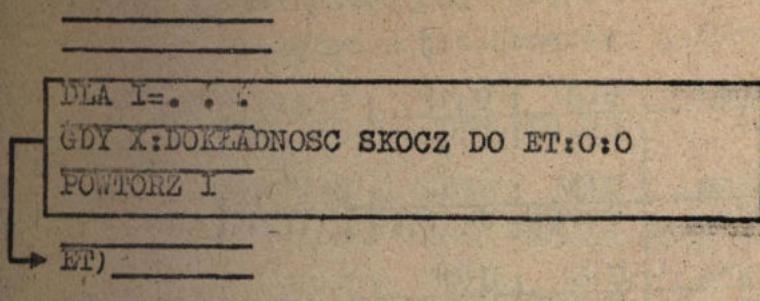
Ponieważ cykl stanowi jedną złożoną instrukcję, wobec tego nie można "wchodzić" do wnętrza cyklu z zewnątrz.

Np. błędny będzie fragment programu:



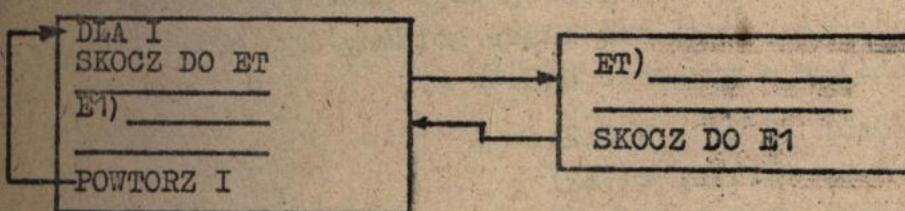
Zle

Niedozwolone są również wyjścia z cyklu przed wykonaniem wszystkich działań określonych liczbą powtórzeń, jak np. w programie:



Zle

Można natomiast "wyjść" z cyklu w celu wykonania określonego podprogramu, ale i wrócić,



bowiem podprogram oznaczony etykietą ET można traktować jak gdyby był on umieszczony wewnątrz cyklu.

Program wczytywania wartości zmiennej indeksowanej można zapisać w postaci:

```
F17A:U(9)
WEJSCIE 3 DRUKARKA LINIA 1
POCZ)
STRUKTURA U(3:3)
DLA I=0:1:2
DLA J=0:1:2
WCZLICZBE U(I:J)
```

POWTORZ J
POWTORZ I

⋮ ⋮ ⋮ ⋮ ⋮

Napisany w pierwszym wierszu rozkaz
F17A:U(9)

powoduje zarezerwowanie w podprogramie F17 dziewięciu następujących komórek:

A(0), A(1), A(2), A(3), A(4), A(5), A(6), A(7), A(8), A(9)

Jak widać, komórki te odróżniają się za pośrednictwem pojedynczego indeksu zmieniającego się od 0 do 9

Rozkaz STRUKTURA U(3:3) przyporządkowuje tym komórkom dwa indeksy według następującej zasady:

Jednoindeksowa	U(0)	U(1)	U(2)
Dwuindeksowa	U(0:0)	U(0:1)	U(0:2)
Jednoindeksowa	U(3)	U(4)	U(5)
Dwuindeksowa	U(1:0)	U(1:1)	U(1:2)
Jednoindeksowa	U(6)	U(7)	U(8)
Dwuindeksowa	U(2:0)	U(2:1)	U(2:2)

Jak widać indeksy przyjmują tyle różnych wartości ile rezerwuje rozkaz STRUKTURA, przy czym pierwszy indeks określa numer wiersza, drugi zaś numer kolumny utworzonej w ten sposób tablicy o wymiarze U(3:3)

U(0:0) U(0:1) U(0:2)
U(1:0) U(1:1) U(1:2)
U(2:0) U(2:1) U(2:2)

Wymiary tablic mogą być określone za pomocą wyrażeń algebraicznych. Np.

STRUKTURA U(N+1:N+1)

Oczywiście N musi być wcześniej określone.

Działania na zmiennych indeksowanych o dwu lub więcej wskaźnikach muszą być poprzedzone określeniem struktury tych zmiennych.

2.11.5. Zmienna indeksowana o wielu wskaźnikach

Przykład F18

Wczytać wartość zmiennej indeksowanej $A(I:J:K)$ dla zakresu indeksów $I=0,1; J=0,1; K=0,1,2.$

Rozwiązanie

Podamy tylko początek programu, zawierający wczytanie bloku. Możemy dokonać tego podobnie jak w przykładzie F17 stosując potrójny cykl

```
F18:A 12 WEJSCIE 3 DRUKARKA
STRUKTURA A(2:2:3)
→ DLA I=0:1:1
→ DLA K=0:1:2
→ DLA J=0:1:1
    CZLICZBE A(I:J:K)
    POWTORZ J
    POWTORZ K
    POWTORZ I
```

Wczytywanie to możemy uprościć jeśli uwzględnimy że kolejne elementy bloku A można także ponumerować jednoindeksowo wg schematu

```
A(0) = A(0:0:0)
A(1) = A(0:0:1)
A(2) = A(0:0:2)
A(3) = A(0:1:0)
A(4) = A(0:1:1)
A(5) = A(0:1:2)
A(6) = A(1:0:0)
A(7) = A(1:0:1)
A(8) = A(1:0:2)
A(9) = A(1:1:0)
A(10) = A(1:1:1)
A(11) = A(1:1:2)
```

Uproszczony program zawiera tylko jedną pętlę

```
F18:A(12) WEJSCIE 3 DRUKARKA
LINIA 1
→ DLA G=0:1:11
    CZLICZBE A G
    POWTORZ G
    STRUKTURA A(2:2:3)
```

2.12. Rozkaz KLUCZ, wydruk kontrolny

Przykład F19:

Wczytane w poprzednim podprogramie wartości zmiennej A(I:J:K) pomnożyć przez liczbę B.

Rozwiązanie

```
F19:A(20)
WEJSCIE 3 DRUKARKA LINIA 1
E1)
STRUKTURA A(2:3:2)
CZLICZBE B
    ➔ DLA I=0:1:1
    ➔ DLA J=0:1:2
    ➔ DLA K=0:1:1
        CZLICZBE A(I:J:K)
        POWTORZ K
        POWTORZ J
        POWTORZ I
    ➔ DLA I=0:1:1
    ➔ DLA J=0:1:2
    ➔ DLA K=0:1:1
        A(I:J:K)=A(I:J:K).B?
        KLUCZ 66:OPUSC
        X=A(I:J:K)
        DRLICZBE(1:2):I:J:K:X
        OPUSC)
        POWTORZ K
        LINIA 1
        POWTORZ J
        LINIA 1
        POWTORZ I
        KONIEC)
        LINIA 2 STOP
        SKOCZ DO E1
        START F19
```

Za rozkazem

A(I:J:K)=A(I:J:K).B?

postaśliśmy znak zapytania? Postawienie tego znaku po rozkazie algebraicznym pozwala na kontrolny wydruk wyników. Nastąpi to wtedy, gdy klucz nr 100 w trakcie tłumaczenia programu będzie włączony, a w trakcie liczenia wyłączony.

Wydruki kontrolne stosujemy na etapie uruchamiania programów, w celu kontroli poprawności obliczeń. Po uruchomieniu programu wydruki te likwiduje się.

Wydruk nastąpi w postaci:

nazwa zmiennej (wartość pojedyńczego indeksu) = wartość
Wartość zmiennej podawana jest w postaci liczby dziewięciocyfrowej z przecinkiem na właściwej pozycji. Po wydruku wartości następuje wydruk 1 linii.

Jeżeli np. obliczona wartość

$$A(0:0:2)=0,00336$$

to wydruk nastąpi w postaci

$$A(2)=,003360000$$

gdyż złożonemu indeksowi (0:0:2) odpowiada indeks (2).^{w/}
0 przed przecinkiem zostało opuszczone.

Rozkaz

KLUCZ 66:OPUSC

oznacza, że jeżeli w trakcie liczenia będzie włączony którykolwiek z kluczy, których suma numerów daje 66 to nastąpi skok do etykiety OPUSC.

Numery kluczy są następujące: 1, 2, 4, 10, 20, 40, 100.
Należy jeszcze zwrócić uwagę, że rozkaz KLUCZ 66:OPUSC znajduje się wewnątrz cyklu, a ponieważ wyjście z cyklu przed jego wykonaniem jest niedozwolone /porównaj 2.11.4/ więc etykieta OPUSC również musi znajdować się wewnątrz tego cyklu.

^{w/} Jeżeli blok A ma strukturę

$$A(I0:I1:I2:I3:I4)$$

to pojedyńczy indeks H zmiennej

$$A(N0:N1:N2:N3:N4)$$

oblicza się według schematu

$$H=((N0 \cdot I1+N1) \cdot I2+N2) \cdot I3+N3) \cdot I4+N4$$

2.13. Rozkaz PODSTAW

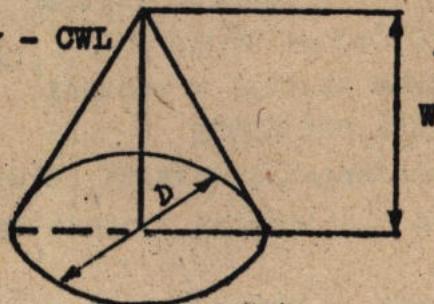
Przykład F20

Ułożyć podprogram obliczania ciężaru stożka według wzoru

$$G = \gamma \cdot \frac{1}{3} \frac{\pi D^2}{4} \cdot W$$

gdzie γ - ciężar właściwy - CWL

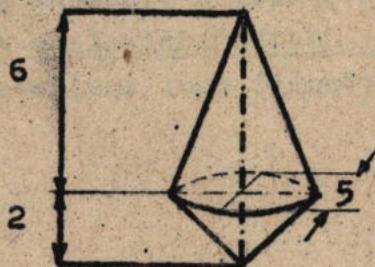
Rozwiązańe



Rys. 7

FCIEZAR:CWL:D:W
FCIEZAR=CWL*D*D*W*3,14159/12
WROC

Skorzystajmy z tego podprogramu do obliczenia ciężaru figury przedstawionej na rys. 8



Rys. 8

dla ciężaru właściwego $\gamma=0,00785$.

