BIURO PROJEKTÓW PRZEMYSŁU MIĘSNEGO WARSZAWA UL. CHOCIMSKA 28

PODRĘCZNIK PROGRAMOWANIA MASZYNY CYFROWEJ ZAM-41

SYSTEM AUTOMATYCZNEGO KODOWANIA SAKO

AUTORZY
MGR INŻ. BARBARA JUNG
MGR INŻ. KAZIMIERZ LEWANDOWSKI

WARSZAWA MAJ 1972 R

III 1.028.132

BIURO PROJEKTÓW PRZEMYSŁU MIĘSNEGO
Warszawa, ul. Chocimska 28

Do użytku służbowego

SYSTEM AUTOMATYCZNEGO KODOWANIA
SAKO

Podręcznik programowania maszyny cyfrowej

ZAM–41

Autorzy:
Mgr inż. Barbara Junq
Mgr inż. Kazimierz Lewandowski

WARSZAWA MAJ 1972

Podręcznik zawiera systematyczny wykład podstawowych elementów informatyki i programowania wraz z przykładami w języku SAKO i jest przeznaczony głównie dla personelu inżynieryjno-technicznego zatrudnionego w biurach projektowych, który po raz pierwszy zapoznaje się z problematyką ETO w procesie projektowania.

Zbiór przykładów zamieszczono w podręczniku w wersji oryginalnej, to znaczy, że zarówno teksty programów jak i wyniki są kserograficzną kopią oryginałów maszynowych.

BIBLIOTEKA NARODOWA BN

III 1.028.132

Okładkę projektował
mgr inż. arch. Roman Dragan

Zdjęcia wykonała
mgr inż. Danuta Walicka

WOSI „WSPÓLNA SPRAWA" PZMP W-wa, ul. Puławska 108, zlec. 453/72 nakł. 500 + 20 egz.

1973 E05397/

Podręcznik niniejszy opracowano na podstawie wykładów prowadzonych w Biurze Projektów Przemysłu Mięsnego w styczniu 1972 r.

Jego wydanie stało się możliwe przede wszystkim dzięki osobistej opiece dyr. Biura mgr inż. Jana Wawra, którego energii, bezkompromisowej postawie i niestrudzonej pracy zawdzięczamy wprowadzenie Elektronicznej Techniki Obliczeniowej jako narzędzia do warsztatu projektanta.

Wielką zachętą były również dla nas codzienne żądania pracowników biura domagających się tej publikacji.

W pracy nad zasadniczym układem merytorycznym preferującym przede wszystkim praktyczne aspekty programowania i zagadnienia inżynierskie korzystaliśmy głównie z krytycznych uwag z-cy dyr. d/s technicznych mgr inż. Emila Jastrzębskiego.

Układ ćwiczeń i ich przejrzystą formę wydawniczą zawdzięczamy inż. Tadeuszowi Tomkiewiczowi, natomiast opracowanie i obliczenie na komputerze zadań przykładowych naszym asystentom: Barbarze Kiersz i Jolancie Makowskiej, będącej również autorką rozdziału "Instrukcja Czynności Operatorskich".

Wszystkim wyżej wymienionym, a także tym, którzy w jakikolwiek sposób przyczynili się do wydania podręcznika, tą drogą składamy serdeczne podziękowania.

Autorzy

SPIS TREŚCI

Str.

1. Przedmowa 9

2. Elementy procesu przetwarzania informacji 10
   2.1. Informatyka (definicja) 11
   2.2. Informacja (definicja) 11
   2.3. Przetwarzanie danych (definicja) 11
   2.4. Komputer (definicja) 11
   2.5. Sygnały dyskretne (definicja) 11
   2.6. Bit-byte 12

3. Podstawowe elementy niezbędne do organizacji procesu obliczeń (przetwarzania) za pomocą komputera 12
   3.1. Człowiek 12
   3.2. Komputer .....
   3.3. Problem 13
   3.4. Model fizyczny lub logiczny problemu 13
   3.5. Algorytm obliczeniowy 13
   3.6. Model matematyczny 13
   3.7. Sieć działań lub schemat blokowy 14
   3.8. Przykład A 14
   3.9. Przykład B 15
   3.10. Program (definicja, autokody, przykład programu) 16
   3.11. Nośniki informacji 18
   3.12. Skrótowy, podstawowy schemat pracy na komputerze 21

4. System binarny dwójkowy, zerojedynkowy 21

5. Autokod (język) SAKO 24

6. Zapis liczb w komputerze 25
   6.1. Liczby całkowite 26
   6.2. Liczby ułamkowe 26
   6.3. Zapis oktalny 27
   6.4. Elementy algebry logiki - algebra Boole'a 28

7. Komputer ZAM-41 32
   7.1. Opis ogólny 32
   7.2. Czołówki programów 43
      7.2.1. Czołówki systemowe 43
      7.2.2. Czołówki języka SAKO 47
   7.3. Instrukcja czynności operatorskich 51
      7.3.1. Podłączanie stacji taśm magnetycznych 51
      7.3.2. Przygotowanie maszyny do wczytania programu 54

Str.

7.3.3. Sprowadzanie Systemu Operacyjnego do PAO i na bęben 55
7.3.4. Obsługa drukarki wierszowej 55
7.3.5. Czynności operatorskie przy monitorze 56
7.4. Wyprowadzanie taśmy binarnej programu 61
7.5. Charakterystyka pamięci taśmowej 61
8. Programowanie 66
8.1. Ogólna struktura programów w języku SAKO 66
8.1.1. Wykaz zdań języka SAKO 69
8.2. Forma i opis pojęć podstawowych 74
8.2.1. Liczba całkowita 74
8.2.2. Liczba ułamkowa 74
8.2.3. Blok liczbowy 74
8.2.3.1. Tablica 76
8.2.3.2. Tablica oktalna 76
8.2.3.3. A (= I, = J, = K, ..., = Z) = W 76
8.2.4. Zmienne proste 77
8.2.5. Zmienne indeksowane 78
8.2.6. Etykiety 78
8.2.7. Funkcje standardowe 78
8.2.8. Wyrażenia arytmetyczne 80
8.2.9. Kolejność wykonywania działań w wyrażeniu arytmetycznym 81
8.2.10. Rozdział 82
8.2.11. Podprogram 82
8.2.12. Struktura 84
8.2.13. Całkowite 84
8.2.14. Koniec 84
8.2.15. Koniec rozdziału 85
8.2.16. Formuła arytmetyczna 85
8.2.17. Pętla 86
8.2.18. Skocz do α 88
8.2.19. Skocz według $\alpha, \beta, \ldots, \omega$ 88
8.2.20. Idź do rozdziału 89
8.2.21. Gdy A > B : α, inaczej β 89
8.2.22. Gdy A = B : α, inaczej β 90
8.2.23. Gdy klucz I : α, inaczej β 90
8.2.24. Stop β 91
8.2.25. Czytaj: lista 92
8.2.26. Czytaj wiersz: lista 93
8.2.27. Czytaj oktalnie: lista 93

Str.

8.2.28.Drukuj (I): lista 94
8.2.29.Drukuj (I,J): lista 94
8.2.30.Drukuj (I,J,K) : lista 95
8.2.31.Drukuj wiersz: lista 96
8.2.32.Drukuj oktalnie: lista 96
8.2.33.Spacja n 96
8.2.34.Tekst:..... 97
8.2.35.Tekst wierszy n: 97
8.2.36.Komentarz 97
8.2.37.Linia n 98
8.2.38.Strona 98
8.2.39.Język SAS, język SAKO 98
8.2.40.Czytaj z bębna od I : lista 101
Pisz na bęben od I : lista 101

9. Przykłady programów w języku SAKO 103

Przykład Nr 1 104
Przykład Nr 2 108
Przykład Nr 3 113
Przykład Nr 4 118
Przykład Nr 5 121
Przykład Nr 6 125
Przykład Nr 7 129
Przykład Nr 8 133
Przykład Nr 9 137
Przykład Nr 10 141
Przykład Nr 11 145
Przykład Nr 12 149
Przykład Nr 13 153

10. Praktyczne wskazówki dla pracowników inżynieryjno-technicznych korzystających z ETO 157

11. Spis błędów sygnalizowanych przez translator języka SAKO (błędy formalne)..... 160

12. Wykaz błędów sygnalizowanych przez funkcje języka i programy obsługujące program wynikowy 164

13. Przyczyny zakończenia programu użytkowego 165

14. Przykład algorytmu 166

Dodatek 169

Przykłady zastosowań komputerów do obliczeń inżynierskich

A) Obliczenie instalacji centralnego ogrzewania 170
B) Obliczenie ustroju prętowego 176
C) Przebieg ciśnień i temperatur w silniku wysokoprężnym 180
D) Najkrótsze drogi w sieci od jednego węzła do wszystkich pozostałych /drzewo ekonomiczne/ 183

1. P R Z E D M O W A

Celem niniejszego podręcznika jest dostarczenie pracownikom inżynieryjno-technicznym minimum praktycznej wiedzy niezbędnej do:

1.1. zrozumienia całego procesu przetwarzania informacji za pomocą komputera przez opanowanie podstawowej terminologii informatyki i podstaw języka autokodu SAKO,

1.2. uzyskania realnej, praktycznej możliwości korzystania zarówno z komputera, jak i z niezbędnych urządzeń towarzyszących,

1.3. nabycia umiejętności sporządzania zarówno graficznych schematów blokowych, jak i pełnowartościowych programów branżowych,

1.4. korzystania z programów już opracowanych lub podprogramów bibliotecznych,

1.5. zrozumienia i dostrzegania możliwości zastosowania ETO w codziennej pracy projektanta każdej branży.

Autorzy podręcznika zdają sobie sprawę, że wobec licznych i na wysokim poziomie stojących publikacji w tej dziedzinie, podręcznik niniejszy może razić fachowców zbytnią prostotą i lapidarnością opisów oraz definicji, jak również stopniem trudności załączonych przykładów.

Jednakże wieloletnia praca w biurach projektów i liczne wykłady prowadzone przez autorów dla personelu inżynieryjno-technicznego zdają się utwierdzać w przekonaniu, że tylko taki system wykładu przynosi pozytywne rezultaty, przynajmniej w pierwszej fazie wprowadzania Elektronicznej Techniki Obliczeniowej do codziennej praktyki w biurze projektowym.

Przyczyną tego stanu rzeczy jest po prostu fakt, że olbrzymia większość inżynierów nie otrzymała na uczelniach nawet "podstawowego wykształcenia" w tej dziedzinie, a sam "zmaterializowany" charakter pracy projektanta nie stwarza okazji do nawyków abstrakcyjnego myślenia.

Z kolei wielu wybitnych specjalistów w dziedzinie E T O odczuwa wyraźne trudności w zetknięciu się z "przyziemną" problematyką inżynierską, czego dowodem może być ubogi do dnia dzisiejszego software inżynierski, którego autorami są jednak w większości inżynierowie ze znajomością E T O, a nie matematycy ze znajomością inżynierii.

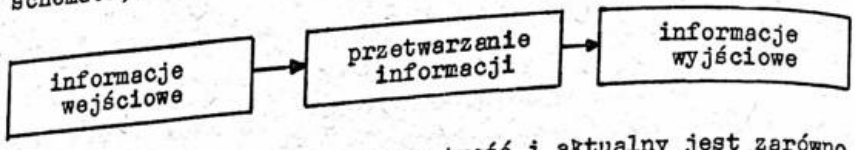
Wieloletnia obserwacja szkoleń stojących na tzw. "wysokim poziomie",a kończących się z reguły fiaskiem i "naturalną śmiercią" tej techniki niemal nazajutrz po zakończeniu wykładów i ćwiczeń, oraz pełny sukces E T O w B.P. Przemysłu Mięsnego, zdaje się wskazywać, że przyjęta forma "prostej rozmowy" z czytelnikiem, a ściślej z projektantem lub stażystą,który nie widział maszyny cyfrowej rozpoczynająca się od abecadła informatyki i programowania jest najwłaściwsza.

Jeżeli praca spełni chociaż w części cele opisane powyżej, będziemy usatysfakcjonowani, a jeżeli czytelnik podczas pierwszego czytania tej pracy przynajmniej raz się uśmiechnie, będziemy zadowoleni.

2. ELEMENTY PROCESU PRZETWARZANIA INFORMACJI

W chwilach refleksji możemy zauważyć, że w codziennej działalności zawodowej wszelki proces myślowy polega na przetwarzaniu informacji, zarówno pochodzącej z zewnątrz, jak i tkwiącej zmagazynowanej w nas samych.

A więc proces myślowy świadomy czy nieświadomy odbywa się wg jednego zasadniczego schematu, a mianowicie:



Ten uproszczony schemat zawiera ogromną treść i aktualny jest zarówno dla odruchów noworodka, jak i pracy fizyka atomowego.

Mówiąc lapidarnie - to samo dzieje się w komputerze, a sposoby organizowania takiego procesu poznamy za chwilę. Każdy dowolnie pomyślany przykład jest typowy.

Pytanie: ile to jest 2 x 8 /informacja wejściowa/.

Chwila zastanowienia, mozolne obliczenie/przetwarzanie informacji/.

Odpowiedź 2 x 8 = 16 /16 jest właśnie informacją wyjściową/.

Czytelniku!

Opracowując projekt branżowy korzystasz z informacji wejściowych takich jak założenia projektowe, normy i normatywy, projekty wstępne innych branż itd.

Projektując przetwarzasz te informacje według reguł, praw i nawyków, które kiedyś przyswoiłeś sobie w trakcie studiów i praktyki /wczytałeś je, zapisałeś w pamięci/ a teraz tworzysz projekt będący zbiorem informacji wyjściowych. Otóż każdy rodzaj informacji możliwy do zapisania w postaci liczb oraz możliwy do przetworzenia według reguł matematyki może być wprowadzony do komputera.

W nim odbywa się w przybliżeniu to, co w naszym umyśle. Ale komputer jest bardzo niedoskonały i może tylko wykonywać bardzo szybko i bezbłędnie nasze polecenia.

Komputer możemy uważać za przedłużenie układu centralnego człowieka. Dysponuje on dużą pamięcią, szybkością wykonywania operacji i możliwością wyboru logicznej alternatywy.

W jaki sposób jest to możliwe i jakie podstawowe elementy są potrzebne do organizacji procesu obliczeń /przetwarzania/? Dowiemy się o tym za chwilę po opanowaniu elementarnych terminów informatyki, bez których nasze dalsze porozumienie się byłoby dosyć utrudnione.

2.1. Informatyka

Dziedzina nauki/ang.computer science/zajmująca się problemami przedstawienia i przekształcenia struktur informacyjnych,systemami do przetwarzania informacji - w szczególności maszynami matematycznymi i ich oprogramowaniem, oraz sposobami wykorzystania /ta ostatnia część definicji interesuje nas najbardziej/.

2.2. Informacja

W języku potocznym jest to wiadomość o przedmiotach, faktach, zjawiskach itp.utożsamiana często z komunikatem, a zaczerpnięta ze świata zewnętrznego; to wyróżnienie jednego z określonej liczby stanów tego samego zbioru, miara prawdopodobieństwa wystąpienia określonego stanu ze zbioru stanów.

Praktycznie informacja i dane używane są jako synonimy. Zbiór danych rozumiemy doskonale intuicyjnie, stykamy się z nim na codzień i dlatego pomijamy definicję.

2.3. Przetwarzanie danych

Przetwarzanie danych jest to usystematyzowany ciąg kolejnych działań wykonywanych na danych, np. sortowanie /porządkowanie i grupowanie/, scalanie, operacje arytmetyczne lub inne manipulacje i przekształcanie przeprowadzane w określonym celu.

Czytelnik wykonuje to codziennie.

2.4. Komputer = maszyna matematyczna = maszyna cyfrowa = elektroniczna maszyna cyfrowa = E M C

Są to synonimy tego samego przedmiotu uszeregowane według natężenia strachu powstającego na sam dźwięk tych słów u niewtajemniczonych.

2.5. Sygnały dyskretne - np. impulsy elektryczne są nośnikami informacji w elektronicznych maszynach cyfrowych /ich układ tworzy informację możliwą do odczytania/.

2.6. Bit - byte

Bit - skrócona angielska nazwa binary digit - cyfra dwójkowa; jest to najmniejsza jednostka informacyjna stanowiąca wyróżnienie jednego z dwu możliwych stanów zbioru jednakowo prawdopodobnych 0 lub 1 systemu dualnego.

Bit oznacza pojedynczą pozycję /miejsce w słowie maszynowym w kodzie binarnym/; szczegółowe wyjaśnienie patrz system binarny;

Byte /wym. bajt/ - oznacza 8-mio bitowe słowo dualne /binarne/ używane do kodowania znaków /cyfr, liter lub znaków specjalnych/, w szczególności jednego znaku alfanumerycznego lub dwóch cyfr.

Na tym kończymy na razie wykaz definicji podstawowych. Czytelnik po pierwszym czytaniu być może nie dostrzeże w nich materialnych treści, co jednak na pewno nastąpi w trakcie studiowania dalszych rozdziałów.

3. PODSTAWOWE ELEMENTY NIEZBĘDNE DO ORGANIZACJI PROCESU OBLICZEŃ /PRZETWARZANIA/ ZA POMOCĄ KOMPUTERA

3.1. Człowiek

Może to być osobnik o zasobie wiedzy dotyczącej E T O mieszczącej się w granicach

- od zera do najlepszej znajomości teoretycznej i praktycznej dotyczącej E T O.

3.2. Komputer

Może to być maszyna o dowolnym typie, lecz o parametrach technicznych wystarczających do rozwiązania zagadnienia.

Najbardziej charakterystycznymi parametrami komputera są:

- pojemność pamięci operacyjnej /liczba słów krótkich/,
- pojemność pamięci pomocniczych /liczba słów krótkich/,
- rodzaj urządzenia wejścia i wyjścia /czytniki, perforatory, drukarki wierszowe, monitory, czytniki kart perforowanych itd./,
- szybkość pracy tych urządzeń,
- języki programowania - zrozumiałe dla komputera tzw. autokody, w których można zaprogramować dany problem.

3.3. Problem

Wszelkie zagadnienia, które chcemy rozwiązać za pomocą operacji matematycznych i logicznych na podstawie praw i zasad obowiązujących w różnych dyscyplinach naukowych.

3.4. Model fizyczny problemu

Przez model fizyczny rozumiemy np.:

most, linię trakcji wysokiego napięcia,
wał korbowy samochodu, układ soczewek mikroskopu itd.

3.5. Algorytm obliczeniowy

Algorytm jest to opis całokształtu obliczeń w postaci układu wzorów z wyjaśnieniem sensu symboli w nim występujących, określeniem kolejności stosowania wzorów, sposobów postępowania, w wyjątkowych wypadkach z podaniem warunków brzegowych i ograniczeń.

Tak więc algorytmem obliczeniowym będzie:

- opis sposobu rozwiązania równania kwadratowego z podaniem wzorów na wyróżnik równania kwadratowego i pierwiastki równań,
- opis ułożenia dla ustroju prętowego układu równań liniowych i sposób ich rozwiązania np.metodą skróconego algorytmu Gaussa,
- opis metody Simplex dla konkretnego przedsięwzięcia itd.

Każdy algorytm winien oznaczać się przynajmniej większością niżej podanych cech:

- prostotą formy,
- podziałem na bloki logiczne,
- likwidacją długich wzorów i ich rozwinięć,
- właściwym wprowadzeniem indeksacji,
- opracowaniem dokładnej listy parametrów wejściowych /informacji wejściowych/ i parametrów wyjściowych - wyników /informacji wyjściowych/,
- podaniem metody sprawdzenia poprawności programu,
- podaniem metody sprawdzenia wyników,
- dyskusją dokładności uzyskiwanych wyników.

Sposób praktycznej realizacji tych licznych, a w większości niezbędnych, postulatów postaramy się pokazać na przykładach.

3.6. Model matematyczny

Przez model matematyczny rozumie się skończoną i pełną formę wzorów matematycznych umożliwiających zaprogramowanie zagadnienia. Potocznie modelu matematycznego i algorytmu nie odróżnia się.

BIBLIOTEKA NARODOWA BN

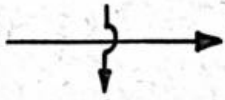
3.7. Sieć działań

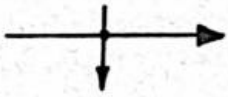
Sieć działań lub schemat blokowy/różnica polega jedynie na stopniu szczegółowości/jest graficznym odzwierciedleniem strony logicznej problemu.

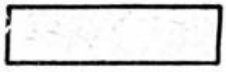
Opracowanie schematu blokowego jest najtrudniejszą częścią przedsięwzięcia. Z chwilą gdy schemat jest zakończony, kodowanie/zapisanie problemu w autokodzie/jest stosunkowo proste.

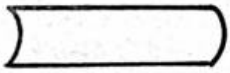
Przyjęte znakowania w sieciach działań i schematach blokowych:

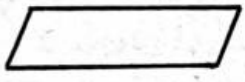
3.7.1. kierunek przepływu informacji

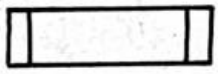
3.7.2.  nie zazębiające się kierunki przepływu informacji

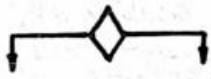
3.7.3.  kierunki przepływu informacji zazębiające się

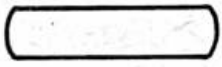
3.7.4.  dowolna operacja przetwarzania /blok operacyjny/

3.7.5.  symbol magazynowania informacji

3.7.6.  operacja wejścia i wyjścia /wprowadzenie i wyprowadzanie informacji/

3.7.7.  blok podprogramu

3.7.8.  blok decyzyjny

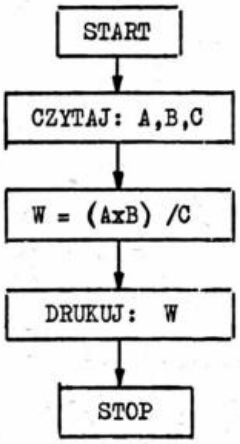
3.7.9.  START, STOP, KONIEC, POCZĄTEK

Dokładne, dynamiczne znaczenie powyższych oznaczeń zrozumie czytelnik studiując podane przykłady.