F20 DRUKARKA

G=FCIEZAR(0,00785:5:6)+FCIEZAR(0,00785:5:2)

DRLICZBE(5:3):G

STOP START F20

Ten sam program możemy napisać w postaci

F20 WEJSCIE 3 DRUKARKA
PODSTAW FCIEZAR(0,00785:5:.)
G=FCIEZAR(.:.:6)+FCIEZAR(.:.:2)
DRLICZBE (5:3):G
LINIA 1 STOP
START F20

Rozkaz

PODSTAW FCIEZAR(0,00785:5:.)

wprowadza do podprogramu FCIEZAR liczbę 0,00785 na CWL oraz 5 na D.

W następnej linii, przy odwołaniu się do podprogramu FCIEZAR, te argumenty, które podstawiono rozkazem PODSTAW następujemy kropkami. Taki zapis ułatwia pracę maszyny, gdyż eliminuje wielokrotne podstawianie tych samych liczb.

Na liście argumentów rozkazu PODSTAW wypisuje się tylko te wielkości, które podlegają podstawieniu. W miejscach przeznaczonych dla pozostałych argumentów, nie podstawionych tym rozkazem - stawia się kropki, zachowując rozdzielające je dwukropki. Kropka i dwukropków można nie stawiać po ostatniej wypisanej na liście wielkości. Omawiany rozkaz w programie moglibyśmy więc zapisać krócej:

PODSTAW FCIEZAR(0,00785:5)

2.14. Podprogram wewnętrz programu

Przykład F21:

Ułożyć program na obliczenie wartości wyrażeń

$$y=3 \cdot x + \ln x$$

$$z=4 \cdot x + \ln(x+1)$$

dla zmiennej x wczytywanej po każdym przyciśnięciu przycisku START.

Rozwiązanie

F21
WEJSCIE 3 DRUKARKA
POCZ)
A=3 CZLICZBE X B=X
PODPROGRAM ET
NAPIS Y=# DRLICZBE(3:4):U
A=4 B=X+1

PODPROGRAM ET
 NAPIS Z= # DRLICZBE(3:4) : U
 LINIA 1 STOP SKOCZ DO POCZ
 ET)
 U=A.X+LOG(B)
 WROC

START F21

Objaśnienie

Ponieważ wzory mają tę samą budowę postaci

$$U = A \cdot X + \ln B$$

wobec tego zapisaliśmy sposób wartości tych wyrażeń w podprogramie oznaczonym etykietą ET i zakończonym rozkazem WROC. Po określeniu A i B do podprogramu tego możemy się odwołać tylko rozkazem

PODPROGRAM ET

2.15. Współpraca z taśma magnetyczna

Przykład F22

Wczytać blok A 20 liczbę B. Zapisać te wielkości na taśmie magnetycznej, odczytując z powrotem do bloku C i komórki C i wydrukować te wielkości.

```

F22:A(20):C(20):D(1)
WEJSIE 3 DRUKARKA
E) CZLICZBE B
DLA I=0:1:19
  CZLICZBE A(I)
POWTORZ I
  PTM 3 NASTAW ADRES 1000
  PTM 3 PISZ Z A:20
  PTM 3 NASTAW ADRES 1000
  PTM 3 CZYTAJ DO C:20
  PTM 3 NASTAW ADRES 301
  PTM 3 PISZ Z B:1
  PTM 3 NASTAW ADRES 301
  PTM 3 CZYTAJ DO D:1
  DRLICZBE(3:4) :D
  DLA I=0:1:19
    DRLICZBE(3:4) :C
POWTORZ I
LINIA STOP SKOCZ DO E
START F22
  
```

Rozkaz

PTM 3 NASTAW ADRES 1000

przygotuje przewijacz taśmy magnetycznej nr 3 do czytania /lub zapisu/ informacji od adresu 1000.

Rozkaz

PTM 3 PISZ Z A:20

spowoduje zapis na taśmie magnetycznej przewijacza nr 3 informacji zapisanych w bloku A o zakresie 20 komórek.

Rozkaz

PTM 3 CZYTAJ DO C:20

spowoduje odczytanie z taśmy magnetycznej na przewijaczu nr 3 informacji i przesłanie ich do bloku C o zakresie 20 komórek. W celu przesłania zmiennej prostej na taśmę magnetyczną zapisujemy ją jako zmienną indeksowaną o zakresie 1, tak uczynione jest w następnych rozkazach.

2.16. Kontrola poprawności zapisu programu

Przed uruchomieniem programu należy dokonać następującego sprawdzenia

Czy nazwy etykiet zaczynają się od litery innej niż F ?

Czy nazwy podprogramów zaczynają się od litery F ?

Czy nazwy zmiennych lub bloków całkowitych zaczynają się od jednej z liter G H I J K L M N O P R ?

Czy nazwy zmiennych lub bloków rzeczywistych zaczynają się od jednej z liter A B C D E S T U V W X Y Z ?

Czy nazwy nie są słowami zastrzeżonymi:

ABS ADRES ARCTG COS CZLICZBE CZSYMBOL CZTEKST CZYTAJ
DLA DOKONCZ DR LICZBE DRUKARKA DR SYMBOL DR TEKST EXP GDY
KLUCZ KOPIA LINIA LOG NAGLOWEK NA PIS NASTAW PISZ
PODPROGRAM PODSTAW POWTORZ PTM PWK RAMKA SIN SKOCZ START
STOP STRONA STRUKTURA WARTOSC WYJSCIE

Czy nie użyto tej samej nazwy dla dwu różnych oznaczeń w obrębie jednego podprogramu ?

Czy dwu różnych podprogramów nie nazwano jednakowo ?

Czy dla zmiennych indeksowanych zarezerwowano bloki liczb ?

- Czy komentarz zaczyna się i kończy apostrofem ?
Czy etykieta zakończona jest nawiasem ?
Czy liczba nawiasów otwartych w wyrażeniu równa jest liczbie nawiasów zamkniętych ?
Czy dwukropki występuje w rozkazach
 GDY W:Y SKOCZ DO e:f:g
 KLU CZ P:e
 DLA Z=W:Y:X
 DR LICZBE(p:q):W:Y:Z
 DR LICZBE(p):W:Y:Z
 STRUKTURA A(W:Y:Z)
Czy dla rozkazu CZTEKST A zarezerwowano blok A ?
Czy rozkaz DRTEKST A był poprzedzony rozkazem CZTEKST A ?
Czy tekst po rozkazie NAPIS zakończony jest znakiem # ?
Czy przed rozkazem STOP występuje rozkaz LINIA 1 ?
Czy rozkaz LINIA p występuje dostatecznie często, tak że nie nastąpi przekroczenie wydruku 120 znaków w wierszu?
Czy rozkaz PISZ Z lub CZYTAJ DO został poprzedzony rozkazem USTAW ADRES ?
Czy w nagłówku podprogramu zarezerwowano miejsca dla argumentów ?
Czy w wywołaniu podprogramu kolejność i charakter argumentów są zgodne ?
Jeśli w wywołaniu podprogramu występuje nazwa bloku ze znakiem # , to czy miejsca dla tego bloku zostały zarezerwowane ?
Czy najwyższy stopień podprogramu nie przewyższa 8 ?
Czy najwyższy stopień cyklu nie przewyższa 8 ?
Czy każdemu rozkazowi DLA I=P:Q:R odnosi się rozkaz POWTORZ I
Czy nie występuje wyskok cyklu przed jego wykonaniem ?
Czy ramki obejmujące powtórzenia nie przecinają się ?
Czy wyrażenie występujące po prawej stronie znaku równości jest określone wcześniej ?
Czy wyrażenia w rozkazie GDY mają wartości określone wcześniej ?
Czy indeksy zmiennych indeksowanych są określone wcześniej ?
Czy podprogram napisany jest wcześniej niż jego wywołanie ?

Czy nie występuje dzielenie przez zero ?

Czy pierwiastek wyciąga się z liczby dodatniej ?

Czy podprogramy zawierają rozkaz WROC ?

Czy dane są przygotowane w takiej kolejności, jaką wskazuje program ?

Czy dane, które powinny być całkowite, są liczbami całkowitymi ?

Czy taśma programu i danych nie zawiera stref dłuższych niż 800 znaków ?

Czy w obrębie strefy nie występuje znak pustej taśmy ?

2.17. Uruchamianie programów

W celu przeprowadzenia obliczeń na maszynie należy wykonać następujące czynności:

A. Przygotować maszynę do translacji z języka SAKO na język MAT:

- a.Wczytać translator SAKO napisany w języku MAT,
- b.Translator zapisać jako program nr 0.

B. Dokonać translacji programu napisanego w języku SAKO na język MAT:

- a.Sprawdzić z taśmy magnetycznej program nr 0 /patrz p.A/
- b.Wyłączyć blokadę nadmiaru, uruchomić perforator nr 2 i taśmę magnetyczną nr 2,
- c.Jeżeli programista życzy sobie wydruków kontrolnych - włączyć klucz 100, jeśli nie - klucz ten wyłączyć,
- d.Podłożyć taśmę z programem w języku SAKO pod czytnik i maciąć klucz START.

W przypadku poprawnego programu, po jego wczytaniu i translacji następuje wydruk programu w języku MAT na taśmie perforowanej wyrowadzonej z perforatora nr 2 w kierunku odwrotnym, aż do zatrzymania maszyny.

C. Wprowadzić program przetłumaczony na język MAT:

Program otrzymany w punkcie B należy wprowadzić za pośrednictwem czytnika jako program nr 1 w języku MAT bez przewijania taśmy.

D. Wykonać program:

- a. Sprowadzić z taśmy magnetycznej program nr 1 /patrz p.C/ oraz uruchomić odpowiednie urządzenia przewidziane przez programistę,
- b. Podłożyć dane do odpowiednich urządzeń wejść,
- c. Nacisnąć klucz START

Jeżeli programista zażądał wydruków kontrolnych i została wykonana w czasie translacji B czynność C, to będą drukowane wydruki kontrolne przy wyłączonym kluczu 100. Włączenie klucza 100 spowoduje skasowanie tych wydruków. To wyłączenie może nastąpić także w czasie pracy maszyny.

E. Przy następnym programie można zaczynać od czynności B.

3. LISTA ROZKAZOW I SŁOWNIK WYRAZOW

3.1. Lista rozkazów języka SAKO dla MIŃSK-22

W każdym rozkazie została podkreślony część niezmienna.

Argumenty rozkazów we wzorcach rozkazów podane są schematycznie i oznaczają:

- Z - zmienną dowolnego typu
- A B C - nazwy bloków
- WYXUV - wyrażenia algebraiczne
- pq - nieujemne liczby całkowite

Lp.	WZORZEC rozkazu przykład	ZNACZENIE Rozkazu	OBJAŚNIENIE Strona
ROZKAZY STERUJĄCE - Stop, skoki, cykle			
1	<u>START</u> nazwa podprogramu START F1 START FPROGRAM	Ostatni rozkaz sygnalizujący koniec programu; przy wykony- waniu programu po naciśnięciu przycisku START wykonywanie programu rozpocznie się od po- czątku podprogramu o nazwie wymienionej po słowie START	14
2	<u>STOP</u>	Zatrzymanie pracy maszyny; po naciśnięciu przycisku START zostaną wykonane na- stępne rozkazy	14
3	<u>SKOCZ DO</u> e SKOCZ DO E1 SKOCZ DO MNIEJSZE	Skok do rozkazu oznaczonego etykieta e.	22
4	<u>GDY W:Y SKOCZ</u> <u>DO e:f:g</u> GDY X:A+C SKOCZ O:E:E1	Dla $W < Y$ skok do rozkazu po- przedzonego etykietą e Dla $W=Y$ skok do rozkazu po- przedzonego etykietą f Dla $W > Y$ skok do rozkazu po- przedzonego etykietą g O - /zero/ zamiast etykiety- skok do następnego roz- kazu.	23

5	<u>KLUCZ</u> _{p:e}	Gdy włączony jest którykolwiek z kluczy, których suma numerów wynosi p - skok do etykiety e Klucz=5:ET2 Numery kluczy 1,2,4,10,20, 40,100	40
6	<u>PODPROGRAM</u> _e	Skok do podprogramu oznaczonego etykietą e; w Programie dopuszczalne są podprogramy stopnia nie większego niż osiem; podprogram bez nagłówka może się mieścić tylko w podprogramie z nagłówkiem.	43
7	<u>WROC</u>	Ostatni rozkaz podprogramu, powodujący powrót z podprogramu podległego do programu nadzorowanego	19 44
8	<u>DLA</u> _{Z=W:Y:X} DLA B=0:1,5:10 DLA Z=R:2:N DLA Z=R+1:2,B+: +3:N-1	Początek cyklu; Rozkaz DLA powoduje wielokrotnie wykonanie odcinka programu zawartego między DLA _{Z=W:Y:X} a rozkazem POWTORZ Z dla wartości zmiennej prostej Z; za pierwszym razem zmenna Z przyjmuje wartość wyrażenia W; przed każdym kolejnym powtórzeniem cyklu wartość Z zwiększa się o Y; gdy $ X-Z < Y/2 $, to cykl powtarzony będzie poraz ostatni, przy czym Z przyjmie wartość X; jeżeli W,Y,X są wyrażeniami, to ich wartości nie mogą ulegać zmianie w trakcie wykonywania cyklu; cykl może wchodzić kolejno w skład co najwyżej 9 cykli.	25 35
9	<u>POWTORZ</u> _Z POWTORZ _B	Ostatni rozkaz cyklu, dokonujący właściwego badania warunku powtarzania; zmienność Z jest identyczna ze zmiennością rozkazu DLA _Z tego samego cyklu.	25 35

ROZKAZY ALGEBRAICZNE I FORMUŁY OPERACYJNE

10	$Z=W$ $A=X+2 \cdot \sin(X)$	Rozkaz algebraiczny; Obliczenie wartości wyrażenia W i podstawienie tej wartości na zmienną Z; jeżeli Z jest całkowita, a wartość wyrażenia W niecałkowita, to na Z zostanie podstawiona wartość W zaokrąglona do najbliższej liczby całkowitej	16 24
11	<u>WARTOSC</u> $Z=L;M;N$ $:0$ WARTOSC $Z=0:1,3$ $:2,7:17$	W czasie translacji kolejne zmienne Z_0, Z_1, Z_2, \dots przyjmą odpowiednio wartości L, M, N, ...	31

PODPROGRAMY

12	<u>Nazwa podprogramu:</u> lista argumentów rozdzielonych dwukropkami <u>FPODPR</u> :X FUN #(2):A#:EPS <u>FCALKA</u> :XMIN :XMAX:FUN#(1) :A(3)	<u>NAGŁÓWEK PODPROGRAMU</u> Nazwa podprogramu zaczyna się od litery F Argumenty mogą mieć postać 1/zmiennej prostej - powoduje to zarezerwowanie jednej komórki pamięci dla tej zmiennej 2/zmiennej indeksowanej po której w nawiasach pisze się liczbę komórek zarezerwowanego bloku 3/nazwy zmiennej, po której występuje znak #. Dla tej zmiennej,/lub bloku/ miejsca zarezerwowane są tam, gdzie występuje wywołanie podprogramu 4/nazwy funkcji /zaczynającej się od litery F/, po której występuje znak # oraz w nawiasach okrągłych liczba argumentów tej funkcji na które dokonuje się podstawienia w tym podprogramie. Dla wszystkich zmiennych niezadeklarowanych w nagłówku podprogramu rezerwuje się jedną komórkę pamięci.	22 19 33 19 33
----	--	--	------------------------------

13	<p>Nazwa podprogramu (lista argumentów rozdzielonych dwukropkiem)</p> <p>FPODPR(X: FUNKCJA#:A#: 0,00012)</p> <p>FCALKA(5,0: XMAX:FUNPODCH)</p>	<p><u>WYWOŁANIE PODPROGRAMU</u></p> <p>powoduje podstawienie argumentów wywołania na argumenty wymienione na liście nagłówka argumentów podprogramu i wykonanie podprogramu.</p> <p>Obydwie listy muszą mieć taką samą kolejność i zgadzać się charakterem</p> <p>1/Jeśli w nagłówku występuje zmienna prosta, to w wywołaniu podajemy wyrażenie, którego wartość zostaje przesłana na tą zmienną</p> <p>2/Jeśli w nagłówku występuje nazwa, a po niej znak #, to w wywołaniu musi wystąpić także nazwa z takim znakiem. Podstawienie oznacza, że w podprogramie nazwa wymieniona w nagłówku zostanie zastąpiona nazwą wymienioną w wywołaniu.</p> <p>Może to być</p> <ul style="list-style-type: none"> a. <u>nazwa zmiennej prostej</u> lub <u>nazwa zmiennej indeksowanej</u> wtedy miejsca dla tych zmiennych zarezerwowane są tam, gdzie nastąpiło wywołanie b. <u>nazwa funkcji</u>. Wtedy ta funkcja powinna być wcześniej opisana podprogramem i zawierać w swoim nagłówku argumenty zgodne z nagłówkiem podprogramu, który z niego korzysta. Ewentualne dalsze argumenty należy określić rozkazem PODSTAW <p>3/Dowolny argument wywołania można zastąpić kropką. Oznacza to niewykonanie podstawienia z zachowaniem poprzednio podstawionej wartości /lub nazwy/. Ostatnie kropki z rozdzielającymi je dwukropkami można pominąć.</p>	<p>33 19 22</p> <p>19</p> <p>33</p>
----	--	---	---