3.8. Przykład A

Sieć działań dla zaprogramowania wzoru

$$W = \frac{A \times B}{C}$$ gdzie A, B, C parametry liczbowe



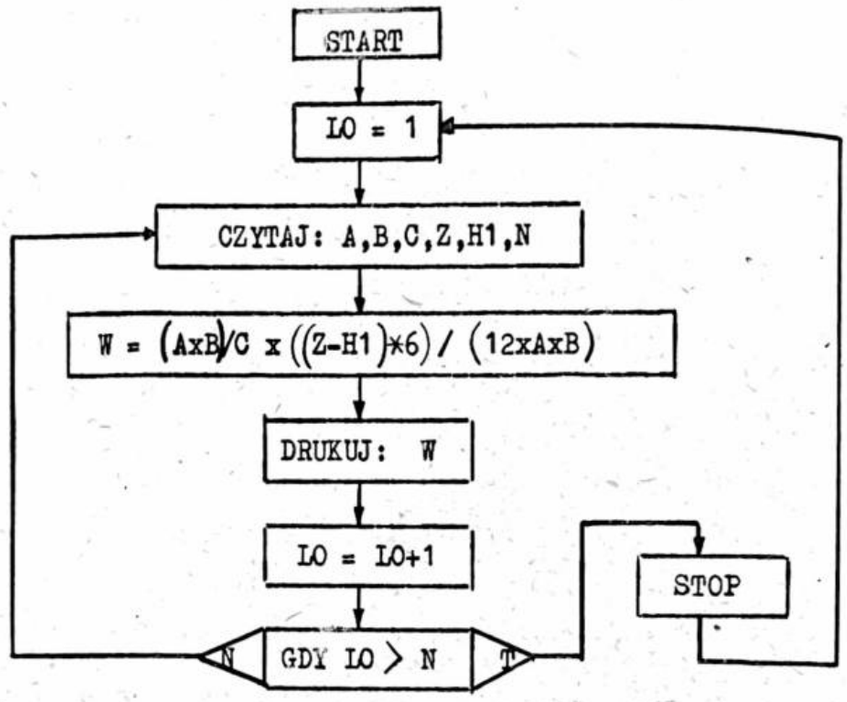
Powyższa sieć działań składa się z kolejnych, logicznych czynności niezbędnych do realizacji obliczenia wielkości W przy zadanych parametrach liczbowych A, B, C i wystarcza całkowicie do zaprogramowania komputera, w którym przebieg czynności będzie identyczny z "wyrysowanym" to znaczy:

1. rozpoczęcie pracy komputera	START
2. wczytanie przez komputer niezbędnych do obliczenia parametrów A, B, C	CZYTAJ: A, B, C
3. obliczenie wielkości W przez wstawienie parametrów A, B, C do wzoru	W = (A x B)/ C
4. drukowanie wartości W	DRUKUJ: W
5. zatrzymanie komputera przerwanie pracy	STOP

3.9. Przykład B

Sieć działań dla zaprogramowania wzoru:

$$W = \frac{A \times B \times (Z - H1)^6}{C \times 12 \times A \times B}$$



Zgodnie z zasadami sporządzania sieci działań kolejność operacji logicz. nych i rachunkowych jest jednoznacznie określona.

Czytelniku!

Wyobraź sobie, że jesteś maszyna cyfrowa rozumiejąca tylko kilka rozkazó i operacji logicznych.

Wyruszając ze STARTU "dojdziesz" do STOPU, idąc posłusznie tylko jedną dr gą wytyczaną strzałkami/znakami ścieżki logicznej/.

Wzór LO = LO + 1 nonsensowny w matematyce jest podstawowym "chwytem" pro gramowania oznaczającym: pozostaw w pamięci wartość LO, dołóż do niej 1 tak powiększoną zachowaj pod tym samym symbolem /adresem/. Dowiemy się o ty dokładnie w pkt. 8.6.7.

3.10. Program /definicja, autokody, przykład programu/

Program jest to bezpośredni, jednoznaczny opis czynności maszyny/ciąg rozkazów/ w języku wewnętrznym maszyny lub w tzw. autokodzie.

Autokod jest to sformalizowany język /np. SAKO, COBOL, FORTRAN, ALGOL/ tłu maczony na język wewnętrzny maszyny za pomocą tzw. translatora to jest programu tłumaczącego.

Dla jednego typu maszyny cyfrowej może być jeden lub więcej translatorów to znaczy, że programy mogą być opracowywane w tylu autokodach, ile jest translatorów.

Podkreśla się, ze programowanie w językach wewnętrznych jest bardzo uciążliwe, a autokody opracowano głównie z myślą o skróceniu i ułatwieniu programowania.

Poniżej podaje się program dla sieci działań z przykładu "B" opracowany w języku SAKO /System Automatycznego Kodowania/, który to autokod jest aktualny dla maszyn produkcji polskiej ZAM-41.

Autokod SAKO będzie też naszym językiem programowania używanym w niniejszym opracowaniu. Nie wszystkie rozkazy będą dla czytelnika zrozumiałe po pierwszym czytaniu.

Jednak po schematycznym zapoznaniu się z listą rozkazów tego autokodu Czytelnik winien powrócić do tego miejsca raz jeszcze, aby uświadomić sobie prostotę tego zapisu.

Podany niżej pełny program przykładu "B" ma na celu ukazanie Czytelnikowi owej tajemniczej "rozmowy z maszyną", o której często słyszy w sposób zakamuflowany i mało komunikatywny.

Tak wygląda program do przykładu "B".

```
ROZDZIAŁ        :  0
CAŁKOWITE       :  SK, N, LO, K
BLOK (0)        :  N, LO, K
BLOK (0)        :  W, A, B, C, H1, Z
1) CZYTAJ       :  N
   LO = 1
2) CZYTAJ       :  A, B, C, Z, H1
3) W = (A x B/C x((Z-H1) ж 6)/(12 x A x B)
4) DRUKUJ /5.2/ : W
   LO = LO + 1
5) GDY LO > N : 6, INACZEJ 7
6) STOP NASTĘPNY
7) SKOCZ DO 2
8) KONIEC       :  0
```

W tym zadaniu N oznaczało liczbę wariantów obliczeń /liczbę grup liczbowych, które należałoby wczytywać aby obliczyć N wariantów obliczeń W dla różnych grup A, B, C, Z, H1 liczb.

Takie rozkazy należało wydać komputerowi aby obliczył nam W jeżeli znamy A, B, C, H1, Z.

LO jest liczbą pomocniczą. Przydatność jej zrozumie Czytelnik w dalszych przykładach.

3.11. Nośniki informacji

Nośnikami informacji, na których wprowadzamy do komputera programy i dane są:

- taśmy perforowane /dziurkowane/
  czyli wąski pasek papieru, na którym w sekwencji dziurek jest wypisane wszystko co przekazujemy E M C, a więc cykle rozkazów oraz dane,
- karty perforowane - /o znormalizowanych wymiarach/
  nośnik informacji spełniający identyczną rolę jak taśma perforowana,
- taśmy magnetyczne przypominające z wyglądu taśmy magnetofonowe; nośniki wyższego rzędu, na których informacje są po prostu namagnesowywane w systemie zero-jedynkowym i jak dotychczas stanowią najbardziej niezawodny i trwały nośnik informacji.

Dwa ostatnie nośniki służą najczęściej do przechowywania zarówno programów, jak i zbiorów liczbowych wykorzystywanych wielokrotnie.

Taśmy i karty perforowane są podstawowymi nośnikami służącymi do wprowadzenia programów i danych do komputera.

Taśmy perforuje się na przyrządzie tzw. /dość niefortunnie/ dalekopisem o zasadzie działania zbliżonej do maszyny do pisania.

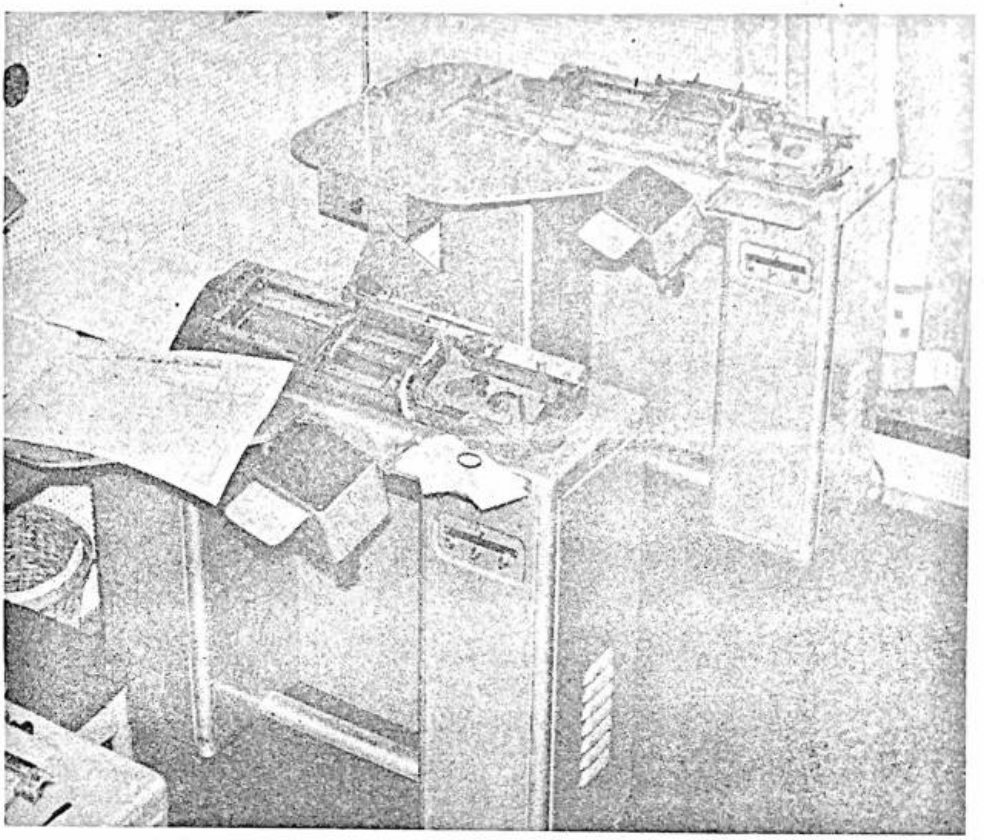
Równocześnie z taśmą wyperforowaną w określoną sekwencję dziurek z dalekopisu otrzymujemy czytelny dla nas tekst wypisany na tzw.tabulogramie - /pas papieru szerokość 27 cm/.

Dalekopis zaopatrzony jest również w czytnik pozwalający na uzyskanie z gotowej taśmy perforowanej kopii tej taśmy oraz tabulogramu.

Karty perforowane przygotowuje się na tzw. dziurkarkach kart. Poniżej zamieszczono zdjęcia zarówno dalekopisu jak i diurkarki.

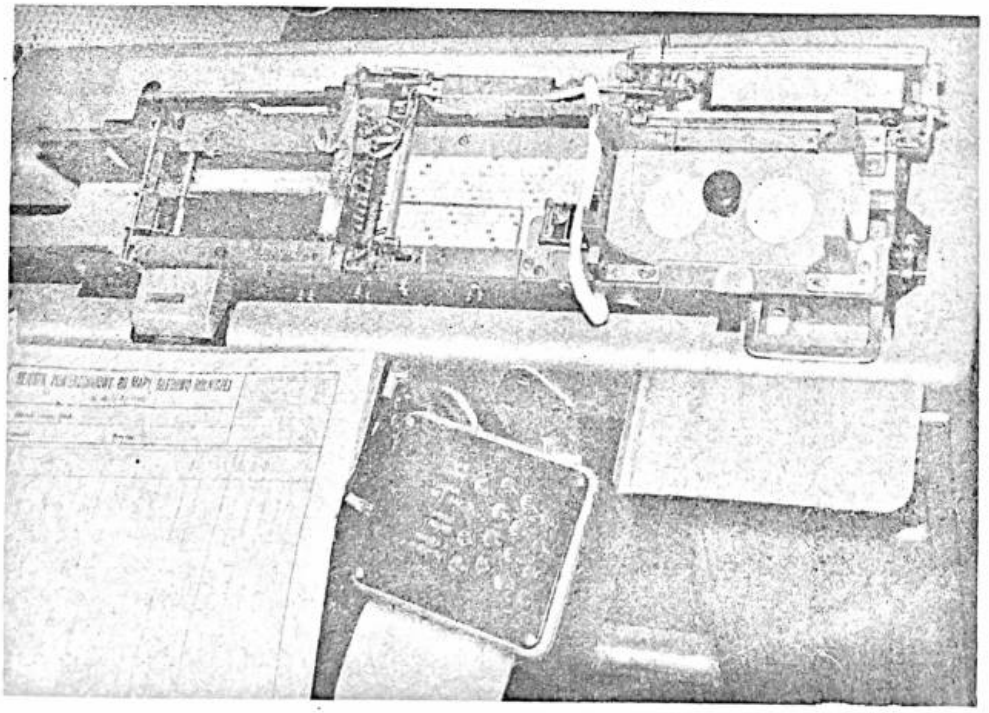
Informacje umieszczone /"wydziurkowane"/ na nośnikach informacji "wchodzą do komputera" i "wychodzą" za pomocą tzw. urządzeń wejścia i wyjścia /patrz opis maszyny ZAM 41 pkt 7/.

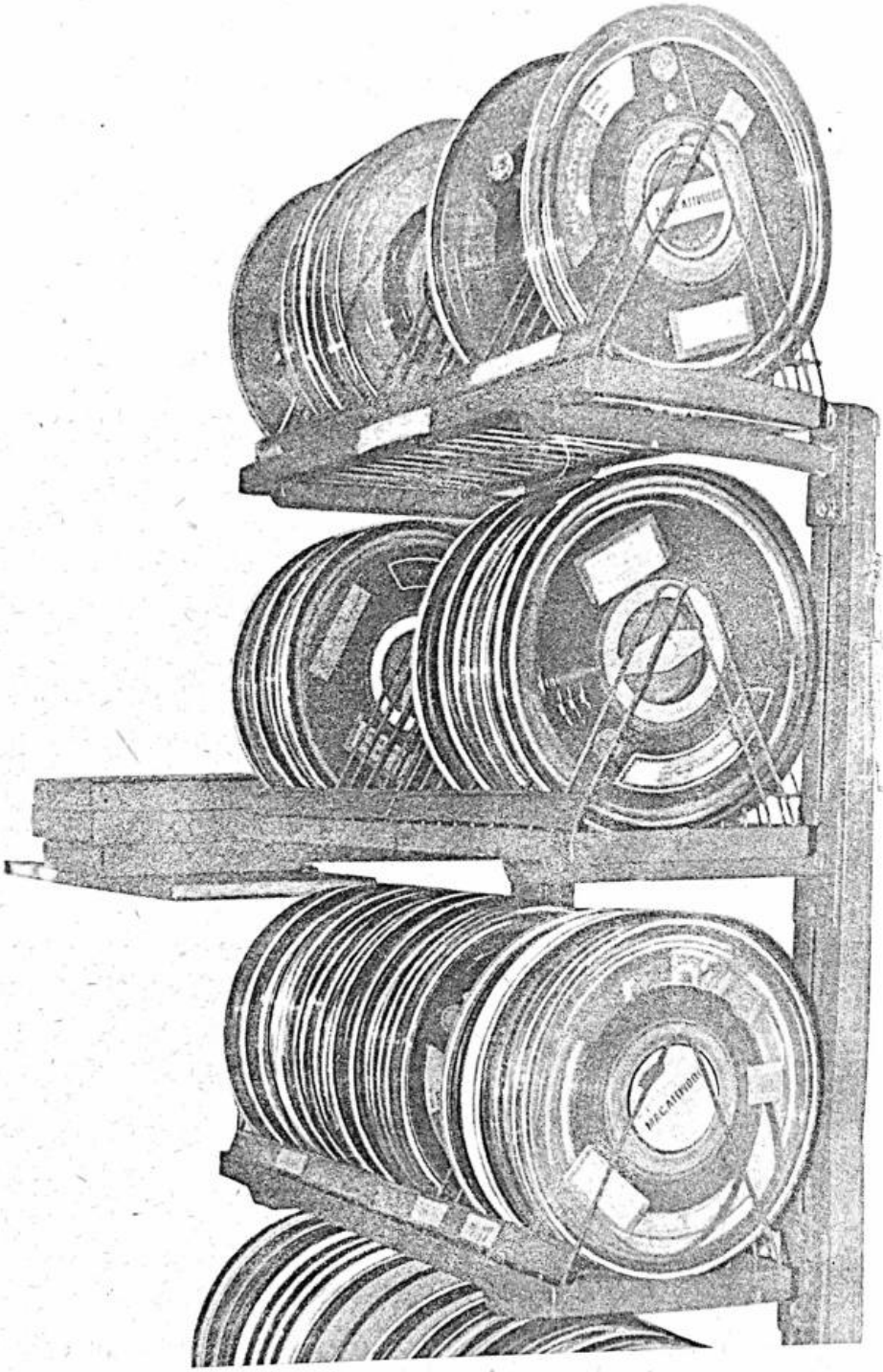
Współczesna komputeryzacja zna również taki nośnik informacji jak głos ludzki /czytanie programu i danych do głośnika komputera/, jednak dotychczasowe doświadczenia nie wydają się potwierdzać słuszności tego kierunku badań.



Dziurkarki kart perforowanych - widok ogólny

Dziurkarka kart perforowanych - pulpit





Podręczny regał taśm magnetycznych

3.12. Skrótowy podstawowy schemat pracy na komputerze

a/ problem lub model fizyczny,
b/ opracowanie modelu matematycznego,
c/ opracowanie schematu blokowego lub graficznej sieci działań,
d/ zakodowanie tego schematu w postaci programu w autokodzie,
e/ wyperforowanie /wydziurkowanie/ tego programu na taśmach lub kartach perforowanych /nośnikach informacji/,
f/ wyperforowanie danych liczbowych na nośnikach informacji,
g/ wprowadzenie programu do komputera za pomocą czytnika taśm lub kart,
h/ translacja /przetłumaczenie/ programu z autokodu na język wewnętrzny maszyny /maszyna "czeka" na dane liczbowe/,
i/ wprowadzenie do komputera danych liczbowych również za pomocą czytnika taśm lub kart perforowanych,
j/ wykonanie obliczeń przez komputer,
k/ wyprowadzenie wyników w postaci taśmy perforowanej, którą trzeba odczytać na dalekopisie, lub w postaci czytelnego tabulogramu wyprowadzonego na drukarkę wierszową.

4. SYSTEM BINARNY /dwójkowy, zerojedynkowy/

W znanym nam od dzieciństwa dziesiętnym systemie liczenia występują cyfry 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Za podstawę systemu przyjmuje się 10^n np.

$$1328 = 1 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 =$$
$$= 1000 + 300 + 20 + 8 = 1328$$

Pozycji każdej cyfry odpowiada pomnożenie tej cyfry przez określoną potęgę dziesięciu.

Otóż w systemie binarnym za podstawę przyjmujemy 2^n a za cyfry pozycyjne tylko 0 i 1. W związku z powyższym

$$1328 = 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 +$$
$$+ 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 =$$
$$= 1024 + 0 + 256 + 0 + 0 + 32 + 16 + 0 + 0 + 0 + 0 = 1328$$

Zestawmy te zapisy w kolumnie pionowej

Zapis dziesiętny	
1×10^3	= 1000
3×10^2	= 300
2×10^1	= 20
8×10^0	= 8
Razem	= 1328

Zapis binarny	
1×2^{10}	= 1024
0×2^9	= 0
1×2^8	= 256
0×2^7	= 0
0×2^6	= 0
1×2^5	= 32
1×2^4	= 16
0×2^3	= 0
0×2^2	= 0
0×2^1	= 0
0×2^0	= 0
Razem	= 1328

Tak więc zapis binarny liczby składa się z sumy jedynek lub zer pomnożonych przez 2^n. Wiemy, że odpowiednie potęgi 2 tworzą następujący ciąg wartości:

2^n	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
.....	128	64	32	16	8	4	2	1

Umawiając się, że odpowiednie pozycje stanowią odpowiednie wartości 2^n, liczbę 1328 w systemie binarnym zapiszemy

$$1328 = 10100110000$$

$$2^{10} \ldots\ldots\ldots\ldots 2^0$$

Ten prosty zapis okazał się przydatny do konstrukcji komputera /zwiększona liczba pozycji nie posiada wielkiego znaczenia/.

Poniżej podajemy prosty sposób przejścia z zapisu w systemie dziesiętnym na binarny.

"A"		"B"
137	1	2^0
68	0	2^1
34	0	2^2
17	1	2^3
8	0	2^4
4	0	2^5
2	0	2^6
1	1	2^7

a więc

$$137 = 1x2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0x2^4 + 2x2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

czyli 137 = 10001001

zapis dziedziętny — zapis binarny

Do celu dochodzimy w następujący sposób:

a/ piszemy żądaną liczbę np. 137 wraz z pionową kreską /kolumna "A"/;

b/ dzielimy tę liczbę przez 2; jeżeli jest to liczba nieparzysta za kreską piszemy 1, jeżeli parzysta 0;

c/ wynik dzielenia piszemy oczywiście bez reszty; tak samo postępujemy z dalszymi wynikami dzielenia;

d/ 1 lub 0 licząc od góry odpowiada właśnie wartości 2 podniesionej do odpowiedniej potęgi /jak w kolumnie "B"/.

Oczywiście komputer sam dokonuje "przekładu" na system binarny i uświadomienie sobie tego faktu jest bardzo korzystne.