14	<u>PODSTAW</u> Nazwa podprogramu (lista argumentów rodzinnych dwukropkiem) PODSTAW FPODPR (.:X+2:B#) PODSTAW F15(-2 :5,0:FUN#)	Powoduje podstawienie tak jak w WYWÓŁANIU PODPROGRAMU, bez wykonania podprogramu.	43
15	<u>STRUKTURA A(W:Y)</u> <u>STRUKTURA A(W:Y:X:V)</u> <u>STRUKTURA U(2:N+1:N)</u>	Okręsła strukturę zmiennej $A[I, K]$ o dwu indeksach. Zapis $A(I, K)$ jest równoważny jednoindeksowemu zapisowi $A(I \cdot Y + K)$ I, K, W, Y są to dowolne wyrażenia. Okręsła strukturę zmienną $A(I, K, L, M)$ o czterech indeksach. Zapis $A(I, K, L, M)$ jest równoważny jednoindeksowemu zapisowi $A(((I \cdot Y + K) \cdot X + L) \cdot V + M)$ We wszystkich podprogramach w których występują zmienne wieloindeksowe musi być określona struktura bloków.	38

FUNKCJE STANDARDOWE

16	<u>SIN(X)</u>	Sinus x	18
17	<u>COS(X)</u>	Cosinus x	18
18	<u>ARCTG(X)</u>	Arcustangens x	18
19	<u>LOG(X)</u>	$\ln_e x$ określony dla $x > 0$	18
20	<u>EXP(X)</u>	Funkcja wykładnicza o podstwie e -EKSPONENT- oznacza e^x	18
21	<u>ABS(X)</u>	Wartość bezwzględna - oznacza $ X $	18
22	<u>PWK(X)</u> $Z=2+\text{EXP}(X)+\text{COS}(2 \cdot X)$ $Y=1/\text{PWK}(X)$	Pierwiastek kwadratowy określony dla $X > 0$ - oznacza \sqrt{X}	18

WSPOLPRACA Z TASMĄ MAGNETYCZNĄ

23	<u>PTM A NASTAW</u> <u>ADRES B</u> PTM 2 NASTAW ADRES 301	Przygotowanie przewijacza o numerze A /A=0,1,...,15/ do czytania lub zapisu informacji od adresu B. Adres B jest adresem taśmy magnetycznej /B=0,1,...,100000/.	44
24	<u>PTM A PISZ Z</u> <u>B:C</u> PTM 3 PIŚ Z B:80	Zapisanie na taśmie magnetycznej na przewijaczu A pola B o zakresie C	44
25	<u>PTM A CZYTAJ DO</u> <u>B:C</u> <u>PTM B CZYTAJ</u> <u>DO U:Z</u>	Odczytanie z taśmy magnetycznej na przewijaczu A informacji i przesłanie ich do pola B o zakresie C	44

ROZKAZY WEJSCIA I WYJSCIA

26	<u>WEJSCIE p</u> <u>WEJSCIE 3</u>	Wybór urządzenia czytającego dla rozkazów CZLICZBE CZTEKST CZSYMBOL p=1 czytnik 1 startstopowo p=2 czytnik 2 startstopowo p=3 czytnik 1 strefowo	11 12
27	<u>CZLICZBE Z</u> <u>CZLICZBE A</u> <u>CZLICZBE B(I+3)</u>	Pobranie jednej liczby z urządzenia czytającego, zdekodowanie jej i przesłanie do Z	13
28	<u>CZTEKST A</u>	Pobranie z urządzenia czytającego do bloku A tekstu aż po symbol # włącznie.	28
29	<u>CZSYMBOL Z</u> <u>CZSYMBOL I</u> <u>CZSYMBOL J(K+7*L)</u>	Pobranie jednego symbolu z urządzenia czytającego i przesłanie na najmniej znaczące pozycje Z Pozostałe pozycje są zerowane	
30	<u>WYJSCIE p</u> <u>WYJSCIE 2</u>	Wybór urządzenia drukującego p=1 perforator 1 p=2 perforator 2	12
31	<u>DRUKARKA</u>	Wydruk wyników na drukarce alfanumerycznej	12

32	<u>KOPIA</u> --#621161040003 <u>KOPIA</u> =0.	Przy wydrukach na drukarkę alfanumeryczną pozycje 0-59 wiersza będą powtarzane na pozycjach 68-127 Likwidacja kopii wydruku	29
33	<u>RAMKA</u> = p	Liczba wolnych wierszy przed treścią strony i po treści strony wynosi P. Rozkaz NAGŁOWEK może zmienić liczbę wolnych wierszy przed treścią strony	29
34	<u>NAGŁOWEK</u> =e	Zamiast wydruku wolnych wierszy przed treścią strony wykonany zostanie podprogram rozpoczynający się od etykiety e i zakończony rozkazem WROC. Jeżeli e=0 przejście na wydruk wolnych wierszy wg rozkazu RAMKA.	
35	<u>STRONA</u> p	Liczba wierszy, przeznaczonych na treść jednej strony -p	29
36	<u>DOKONCZ</u>	Dokończenie treści drukowanej strony wólmymi wierszami.	29
37	<u>DRLICZBE</u> (p:q) :W:Y:Z DRLICZBE(5:3) :A:B:C DRLICZBE(4:4) :A.X.X/B	Wydruk w systemie dziesiętnym rzeczywistych wartości wyrażeń w postaci znak, p cyfr przed przecinkiem przecinek, q cyfr po przecinku $p+q \leq 9$ znak "+" oraz początkowe zera są pomijane znak "-" jest dostawiany do pierwszej cyfry. Po liczbie drukowane są 2 spacje	13
38	<u>DRLICZBE</u> (p):W:Y:Z DRLICZBE(7):X :Y+1 DRLICZBE(5):A .X/(Y+2)	Wydruk w systemie dziesiętnym wartości całkowitych wyrażeń w postaci: znak, p cyfr początkowe zera oraz znak "+" są pomijane. Znak "-" dostawiany jest do pierwszej znaczącej cyfry $p \leq 11$ Po liczbie drukowane są 2 spacje.	14

39	DRTEKST A DRTEKST B	Wydruk tekstu wczytanego uprzednio rozkazem CZTEKST A - do bloku A	28
40	DRSYMBOL W	Wydruk symbolu W według kodu M2/CCIT	
41	SPACJA W SPACJA 1 SPACJA R+2	Wydruk W spacji W jest dowolnym wyrażeniem	29
42	LINIA W LINIA 1 LINIA H+1	Wydruk symbolu karetka i W symboli linia W jest dowolnym wyrażeniem	13
43	NAPIS dowolny tekst # NAPIS Wynikiu 5.7 #	Wydruk tekstu napisanego po spacjach za słowem NAPIS. Pierwsze spacje zawarte w tekście są pomijane.	24
44	'dowolny tekst nie zawiera- jący apostro- fu'	Komentarz ujęty w apostrofy Przy czytaniu jest ignorowany.	32
45	? A(I:J:K)= W+X.Y?	Pytajnik postawiony bezpo- rednią za rozkazem algebra- icznym; powoduje wydruk war- tości tego wyrażenia - jeśli w trakcie tłumaczenia progra- mu Klucz nr 100 jest włączony, a w trakcie liczenia wyłączo- ny. Postać wydruku: Nazwa zmiennej /indeks/=war- tość i nowa linia lub Nazwa zmiennej=wartość i nowa linia. Indeks wielowskaźnikowy prze- liczany jest zgodnie z działa- niem rozkazu. STRUKTURA na je- dnowskaźnikowy.	40

3,2. Słownik wyrazów

Str.

Adres	7
Argument	18
Blok	29
Cyfra	4
Cykl	26
Czytnik	5
Czytanie strefowe	12
Dalekopis	4
Dane	3 11
Drukarka	5
Etykieta	22
Funkcja definiowana	19 21
Funkcja standardowa	18
Indeks	29
Język wewnętrzny	3
Ked	6
Kolejność działań	22
Komentarz	31
Komentarz w danych	32
Komórki pamięci	7
Liczby	9
Lista argumentów	22 34
Nagłówek podprogramu	26
Nazwa	11
Pamięć operacyjna /szybka/	7
Pamięć zewnętrzna	7
Parametr	14
Podstawienie	42
Przesłanie wartości	16 24
Pusta taśma	12
Rozkaz	13
SAKO	3
Skok zmiennej	26
Skok bezwarunkowy	22
Skok warunkowy	23
Stopień podprogramu	21
Strefa	12
Taśma perforowana	5 12
Wejście	4 11
Wyjście	4 12
Wyniki	4 11
Wyrażenie algebraiczne	15
Zakres zmienności	30
Zmienna	3
Zmienna prosta	13
Zmienna całkowita	13 30
Zmienna rzeczywista	13 30
Zmienna indeksowana	29 34 39
Znak dalekopisu	4