Poniżej zamieszczono reguły działań arytmetycznych w systemie binarnym.

dodawanie	odejmowanie	mnożenie
0 + 0 = 0	0 - 0 = 0	0 x 0 = 0
1 + 0 = 1	0 - 1 = 1	0 x 1 = 0
0 + 1 = 1	1 - 0 = 1	1 x 0 = 0
1 + 1 = 0	1 - 1 = 0	1 x 1 = 0

1 przechodzi na następne "miejsce"

Dzielenie: np. 165 : 5

10100101 : 101 = 100001

101

00101

101

przykład: 165 + 21 = 186

165	1	$/2^0/$
82	0	
41	1	$/2^2/$
20	0	
10	0	
5	1	$/2^5/$
2	0	
1	1	$/2^7/$
0		

21	1	$/2^0/$
10	0	
5	1	$/2^2/$
2	0	
1	1	$/2^4/$

10100101 = 165
10101 = 21 +
10111010 = 186

Sprawdzenie:

$$10111010 = 1x2^7+0x2^6+1x2^5+1x2^4+1x2^3+0x2^2+1x2^1+0x2^0 = 128+32+16+8+2 = 186$$

Uwaga

W komputerze wszelkie obliczenia rachunkowe przeprowadza się za pomocą czterech działań arytmetycznych, stąd nazwa obliczenia numeryczne - taki jest sens tego terminu.

5. A U T O K O D /j ę z y k/ S A K O

Dla rozwiązania na maszynie cyfrowej problemu niezbędne jest zakodowanie /zaprogramowanie/ metody jego rozwiązania w języku dla niej zrozumiałym.

O ile sam problem może być złożony, o tyle po opracowaniu modelu matematycznego i schematu blokowego samo jego zakodowanie wymaga tylko wiadomości o charakterze raczej formalnym i odpowiedniej wprawy.

Stopień trudności w opanowaniu S A K O nie jest większy niż przy uczeniu się innych autokodów i średnio uzdolniony osobnik ze średnim wykształceniem jest w stanie, po pewnym czasie, opanować ten język do perfekcji.

Podkreśla się jednak, że nauczenie się autokodu nie polega tylko na pamięciowym opanowaniu rozkazów i rozumieniu ich funkcji oraz możliwości, ale nade wszystko na praktycznej umiejętności ich stosowania i maksymalnego wykorzystania przy kodowaniu skomplikowanych problemów różnych dyscyplin naukowych. Toteż do czynnego opanowania dowolnego autokodu konieczna jest praktyka nawet wieloletnia w programowaniu w ogóle.

Język S A K O jest autokodem opracowanym dla maszyn produkcji polskiej rodziny ZAM, a takie walory jak prostota rozkazów i polska terminologia czynią go wielce przydatnym w rozwiązywaniu problemów numerycznych. Opanowanie go przez pracowników różnych branż, jako pierwszego w ich praktyce je-

zyka programowania, ułatwia w znacznym stopniu uczenie się innych autokodów.

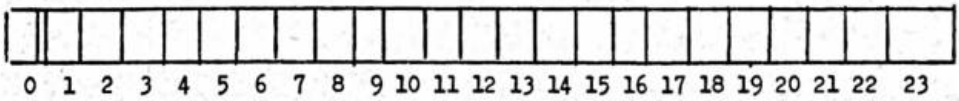
Nie należy oczywiście zapominać, że program napisany w autokodzie musi być dopiero przetłumaczony na program wynikowy w języku wewnętrznym maszyny. Dokonuje tego tzw. translator - czyli specjalny program tłumaczący. W każdym współczesnym komputerze istnieje na ogół kilka translatorów, co umożliwia programowanie w kilku autokodach.

6. ZAPIS LICZB W KOMPUTERZE

Jak już wspomniano wyżej w komputerach stosuje się najczęściej zapis binarny /dwójkowy/.

Pamięć komputera wyobrażamy sobie jako czystą kartę papieru, na którą dokonujemy wpisu liczb.

Pamięć składa się ze słów /komórek/ pamięci. Możemy je sobie wyobrazić w postaci prostokątów jak na rysunku.

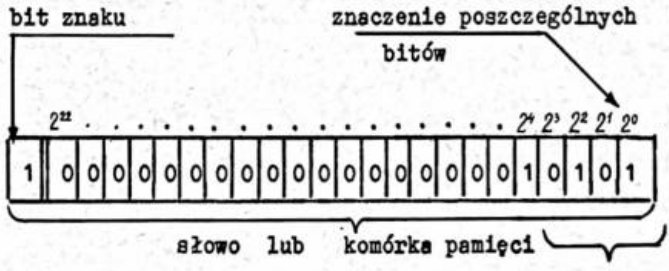


Otóż poszczególne miejsca w komórce pamięci nazywamy bitami patrz pkt 2.6. Liczba bitów określa więc długość komórki pamięci /słowa/ komputera, która może być różna dla różnych typów maszyn.

W ZAM-41 długość komórki /słowa/ wynosi 24 bity /24 miejsca/

w które można wpisywać jedynki lub zera ! ! !

Tak więc liczba - 21 zostanie zapisana w pamięci w następujący sposób /pierwszy bit jest zarezerwowany dla znaku liczby; 0 oznacza +,1 oznacza -/:



$$21 = 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$$

Tak wygląda zasada zapisu liczb w maszynie.

Pamięć komputera składa się z komórek, w których znajdują się tylko zera i jedynki.

Liczba komórek pamięci określa jej pojemność.

Czytelniku !

Jakkolwiek zapis w pamięci komputera odbywa się w sposób automatyczny, dla porządku musimy podać krótki, lecz dokładny i trochę skomplikowany opis zapisu liczb i rozkazów w komórkach pamięci.

Jeżeli nie wszystko będzie zrozumiałe, należy powrócić do tego fragmentu po przeczytaniu opisu maszyny i listy rozkazów w języku S A K O wraz z objaśnieniami.

6.1. <u>Liczby całkowite</u> zapisywane są w jednej komórce pamięci /słowie krótkim/ to znaczy na długość 23 bitów oraz na bicie zerowym, który jest przeznaczony na znak.

W związku z tym maksymalna liczba całkowita, jaką można wpisać do komórki pamięci, będzie wtedy, gdy w każdym bicie z wyjątkiem zerowego umieścimy 1.

Jak łatwo obliczyć wartość takiej liczby będzie wynosiła

$$2^{23} + 2^{22} + \ldots\ldots\ldots\ldots\ldots + 2^{0} = 8\ 388\ 607 = 2^{23} - 1$$

6.2. Liczby ułamkowe /rzeczywiste/ zapisywane są w dwu komórkach pamięci /słowo długie/ w postaci:

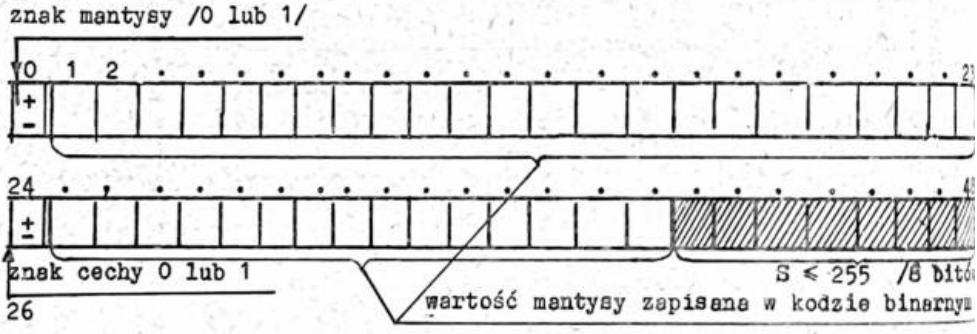
$$\boxed{\text{liczba} = m \cdot 2^{S}}$$

m - mantysa liczby; jej wartość $0{,}5 \leqslant m < 1$
s - cecha liczby $-255 \leqslant s \leqslant 255$

Tak więc największa liczba rzeczywista, którą możemy zapisać w maszynie jest:

$$2^{255} \approx 10^{77}$$

Graficzne wyobrażenie tej liczby /"dwukomórkowej"/ jest następujące:



W przypadku zapisywania w komórce rozkazu czyli słowa zapisanego w znakach alfanumerycznych tzn. za pomocą liter i cyfr znaczenie poszczególnych bitów jest następujące:

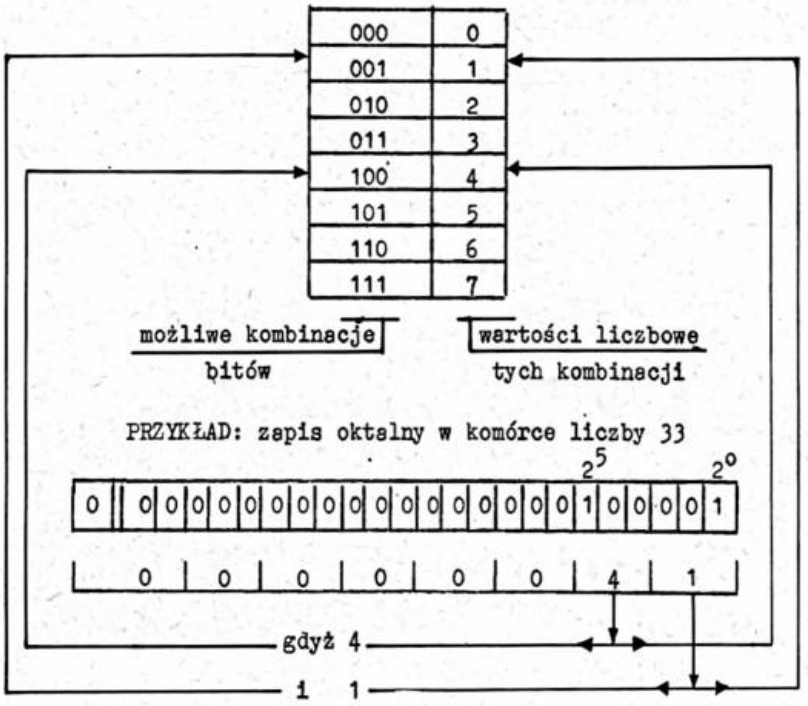
bit nr 0 - znak rozkazu
bit nr 1 - bit modyfikacji
bit nr 2 - bit pośredniego adresowania
bit nr 3 ÷ 8 - część operacyjna rozkazu
bity nr 9 ÷ 23 - część adresowa rozkazu

W przypadku specjalnego zainteresowania wewnętrzną strukturą tej maszyny, znaczenie tych terminów znajdzie Czytelnik w publikacji Instytutu Maszyn Matematycznych pt. "ZAM-41 Oprogramowanie" - część I.

Do programowania w języku S A K O wiadomości te nie są konieczne.

6.3. Zapis oktalny /zapis w systemie ósemkowym/

Niekiedy dokonuje się w maszynie zapisu /i wydruku/ w postaci oktalnej. W tym przypadku słowo zostaje podzielone po trzy bity, w których to grupach może być 7 kombinacji jedynek i zer wg tabeli zamieszczonej niżej:



Uwaga !

Znaku przy liczbach oktalnych nie ma !
Liczba 33 w systemie oktalnym będzie zapisana w następujący sposób:

00000041

Czytelniku !

To nie jest trudne ! Dla wprawy zapisz liczbę 1231. Najpierw zapisz w staci binarnej, jak przedtem, a następnie oktalnej. /która grupa po bity oznacza którą liczbę z tabeli ?/.

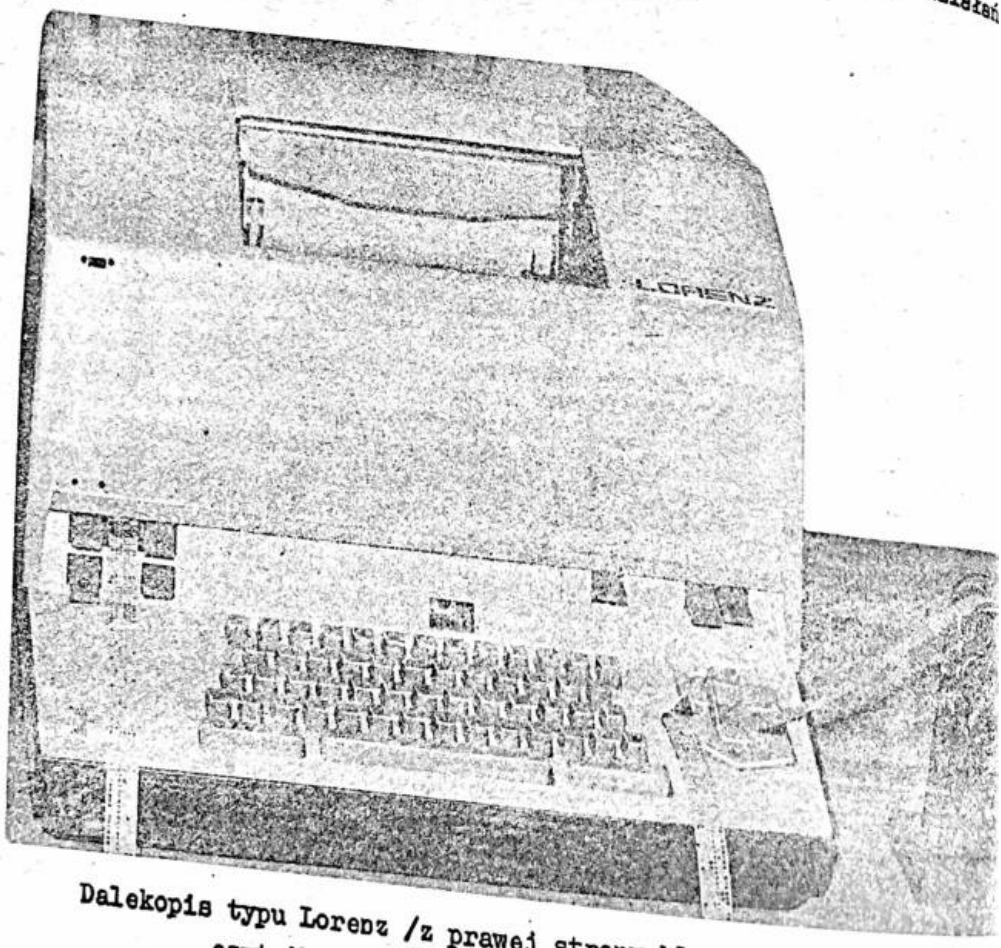
6.4. Elementy algebry logiki - albebry Boole'a

Algebra Boole'a /bulowska/ jest teorią działań nie wyprowadzającą poza strakcyjny zbiór dwuelementowy.

Elementy te zwane wartościami bulowskimi zaznaczamy symbolicznie przez ro /0/ i jedynkę /1/.

Czytelnik zapoznał się z najprostszym użyciem tych elementów w zapisie narnym /zero-jedynkowym/ sygnalizującym obecność jednego z dwu możliw stanów "coś jest /1/, lub czegoś nie ma /0/".

W języku SAKO wystarczająca będzie znajomość trzech podstawowych działań lowskich na zmiennych p i q /zdaniach składowych/.



Dalekopis typu Lorenz /z prawej strony klawiatury widoczny czytnik taśmy perforowanej/

- "i" tzw. koniunkcja lub iloczyn logiczny symbol $\wedge$ np. $p \wedge q$, co przedstawia zdanie p i q, które jest wówczas prawdziwe, gdy p i q są <u>jednocześnie</u> prawdziwe
- "albo" tzw. alternatywa lub suma logiczna, symbol np. $p \vee q$, co przedstawia zdanie p lub q, które jest wówczas prawdziwe, gdy przynajmniej <u>jedno</u> ze zdań składowych jest prawdziwe
- "negacja" czyli zaprzeczenie - symbole $\sim p$, $\bar{p}$, $-p$ co oznacza <u>"nie p"</u>

<u>przykłady</u> elementarnych działań bulowskich

<u>koniunkcje</u>	<u>alternatywa</u>	<u>negacja</u>
$-5 < x \wedge x < 10$	$-5 \leqslant x \vee x < 10$	$\sim p$

W języku SAKO przez <u>słowo bulowskie</u> rozumie się słowo, którego kolejne cyfry 0 lub 1 są traktowane jako elementy algebry Boole'a. W związku z tym przedstawione działania bulowskie znajdują szczególne zastosowanie w rozważaniach i operacjach na słowach i bitach maszyny.

<u>Identyczność</u>

zapis $\boxed{A \equiv B}$

co oznacza identyczność dwu słów, a więc identyczność bitów na odpowiednich miejscach.

0	1	1	0	0	....	$\equiv$	0	1	1	0	0	....
słowo A							słowo B					

<u>Suma bulowska</u>

zapis $\boxed{A \equiv B + C}$

oznacza, że wynikiem tej operacji jest słowo bulowskie powstałe według alternatywy /sumy logicznej/ tzn., że bitem wynikowym jest cyfra 1 wtedy i tylko wtedy, gdy przynajmniej jeden bit na odpowiednich miejscach w słowach logicznie sumowanych jest jedynką. Sumowanie wyobrażamy sobie osobno dla każdej "pionowej kolumny" bitów jak na rysunku.

"pionowa kolumna"
bitów

B	0	1	0	1	0	0	.....
C	1	0	0	1	0	1	.....
A	1	1	0	1	0	1	.....

Iloczyn bulowski

zapis $\boxed{A \equiv B \times C}$

oznacza, że wynikiem tej operacji jest słowo bulowskie powstałe według koniunkcji /iloczynu logicznego/ tzn., że bitem wynikowym będzie wówczas jedynka, gdy obydwa bity mnożone logicznie będą równocześnie jedynką

0	1	0	0	1	1	1		B
0	1	1	0	1	0	1		C

słowa mnożone logicznie

słowo wynikowe A

0	1	0	0	1	0	1	

Negacja

zapis $\boxed{A \equiv -B}$

oznacza, że wynikiem tej operacji jest słowo bulowskie A, które powstaje przez zmianę bitów na przeciwne w każdej pozycji.

0	1	0	1	1		B
1	0	1	0	0		A słowo wynikowe

Tak więc z lewej strony relacji ≡ powstaje w wyniku słowo bulowskie, które jest zwykłym słowem maszynowym tzn. komórką zapisaną w systemie zerojedynkowym /nie może to być zmienna indeksowana/, lecz o układzie bitów powstałym w zupełnie inny sposób niż przy operacjach arytmetycznych.

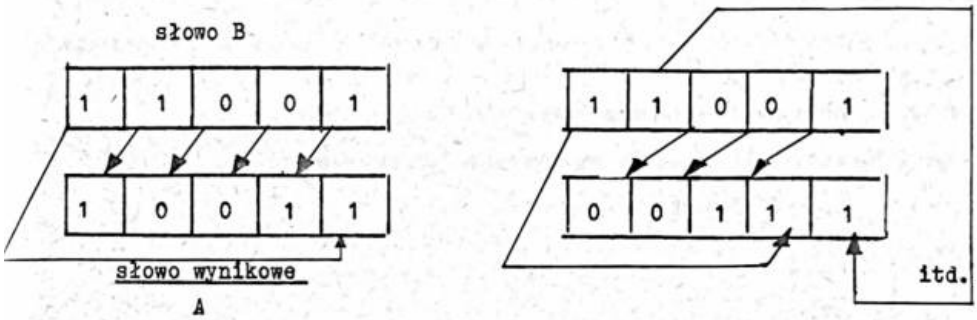
Przesunięcie cykliczne w lewo

zapis $\boxed{A \equiv B * n}$ n - liczba całkowita dodatnia

Wynikiem przesunięcia słowa B w lewo jest nowe słowo bulowskie A o układzie bitów przesuniętym jak na rysunku.

$A \equiv B * 1$ przesunięcie w lewo o 1

$A \equiv B * 2$ przesunięcie w lewo o 2



Dla uproszczenia przyjęto na rys. słowo 5-cio bitowe.

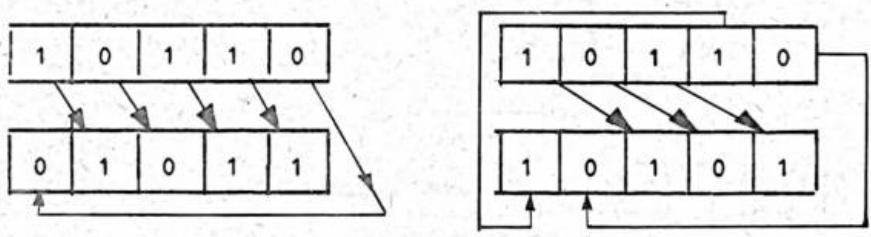
Przesunięcie cykliczne w prawo

zapis $\boxed{A \equiv B * (-n)}$ n - liczba całkowita dodatnia

Wynikiem przesunięcia słowa B w prawo jest nowe słowo bulowskie o układzie bitów przesuniętym jak na rys.:

$A \equiv B * (-1)$ przesunięcie w prawo o 1

$A \equiv B * (-2)$ przesunięcie w prawo o 2



Czytelnik zapewne domyśla się, że po przesunięciu w lewo lub w prawo o n bitów nowopowstałe słowo ma po prostu inną wartość w przypadku, gdy rozważamy zapis liczby.

Reasumując formuły bulowskie:

A ≡ B
A ≡ B + 0
A ≡ B x 0
A ≡ B* n
A ≡ B*(-n)

W kodzie M-2 znak ≡ zastąpiony jest znakiem [

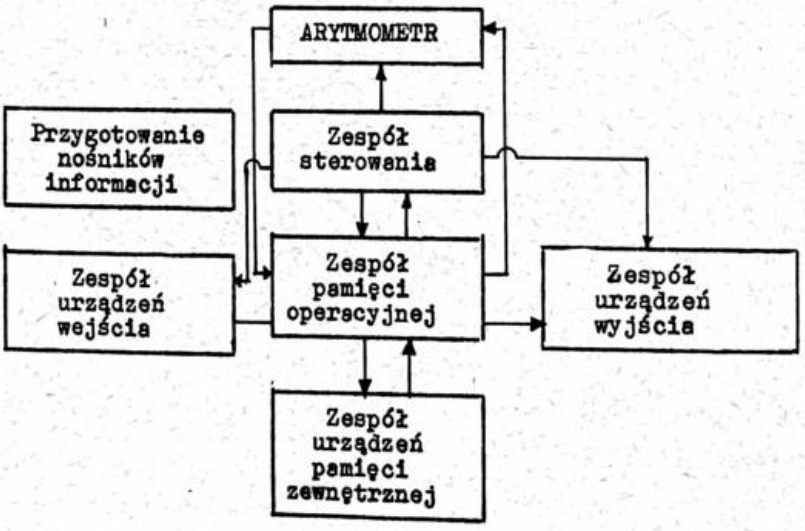
w tej postaci używane są w programach napisanych w języku SAKO i znajdu zastosowanie w zagadnieniach analitycznych jak badanie funkcji, zbiorów l czbowych, zbieżności szeregów itd.

W przykładzie Nr 13 pokazano najprostsze ich zastosowanie.

7. KOMPUTER ZAM-41

7.1. Opis ogólny

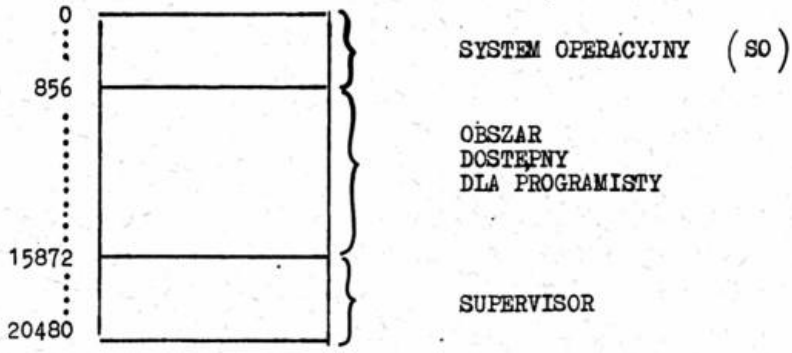
Schemat ideowy każdego komputera jest zamieszczony poniżej na rysunku. Wi kość pamięci operacyjnej PAO i bębnowej mierzy się liczbą słów krótkic Jednostką jest 1 kilo = 1K = 1024 słowa.



OGÓLNY SCHEMAT KOMPUTERA

Maszyna ZAM-41 jest komputerem II generacji produkcji polskiej. Posiada pamięć operacyjną /PAO/ ferrytową montowaną w blokach po 4K słów i maksymalnie posiada 20 K słów. Słowo posiada 24 bity.

Poniżej przedstawiono podział pamięci operacyjnej, z którego wynika,



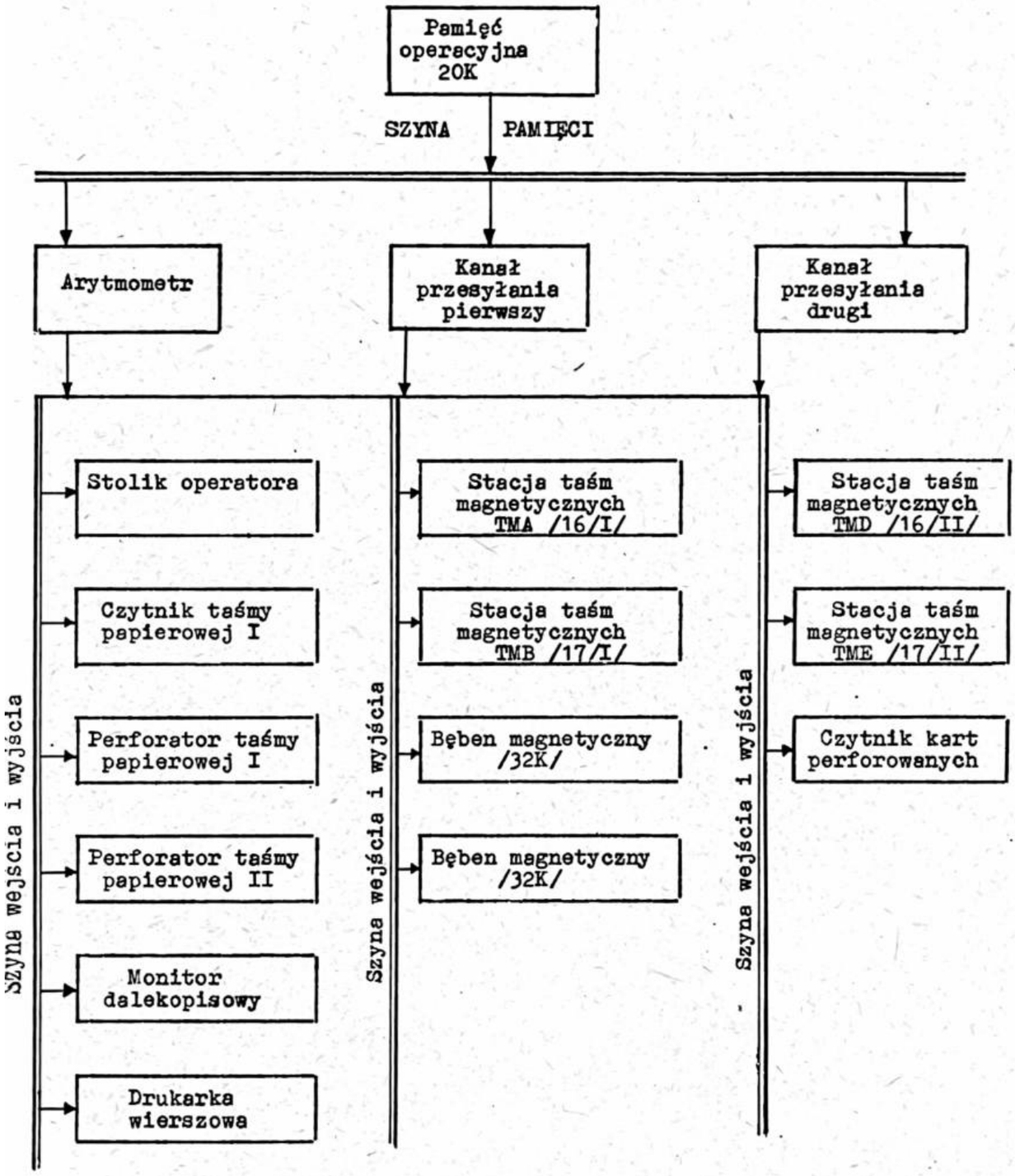
że programista dysponuje ~ 15000 słów, gdyż do SO programista nie ma dostępu.

Maszyna ZAM-41 posiada dwa rodzaje pamięci zewnętrznej /pomocniczej/: pamięć bębnową i taśmową.

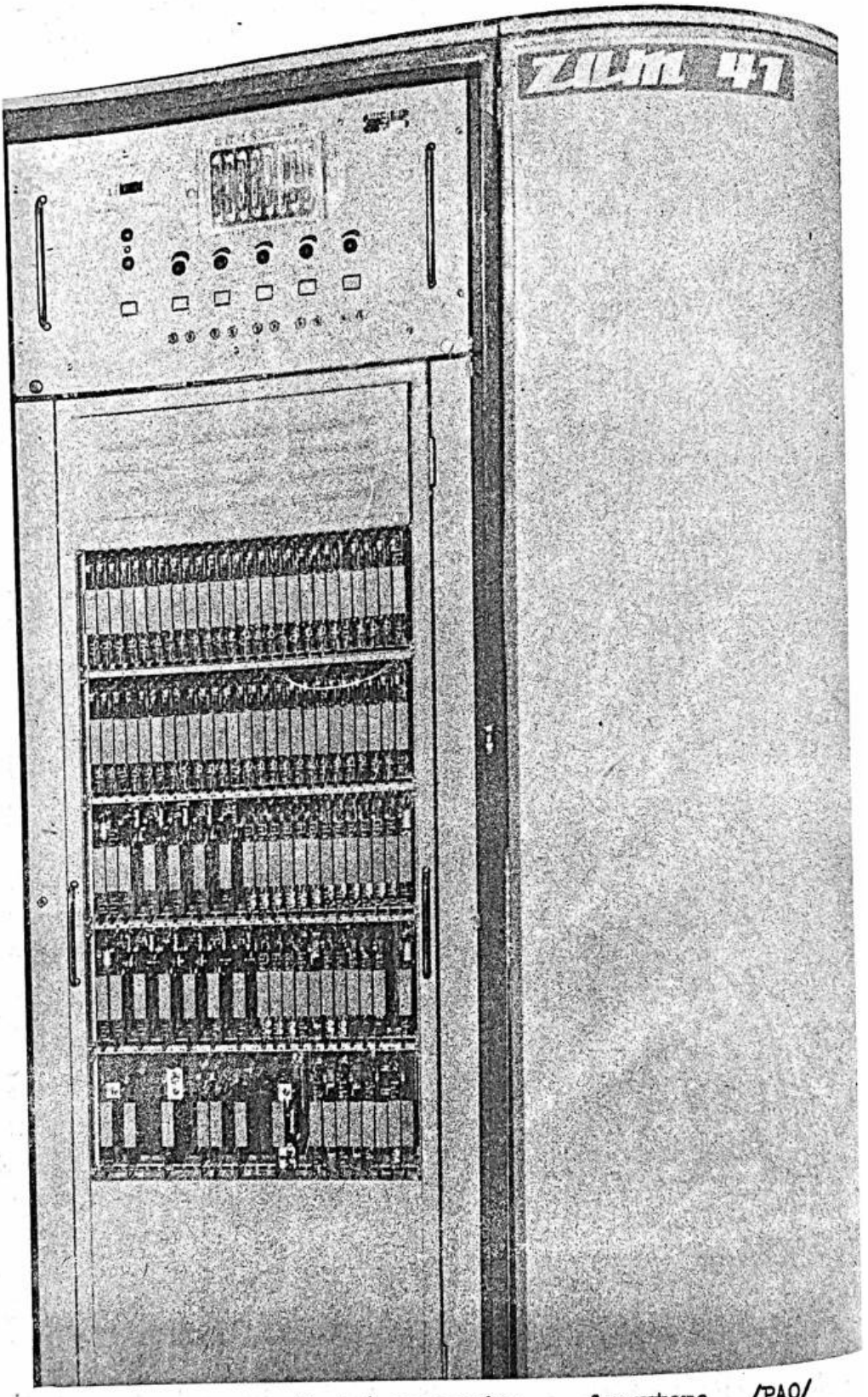
Maksymalnie dostępna dla programisty pamięć bębnowa wynosi ~42000 słów przy zestawie dwóch bębnów (32768 słów/bęben; szybkość przesyłania 6000 słów/sek.).

Programista może korzystać z 5-ciu stacji taśm magnetycznych. Maksymalna długość taśmy magnetycznej wynosi ~ 750 m, co odpowiada pojemności ~ 10 000 000 znaków. Na jednej stacji taśm magnetycznych jest zawsze założona taśma z wszystkimi translatorami języków /"dusza" maszyny - software - bez niego maszyna nie przedstawia żadnej wartości dla użytkownika/.

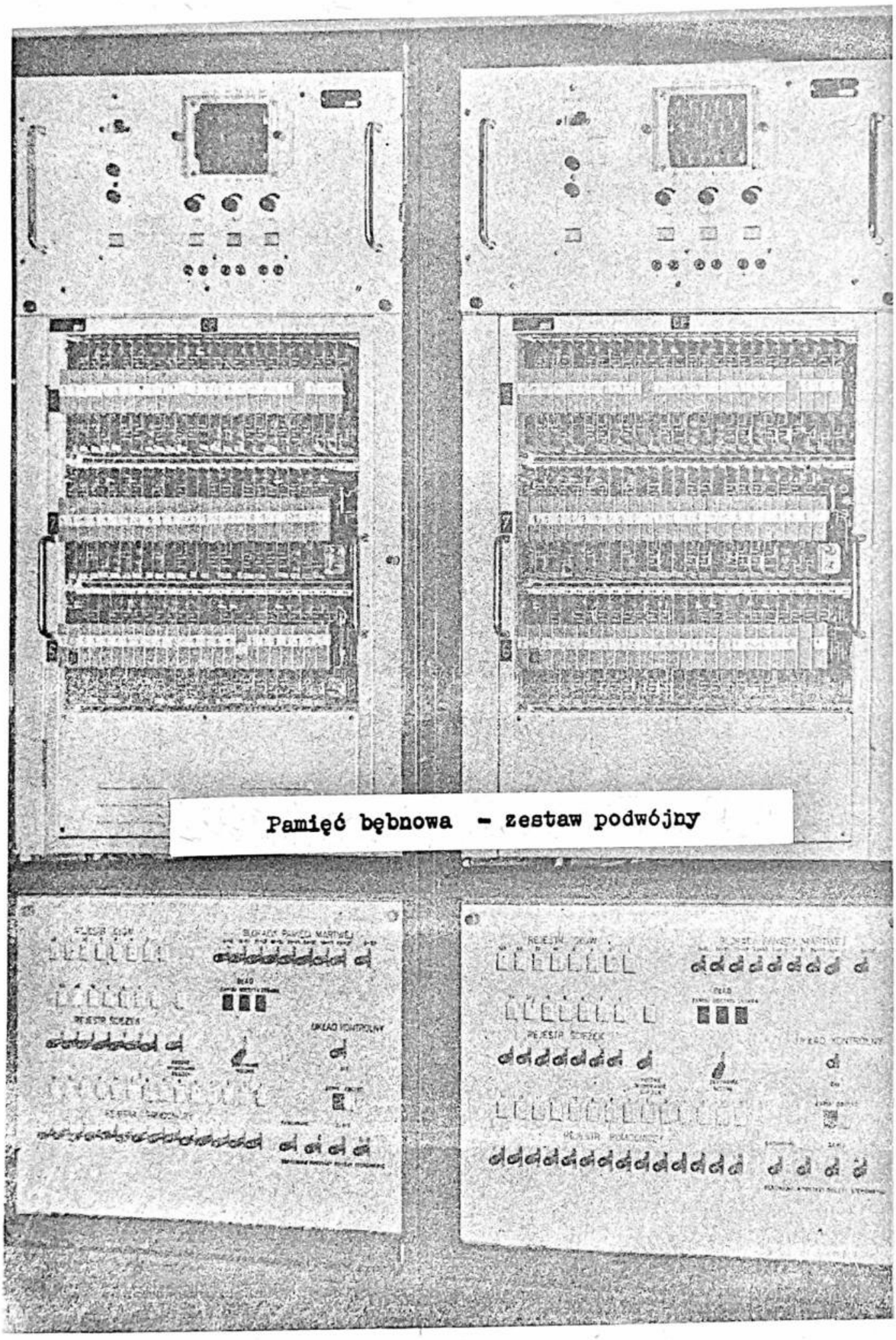
Ogólny widok maszyny ZAM-41 /z prawej strony widoczny monitor dalekopisowy/



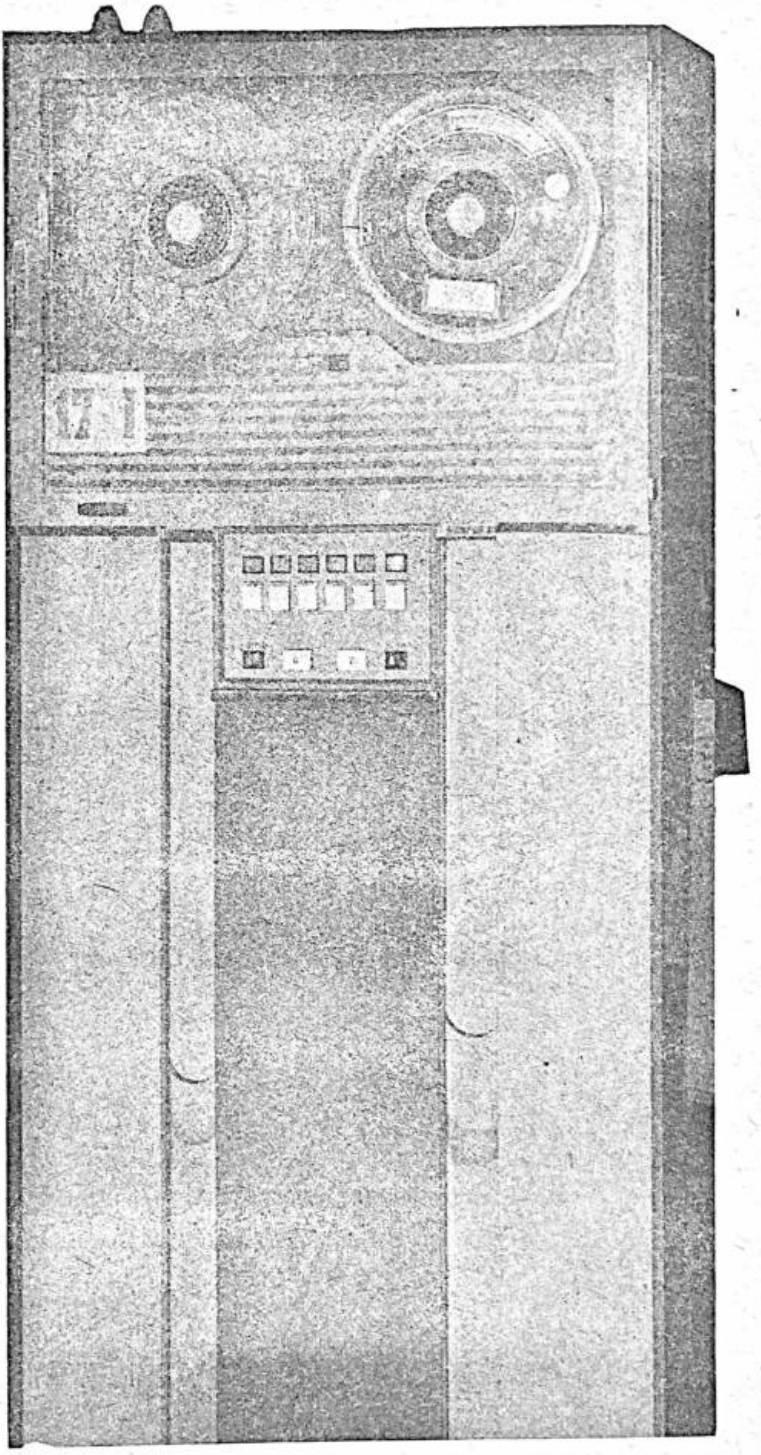
SCHEMAT KOMPUTERA ZAM-41



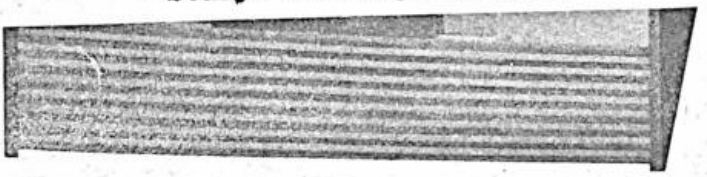
Pamięć operacyjna - ferrytowa /PAO/

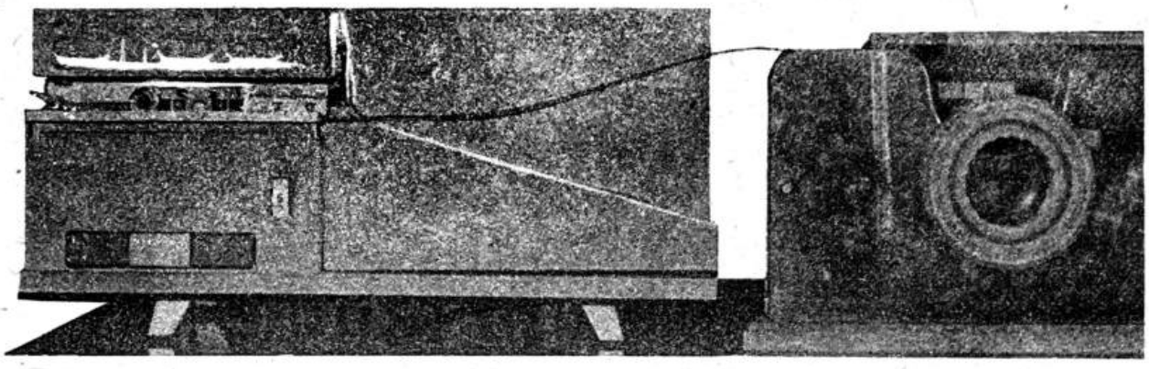


Pamięć bębnowa - zestaw podwójny

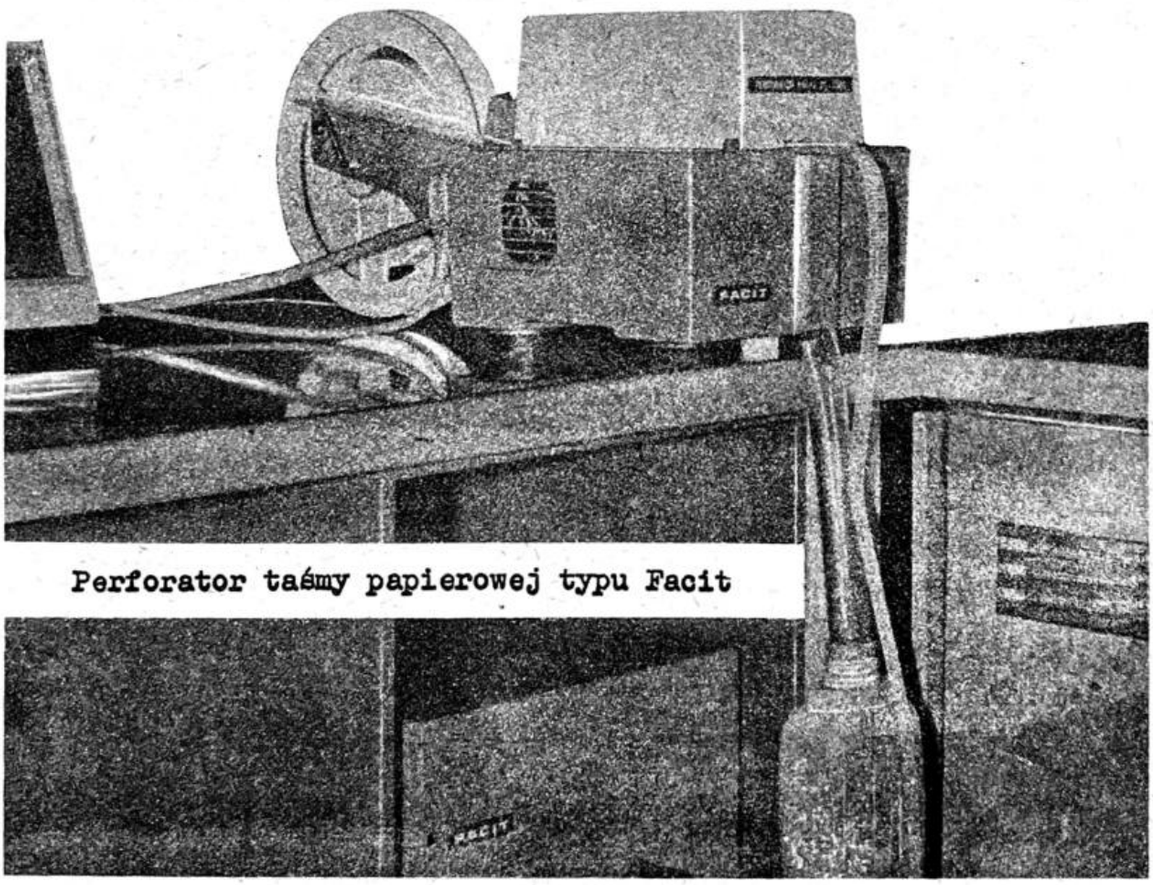


Stacja taśm magnetycznych





Czytnik taśmy perforowanej /typ CT 1001/



Perforator taśmy papierowej typu Facit

Oznaczenia stacji taśm magnetycznych:

Oznaczenie techniczne	Numer symboliczny urządzenia używany przez programistę		Oznaczenie używane w czołówkach programów
	czytanie z TM	pisanie na TM	
16/I	8	9	TMA
17/I	10	11	TMB
18/I	12	13	TMC
16/II	14	15	TMD
17/II	16	17	TME
18/II	18	19	TMF

Maszyna posiada następujące urządzenia WEJŚCIOWE:

1/ dwa czytniki taśmy perforowanej pięcio- i ośmio- kanałowej (1000 znaków /sek),

2/ czytnik kart perforowanych (400 kart/min) oraz następujące urządzenia WYJŚCIOWE:

1/ dwa perforatory taśmy papierowej (150 znaków/sek),

2/ drukarka wierszowa (600 wierszy/min; szerokość wiersza 120 znaków; repertuar możliwych znaków = 64 patrz:tablica str. 69).