4. BIBLIOTEKA PODPROGRAMOW

3.1. Działania na liczbach zespolonych	61
3.1.1. Sumowanie liczb zespolonych	61
2. Odejmowanie liczb zespolonych	61
3. Mnożenie liczb zespolonych	61
4. Dzielenie liczb zespolonych	62
5. Moduł liczby zespolonej	62
6. Pierwiastek liczby zespolonej	62
7. Tangens argumentu liczby zespolonej	62
3.2. Funkcje elementarne	62
3.2.1. Funkcje odwrotne względem funkcji trygonometrycznych	62
3.2.2. Funkcje potęgowe	63
3.2.3.1. Obliczenie wartości $\sinh x$	63
2. Obliczenie wartości $\cosh x$	63
3. Obliczenie wartości $\tgh x$	63
4. Obliczenie wartości $\operatorname{arcsinh} x$	64
5. Obliczenie wartości $\operatorname{arccosh} x$	64
6. Obliczenie wartości $\operatorname{arctgh} x$	64
3.3. Algebra wielomianów	64
3.3.1. Obliczenie wartości wielomianu	64
2. Wyznaczenie współczynnika wielomianu	65
3. Wyznaczenie pierwiastków wielomianu	66
3.4. Działania na macierzach	67
3.4.1. Obliczenie wartości wyznacznika	67
2. Transpozycja macierzy	68
3. Odwracanie macierzy	68
4. Dodawanie macierzy	70
5. Mnożenie macierzy	71
6. Diagonalizacja macierzy symetrycznej	71
7. Wartości własne i wektory własne macierzy symetrycznej	72
8. Rozwiązywanie układu równań algebraicznych	74
3.5. Rachunek różniczkowy i całkowy	75
3.5.1. Rozwiązywanie układu równań różniczkowych	75
2. Obliczenie wartości całki oznaczonej	76
3.6. Programowanie liniowe	77
3.6.1. Zadanie transportowe	77

Korzystając z podprogramów, należy pamiętać że:

1. Podprogramy muszą być napisane przed programem, w którym następuje ich wywołanie.

2. Wywołanie podprogramów ma postać

NAZWA PODPROGRAMU (lista parametrów rozdzielonych dwukropkami)

Podprogramy funkcyjne, w których obliczoną wartość funkcji podstawi się na nazwę podprogramu można wywoływać w wyrażeniu:

ZMIENNA=Wyrażenie algebraiczne, w którym występuje

NAZWA PODPROGRAMU (lista parametrów rozdzielonych dwukropkami)

3. Charakter parametrów podstawionych w wywołaniu musi być zgodny z charakterem parametrów podprogramu.

a. Typ parametru /całkowity, rzeczywisty/ określa jego nazwa

b. Bloki zmiennych indeksowanych powinny mieć odpowiednie struktury

c. Na parametr będący funkcją, można podstawić tylko funkcję

d. Parametry - na które podstawi się wyniki muszą być oznaczone znakiem #

3.1. Działania na liczbach zespolonych

W poniższych podprogramach składowe zmiennych zespolonych oznaczone są dwiema literami. Druga litera określa rodzaj składowej; R - oznacza część rzeczywistą, I - oznacza część urojoną. Liczby A, B, C są odpowiednio równe $A=AR+i\cdot AI$ $B=BR+i\cdot BI$ $C=CR+i\cdot CI$

3.1.1. Sumowanie liczb zespolonych

Podprogram - FZS - oblicza sumę liczb zespolonych $C=A+B$

FZS:AR:AI:BR:BI:CR#:CI#
CR=AR+BR
CI=AI+BI
WROC

WYWOŁANIE: FZS(AR:AI:BR:BI:CR#:CI#)
FZS(3:2:-5:0,2:CR#:CI#)

3.1.2. Odejmowanie liczb zespolonych

Podprogram - FZR - oblicza różnicę liczb zespolonych

C=A-B
FZR:AR:AI:BR:BI:CR#:CI#
CR=AR-BR CI=AI-BI
WROC

WYWOŁANIE: FZR(AR:AI:BR:BI:CR#:CI#)

3.1.3. Mnożenie liczb zespolonych

Podprogram - FZIL - oblicza iloczyn liczb zespolonych $C=A\cdot B$

FZIL:AR:AI:BR:BI:CR#:CI#
CR=AR.BR-AI.BI CI=AR.BI+AI.BR
WROC

WYWOŁANIE: FZIL(AR:AI:BR:BI:CR#:CI#)

3.1.4. Dzielenie liczb zespolonych

Podprogram - FZDZ - oblicza iloraz liczb zespolonych $C=A/B$

FZDZ:AR:AI:BR:BI:CR#:CI#
W=BR.BR+BI.BI
GDY W:0 SKOCZ DO E:0:E
NAPIS
DZIELENIE PRZEZ ZERO

SKOCZ DO KONIEC
E)
CR=(AR.BR+AI.BI)/W
CI=(AI.BR-AR.BI)/W

KONIEC)

WROC

WYWOŁANIE: FZDZ(AR:AI:BR:BI:CR#:CI#)

Jeżeli mianownik jest równy zero, zostaje wydrukowany tekst "Dzielnie przez zero".

3.1.5. Moduł liczby zespolonej

Podprogram funkcyjny - FZMOD - oblicza moduł liczby zespolonej A.

FZMOD:AR:AI

FZMOD=PWK(AR.AR+AI.AI)

WROC

WYWOŁANIE: FZMOD(AR:AI)

3.1.6. Pierwiastek z liczby zespolonej

Podprogram - FZPIERW - oblicza pierwiastek z liczby zespolonej $C = \sqrt{A}$

FZPIERW:AR:AI:CR#:CI#

A=PWK(AR.AR+AI.AI)

CR=PWK((A+AR)/2)

CI=PWK((A-AR)/2)

GDY AI:0 SKOCZ DO 0:E:E

CI=-CI

E)

WROC

WYWOŁANIE: FZPIERW(AR:AI:CR#:CI#)

3.1.7. Tangens argumentu liczby zespolonej.

Podprogram funkcyjny - FZTG - oblicza tangens argumentu liczby zespolonej.

FZTG:AR:AI

FZTG=AI/AR

WROC

WYWOŁANIE: FZTG(AR:AI)

3.2. Funkcje elementarne

3.2.1. Funkcje odwrotne względem funkcji trygonometrycznych

Podprogram funkcyjny - FARSIN - oblicza wartość funkcji $\text{arc sin } x$

```
FARSIN:X  
FARSIN=ARCTG(X/PWK(1-X*X))  
WROC
```

WYWOŁANIE: FARSIN(X)

X - argument funkcji w radianach $-1 < x < +1$

3.2.2. Funkcje potęgowe

Podprogram funkcyjny - FPOTADOB - oblicza wartość funkcji
 $y=a^b$

```
FPOTADOB:A:B  
FPOTADOB=EXP(B.LOG(A))  
WROC
```

WYWOŁANIE: FPOTADOB(A:B)

A - podstawa. Wynik określony dla $A > 0$

B - wykładnik potęgi

3.2.3. Funkcje hiperboliczne i ich funkcje odwrotne

3.2.3.1. Obliczenie wartości funkcji sinh x zawiera podprogram funkcyjny - FSH

```
FSH:X  
Y=EXP(X)  
FSH=(Y-1/Y)/2  
WROC
```

WYWOŁANIE: FSH(X)

3.2.3.2. Obliczenie wartości funkcji cosh x podaje podprogram funkcyjny - FCH

```
FCH:X  
Y=EXP(X)  
FCH=(Y+1/Y)/2  
WROC
```

WYWOŁANIE: FCH(X)

3.2.3.3. Obliczenie wartości funkcji tgh x zawiera podprogram funkcyjny - FTGH

```
FTGH:X  
Y=EXP(X)        Z=1/Y  
FTGH=(Y-Z)/(Y+Z)  
WROC
```

WYWOŁANIE: FTGH(X)

3.2.3.4. Obliczenie wartości funkcji arc sinh x
zawiera podprogram funkcyjny - FARSH

FARSH:X
Y=PWK(X,X+1)
FARSH=LOG(X+Y)
WROC

WYWOŁANIE: FARSH(X)

3.2.3.5. Obliczenie wartości funkcji arc cosh x
zawiera podprogram funkcyjny - FARCH

FARCH:X
Y=PWK(X,X-1)
FARCH=LOG(X+Y)
WROC

WYWOŁANIE: FARCH(X)

3.2.3.6. Obliczenie wartości funkcji arc tgh x
zawiera podprogram funkcyjny - FARTGH

FARTGH:X
FARTGH=0,5.LOG ((1+X)/(1-X))
WROC

WYWOŁANIE: FARTGH(X)

3.3. Algebra wielomianów

3.3.1. Obliczenie wartości wielomianu

Podprogram funkcyjny - FWIELOMIAN - oblicza wartość wielomianu n-tego stopnia o postaci:

$$Y(X)=A(0) \cdot X^0 + A(1) \cdot X^1 + A(2) \cdot X^2 + \dots + A(n) \cdot X^n$$

```
FWIELOMIAN:X:A#:N
Y=0,0
DLA I=N: (-1):0
Y=Y.X+A(I)
POWTORZ I
FWIELOMIAN=Y
WROC
```

WYWOŁANIE: FWIELOMIAN(X:A#:N)

gdzie:

X - zmienna rzeczywista

A# - nazwa bloku współczynników rzeczywistych A(0), A(1), ..., A(N)

N - całkowita = stopień wielomianu

3.3.2. Wyznaczanie współczynników wielomianu

Podprogram - FWSPOŁCZYNNIKIWIELOM - wyznacza współczynniki wielomianu n-tego stopnia o postaci

$$Y(X) = A(0) \cdot X^0 + A(1) \cdot X^1 + A(2) \cdot X^2 + \dots + A(n) \cdot X^n$$

przyjmującego zadane wartości w $n+1$ punktach

FWSPOŁCZYNNIKIWIELOM: X#: Y#: N#: EPS

DLA K=0:1:N-1

DLA I=K+1:1:N

$Y(I) = (Y(I) - Y(K)) / (X(I) - X(K))$

POWTORZ I

POWTORZ K

$Y(N+1) = 0$

DLA K=N-1:-1:0

DLA I=K:1:N

$Y(I) = Y(I) - Y(I+1) \cdot X(K)$

POWTORZ I

POWTORZ K

I=N

OBNIZ

GDY ABS(Y(I)) : EPS SKOCZ DO 0:0:E

$Y(I) = 0$ I=I-1 SKOCZ DO OBNIZ

E) N=I

WROC

WYWOŁANIE: FWSPOŁCZYNNIKIWIELOM(X#:Y#:N#:EPS)

gdzie:

X# - blok zmiennych $X(0), X(1), X(2), \dots, X(N)$

Y# - blok wartości funkcji $Y(0), Y(1), Y(2), \dots, Y(N)$

N# - całkowita=stopień wielomianu /zakładany i wynikowy/

EPS - dokładność obliczania współczynników

Po wykonaniu programu zawartości bloków X i Y zostają zniszczone. Wartości współczynników $A(0), A(1), \dots, A(N)$ zostają wpisane do bloku Y na miejsca $Y(0), Y(1), \dots, Y(N)$.

W programie nadziednym dla bloku Y należy zarezerwować o jedną komórkę /komórka robocza/ więcej niż dla bloku X.

Stopień wielomianu N jest obniżany przez odrzucenie końcowych wyrazów, dla których współczynniki $A(i) \leq EPS$
 $i=N, N-1, N-2, \dots$

3.3.5. Wyznaczanie pierwiastków wielomianu

Podprogram -FPIERWIASKI - oblicza wszystkie pierwiastki wielomianu zespolonego n-tego stopnia o postaci:

$$W(X+iY) = [A(0)+iB(0)] \cdot [X+iY]^n + [A(1)+iB(1)] \cdot [X+iY]^{n-1} + \dots + [A(n)+iB(n)] \cdot [X+iY]^0$$

```
FPIERWIASKI:N:A#:B#:EPS
ZR=0,0 ZI=0,0
UTART)
DZR=0,1 DZI=0,0 AKO=100000000,0
E1) DWR=0,0 DWI=0,0 WR=0,0 WI=0,0
DLA I=0:1:N
DWR1=DWR.ZR+WR-DWI.ZI
DWI=DWR.ZI+WI+DWI.ZR
DWR=DWR1
WR1=WR.ZR+A(I)-WI.ZI
WI=WR.ZI+B(I)+WI.ZR
WR=WR1
POWTORZ I
AK=ABS(WR)+ABS(WI)
GDY AK:AKO SKOCZ DO 0:SN:S
SN) AKO=AK
AM=ABS(DWR)+ABS(DWI)
GDY AK:4.AM SKOCZ DO NEW:0:PODWOJ
PODWOJ)
DZR=DZR+DZR DZI=DZI+DZI
SKOCZ DO LICZ
NEW)
AP=DWR.DWR+DWI.DWI
DZR=-(WR.DWR+WI.DWI)/AP
DZI=+(WR.DWI-WI.DWR)/AP
SKOCZ DO DOKLAD
S) ZR=ZR1 ZI=ZI1
DZR1=-(DZR+3,0.DZI)/4 DZI=(3,0.DZR-DZI)/4
DZR=DZR1
DOKLAD)
GDY (ABS(DZR)+ABS(DZI)) :EPS SKOCZ DO PIERW:PIERW:LICZ
LICZ) ZR1=ZR ZI1=ZI
ZR=ZR+DZR ZI=ZI+DZI
SKOCZ DO E1
PIERW)
A(N)=ZR B(N)=ZI N=N-1 LINIA 1
GDY N:0 SKOCZ DO KONIEC:KONIEC:0
DLA I=1:1:N
A(I)=A(I)+A(I-1).ZR-B(I-1).ZI
```

$B(I) = B(I) + B(I-1) \cdot ZR + A(I-1) \cdot ZI$
POWTORZ I
SKOCZ DO UTART
KONIEC)
WROC

WYWOŁANIE: FPIERWIASKI(N:A#:B#:EPS)

gdzie:

- N - całkowita=stopień wielomianu,
 $A\#, B\#$ - nazwy bloków zmiennych rzeczywistych zawierających kolejne składowe rzeczywiste i urojone współczynników wielomianu $A(N+1), B(n+1)$
EPS - rzeczywista=założona dokładność obliczania pierwiastków /należy zakładać bliską dokładność maszyny - w celu uzyskania dokładnych wyników/

W trakcie wykonywania programu, zawartości bloków $A\#$ i $B\#$ ulegają zniszczeniu, a na ich miejsce wpisują się kolejno znalezione składowe pierwiastków:

$A(1)+iB(1), A(2)+iB(2), \dots, A(n)+iB(n)$
/wartość $A(0)+iB(0)$ - nie oznacza pierwiastka/.

3.4. Działania na macierzach

3.4.1. Obliczenie wartości wyznacznika

Podprogram funkcyjny - FWYZNACZNIK - oblicza wartość wyznacznika macierzy A.

```
FWYZNACZNIK:N:A#
STRUKTURA A(N+1:N+1)
W=1,O
DLA I=0:1:N
C=0
DLA J=I:1:N
GDY ABS(C):ABS(A(J:I)) SKOCZ DO O:O:E
C=A(J:I) K=J
E)
POWTORZ J
GDY C:O SKOCZ DO ET:O:ET
W=0 SKOCZ DO KONIEC
ET)
GDY I:K SKOCZ DO O:E1:O
DLA J=I:1:N
D=A(I:J) A(I:J)=A(K:J) A(K:J)=D
POWTORZ J
W=-W
```

E1)

W=W.C

GDY I:N SKOCZ DO O:KONIEC:O

DLA J=I+1:1:N

D=A(J:I)/C

DLA K=I+1:1:N

A(J:K)=A(J:K)-D.A(I:K)

POWTORZ K

POWTORZ J

POWTORZ I

KONIEC)

FWYZNACZNIK=W

WROC

WYWOŁANIE: FWYZNACZNIK(N:A#)

gdzie:

N - całkowita określająca wymiar macierzy

A# - macierz kwadratowa o strukturze A(N+1:N+1)

3.4.2. Transpozycja

Podprogram FTRANSPOZYCJA - wyznacza macierz transponowaną

A^T danej macierzy A.

FTRANSPOZYCJA:N:A#

STRUKTURA A(N+1:N+1)

DLA I=0:1:N-1

DLA K=I+1:1:N

W=A(I:K) A(I:K)=A(K:I) A(K:I)=W

POWTORZ K

POWTORZ I

WROC

WYWOŁANIE: FTRANSPOZYCJA(N:A#)

gdzie:

N - całkowita - określająca wymiar macierzy A

A# - macierz o strukturze A(N+1:N+1)

Macierz transponowana podstawiana jest w miejsce macierzy wyjściowej.

3.4.3. Odwracanie macierzy

Podprogram - FODWRACAMAC - wyznacza macierz odwrotną A^{-1}

danej macierzy A.

FODWRACAMAC:N:A#:P#:H#

STRUKTURA A(N:N) H=O

DLA I=0:1:N-1

C=0
DLA K=I:1:N-1
GDY ABS(A(K:I)) :ABS(C) SKOCZ DO E1:E1:0
C=A(K:I)
R=K

E1)
POWTORZ K
GDY C:0 SKOCZ DO E2:0:E2
NAPIS
MACIERZ WYJSCIOWA JEST OSOBLIWA
H=1 SKOCZ DO KONIEC
E2)
GDY R:I SKOCZ DO 0:LICZ:0

DLA J=0:1:N-1
C=A(R:J) A(R:J)=A(I:J) A(I:J)=C
POWTORZ J
LICZ) P(I)=R
C=A(I:I)
A(I:I)=1,0
DLA J=0:1:N-1
A(I:J)=A(I:J)/C
POWTORZ J

DLA K=0:1:N-1
GDY K:I SKOCZ DO 0:TAK:0
D=A(K:I)
A(K:I)=0
DLA J=0:1:N-1
A(K:J)=A(K:J)-D.A(I:J)
POWTORZ J
TAK)
POWTORZ K
POWTORZ I

DLA I=N-1:(-1):0
R=P(I)
GDY R:I SKOCZ DO 0:NIEPRZESTAWIAJ:0
DLA J=0:1:N-1
C=A(J:R) A(J:R)=A(J:I) A(J:I)=C
POWTORZ J
NIEPRZESTAWIAJ)
POWTORZ I
KONIEC)
WROC

WYWOŁANIE: FODWRACAMAC N:A#:P#:H#

gdzie:

- N - całkowita określająca wymiary macierzy A i P
A# - nazwa macierzy odwracana o strukturze A(N:N)

- P# - całkowite. Nazwa bloku roboczego o N komórkach
 H# - Nazwa zmiennej całkowitej. Parametr ten przyjmuje wartość H=0 dla macierzy A niesobliwej i H=1 dla macierzy osobliwej.

Obliczona macierz odwrotna A^{-1} podstawiana jest na miejsca macierzy wyjściowej A. W przypadku macierzy A osobliwej drukowany jest tekst: MACIERZ WYJSCIOWA JEST OSOBLIWA.