Urządzenia te posiadają tzw. bufory, po zapełnieniu których informacje dopiero wyprowadzane są na zewnątrz.

Lista rozkazów maszyny składa się z ~100 rozkazów, przy pomocy których został opracowany autokod SAKO. Maszyna pracuje w zmiennym przecinku,co pozwala uzyskiwać duże dokładności obliczeń.

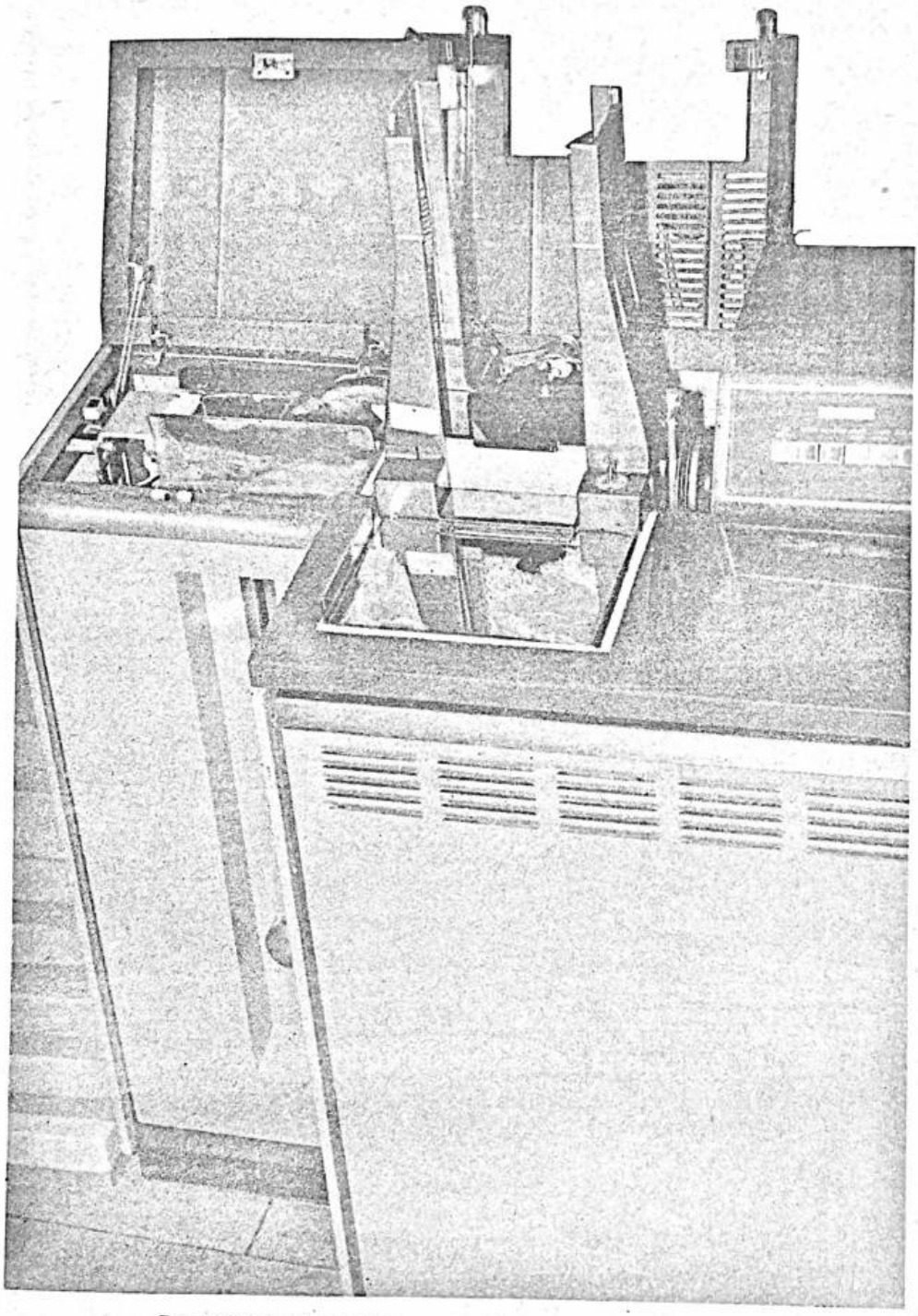
Średnia szybkość obliczeń numerycznych wynosi ~40 000 operacji/sek.natomiast średnia szybkość przetwarzania danych ~50 000/sek.

Poszczególne urządzenia zewnętrzne maszyny mogą współpracować z PAO bezpośrednio, bądź za pośrednictwem tzw. kanału przesyłania. Na str. 35 zamieszczono schemat komputera ZAM-41.

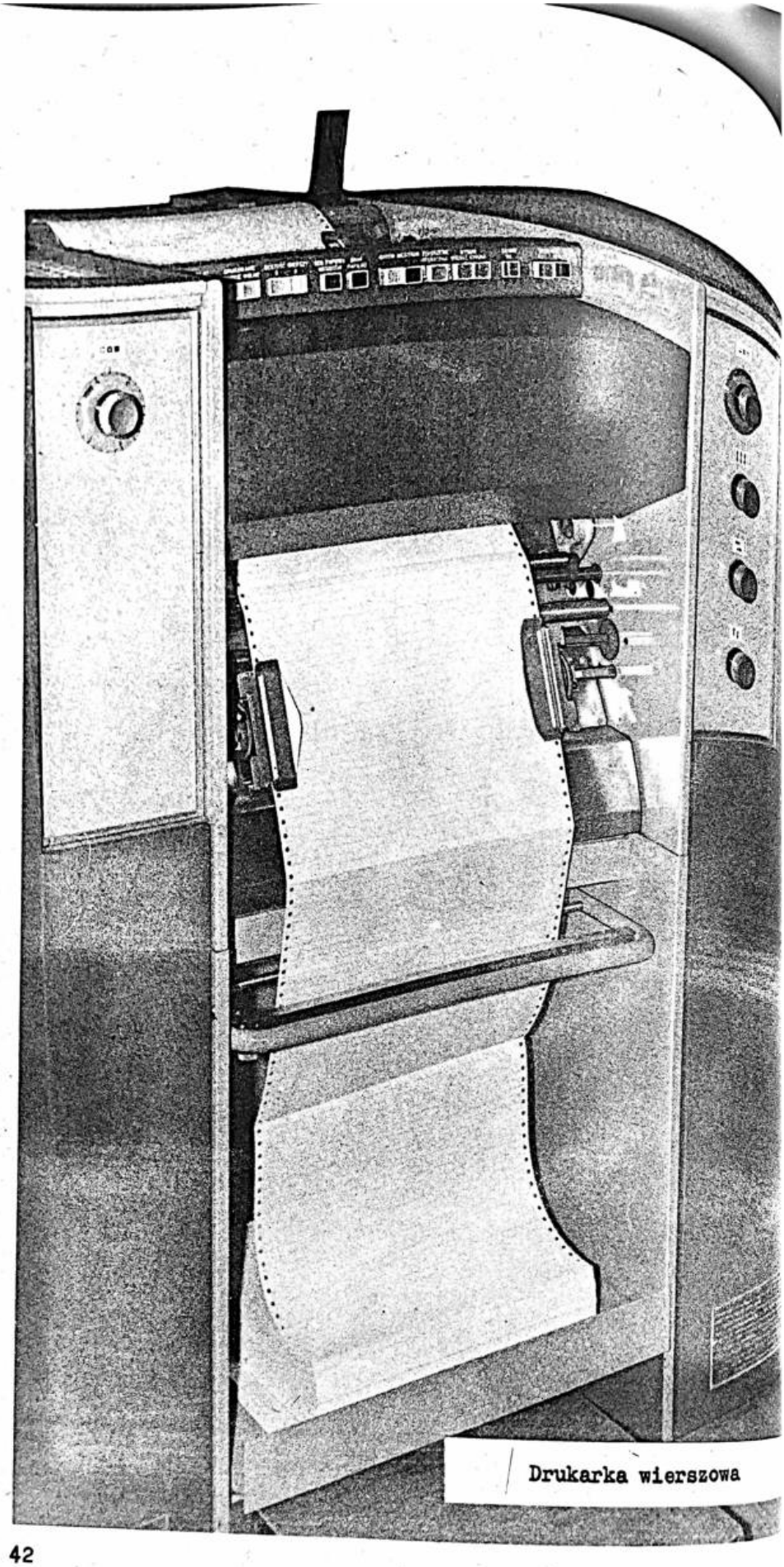
System operacyjny (SO) służy do sterowania przy wykonywaniu programu. W procesie wprowadzania i wykonywania programów można wyróżnić następujące czynności:

1/ wprowadzenie /translacja/ programu użytkowego

2/ wykonywanie programu użytkowego



Czytnik kart perforowanych



Drukarka wierszowa

3/ informowanie o przyczynie zakończenia programu
4/ czynności organizacyjne /typu "post mortem",opisu programu itp./.

Każdą z tych czynności steruje odpowiedni program /w szczególności wykonywaniem programu użytkowego steruje on sam/, kolejnością przekazywania sterowania tym programem steruje SO. System operacyjny ułatwia pracę programu, gdyż obsługuje wszystkie urządzenia wejścia i wyjścia /WE - WY/.

Do miejsc PAO, gdzie znajduje się System Operacyjny, programista nie ma dostępu.

7.2. Czołówki programów

Każdy program w języku SAKO musi być poprzedzony dwoma czołówkami: czołówką systemową /w Języku Operacyjnym Maszyny JOM/ i sakowską. Na podstawie czołówek SO steruje odpowiednio pracą maszyny.

7.2.1. Czołówki systemowe

Czołówka systemowa zawiera:

1/ opis problemu dla maszyny
2/ opis wszystkich wejść i wyjść dla translatora i programu
3/ tytuł problemu /dla identyfikacji zadań wykonanych na maszynie przez operatora/.

Każde zdanie czołówki kończy się znakiem ; lub Ω z wyjątkiem ostatniego zdania, które musi kończyć znak kropki.

Ad. 1. W problemie podajemy język, w którym napisany jest program, deklarujemy krotność wykonania programu i ewentualnie deklarujemy dodatkowe czynności opisowe dotyczące organizacji programu i jego poprawności działania, które umożliwiają szybkie wychwycenie błędów merytorycznych w programie podczas jego uruchamiania.

Zdanie w języku JOM opisujące problem może przykładowo być przedstawione następująco:

P :	SAKO,	PROGRAM ,	ZAKO;
oznacza problem	wywołanie translatora	wykonanie programu	czynności organizacyjne /"post mortem", przyczyny zakończeń/

W problemie może wystąpić maksymalnie 6 zadań. Jeżeli program jest już uruchomiony, translatora zako można nie wywoływać, a słowo program może wystą-

pić wtedy 5 razy, co oznacza możliwość pięciokrotnego wykonania progra
Jest to aktualne, gdy program kończy się zdaniem STOP NASTĘPNY I IDŹ D
ROZDZIAŁU: nr pierwszego rozdziału programu /przy programie wielo
rozdziałowym/ lub zdaniem STOP α, gdzie α jest etykietą pierwszego rozk
zu programu /przy programie jednorozdziałowym/. Można uniknąć deklarowania
krotności wykonania programu, jeżeli program kończymy zdaniem IDŹ DO ROZ
DZIAŁU: nr pierwszego rozdziału programu /przy programie wielorozdz
.działowym/ lub SKOCZ DO α, gdzie jest etykietą pierwszego rozkazu pro
gramu.

Wystarczy wtedy zadeklarować raz w problemie słowo program. Następnie de
klarujemy wszystkie WE-WY zarówno dla translatora, jak i programu /wejścia
danych/. Jeżeli nie zadeklarujemy urządzeń WE-WY, maszyna będzie pracować
w tzw. standardzie, czyli wejście zarówno dla translatora, jak i program
będzie z I czytnika taśmy perforowanej, a wyjście z I perforatora taśmy per
forowanej, urządzenia te mają numery symboliczne zerowe i związane są z
kodem M2.

W problemach inżynieryjno-technicznych /numerycznych/ wyniki z reguły wypro
wadza się na drukarkę wierszową i wtedy np.możemy numer zero nadać drukarce
wierszowej deklarując ją jako wyjście w następujący sposób:

wyp:	0 =	pdw	(dx,	ddx1);
↓	↓	↓	↓	↓
oznacza wyjście programu	numer symboliczny urządzenia	program drukarki wierszowej	nazwa drukarki wierszowej	nazwa dekodera drukarki wierszowej

Wyprowadzenie wyników na perforator taśmy odbywa się przykładowo przez następujący zapis:

wyp : 0 =	pptp	(ptp 1	dfep);
	↓	↓	↓
	program pisania taśmy perforowanej	perforator taśmy perforowanej I	dekoder Ferrantiego dla pisania

lub na przykład:

wyp : 1 =	pptp	(ptp2	dm2p) ;
		↓	↓
		perforator taśmy perforowanej II	dekoder dla kodu M2 /perforowania/

Wyjście translatora można opisać następująco:

wyt : 0 = pdw (dx, ddx1);

lub

wyt : 0 = pptp (ptp1, dfep) ;

lub

wyt : 0 = pptp (ptp1, dm2p) ;

oznacza
wyjście
translatora

Jeżeli wyp jest identyczne z wyt wystarczy pisać wy.

Przykładowo poniżej załączono deklaracje możliwych wejść programu /danych/.

wep : 0 =	pctp	(ctp1	dfec) ;
oznacza wejście programu	program czytania taśmy perforowanej	czytnik taśmy perforowanej I	dekoder kodu Ferrantiego dla czytania danych

wep : 0 : pctp (ctp 1, dfe6) ;

dekoder dfe6 może być również używany do danych liczbowych przy perforowaniu w kodzie Ferrantiego, lecz należy go koniecznie stosować w wypadku perforowania programu w kodzie Ferrantiego. Napiszemy wtedy

wet : 0 = pctp (ctp1, dfe6) ;

Nie omawiamy tu wejścia danych z czytnika kart, gdyż w problemach numerycznych na tej maszynie nie ma on zastosowania; gdyby jednak Czytelnik chciał się zapoznać ze sposobem jego użycia, może skorzystać z wydawnictwa IMM "ZAM-41 - Oprogramowanie część I" na stronach: 16, 119 i 138.

Poniżej zamieszczamy przykład gdzie zadeklarowano 2 rodzaje wejść danych i 2 rodzaje wyjść wyników.

wep : 0 = pctp (ctp1, dfec) ;
wep : 1 = pctp (ctp2, dm2c) ;
wyp : 0 = pdw (dx, ddx1) ;
wyp : 1 = pptp (ptp1, dm2p);

Wszystkie zadeklarowane wejścia i wyjścia oznaczone numerem symbolicznym większym niż 0, należy dołączać do programu, tzn. dołączać ich podprogramy. Służy do tego tzw. funkcja SAKO DOL (X, Y) /patrz str.79 /.

Z = DOL (NR symboliczny WEJŚCIA, * A)

adres początku pola, do którego podprogram będzie zapisany.

Wartością funkcji Z jest pierwszy wolny adres za dołączonym wejściem.

Dla wyjścia:

Z = DOL (NR SYMB. WYJŚCIA + 16, * A)

W naszym przykładzie dołączenie przedstawiałoby się następująco:

CAŁKOWITE: * A, Z, K
BLOK (409): A
Z = DOL (1, * A)
K = DOL (17, Z) lub Z = DOL (17, Z)

Przy dołączaniu podprogramów WE-WY należy znać rozmiary bloku /pola/, jakie należy zarezerwować. Poniżej podano liczbę słów, jakie zajmują podprogramy niektórych urządzeń i związane z nimi dekodery.

Nazwa	Skrót	Liczba słów
Podprogram czytania taśmy perforowanej	PCTP	128
Podprogram drukarki wierszowej	PDW	148
Podprogram perforowania taśmy	PPTP	148
Dekoder drukarki wierszowej	DDX1 lub DDU1	79
Dekoder M2 /pisanie/	DM2P	95
Dekoder M2 /czytanie/	DM2C	54
Dekoder Ferrantiego /czytanie danych/	DFEC	54
Dekoder Ferrantiego /pisanie danych/	DFEP	100
Dekoder Ferrantiego /pisanie programu i danych/	D6FE	100
Dekoder Ferrantiego /czytanie programu i danych/	DFE6	54
Dekoder jednostkowy	DJED	10

Jeżeli mamy np. dwa różnego rodzaju wejścia lub wyjścia, stosując funkcję SAKO SEL (X), gdzie X jest numerem symbolicznym urządzenia, wybieramy urządzenie, na które będą wyprowadzone wyniki lub czytane dane za pomocą rozkazów DRUKUJ I CZYTAJ występujące po zadeklarowaniu tej funkcji.

Urządzenia te są aktualne aż do nowego wprowadzenia funkcji SEL (X). Można również odłączyć odpowiednie wejście lub wyjście. Funkcję tę pełni funkcja ODL (X), która jednocześnie wypisuje resztę informacji zawartych w buforach /tzw. "czyszczenie buforów"/.

7.2.2. Czołówki języka SAKO

Umieszcza się je przed programem napisanym w języku SAKO w celu uzyskiwania dodatkowych informacji, zarówno o rozmieszczeniu, jak i o działaniu programu użytkowego. Przy ich pomocy można również tworzyć programy binarne na taśmie magnetycznej /przetłumaczone na język wewnętrzny maszyny w postaci binarnej/.

Zdanie czołówki składa się ze znaków identyfikujących zdanie i ewentualnych parametrów zdania. W zwrotach identyfikowane są 4 pierwsze litery. Parametry zdań czołówek składają się z liter i cyfr. Zdania czołówki SAKO dzielimy na:

1/ związane z translacją
2/ związane z wykonaniem
3/ kończące czołówkę.

Zdania związane z translacją są następujące:

OPISZ PROGRAM

żądanie wypisania na zadeklarowanym urządzeniu wyjścia translatora /czołówka systemowa/ informacji o programie takich, jak długości rozdziałów z podziałem na program i zmienne, długość całego programu, liczba wolnych miejsc na bębnie po zapisie programu.

CHOWAJ SŁOWNIKI

Żądanie przechowywania wraz z programem wynikowym słowników etykiet rozdziałów. Dzięki temu zdaniu jest możliwe wyprowadzenie "post mortem".

URUCHAMIANIE : nr rozdziału

np. uruchamianie: 123
uruchamianie

Żądanie aby w czasie tłumaczenia rozdziału określonego przez parametr b tłumaczone zdania opatrzone znakami ? lub (apostrof). Brak parametru ozn cza, że zdanie odnosi się do wszystkich rozdziałów.

PAO : pojemność

Np. pao : 11000
pao : 7000

Oznacza żądanie takiego tłumaczenia rozdziałów programu, aby mogły one w konywać się w pamięci operacyjnej o pojemności określonej przez parame Parametr wskazuje maksymalny adres PAO zwiększony o 1 przeznaczony dla r działu. Brak tego zdania w czołówce spowoduje, że rozdziały będą transl wane na taką pamięć jaka jest podłączona do maszyny.

BĘBEN : pojemność

np. bęben : 35000

Oznacza żądanie, aby podczas translacji wykorzystywany był obszar bębna adresu określonego przez parametr /wyłącznie/.

POST MORTEM : szerokość arkusza

np. port mortem : 65
post mortem

Oznacza żądanie, aby w przypadku wykonywania programu po jego wykonaniu po przerwaniu wskutek błędu wypisywane były informacje o stanie wszystki zmiennych w rozdziale w którym zakończone zostało działanie programu. Jeż li blok zawiera więcej niż 15 elementów, zostają wypisane wartości tyl 15 elementów. Należy zwrócić uwagę, że aby otrzymać "post mortem", nale w czołówce systemowej w problemie zadeklarować zdanie ZAKO, a w czołó sakowskiej napisać zdanie CHOWAJ SŁOWNIKI. Parametr określa szerokość a kusza wydawniczego w postaci liczby znaków w wierszu. Brak parametru je równoważny szerokości 120 znaków.

NAZWA: nazwa

np. nazwa : WĘDLIN
nazwa : PROG 12

Jest to przyporządkowanie tłumaczonemu programowi nazwy określonej przez parametr. Nazwa wystąpi przy realizacji zdania OPISZ PROGRAM i jest bardzo ważna przy zapisywaniu binarnym programu na taśmę magnetyczną (identyfikacja programu w zbiorze programów na taśmie magnetycznej). W przypadku braku nazwy przy realizacji opisz program zostanie przyporządkowana nazwa PROGPR.

Nazwą może być dowolny tekst złożony z co najmniej 6 znaków /jedynie litery i cyfry/.

Zdania związane z wykonaniem są następujące:

LIKWIDUJ STOP

Oznacza żądanie, aby rozkaz STOP α w programie nie powodował zakończenia programu a tylko jego zawieszenie. Zlikwidowanie zawieszenia z monitora przez wyklawiszowanie 0 - S ? spowoduje kontynuację programu poczynając od etykiety o numerze α

KLUCZE : lista numerów kluczy

np. klucz: 23
klucze: 1, 15, 17, 0, 22

Oznacza żądanie ustawienia przed startem programu stanu początkowego rejestru kluczy. Jedynki zostaną ustawione na pozycje określone parametrami.

WYMAGANY BĘBEN : pojemność

np. wymag : 9000

Oznacza żądanie, aby podczas wykonywania programu wynikowego rozdziały umieszczone na bębnie znajdowały się w obszarze bębna o adresach nie mniejszych od określonego w parametrze zdania. W przypadku niemożności zrealizowania tego żądania sygnalizowany jest błąd 210. Brak tego zdania jest równoważny zdaniu WYMAGANY BĘBEN : 0.

NIWELUJ NADMIARY : tak lub nie

np. niwe : nie
niweluj : nie
niweluj nadmiary : tak

Oznacza wybranie jednego ze sposobów reagowania na pojawienie się
lub niedomiaru /liczba mniejsza od 2^{-256}/ w trakcie wykonywania
zmiennoprzecinkowych.

NIE - polega na sygnalizacji błędu wykonania i przerwanie wykonywa
gramu

TAK - polega na zastępowaniu wyników działań zmiennoprzecinkowych

a/ w przypadku niedomiaru na zero zmiennoprzecinkowe + 0.0E40
maszynie ZAM-41 zapisane jest następująco:

zera	zera	cecha - 255
0 – 23	24 – 39	40 – 47

b/ w przypadku nadmiaru dodatniego na maksymalną liczbę zmieszc
cinkową

c/ w przypadku nadmiaru ujemnego na minimalną liczbę zmiennoprz
kową.

Parametr TAK można zastąpić jedynką
NIE można zastąpić zerem

Brak parametru oznacza TAK

IGNORUJ BŁĘDY

Zezwolenie na wykonywanie programu, w którym występowały błędy for
Nie dotyczy to błędów kończących translację. Zdania kończące czołówk
następujące:

TRANSLACJA

Oznacza zdanie kończące czołówkę SAKO, które powoduje, że rozdziały
gramu SAKO zostaną przetłumaczone i rozmieszczone na bębnie w postaci
gotowanej do wykonania.

ZAPISZ : nazwa TM, nr szpuli, lokalizacja na TM

np. zapisz : tma, 1, poc
zapisz : tmb, 1, kon
zapisz : tmd, 1, kielba

Oznacza zdanie kończące czołówkę SAKO i powodujące, że następujący za nim program SAKO zostanie umieszczony na taśmie magnetycznej wyznaczonej przez 2 pierwsze parametry. Trzeci parametr określa lokalizację programu na TM przy czym:

POC - oznacza od początku szpuli

KON - oznacza za ostatnim programem, który znajduje się na tej szpuli,

w innym wypadku oznacza żądanie wpisania programu na miejsce programu o wskazanej nazwie.

W tym przypadku usunięty zostanie z taśmy magnetycznej program o ustalonej nazwie oraz wszystkie programy, które były umieszczone za nim.

Program zapisany na taśmie magnetycznej jest gotowy do wykonania. Wraz z programem przechowywane są niektóre zdania czołówki SAKO a mianowicie:

wymagany bęben

stan kluczy

wskaźnik niwelowania nadmiarów.

POBIERZ : nazwa TM, nr szpuli, nazwa programu, lista rozdziałów sprowadzanych na bęben

np. pobierz : tma, 1, WĘDLIN, R0, R1, S0, S1

Oznacza żądanie przygotowania do wykonania programu określonego przez trzeci parametr umieszczony na taśmie magnetycznej, określonej przez dwa pierwsze parametry. Następne parametry określają rozdziały i ewentualnie słowniki etykiet rozdziałów, które powinny być omieszczone na bębnie.

Numer rozdziału poprzedzony jest literą R, a słownika odpowiedniego rozdziału S.

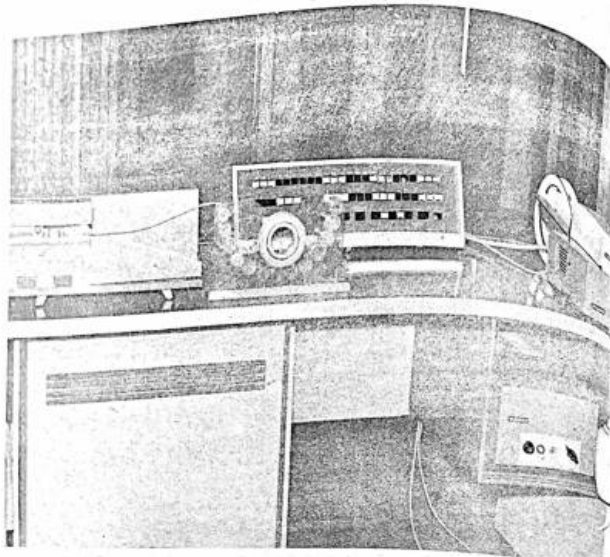
K O N I E C

Oznacza zdanie kończące czołówkę oraz działanie translatora SAKO. Stosuje się go w celu dokonania zmian parametrów czołówki programu SAKO przetłumaczonego wcześniej, znajdującego się na bębnie.

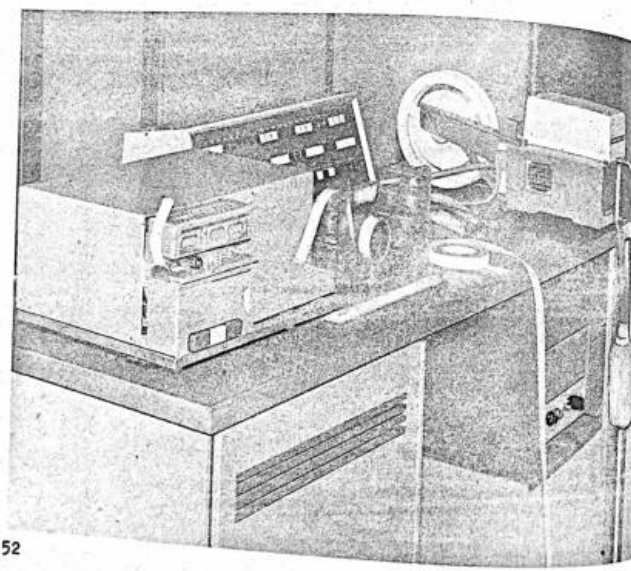
7.3. Instrukcja czynności operatorskich

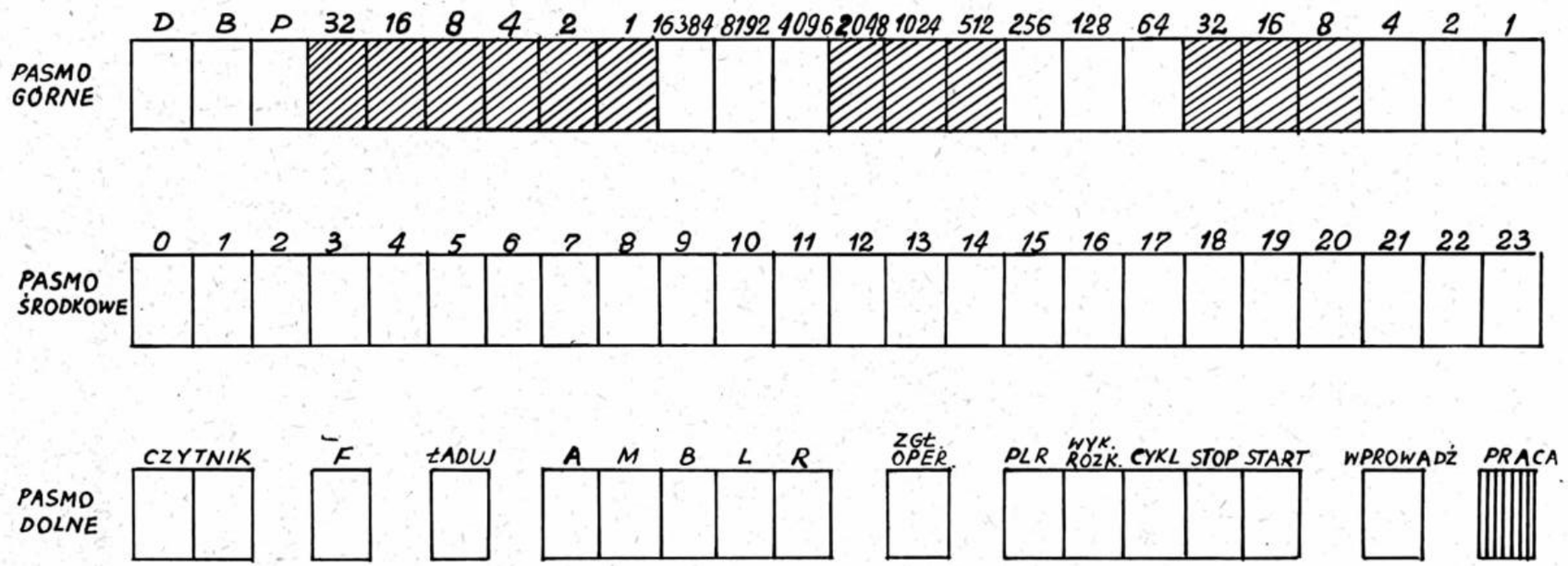
7.3.1. Podłączenie stacji taśmy magnetycznej.

Przyciski na stacji T M /str.38 /



Stolik operatora maszyny ZAM-41





PULPIT OPERATORA EMC ZAM -4-1

WS	←	I	→	⇉	M
Gotów	0	Z	MOC		
lampka	wyłą-czanie	zasi-lanie	lampka		

W celu włączenia stacji TM należy:

1/ wcisnąć przycisk Z i WS - powoduje to podłączenie stacji do zasila

2/ wcisnąć klawisz stop [I] i czekać do chwili, gdy zapali się lampka G TÓW

3/ przewinąć taśmę najpierw w lewo [←], potem w prawo [→] w celu spr dzenia, czy taśma została poprawnie założona;
przewidnięcie w prawo powoduje odwinięcie taśmy do nalepki początku my (przyklejona folia aluminiowa) stanowi początek taśmy

4/ wcisnąć klawisz STOP [I] po zakończonym przewijaniu taśmy

5/ wcisnąć klawisz [M] w celu podłączenia stacji do maszyny.

O gotowości stacji TM do współpracy z komputerem sygnalizuje lampka GOT

Przed rozpoczęciem współpracy ze stacją TM należy jeszcze sprawdzić poł żenie przełacznika ZAPIS - BLOKADA ZAPISU, znajdującego się w tylnej cz ci stacji.

Jeśli na taśmę zapisujemy program lub zbiór danych,przełącznik należy us wić w położeniu ZAPIS;

jeśli natomiast odczytujemy informacje z TM, przełącznik należy ustawi położeniu BLOKADA ZAPISU.

7.3.2. Przygotowanie maszyny do wczytania programu.

Wczytanie programu do maszyny następuje przy wciśniętych klawiszach 19, 21 oraz przy wciśnięciu klawisza ZGŁOSZENIE OPERATORA /ZO/ /str 53 pa dolne/.

Należy podkreślić, że bez pomocy klawisza ZGŁOSZENIE OPERATORA komputer przyjmie żadnej informacji.

Na pulpicie stolika operatora klawisz

19	oznacza	ciąg dalszy
20	"	czytnik
21	"	perforator
22	"	problem
23	"	program

Dla chwilowego wyłączenia czytnika czy perforatora, należy wycisnąć odpowiadające im przyciski 20 lub 21 i ZO.

Manipulacja tymi przyciskami pozwala na chwilową przerwę w pracy danego urządzenia.

Po zakończeniu liczenia jednego programu, przed wczytaniem następnego należy "wyrzucić" problem tzn. pierwszy program.

Dokonujemy tego przez:

- wyciśnięcie klawisza 19
- wciśnięcie klawisza 22 i ZGŁOSZENIA OPERATORA

Jeśli zaś chcemy poznać zawartość pamięci lub bębna po zakończeniu liczenia, czy przy uruchamianiu programu - "wyrzucamy" program, poprzez:

- wyciśnięcie klawisza 19
- wciśnięcie klawisza 23 i ZGŁOSZENIA OPERATORA

Na drukarce zostaje wypisany "POST MORTEM". Aby móc wczytać kolejny program, klawisze muszą powrócić do pierwotnego położenia.

7.3.3. Sprowadzanie Systemu Operacyjnego do PAO i na bęben

Ściąganie systemu operacyjnego z TM do PAO i na bęben może odbyć się jednym z niżej podanych sposobów:

1/ za pomocą "ściągaczki systemu" - tzn. specjalnego programu wyperforowanego na taśmie; wymaga on wielu manipulacji na klawiszach Stolika Operatora i współpracy z PAO przez monitor;

2/ przez regenerację Systemu za pomocą klawisza 16 i ZGŁOSZENIA OPERATORA; tę metodę można zastosować jedynie wtedy, gdy "pali się" lampka PRACA maszyny;

3/ w przypadku, gdy jest STOP maszyny, tzn. nastąpiło chwilowe, nieprzewidziane przerwanie pracy komputera, system wznawiamy przez:

- wyzerowanie wszystkich rejestrów /wcisnąć klawisz A, M, B, L, R i ŁADUJ/,
- przesłanie liczby 15 /wcisnąć klawisze 20, 21, 22, 23/ do licznika rozkazów,
- wciśnięcie klawisza START.

7.3.4. Obsługa drukarki wierszowej

Jeżeli w czołówce programu zadeklarowana jest drukarka wierszowa jako urządzenie wyjściowe, należy przygotować ją do współpracy z częścią centralną maszyny cyfrowej, gdyż w przeciwnym wypadku zostanie samoczynnie wstrzyma-

praca komputera; na stoliku operatora pojawi się "alarm", tzn. że
"wykryła" brak określonego urządzenia i "domaga się" podłączenia go.

Drukarkę, podobnie jak wszystkie urządzenia we - wy, uruchamiamy
wciśnięcie, w ustalonej kolejności, odpowiednich przycisków znajdu
się w przedniej części urządzania /patrz zdjęcie na str.42 /.

Przyciski drukarki są najczęściej różnokolorowe i oświetlone od spo
znacznie ułatwia jej obsługę.

Podstawową czynnością jest podłączenie urządzenia do zasilania;pojaw
NIEGOTÓW zasygnalizowany zapaleniem się lampki. Stan ten oznacza, że
karka znalazła się w zestawie urządzeń współpracujących z Systemem O
cyjnym, ale nie jest przygotowana do wydruku wyników czy żądanych t

Przygotowanie drukarki do współpracy z EMC uzyskać można przez

- ustawienie początku strony
- wciśnięcie klawisza GOTÓW
- wciśnięcie ZGŁOSZENIA OPERATORA

zapala się lampka PRACA oznajmiając ostateczne przygotowanie drukarki
szowej do pracy. Wciśnięcie lub wyciśnięcie klawisza ZGŁOSZENIE OPE
decyduje o pracy drukarki lub jej chwilowym przerwaniu; wyciśnięcie
PRACĘ, tzn. odłącza drukarkę od części centralnej komputera.

Tymczasowe przerwanie pracy drukarki niezbędne jest przy takich czyn
ciach operatorskich, jak np. zmiana papieru.

7.3.5. Czynności operatorskie przy monitorze

Monitor jest urządzeniem, które umożliwia komunikację człowiek kompu
odwrotnie. Nawiązanie kontaktu z maszyną możliwe jest przez wzbudzeni
nitora za pomocą przycisku ZGŁOSZENIA WYŁĄCZENIA (sygnalizacja świetl
formuje o możliwości przyjęcia podanej instrukcji). Przez monitor moż

- podać maszynie dalsze instrukcje związane z aktualnie liczonym
  gramem
- zatrzymać pracę maszyny przez podanie ustalonej formuły

  O - Z * (O → zero) tzw. zawieszenie maszyny

  lub ją wznowić

  O - S * tzw. odwieszenie maszyny

- dowiedzieć się o przyczynę "alarmu" na stoliku operatora np.:

  O - CO * (O → zero)

  monitor : 0 alarm CP-7.

Podawane polecenia muszą mieć ustaloną strukturę, gdyż w przeciwnym razie nie będą przyjęte i monitor odpisze NIE ROZUMIEM.

Monitor może przekazywać operatorowi polecenia części centralnej komputera, czuwając w ten sposób nad prawidłowym przebiegiem całości programu. Są to polecenia typu, np.:

: ÓDWIEŚ I PODŁÓŻ TAŚMĘ, : - ROZLICZENIE MATERIAŁOWE
: ZAŁÓŻ TAŚMĘ - KOSZTY - NA STACJE 17 - I

zaprogramowane przez programistę.

W przypadku awarii monitora praca maszyny jest niemożliwa.

Poniżej zamieszczono 4 programy w języku RJES bardzo przydatne zarówno programistom jak i operatorom EMC.

PROGRAM SŁUŻĄCY DO REPRODUCJI TAŚMY PERFOROWANEJ (JĘZYK PJES - PODSTAWOW JĘZYK SYMBOLICZNY)

```
wet:0=pctp(ctp1,dm2c);
wyt:0=pdw(dw,ddu1);
wep:0=pctp(ctp1,djed);
wyp:0=pptp(ptp1,djed);
p:2,0.
vpb[]
p31
p31 16
sko q-2
vkb
vst[]
```

PROGRAM SŁUŻĄCY DO REPRODUKCJI TAŚMY PERFOROWANEJ W DWÓCH EGZEMPLARZACH (JĘZYK PJES)

```
p:2,0;
wet:0=pctp(ctp1,dm2c);
wyt:0=pdw(dx,ddu1);
wep:0=pctp(ctp1,djed);
wyp:0=pptp(ptp1,djed);
wyp:1=pptp(ptp2,djed);
t:reprodukcja tasmy papierowej na 2 perforatory.

vpb[e1]
sko e1
e0=q
q=q+400
e1)ueb e0
pmb 38
sdy 1/11+1/13+1/17+8
p31
p31 16
p31 17
sko q-3
vkb

vst[]
```

PROGRAM SŁUŻĄCY DO WYDRUKU ZAPISU NA TAŚMIE PERFOROWANEJ W KODZIE M-2 NA DRUKARKĘ WIERSZOWĄ (JĘZYK PJES)

```
p:2,0;
wy:0=pdw(dx,ddu1);
tytul:wydruk tasmy 5-kanatowej na dw (kod m-2).
vpb[e4]

sko q+2

e0)+0  :robocza
e4)ueb 60
p31 16
ueb 7
pmb e0
ueb 58
p31 16
umb e0
sob q-4  :8 linii

ueb 55
e3)pmb e0
e1)p31
prb 58
sko e2
p31 16
sko e1
e2)p31 16
umb e0
sob e3

sko e4
vkb
vst[]
```

Uwaga: przed wydrukiem należy ustawić nową stronę na drukarce.

PROGRAM SŁUŻĄCY DO WYDRUKU ZAPISU NA TAŚMIE PERFOROWANEJ W KODZIE FERRANTIEGO NA DRUKARKĘ WIERSZOWĄ (JĘZYK PJEI)

```
p:2,0;
wy:0=pdw(dx,ddu1);
wep:0=pctp(ctp1,dfe6);
tytuł:wydruk taśmy 5-kanałowej na dw (kod ferrantiego).
vpb[e4]

sko q+2

e0)+0  :robocza
e4)ueb 60
p31 16
ueb 7
pmb e0
ueb 58
p31 16
umb e0
sob q-4  :8 linii

ueb 55
e3)pmb e0
e1)p31
prb 58
sko e2
p31 16
sko e1
e2)p31 16
umb e0
sob e3

sko e4
vkb
vst[]
```

Uwaga: przed wydrukiem należ ustawić nową stronę na drukarce.

7.4. Wyprowadzanie taśmy binarnej programu

Taśma binarna programu jest to program przetranslowany na język wewnętrzny maszyny i wyprowadzony w postaci binarnej na taśmie perforowanej. Jest to bardzo korzystna postać programu do wprowadzenia go do maszyny z powodu unikania czasu translecji /który bywa niejednokrotnie długi/. Poniżej zamieszczono dwie czołówki: jedną dla otrzymywania taśmy binarnej programu i drugą do wprowadzenia jej do maszyny. Założono przy tym, że program w języku SAKO i dane są na taśmie perforowanej.

Czołówka do otrzymywania taśmy binarnej:

```
p    : sako, wybi ;
wyt  : 0 = pdw (dx, ddx1);
wyt  : 1 = pptp (ptp1, dm2p);
wyt  : 2 = pptp (ptp1, djed);
wet  : pctp ctp1,(dm2c.)
                 lub dfe6
```

Czołówka do wprowadzenia taśmy binarnej do maszyny i jednokrotnego wykonania programu:

```
p    : webi, program, pjeg (zak);
wy   : 0 = pdw (dx, ddx1);
wyt  : 1 = pptp (ptp1, dm2p);
wet  : 1 = pcpt (ctp1, djed);
wep  : 0 = pctp (ctp1, dfe6).
                       lub dm2c
```

7.5. Charakterystyka pamięci taśmowej

Informacje na taśmie magnetycznej zapisane są w postaci tzw. bloków taśmowych. Bloki te mogą zawierać od 2 do 2048 słów.