3.4.4. Dodawanie macierzy

Podprogram - FSUMAC - oblicza sumę macierzy C=A+B

FSUMMAC : A#:IA:KA:B#:IB:KB:C#:IC#:KC#
 STRUKTURA A (IA+1:KA+1)
 STRUKTURA B (IB+1:KB+1)

GDY IA:IB SKOCZ DO 0:0:E
 IC=IB SKOCZ DO KP

E) IC=IA
 KP) GDY KA:KB SKOCZ DO 0:0:ET
 KC=KB SKOCZ DO SUMA
 ET) KC=KA
 SUMA)

STRUKTURA C (IC+1:KC+1)

DLA I=0:1:IC

DLA K=0:1:KC

C(I:K)=0

POWTORZ K

POWTORZ I

DLA I=0:1:IA

DLA K=0:1:KA

C(I:K)=A(I:K)

POWTORZ K

POWTORZ I

DLA I=0:1:IB

DLA K=0:1:KB

C(I:K)=C(I:K)+B(I:K)

POWTORZ K

POWTORZ I

WROC

WYWOŁANIE: FSUMAC(A#:IA:KA:B#:IB:KB:C#:IC#:KC#)

gdzie:

IA, KA, IB, KB, IC, KC - całkowite określające wymiary macierzy A, B, C
 A#, B#, C# - nazwy macierzy o elementach rzeczywistych i strukturach podanych w programie

Struktura macierzy C(IC+1:KC+1) - zostaje określona w trakcie liczenia. Po wykonaniu programu na IC, KC i C podstawione zostaną wartości określające strukturę macierzy C i elementy macierzy C=A+B. Pozostałe parametry nie ulegają zmianie.

3.4.5. Mnożenie macierzy

Podprogram - FMNOZMAC - oblicza iloczyn macierzy C=A*B

```
FMNOZMAC:IA:KA:A#:IB:KB:B#:C#
STRUKTURA A(IA:KA)
STRUKTURA B(IB:KB)
STRUKTURA C(IA:KB)
'Musi zachodzić równość KA=IB'
DLA M=0:1:IA-1
DLA N=0:1:KB-1
S=0,0
DLA J=0:1:KA-1
S=S+A(M:J)*B(J:N)
POWTORZ J
C(M:N)=S
POWTORZ N
POWTORZ M
WROC
```

WYWOŁANIE: FMNOZMAC (IA:KA:A#:IB:KB:B#:C#)

gdzie:

IA, KA, IB, KB - całkowite - określające wymiary macierzy A, B, C

A#, B#, C# - nazwy macierzy o strukturach podanych w nagłówku podprogramu.

3.4.6. Diagonalizacja macierzy

Podprogram - FDIAGONALIZACJA - wyznacza wartości własne symetrycznej macierzy A metodą Jacobiego.

```
FDIAGONALIZACJA:N:EPS:A#
STRUKTURA A(N:N)
POCZ)
C=0,0
DLA L=0:1:N-2
DLA J=L+1:1:N-1
GDY ABS(C):ABS(A(L:J)) SKOCZ DO 0:0:ET
```

```

C=A(L:I)    I≠L    K=J
ET)
POWTORZ J
POWTORZ L
D=A(K:K)-A(I:I)
GDY ABS(C):EPS SKOCZ DO KONIEC:O:O
GDY ABS(2.C):ABS(D) SKOCZ DO E1:O:O
C=0,707106781 S=C SKOCZ DO OBROT
E1)
T=A(I:K)/(2.(A(K:K)-A(I:I)))
S=2.T/(1+T.T)
C=(1-T.T)/(1+T.T)
OBROT)
DLA J=0:1:N-1
W=A(J:I).C-A(J:K).S
A(J:K)=A(J:I).S+A(J:K).C
A(J:I)=W
POWTORZ J
DLA J=0:1:N-1
W=A(I:J).C-A(K:J).S
A(K:J)=A(I:J).S+A(K:J).C
A(I:J)=W
POWTORZ J
A(I:K)=(A(I:K)+A(K:I))/2
A(K:I)=A(I:K)
SKOCZ DO POCZ
KONIEC)
WROC

```

WYWOŁANIE: FDIAGONALIZACJA(N:EPS:A#)

gdzie:

N - całkowita określająca wymiar macierzy

A# - macierz współczynników rzeczywistych o strukturze
A(N:N)

EPS - rzeczywista określająca dokładność wyznaczania wartości własnych

Wyznaczone wartości własne podstawione są w miejsca elementów przekątniowych A(I:I).

3.4.7. Wartości własne i wektory własne macierzy

Podprogram - FWWW - wyznacza wartości własne i wektory własne symetrycznej macierzy A.

FWWW:N:EPS:A#:AM#

STRUKTURA A(N:N)

STRUKTURA AM(N:N)

DLA $J=0:1:N-1$
 DLA $L=0:1:N-1$
 $AM(J:L)=0,0$
 GDY $J:L$ SKOCZ DO $E:0:E$
 $AM(J:L)=1,0$
 E)
 POWTORZ L
 POWTORZ J
 POCZ)
 $C=0,0$
 DLA $L=0:1:N-2$
 DLA $J=L+1:1:N-1$
 GDY $ABS(C):ABS(A(L:J))$ SKOCZ DO $O:0:ET$
 $C=A(L:J) I=L K=J$
 ET)
 POWTORZ J
 POWTORZ L
 $D=A(K:K)-A(I:I)$
 GDY $ABS(C):EPS$ SKOCZ DO KONIEC,
 $GDY ABS(2.C):ABS(D)$ SKOCZ DO $E1:0:0$
 $C=0,707106781 S=C$ SKOCZ DO OBROT
 $E1)$
 $T=A(I:K)/2.D$
 $S=2.T/(1+T.T)$
 $C=(1-T.T)/(1+T.T)$
 OBROT)
 DLA $J=0:1:N-1$
 $W=A(J:I).C-A(J:K).S$
 $A(J:K)=A(J:I).S+A(J:K).C$
 $A(J:I)=W$
 POWTORZ J
 DLA $J=0:1:N-1$
 $W=A(I:J).C-A(K:J).S$
 $A(K:J)=A(I:J).S+A(K:J).C$
 $A(I:J)=W$
 POWTORZ J
 $A(I:K)=(A(I:K)+A(K:I))/2$
 $A(K:I)=A(I:K)$
 DLA $J=0:1:N-1$
 $W=AM(J:I).CcAM(J:K).S$
 $AM(J:K)=AM(J:I).S+AM(J:K).C$
 $AM(J:I)=W$
 POWTORZ J
 SKOCZ DO POCZ
 KONIEC)
 WROC

METODA: Jacobiego /zmodyfikowana/.

WYWOŁANIE: FWWW(N:EPS:A#:AM#)

PARAMETRY PODPROGRAMU:

N - całkowita określająca wymiary macierzy

EPS - rzeczywista określająca dokładność wyznaczania wartości i wektorów własnych

A# - macierz współczynników rzeczywistych o strukturze (N:N)

AM# - macierz współczynników rzeczywistych wektorów własnych

Po wykonaniu programu elementy przekątniowe macierzy A są wartościami własnymi, a odpowiadające im kolumny macierzy AM zawierają współrzędne wektorów własnych.

3.4.8. Rozwiązanie układu równań algebraicznych

Podprogram - FUKLADROWNAN - rozwiązuje układ równań algebraicznych liniowych, metodą elementów głównych Gaussa.

```
FUKLADROWNAN:A#:N:M
STRUKTURA A(N:N+M)
DLA L=0:1:N-1
C=0
DLA I=L:1:N-1
GDY ABS(C):ABS(A(I:L)) SKOCZ DO O:E1:E1
C=A(I:L) IT=I
E1) POWTORZ I
GDY ABS(C):0.000000001 SKOCZ DO SPRZECZNY;SPRZECZNY:0
A(IT:L)=A(L:L)
A(L:L)=C-1,0
DLA K=L+1:1:N+M-1
D=A(IT:K)
A(IT:K)=A(L:K)
A(L:K)=D
D=D/C
DLA I=0:1:N-1
A(I:K)=A(I:K)-D.A(I:L)
POWTORZ I
POWTORZ K
POWTORZ L
WROC
SPRZECZNY) NAPIS
UKLAD SPRZECZNY
#
WROC
```

WYWOŁANIE: FUKLADROWNAN (A#:N:M)

gdzie:

N - całkowita, określająca liczbę równań

M - całkowita, określająca liczbę kolumn wyrazów wolnych

A# - nazwa macierzy współczynników z dodatkowymi M kolumnami wyrazów wolnych

N) Przykład układu równań dla N=3 M=2

$$a_{00}x_0 + a_{01}x_1 + a_{02}x_2 = a_{03}, \quad a_{04}$$

$$a_{10}x_0 + a_{11}x_1 + a_{12}x_2 = a_{13}, \quad a_{14}$$

$$a_{20}x_0 + a_{21}x_1 + a_{22}x_2 = a_{23}, \quad a_{24}$$

Układ należy traktować jako M równoczesnych zadań, w których wyrazy wolne umieszczone są w kolejnych kolumnach wyrazów wolnych. Po wykonaniu działań wartości niewiadomych x_0, x_1, x_2 zostają wypisane na miejsca odpowiadających im wyrazów wolnych tj. na

$$a_{03} \quad a_{13} \quad a_{23} \quad i \quad a_{04} \quad a_{14} \quad a_{24}$$

3.5. Rachunek różniczkowy i całkowy

3.5.1. Rozwiązywanie układu równań różniczkowych metodą Runge Kutta Gille

Podprogram - FRKG - służy do rozwiązywania układu N równań różniczkowych zwyczajnych w postaci:

$$\frac{dx_i}{dx_0} = f_i(x_0, \dots, x_N) \quad \text{dla } i = 0, 1, \dots, N \quad /1/$$

$$f_0 = 1$$

```
FRKG:FPRA#:2:X#:AK#:N:AH:AQ#:ALFA(4):BETA(4):AMA(4)
WARTOSC ALFA=0,5:0,292893219:1,707106781 :0,16666666666
WARTOSC BETA=0,5:0,292893219:1,707106781 :0,33333333333
WARTOSC AMA=0,5:0,292893219:1,707106781 :0,5
```

```
FPRA(X#:AK4)
DLA J=0:1:3
AL=ALFA(J) BE=BETA(J) AM=AMA(J)
DLA I=0:1:N
AK(I)=AH.AK(I)
```

$AR = AL \cdot AK(I) - BE \cdot AQ(I)$
 $X(I) = X(I) + AR$
 $AQ(I) = 3,0 \cdot AR + AQ(I) - AM \cdot AK(I)$
 POWTORZ I
 POWTORZ J
 WROC

Przy danych wartościach początkowych x_1, x_2, \dots, x_N w punkcie x_0 , podprogram oblicza wartości x_1, x_2, \dots, x_N w punkcie $x_0 + AH$, gdzie AH - krok całkowania.