Na taśmę magnetyczną można przesłać z PAO bloki informacji /zapis na taśmę/ lub z taśmy magnetycznej można przesłać bloki informacji lub ich części do PAO /odczyt z taśmy/. Bloki taśmowe powstają w czasie pisania na taśmę. Po każdej operacji czytania taśma magnetyczna przesuwa się do następnego bloku taśmowego niezależnie od tego, jaka część bloku została przeczytana do PAO. Bloki taśmowe na taśmie magnetycznej przedzielone są tzw.przerwami międzyblokowymi, które nie zawierają żadnych informacji. Istnieją operacje pozwalające kasować /tzn. przekształcać w przerwę międzyblokową/ standardowy /ok. 8 cm/ odcinek taśmy, posuwać taśmę magnetyczną o jeden blok do przodu lub do tyłu i odwijać /tzn. cofać do początku/ taśmę magnetyczną. Dla ułatwienia opisu operacji dotyczących taśm magnetycznych ustalamy nu -

merację bloków taśmowych danej taśmy kolejnymi liczbami naturalnymi. Operacje czytania i pisania na taśmie magnetycznej jak również ruch w przód i w tył taśmy sterowany jest przez tzw. głowicę pisząco- czytającą. Numer bloku taśmowego, na którego początku ustawiona jest głowica, jest numerem bloku taśmowego, który ma być odczytywany lub zapisywany. Po każdej operacji czytania, pisania i posuwania numer ten zwiększa się o 1, po operacji cofania zmniejsza się o 1, zaś po operacji odwijania zeruje się. Operacja kasowania nie zmienia tego numeru. Wymienione powyżej operacje wchodzą w skład języka PJES i M-SAS i korzystając z nich można współpracować z taśmą magnetyczną. W języku SAKO nie ma niestety rozkazu typu PISZ NA TAŚMIE lub CZYTAJ Z TAŚMY jak dla bębna.

Poniżej załączono łącznie z opisem podprogramy w języku SAKO - MSAS służące do współpracy z taśmą magnetyczną.

Dołączając je do programów w języku SAKO można zapisywać na taśmę lub odczytywać z taśmy bloki liczbowe krótkie lub długie. Można również dowolne teksty zapisywać na taśmę lub odczytywać z taśmy przy pomocy tych podprogramów.

Opis

podprogramu sakowskiego do zapisu i odczytu liczb krótkich i długich z taśmy magnetycznej

Wywołanie podprogramu

(Z) = TM (ZO, BTM, ✳A, LICZ)

Zmienne : Z, ZO, BTM, LICZ są całkowite

Objaśnienia do poszczególnych zmiennych

ZO = 0 - przy zapisie na taśmę magnetyczną
ZO = 1 - przy odczycie z taśmy magnetycznej
BTM - liczba liczb do zapisania na TM lub odczytywania z TM
✳A - nazwa bloku zapisywanego lub odczytywanego z TM
LICZ - numer bloku taśmowego zapisywanego lub odczytywanego z TM /liczony od 1/.

Po zadziałaniu podprogramu wartością funkcji Z jest numer odczytywanego lub zapisywanego bloku taśmowego /liczony od 1/.

Uwagi

1/ numer stacji zadaje się ustawieniem klucza w czołówce sakowskiej lub przez monitor zgodnie z wymaganiami systemu operacyjnego

8 dla 16/I
10 dla 17/I
12 dla 18/I
14 dla 16/II
16 dla 17/II
18 dla 18/II

2) w blokach, które będą zapisywane lub odczytywane z taśmy należy rezerwować o jedno miejsce więcej niż przewiduje programista

3) po założeniu taśmy magnetycznej na stację należy odwinąć ją do początku (technicznie lub programowo

4) w podprogramie używane są przesłania o numerach 30 i 31

5) przyczyna zakończenia 37 - oznacza technicznie złe przesłanie, przyczyna zakończenia 333 - oznacza zły zapis na taśmie magnetycznej.

PODPROGRAM WSPÓŁPRACY Z TAŚMĄ MAGNETYCZNĄ DOŁĄCZANY DO PROGRAMÓW W JĘZYKU SAKO

(zapis i odczyt liczb długich)

```
PODPROGRAM:(Z)=TM(Z0,BTM,TA,LICZ)
CALKOWITE:NRST,Z0,BTM,TA,LICZ,Z
T1)GDY KLUCZ NRST:7,INACZEJ 6
6)POWTORZ OD 1:NRST=B(Z)18
7)JĘZYK SAS:NRST,Z0,BTM,TA,LICZ,Z
SKO B0
A30)STO 333
A1)+1
A0)+0.0E+0
A00)+0
A3)+2
A2)+0
A7)SDY 1/11
A8)SDY 1/10+30
A9)SDY 1/10+31
B0)SDY 1/11+3/20
C3)+0
A20)SDY 1/11+4
W1)+0
B1)+0
A22)+0
SKO B2
B0)UMA A00
UMA NRST
DOA Z0
LAR 6
PZA A2
UMA A20
DOM A2
PMA A21
PMA A22
UMA A7
DOA A2
PZA B1
UMA Z0
SZA P0
UMB A00
C0)UMA A8
DOA A2
PZA C4
UMA BTM
DOA BTM
DOA A3
PZA C4+2
C4)+0
SKO A7
+10
UMA D0
DOA A2
PZA P2
C1)UMB LICZ
DOB LICZ
ODB A1
P2)+0
A21)+0
C5)SDY 30
```

c.d.

```
SDY 3/10+1/15+30
A11)STO 37
UAM A0
UMB BTM
ODB A1
PMB C3
UMB A00
DOZ A7
DOP A
DOP A
UMB C3
SOB 7-6
ODZ A7
SZA A10
UMA W1
ODA A1
SZA A30
DOP W1
UMA A
ODA BTM
ODA BTM
PZA A
SKO C5
A10)UMA A00
A12)PZA W1
SKO D1
P0)UMB A00
UMA A9
DOA A2
PZA P1
UMA BTM
DOA BTM
DOA A3
PZA P1+2
P1)+0
SKO A7
+10
UAM A0
UMB BTM
ODB A1
PMB C3
UMB A00
DOZ A7
DOP A
DOP A
UMB C3
SOB 7-6
PAD A7
UEB 1
B2)SDY 31
SDY 3/10+1/15+31
SKO B1
SOB 7-5
D1)UMA LICZ
PZA Z
JĘZYK SAKO
WROC
```

PODPROGRAM WSPÓŁPRACY Z TAŚMĄ MAGNETYCZNĄ DOŁĄCZANY DO PROGRAMÓW W JĘZYKU SAKO
(zapis i odczyt liczb krótkich)

```
PODPROGRAM:(Z)=TM(Z0,BTM,?A,LICZ)
CALKOWITE:NRST,Z0,BTM,?A,LICZ,Z
?1)GDY KLUCZ NRST:7,INACZEJ 6
6)POWTORZ OD 1:NRST=8(2)18
7)JEZYK SAS:NRST,Z0,BTM,?A,LICZ,Z
SKO B0
A30)STO 333
A1)+1
A0)+0
A3)+2
A2)+0
A7)SDY 1/11
A8)SDY 1/10+30
A9)SDY 1/10+31
D0)SDY 1/11+3/20
C3)+0
A20)SDY 1/11+4
W1)+0
B1)+0
A22)+0
SKO B2
B0)UMM A0
UMA NRST
DOA Z0
LAR 6
PZA A2
UMM A20
DOM A2
PMM A21
PMM A22
UMA A7
DOA A2
PZA R1
UMA Z0
SZA P0
UMR A0
C0)UMA A8
DOA A2
PZA C4
UMA BTM
DOA A1
PZA C4+2
C4)+0
SKO A?
+10
UMA D0
DOA A2
PZA P2
C1)UMB LICZ
DOB LICZ
ODB A1
P2)+0
A21)+0
C5)SDY 30
```

n.d.

```
SDY 3/10+1/15+30
A11)STO 37
UMA A0
UMB BTM
ODB A1

PMB C3
UMB A0
DOA A?
DOP A
UMB C3
SOB ?-5
ODA A?
SZA A10
UMA W1
ODA A1
SZA A30
DOP W1
UMA A
ODA BTM
PZA A
SKO C5
A10)UMA A0
A12)PZA W1
SKO D1
P0)UMB A0
UMA A9
DOA A2
PZA P1
UMA BTM
DOA A1
PZA P1+2
P1)+0
SKO A?
+10
UMA A0
UMB BTM
ODB A1
PMB C3
UMB A0
DOA A?
DOP A
UMB C3
SOB ?-5
PZA A?
UEB 1
B2)SDY 31
SDY 3/10+1/15+31
SKO B1
SOB ?-3
D1)UMA LICZ
PZA Z
JĘZYK SAKO
WROC
```

8. PROGRAMOWANIE

Autorzy zdają sobie sprawę, że czytelnik pragnąłby wreszcie rozpocząć programowanie. Ażeby przejść do tej czynności, należy zapoznać się z listą zdań języka S A K O oraz zrozumieć ich znaczenie.

Zarówno rozkazy jak i deklaracje powodują po wczytaniu określone skutki.

Umiejętność programowania polega na:

- schematycznej znajomości komputera (bez szczegółów konstrukcyjnych)
- zapoznaniu się z listą deklaracji i rozkazów autokodu, poznaniu ich znaczenia (skutków, które wywołują w komputerze)
- nabyciu praktycznych umiejętności (co jest najważniejsze) opisywania problemu za pomocą deklaracji i rozkazów, które są na liście zdań autokodu.

Nabycie tych umiejętności bez ćwiczeń jest niemożliwe, a sama znajomość listy rozkazów i deklaracji to max 5% umiejętności programowania. Taka kolejność nauki obowiązuje dla każdego autokodu i każdego typu komputera.

Dokładnego znaczenia wszystkich rozkazów przy pierwszym czytaniu czytelnik może nie rozumieć. Radzimy po pierwszym przeczytaniu tego materiału przejść do przykładów umieszczonych na końcu książki. Do każdego przykładu załączono komentarz, który wyjaśnia nieomal każdy rozkaz tam użyty.

Po przeczytaniu komentarza należy wrócić do tego rozdziału i odszukać ewentualnie opis tej deklaracji czy rozkazu, którego funkcja i użycie budzi wątpliwości.

W pierwszej fazie analizy przykładów radzimy również nie zajmować się formą ich czołówek, lecz rozpocząć analizę od deklaracji ROZDZIAŁ: nr rozdziału.

8.1. Ogólna struktura programów w języku S A K O

Program składa się z ciągu zdań pisanych w odpowiedniej kolejności, składających się z trzech następujących kategorii:

deklaracji - podających informacje odnośnie budowy programu oraz znaczenie używanych symboli. Deklaracje spełniają swoją rolę przy wprowadzaniu programu do maszyny, są natomiast pomijane przy wykonaniu programu przez maszynę

rozkazów - określających czynności maszyny w trakcie wykonywania programu

komentarzy - objaśnień mających charakter tylko pomocniczy dla programisty i nie branych "pod uwagę" przez maszynę.

Każdy wiersz programu w języku S A K O kończy się znakiem nowej linii. Kolejność wykonania rozkazów napisanych przez nas nie jest bynajmniej taka sama jak kolejność wykonywania ich przez maszynę.

Od czego więc zależy ta ostatnia ?

Zależy od nas, jeżeli program układamy samodzielnie lub od programisty, któremu program zleciliśmy do opracowania. O kolejności decyduje więc struktura programu i tak zwane rozkazy sterujące, które narzucają rygorystyczny porządek. Gdy w strukturze programu nie ma rozkazów sterujących, maszyna wykonuje wszystkie rozkazy w takiej kolejności, w jakiej zostały napisane w programie.

Sytuacje takie zdarzają się jedynie w teorii i w programach szkoleniowych. W przypadku ogólnym kolejność jest na ogół narzucona przez tzw. ścieżki logiczne, to jest alternatywne kolejności pewnych cyklów rozkazów.

Jak zaznaczono powyżej, program składa się z ciągu zdań. Zanim przejdziemy do ich omawiania, musimy się zaznajomić z ich składnią.

Zdania S A K O mogą być zbudowane ze znaków dostępnych na dalekopisie. Dalekopis jest to przyrząd nie związany bezpośrednio z maszyną, o sposobie użytkowania podobnym do maszyny do pisania. W trakcie pisania dowolnego tekstu z dalekopisu wysuwa się również taśma perforowana (nośnik informacji), na której w sekwencji dziurek jest wypisane to samo co w tekście uwidocznionym na tabulogramie - arkuszu papieru jak w maszynie do pisania.

Do pracy na komputerze służy więc taśma perforowana, gdyż tylko za jej pośrednictwem można dokonać wczytania programu lub danych. Natomiast tabulogram służy nam wyłącznie jako element pomocniczy w celu śledzenia procesu tłumaczenia, wczytywania danych i wykrywania błędów.

Na dalekopisie dostępne są następujące znaki:

alfabetu łacińskiego składającego się z 24 liter:

A, B, C, D, E ... Z (tylko duże lub tylko małe litery w zależności od typu dalekopisu)

cyfr:

0 1 2 3 4 5 6 7 8 9

znaków działań i relacji:

+ dodawanie

- odejmowanie

◊ mnożenie w kodzie M-2

x mnożenie w kodzie Ferrantiego

/ dzielenie

* potęgowanie

= "staje się"

> większy w kodzie Ferrantiego

] większy w kodzie M-2

≡ znak równoważności /znak identyczności w kodzie Ferrantiego/

[znak równoważności w kodzie M-2

Separatorów i nawiasów

. , : ( )

Specji (odstępów pomiędzy znakami)

Pewne ciągi znaków będziemy nazywać wyrażeniami.

Przykładami są:

liczby
zmienne
funkcje
listy zmiennych
wyrażenia arytmetyczne
wyrażenia bulowskie

I wreszcie w programie występują tzw. teksty będące ciągami dowolnych znaków o określonym znaczeniu.

Na przykład:

'NR ... OBLICZANIE ILOCZYNU ZBIOROW
OBLICZANIE ROWNANIA TRZECIEGO STOPNIA
NR OBLICZEN ... WARSZAWA DN ...
PROGRAM TRANSPORTOWY

Teksty są również programowane w trakcie pisania programu i forma ich zależy wyłącznie od programisty i pewnych ograniczeń związanych z budową maszyny.

Wprawny programista obchodzi z łatwością te ograniczenia, uzyskując prawie całkowitą dowolność wydruku.

Uwaga!

Podana wyżej terminologia ciągu znaków, deklaracji, rozkazów, komentarzy jak również zdań i wyrażeń jest aktualna dla innych języków programowania i opanowanie jej będzie pomocne przy dalszym studiowaniu informatyki.

Ze względu na fakt, że praca niniejsza jest pisana pod kątem praktycznej jej wykorzystania przez użytkowników, do których dyspozycji będzie maszyna ZAM-41, podaje się znaki dalekopisowe w kodzie Ferrantiego i międzynarodowym M-2.

KODY FERRANTIEGO I M-2

DLA TAŚMY PERFOROWANEJ PIĘCIO-KANAŁOWEJ

LP	KOD FERRANTIEGO LITERY LS	KOD FERRANTIEGO CYFRY FS	TASMA 5	4	3	•	2	1	KOD M-2 (MIĘDZYNARODOWY) LITERY LS	KOD M-2 (MIĘDZYNARODOWY) CYFRY FS
0	ZNAK CYFR (FS)					•			BLANK (BL)	
1	A	1				•		●	E	3
2	B	2				•	●		LINIA (LF)	
3	C	*				•	●	●	A	—
4	D	4			●	•			SPACJA (SP)	
5	E	(			●	•		●	S	' (APOSTROF)
6	F	)			●	•	●		I	8
7	G	7			●	•	●	●	U	7
8	H	8		●		•			POWRÓT KARETKI (CR)	
9	I	≡		●		•		●	D	✠
10	J	=		●		•	●		R	4
11	K	—		●		•	●	●	J	; lub Ł
12	L	∨		●	●	•			N	,
13	M	LINIA (LF)		●	●	•		●	F	[lub ą lub □
14	N	SPACJA (SP)		●	●	•	●		C	:
15	O	⊃		●	●	•	●	●	K	(
16	P	0	●			•			T	5
17	Q	>	●			•		●	Z	+
18	R	:	●			•	●		L	)
19	S	3	●			•	●	●	W	2
20	T	→	●		●	•			H	◇ lub ł
21	U	5	●		●	•		●	Y	6
22	V	6	●		●	•	●		P	0
23	W	/	●		●	•	●	●	Q	1
24	X	×	●	●		•			O	9
25	Y	6	●	●		•		●	B	* lub ?
26	Z	+	●	●		•	●		G	] lub ę lub ⊟
27	ZNAK LITER (LS)		●	●		•	●	●	ZNAK CYFR (FS)	
28	•	• (KROPKA)	●	●	●	•			M	•
29	?	∩	●	●	●	•		●	X	/
30	⊂	POWRÓT KARETKI (CR)	●	●	●	•	●		V	=
31	LIKWIDACJA BŁĘDNEGO ZNAKU (ER-ERRASE)		●	●	●	•	●	●	ZNAK LITER (LS)	

8.1.1. WYKAZ ZDAŃ JĘZYKA SAKO

DEKLARACJE JĘZYKA SAKO

ROZDZIAL: n
BLOK (n, m, ..., k): lista
STRUKTURA (i, j, ..., l): lista
TABLICA (n, m, ..., k): nazwa
TABLICA OKTALNA (n, m, ..., k): nazwa
CALKOWITE : lista
JEZYK SAS : lista
JEZYK SAKO
KONIEC : n
KONIEC ROZDZIALU
PODPROGRAM: (lista wyników) = nazwa (lista danych)

ROZKAZY WSPÓŁPRACY Z BĘBNEM

CZYTAJ Z BEBNA OD I : lista
PISZ NA BEBEN OD I : lista

KOMENTARZE

K) dowolny tekst

<u>Uwaga</u>! Lista nie mieszcząca się w 1 wierszu tabulogramu może być przeniesiona do następnego przez znak ,—

ROZKAZY ARYTMETYCZNE I BULOWSKIE

FORMUŁA ARYTMETYCZNA

A = B

A – zmienna ułamkowa, całkowita, prosta lub indeksowana

B – wyrażenie arytmetyczne

FORMUŁA BULOWSKA

A ≡ B

lub

A [B

A – zmienna ułamkowa, całkowita, prosta

B – wyrażenie bulowskie

≡ znak w kodzie Ferrantiego

[znak w kodzie M2

FORMUŁA OPERACYJNA

(lista wyników) = nazwa (lista argumentów)

PODSTAW : nazwa (lista argumentów)

Nadawanie wartości B wszystkim lub niektórym elementom bloku A

A(=I,=J,....) = B

lub

A(=I,J,K,....) = B

zapis =I oznacza – dla wszystkich I

B – wyrażenie arytmetyczne

ROZKAZY STERUJACE

SKOCZ DO α
SKOCZ WEDLUG I : $\alpha, \beta, \ldots, \omega$
POWTORZ OD α : I = J (K) L
IDZ DO ROZDZIALU : n
WROC
GDY A > B : α, INACZEJ β
GDY A = B : α, INACZEJ β
GDY KLUCZ I : α, INACZEJ β
STOP α

Uwagi

α – etykieta (numer rozkazu) lub słowo NASTĘPNY

Znak ">" w kodzie Ferrantiego = znak "]" w kodzie M2

ROZKAZY
WEJŚCIA I WYJŚCIA

	CZYTAJ : lista
	CZYTAJ WIERSZ : lista
	CZYTAJ OKTALNIE : lista
	DRUKUJ (I) : lista
*)	DRUKUJ (I,J): lista
	DRUKUJ (I,J,K) : lista
	DRUKUJ WIERSZ : lista
	DRUKUJ OKTALNIE : lista
	TEKST :
	TEKST WIERSZY n :
**)	SPACJA I
	LINIA I
	STRONA

Uwagi

*) Dopuszczalny jest zapis
DRUKUJ(I.J): lista

**) I > 0

8.2. Forma i opis pojęć podstawowych

8.2.1. Liczba całkowita

Znaczenie jak w matematyce. Nie wolno dla tej liczby używać kropki zwanej pozycyjną.

Przykłady:

2
- 500
003 /przed pierwszą cyfrą znaczącą mogą występować zera/
+ 14630

8.2.2. Liczba ułamkowa

Znaczenie jak w matematyce. Kropkę pozycyjną należy bezwzględnie umieszczać, w przypadku istnienia części ułamkowej.

Przykłady:

2.23
- 4.56782
551
053.00
- 0.25
-. 275
+ 62.0
+ 62.
+ 62
62
0.0
0.
.0
0

Ponadto istnieje możliwość przedstawienia liczby rzeczywistej w zmiennym przecinku np.:

1.27E + 100 co oznacza 1.27×10^{100}
-0.53789E - 02 co oznacza -0.53789×10^{-2}

8.2.3. Blok liczbowy

Blokiem nazywamy skończoną i uporządkowaną grupę liczb oznaczoną wspólnym symbolem.

Liczby wchodzące w skład bloku nazywamy elementami bloku. Położenie elementu w bloku określa układ <u>indeksów</u>.

W przykładach na końcu podręcznika podaje się zapis bloku już w formie używanej w języku S A K O.

BLOK (n,m, ... k) : lista	deklaracja

gdzie n, m, ... k oznaczają indeksy /liczby naturalne lub wyrażenia arytmetyczne o wartości naturalnej/; lista oznacza zbiór symboli stanowiących nazwy bloków.

Liczbę indeksów nazywamy wymiarami bloku.

Np. BLOK (40) : A - jednowymiarowy
BLOK (40, 40) : B - dwuwymiarowy
BLOK (30, 30, 30) : Z - trójwymiarowy

W praktyce rzadko używamy bloków większych od trójwymiarowych, aczkolwiek w SAKO można posługiwać się blokami o dowolnych wymiarach. Zakresem indeksu nazywamy liczbę kolejnych liczb naturalnych począwszy od zera.