Podprogram - FRKG - korzysta z podprogramu obliczającego wartości funkcji f_i . Podprogram ten należy pisać osobno przy rozwiązywaniu każdego układu.

WYWOŁANIE: FRKG(FPRA#:X#:AK#:N:AH:AQ#)

gdzie:

FPRA# - nazwa podprogramu obliczającego wartości f_i prawych stron układu /1/. W nagłówku tego podprogramu pierwszym parametrem musi być nazwa bloku wartości funkcji f_i zakończona znakiem #, jako drugi należy podać nazwę bloku pochodnych funkcji.
 Przykład nagłówka: FPRZYKL:X#:XPRIM#

Dalsze parametry podprogramu FPRA muszą być podstawione rozkazem PODSTAW

X#	- blok wartości funkcji f_i
AK#	- blok roboczy o wymiarze AK(N+1)
N	- całkowita określająca ilość równań
AH	- rzeczywista = krok całkowania
AQ#	- blok roboczy o wymiarze AQ(N+1)

Przed pierwszym krokiem całkowania należy wyzerować ten blok.

Po każdym kroku całkowania blok X# zawiera obliczone wartości $X(I)$ dla $I = 1, 2, \dots, N$ dla parametru $X(0) = X(0) + AH$

3.5.2. Obliczenie wartości całki oznaczonej

Podprogram funkcyjny - FCALKAG - oblicza wartość całki oznaczonej metodą Gaussa-Legendre

```

FCALKAG:XMIN:XMAX:FUN#(1):A(3):Z(3)
WARTOSC A=0,555555555:0,888883888:0,555555555
WARTOSC Z=-0,774596669:0,000000000:0,774596669
S=0
DLA I=0:1:2
X=(XMIN+XMAX)/2,0+Z(I).(XMAX-XMIN)/2
S=S+A(I).FUN(X)
POWTORZ I
FCALKAG=S.(XMAX-XMIN)/2
WROC

```

WYWOŁANIE: FCALKAG (XMIN:XMAX:FUN#)

gdzie:

XMIN - rzeczywista = dolna granica całkowania

XMAX - rzeczywista = górna granica całkowania

FUN# - nazwa podprogramu funkcyjnego /lub funkcji standar-
dowej/ podstawianej na funkcję podcałkową.

Pierwszym parametrem tego podprogramu musi być zmien-
na niezależna X. Pozostałe parametry muszą być okreś-
lone rozkazem PODSTAW przed odwołaniem się do pod-
programu FCALKAG.

Przykład wywołania:

Z=FCALKAG (5:7:FUNPODC#)

3.6. Programowanie liniowe

3.6.1. Zadanie transportowe

Podprogram - FTRANSPORT - rozwiązuje zbilansowane zadanie
transportowe.

Sformułowanie zadania:

W magazynach o numerach $0, 1, \dots, m$ znajdują się zapasy towa-
ru x_0, x_1, \dots, x_m

Odbiorcy o numerach $m+1, m+2, \dots, n$ zgłaszają zapotrzebowan-
ie na ten towar w wysokościach $x_{m+1}, x_{m+2}, \dots, x_n$.

Wszystkie dane x_0, x_1, \dots, x_n muszą być nieujemne, i zapo-
trzebowania powinny zbilansować się z zapasami tzn.

$$x_0 + x_1 + \dots + x_m = x_{m+1} + x_{m+2} + \dots + x_n$$

Dane są koszty C_i przesłania jednostki towaru z magazynu do
odbiorcy, które można zapisać na skrzyżowaniu wiersza maga-
zynu z kolumną odbiorcy w tablicy, którą poniżej podano

przykładowo dla $m=2$ $n=6$. Macierz ponumerowano wierszami.

	x_0	c_0	c_1	c_2	c_3
1	x_1	c_4	c_5	c_6	c_7
$m = 2$	x_2	c_8	c_9	c_{10}	c_{11}
Zapas	x_3	x_4	x_5	x_6	
Zapotrzeb.	3	4	5	6 = n	

Odbiorcy

Należy tak poprzesyłać towary, aby opróżnić magazyny, zaspakoić odbiorców i aby koszt przesyłki był jak najniższy.

FTRANSPORT:N:M:IX#:IC#:P#:IV#:IF#

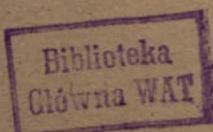
```

IT=0  K=M+1  J=0  IV(0)=0
ID=1  IE=N-M  IW=IE.(M+1)-1
E1)
IV(K)=IC(J)-IV(IT)  L=IX(IT)-IX(K)
GDY L:0 SKOCZ DO 0:O:ET
R=IT  IT=K  K=R
R=ID  ID=IE  IE=R
ET)
IX (IT)=ABS (L)
P(K)=IT
K=K+1
GDY J:IW SKOCZ DO 0:CZYITEROWAC:0
J=J+ID
SKOCZ DO E1

CZYITEROWAC)
J=0  I=0
POWI)
IDO=IV(I)  K=M+1
POK)
IF=IC (J)-IDO-IV(K)
GDY IF:0 SKOCZ DO POPRAW:0:0
J=J+1  GDY K:N  SKOCZ DO 0:WYKI:0
K=K+1
SKOCZ DO POK
WYKI)  GDY I:M  SKOCZ DO 0:WYI:0
I=I+1

```

SKOCZ DO POWI
 WYI) SKOCZ DO KONIEC
 POPRAW
 H=0 R=I
 E3)
 J=P(I) P(I)=K G=IX(I) IX(I)=H H=G
 GDY I:IT SKOCZ DO 0:E4:0
 K=I I=J
 SKOCZ DO E3
 E4)
 G=6870000000
 I=R
 E5) I=P(I)
 GDY G:IX(I) SKOCZ DO E6:E6:0
 G=IX(I) IT=I
 E6) I=P(I)
 GDY I:R SKOCZ DO E5:0:E5
 E7) IX(I)=IX(I)+G G=G-G I=P(I)
 GDY I:R SKOCZ DO E7:0:E7
 ICC=JV(0)
 DLA K=0:1:N
 I=K IV(K)=IV(K)-ICC
 E8)
 GDY I:R SKOCZ DO 0:E9:0
 GDY I:IT SKOCZ DO 0:E1:0
 I=P(I) SKOCZ DO E8
 E9)
 IV(K)=IV(K)+IF
 E10)
 GDY K:M SKOCZ DO E11:0:E11
 IF=-IF
 E11)
 POWTORZ K
 SKOCZ DO CZYITEROWAC
 KONIEC) IF=0
 DLA I=0:1:N
 K=P(I) IF=IF+IX(I).(IV(I)+IV(K))
 POWTORZ I
 'W X(I) ZNAJDUJE SIE PRZESYLKA Z MAGAZYNU I DO ODBIORCY
 P(I) I=0,1,...,N'
 WROC
 WYWOLANIE: FTRANSPORT(N:M:IX#:IC#:P#:IV#:IF#)
 gdzie
 M - numer ostatniego magazynu /licząc od zera/
 N - numer ostatniego odbiorcy
 IX# - blok zapasów i zapotrzebowania IX(0), IX(1), ..., IX(N)
 IC# - blok kosztów transportu IC(0), IC(1), ..., IC((M+1)(N-M)-1)



P# - blok numerów /roboczy/ $P(0), P(1), \dots, P(M),$
 $P(M+1), \dots, P(N)$

IV# - blok potencjałów /rpbczy/ $IV(0), IV(1), \dots, IV(M),$
 $IV(M+1), \dots, IV(N)$

IF# - całkowita przyjmująca wartość funkcji celu - określająca koszt ogólny transportu.

W wywołaniu należy określić parametry: M, N, IX#, IC#
Po wykonaniu programu elementy bloku IX# i całkowita IF zawierają:

$X(0) =$ przesyłka z magazynu 0 do odbiorcy $P(0)$

• •

$X(M) =$ - " - - " - M - " - P(M)

$X(M+1) =$ - " - - " - P(M+1) - " - P(M+1)

• •

$X(N) =$ - " - - " - P(N) - " - N

IF = wartość funkcji celu = ogólny koszt transportu.