Każdy blok jest wczytywany do maszyny wierszami. Blok może być utworzony wyłącznie z liczb całkowitych lub wyłącznie z ułamkowych. Rozpatrujemy tylko bloki prostokątne, tzn. takie, gdzie każdy indeks przebiega ustalony zbiór wartości, niezależnie od wartości innych indeksów.

Deklaracja blok umieszczona w programie powoduje zarezerwowanie w pamięci komputera odpowiedniej liczby słów na umieszczenie wszystkich elementów tych bloków, które są wymienione na liście. Będzie to rezerwacją słów krótkich w przypadku liczb całkowitych i słów długich w przypadku liczb rzeczywistych /ułamkowych/.

Przykłady:

BLOK (5) : A

Nazwa bloku - A
Liczba miejsc /komórek/ pamięci potrzebnych do przechowania elementów bloku wynosi 6, gdyż miejsca liczymy od zera.

Miejsca wyobrażamy sobie jak na rysunku:

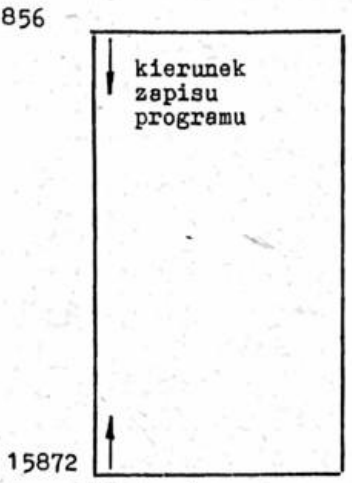
BLOK (5) : A

0	1	2	3	4	5
A(0)	A(1)	A(2)	A(3)	A(4)	A(5)

W każdym miejscu zarezerwowanego bloku może więc tkwić jeden element bloku /A (0), A (1) ... A (5) - elementy bloku/;
liczby od 0 do 5 oznaczają indeksy poszczególnych elementów.

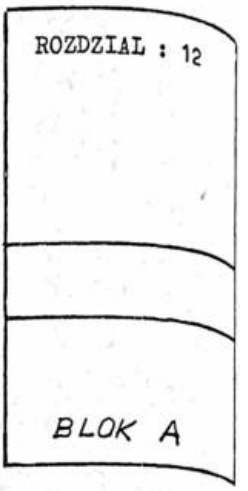
Tak więc dla bloku liczbowego A

1.25 3.56 6.28 0.0 12.3 14.8



Kierunek zapisu
bloków i zmiennych

np. ROZDZIAL : 12
BLOK (50): A
BLOK (9): B
——
——
——
KONIEC ROZDZIALU
ROZDZIAL : 13
BLOK (50): A
BLOK (9): B
——
——
——
KONIEC ROZDZIALU



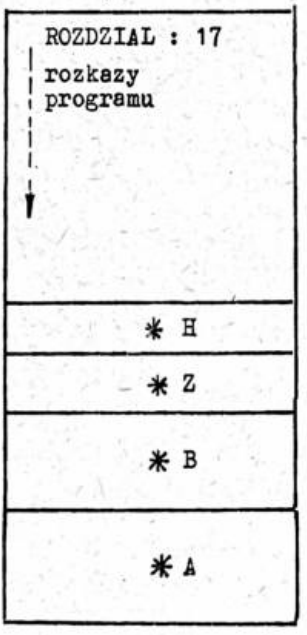
Deklarując z kolei ROZDZIAL : 14 bez deklaracji BLOK (50) : A i BLOK (9) : B nie będziemy mogli używać do operacji w nim zawartych żadnych zmiennych indeksowanych w postaci A (K) lub B (K). Jeżeli natomiast w ROZDZIALE : 14 zadeklarowaliśmy w miejsce bloków A, B, bloki Z1 i Z2 w postaci

ROZDZIAL : 14
BLOK (50) : Z1
BLOK (9) : W2

to wówczas używając w operacjach nazw zmiennych indeksowanych Z1 (K) i W2 (K) możemy korzystać z odpowiednich wartości bloków A i B, gdyż zmieniły się jedynie nazwy, a nie adresy komórek poszczególnych elementów bloków.

Dla każdego rozdziału maszyna rezerwuje bloki poczynając od ostatniego miejsca pamięci operacyjnej dostępnej dla programisty.

np. ROZDZIAL : 15
BLOK (119) : A, B
BLOK (49) : Z
——
——
——
KONIEC ROZDZIALU
ROZDZIAL : 16
BLOK (119) : A, B
BLOK (49) : Z
——
——
——
KONIEC ROZDZIALU
ROZDZIAL : 17
BLOK (119) : A, B
BLOK (49) : Z
BLOK (20) : H
——
——
——



położenie elementów w pamięci maszyny wyobrażamy sobie jak następuje

BLOK (5) : A

1.25	3.56	6.28	0.0	12.3	14.8

a poszczególne elementy: A (0) = 1.25, A (1) = 3.56 ...
A (5) = 14.8

BLOK (4) : Z, H1

oznacza dwa bloki liczbowe o jednakowych liczbach elementów po 5,które wyobrażamy sobie jak niżej:

BLOK (4) : Z

BLOK (4) : H1

Z (0)	Z (1)	Z (2)	Z (3)	Z (4)
H1(0)	H1(1)	H1(2)	H1(3)	H1(4)

gdzie Z (0) ..... Z (4)
H1 (0) ..... H (4) poszczególne elementy bloków

BLOK (3,2) : W

Oznacza dwuwymiarowy blok o wym. 4 x 3 jak np. poniżej, gdzie 4 oznacza liczbę wierszy, a 3 liczbę kolumn.

BLOK (3,2) : W

W (0,0)	W (0,1)	W (0,2)
W (1,0)	W (1,1)	W (1,2)
W (2,0)	W (2,1)	W (2,2)
W (3,0)	W (3,1)	W (3,2)

- w deklaracji BLOK może znajdować się więcej niż jedna nazwa np.
BLOK (3,2) : W, H, X1 oznaczałoby trzy bloki liczbowe o wymiarach 4 x 3 każdy.

Jak zaznaczono wyżej wartości zmiennych i bloków zadeklarowane w jednym rozdziale tracą swą aktualność w następnym rozdziale. Niemniej jednak zapis ich w pamięci operacyjnej istnieje nadal i może być wykorzystywany.

Przeniesienie do następnego rozdziału wartości zmiennych i bloków dokonuje się za pomocą użycia w nim identycznych deklaracji i w identycznym porządku. W rozdziale 7 opisano również charakterystykę zapisu programu w PAO, co wizualnie wyobrażamy sobie jak następuje:

Na przykładzie widzimy, że zarówno w rozdziale 15 jak i 16 możemy zmiennych indeksowanych A (K), B (K) i Z (K), natomiast w rozdziale 1 datkowo H (K) o ile oczywiście w blokach tych znajdują jakieś interes nas wartości liczbowe wprowadzone tam za pomocą rozkazu CZYTAJ /patrz 8.2.25/ lub obliczane w trakcie działania programu.

8.2.3.1. TABLICA (n, m ... k) : nazwa

Deklaracja określa strukturę i nazwę bloku liczbowego całkowitego lub rzywistego, którego wartości wprowadzamy w programie, a nie za pomocą kazu CZYTAJ. Elementy tablicy wprowadzamy natychmiast po tej deklar kończąc zapis znakiem w nowej linii.

np. TABLICA (2,3) : Z

```
2  5  6  7
3 12 -5  4
4  5  6  7
*
```

W programie poszczególne elementy tablicy oznaczamy jak zmienne indeksowane tzn. np. Z (1,0) co oznacza liczbę 3.

8.2.3.2. TABLICA OKTALNA (n, m ... k) : nazwa

Deklaracja jest identyczna z poprzednią z tym, że jest utworzona ze sł bulowskich krótkich lub długich zapisanych oktalnie.

np. TABLICA (1) : ZHP

```
3624317777733325                    4321567035777777
*
```

W obydwu przypadkach liczba indeksów jest ograniczona tylko pojemnością mięci maszyny.

8.2.3.3. A (= I, = J, = K ,..., = Z) = W

W - zmienna prosta, indeksowana, wyrażenie arytmetyczne lub explicite d wolna liczba.

Taki zapis jest równoważny następującemu:

```
**-  **) A (I, J, K,..., Z) = W
         POWTORZ : I = zakres - 1 0
         POWTORZ : J = zakres - 1 0
         POWTORZ : K = zakres - 1 0
         ==========
         POWTORZ : Z = zakres - 1 0
```

Przez zakres rozumiemy wymiary bloku lub ustawioną strukturę. Można również używać następujących zapisów:

np. A (I, = J) = W

co oznacza "dla wszystkich J" przy zadanym uprzednio I.

np. A (= I, = J) = 16,5

tzn., że wszystkim elementom bloku A nadaje się automatycznie wartość 16,5.

np. B (= I, = J, = K) = 0 /wyzerowanie bloku trójwymiarowego/

np. B (I, = J, = K) = 0 /wyzerowanie pewnych elementów bloku B/

8.2.4. Zmienne proste

Zmiennymi nazywamy symbole przyjmujące w trakcie obliczeń różne wartości liczbowe.

Stosowanie zmiennych /symboli/ jest niezbędne we wszystkich autokodach ze względu na fakt, że programy opracowuje się na symbolach, które dopiero po wprowadzeniu danych i rozpoczęciu procesu liczenia przybierają wartości liczbowe. Te same nazwy zmiennych użyte w różnych rozdziałach programu posiadają całkowicie niezależne znaczenie. W tym samym rozdziale programu programu głównego i wchodzącego do tego rozdziału podprogramu mogą występować te same nazwy zmiennych o różnym znaczeniu.

Każda zmienna prosta to ciąg liter, cyfr i spacji rozpoczynający się zawsze od litery.

Każdej zmiennej prostej użytej w programie jest przypisana jedna komórka pamięci maszyny długa lub krótka w zależności od zadeklarowania /patrz deklaracja CAŁKOWITE/.

Maszyna bierze pod uwagę tylko cztery pierwsze znaki do identyfikacji zmiennej.

Przykłady prawidłowych zmiennych prostych:

A
ZH
B1W2
ZHP
W113
Z1ZZ
X1

Przykłady nieprawidłowych zmiennych prostych:

1Z	rozpoczęcie od cyfry
2BW	- " -
PIES	ta sama zmienna, gdyż maszyna
PIESEK	identyfikuje tylko cztery pierwsze znaki

8.2.5. Zmienne indeksowane są to nazwy elementów bloku wyznaczone wartościami indeksów. Symbol zmiennej jest identyczny z nazwą bloku, którego element zmienna ta określa.

Po nazwie występuje para nawiasów, wewnątrz których znajdują się indeksy rozdzielone przecinkami. Maszyna również bierze pod uwagę tylko cztery pierwsze znaki nazwy do identyfikacji. Indeksy mogą przybierać oczywiście tylko wartości całkowite nieujemne i mogą występować w programie jako liczby lub zmienne proste, których wartość określono w innym punkcie programu. Przykłady zmiennych indeksowanych:

A (15)
ZKL (20, 30, 0)
B (J, 4, W)
WH1 (J, K)
ZKB7 (M, N, P)

gdzie J, K, M, N, P, W - zmienne proste całkowite, nieujemne.

Posługiwanie się zmiennymi indeksowanymi możliwe jest tylko po uprzednim zadeklarowaniu bloku.

Indeksami mogą być również zmienne indeksowane lub wyrażenie arytmetyczne o wartościach całkowitych nieujemnych.

8.2.6. Etykiety (numery rozkazów)

Są to symbole służące do oznaczania rozkazów SAKO. Numer wypisuje się z lewej strony rozkazu (patrz przykłady), oddzielając go nawiasem zamykającym. Nie jest konieczne numerowanie wszystkich rozkazów programu, lecz przede wszystkim tych, które związane są z rozkazami sterującymi w danym rozdziale wg ścieżek logicznych sieci działań. Numer rozkazu (etykieta) jest to ciąg liter, cyfr i spacji rozpoczynający się od cyfry (przeciwnie jak zmienna). Maszyna identyfikuje jedynie cztery pierwsze znaki;

przykłady numerów (potocznie zwanych etykietami)

15
1W6B
12Z
103
14HZ

8.2.7. Funkcje standardowe

W języku SAKO jak i w innych autokodach istnieją tzw. funkcje standardowe, których nie potrzeba programować, gdyż znajdują się one w translatorze autokodu. Stanowi to wielkie udogodnienie w programowaniu, gdyż za pomocą nazwy składającej się z trzech znaków w programie wywołujemy pewien podprogram /cykl rozkazów/ określający wartość tej funkcji. Są to tzw. funkcje

FUNKCJE JĘZYKA SAKO

SIN (X)	sinus
COS (X)	cosinus
TG (X)	tangens
ASN (X)	arcus sinus
ACS (X)	arcus cosinus
ATG (X)	arcus tangens
ARC (X, Y)	arcus (kąt, którego tangens $=\frac{Y}{X}$)
PWK (X)	pierwiastek kwadratowy
PWS (X)	pierwiastek sześcienny
LN (X)	logarytm naturalny
EXP (X)	eksponent
MAX (X, Y, ... Z)	oznacza największą z liczb X, Y, ... Z
MIN (X, Y, ... Z)	oznacza najmniejszą z liczb X, Y, ... Z
MOD (X, Y)	reszta powstała z dzielenia X przez Y (dla całkowitych X i Y)
SGN (X, Y)	oznacza liczbę |X| sgn Y
ABS (X)	oznacza |X|
ENT (X)	część całkowita liczby X
DIV (X, Y)	część całkowita ilorazu X przez Y (dla całkowitych X i Y)
SEL (X)	Wybierz urządzenie WE lub WY o numerze symbolicznym X
DOL (X, Y)	Dołącz urządzenie o numerze symbolicznym X w pole o nazwie Y
ODL (X)	odłącz urządzenie o numerze symbolicznym X
SUM (I, N, W)	$\sum_{i=0}^{N} W_i$
ILN (I, N, W)	$\prod_{i=0}^{N} W_i$

języka w przeciwieństwie do funkcji rozumianych jako definicje, których wykorzystanie należy ująć w specjalnie opracowanym podprogramie.

Na poprzedniej stronie podajemy wykaz tych funkcji wraz z objaśnieniem. Sposób ich użycia pokazujemy na przykładach. Należy po prostu określić argument funkcji, której zawartość zostanie dla tego argumentu obliczona już w sposób automatyczny.

Np. A = sin (B)

Oznacza to, że w komórce A znajdzie się wartość sinusa kąta B wyrażonego w radianach, /gdyż komputer pracuje jedynie na liczbach i nie uwzględnia żadnych jednostek/.

Np. A = - 5
Z = ABS (A)

co oznacza, że w komórce Z znajdzie się wartość + 5.

Funkcje SEL, DOL, ODL zostały już omówione w rozdziale 7 i dotyczą organizowania "wejścia" i "wyjścia" komputera. Przy pomocy tych funkcji można realizować wszystkie obliczenia np. potęgowanie, w którym wykładnik potęgi jest liczbą rzeczywistą realizujemy w następujący sposób:

$X^y = e^{y \ln x}$ stosując do zapisu w programie

EXP(Y◊ LN(X));

Funkcja SUM służy do sumowania elementów bloku W od indeksu zerowego do N-tego.

Funkcja ILN służy do wymnażania przez siebie elementów bloku od indeksu zerowego do N-tego.

8.2.8. Wyrażenia arytmetyczne

Wyrażenie arytmetyczne posiada sens <u>ogólnie przyjęty w matematyce</u> np.

B + Z + 1
20◊A/C * 3
Z - H◊A (6,3) - 12◊Z - HB (J) itd.

Nawiasy określają kolejność wykonywania działań lub zamykają argumenty funkcji np. SIN (X), względnie zamykają indeksy zmiennych indeksowanych.

Wyrażenie arytmetyczne po wykonaniu posiada więc określoną wartość, którą może być liczba całkowita lub ułamkowa.

Podane wyrażenia arytmetyczne występując samodzielnie nie miałyby określonego znaczenia i nie byłyby przyjęte przez translator. Posiadają one sens w przypadku występowania zmiennej <u>z lewej strony wyrażenia</u> względnie jako indeksy zmiennych indeksowanych.

przypadek 1. LP = 2 x A ✳ 3/B (obliczanie wartości LP)
przypadek 2. A (2 x (Z - H), 2XYZ) indeksacja zmiennej A.

Pomimo jasnej zasady, zachowanie poprawności wyrażeń arytmetycznych w autokodzie wcale nie jest łatwe.

<u>przykłady poprawnych wyrażeń arytmetycznych:</u>

2 x Z/ (W - H) ◇ (Z - 15A✳3)
15Z - PWK (Z + HZ1X) + PWS (2◇J - 3)

przykłady niepoprawnych wyrażeń arytmetycznych

(X2 - X1) (125 x 6 x Z - W✳6) — brak ◇ pomiędzy nawiasami; nie wolno pomijać żadnych relacji matematycznych

12 x XYxZ - 15 KW x ABS (H) — K i W to dwie zupełnie odmienne zmienne; przy braku znaku mnożenia pomiędzy nimi maszyna potraktuje je

jako jedną nową zmienną <u>KW</u>.

8.2.9. Kolejność wykonywania działań w wyrażeniu arytmetycznym

W autokodzie SAKO przyjmuje się następującą kolejność działań wg tzw.mocy:

1. obliczanie wartości funkcji i wartości zmiennych indeksowanych
2. potęgowanie ✳
3. mnożenie ◇ /lub "x" w kodzie Ferrantiego/
   dzielenie /
4. odejmowanie -
   dodawanie +

Z dwu działań wykonuje się najpierw to, które jest zamknięte w większej ilości nawiasów. Jeżeli więcej niż jedno działanie jest zamknięte w jednakowej ilości nawiasów, to wykonuje się najpierw to, wewnątrz którego jest działanie największej mocy.

W przypadku gdy i moc działań wewnątrz jest równa to obliczenie wykonuje się w kolejności od lewej do prawej.

przykłady:

A/B - Z (najpierw A/B, później - Z)
(AW/H) x ((H - Z) + X) najpierw ((H-Z) + X) później (AW/H)
A x B/W (najpierw A x B później /W)
((Z✳H - (W✳Y - ZK)) - D (J,X)✳(PB x DE (K,4)))
kolejność: D (J,X) - określenie elementu zbioru D
DE(K,4) - - " - DE
W✳Y - potęgowanie
(W✳Y - ZK) itd.

Pospolitym błędem początkujących programistów jest umieszczanie niejed- kowej liczby nawiasów zwróconych w lewą i prawą stronę, to jest /błąd ! !/

Uwaga !

Wszystkie zmienne występujące w wyrażeniu muszą być uprzednio określo- /mieć wartości/ przez wprowadzenie, wczytanie lub obliczanie.

- Zbędne nawiasy /przy zgodnej ilości skierowań w przeciwne strony/ czynią zapisu błędnym
- nie wolno pomijać żadnej relacji matematycznej
- każdą liczbę wymierną możemy zapisać w postaci ułamka np. 5 jako 5/1 0.5 jako 5/10 itd.
- wszelkie dzielenia przez 0 powodują zatrzymanie się maszyny
- w trakcie programowania należy zwrócić szczególną uwagę na podział liczb na całkowite i ułamkowe,
  np. wyrażenie A/B jest zawsze ułamkowe. Natomiast A + B; A-B; AxB; A+B jest tylko wtedy całkowite, gdy A i B są całkowite.

8.2.10. ROZDZIAL : n

Rozdział jest to część programu, która może pomieścić się w całości w pa- mięci operacyjnej maszyny ZAM-41;

n - jest to nazwa rozdziału będąca liczbą naturalną < 32 000

W jednym programie może być maksymalnie 50 rozdziałów. Wszystkie deklara- cje tracą znaczenie przy przejściu do następnego rozdziału. Jeżeli piszemy program jednorozdziałowy, deklaracja niniejsza jest zbyteczna. Deklaracja ta jest aktualna do deklaracji KONIEC ROZDZIAŁU lub KONIEC : n.

8.2.11. PODPROGRAM: (lista wyników) = nazwa (lista argumentów)

Na liście wyników nie mogą występować zmienne indeksowane.

Deklaracja ta określa ciąg następujących po niej rozkazów powodujących rozwiązanie jakiegoś zagadnienia wyodrębnionego, które jest samodzielnym małym programem możliwym do podłączania do innych programów.

Wyodrębnianie cyklu rozkazów w postaci podprogramu jest wówczas opłacalne gdy któraś z faz obliczeń problemu powtarza się kilkakrotnie w jednym rozdziale.

Każdy podprogram kończy się rozkazem WROC, który powoduje powrót do miejsca programu, w którym dany podprogram był wywoływany.

Podprogram jest aktualny tylko w danym rozdziale, w którym jest dołączony i umieszcza się go <u>po wszystkich rozkazach</u> danego rozdziału.

W jednym rozdziale może być dołączonych szereg podprogramów. Posiadanie podprogramu np. rozwiązywania układu równań liniowych pozwala dołączyć go do innych programów, gdzie konieczne jest rozwiązywanie takiego układu jako jednej z faz obliczeniowych. Zaoszczędza to czas programisty.

np. PODPROGRAM : (X1, X2) = PKW (A,B,C)

dla obliczania równania kwadratowego

A, B, C - współczynniki równania kwadratowego;

PKW - nazwa podprogramu mogąca składać się najwyżej z trzech znaków i rozpoczynająca się od litery.

Formuła operacyjna /wywoławcza/ podprogramu wygląda jak następuje:

(lista wyników) = nazwa (lista argumentów)

np. (X1, X2) = ROK (A, B, C)

Formuła operacyjna

PODSTAW: nazwa (lista argumentów)

powoduje podstawienie wartości liczbowych zmiennych z programu głównego do podprogramu. Po tej formule formuła wywoławcza podprogramu ma postać:

(lista wyników) = nazwa (•).

W przypadku gdy podprogram wylicza jedną tylko wartość wówczas jego deklaracja przedstawia się następująco:

PODPROGRAM : nazwa (lista argumentów)

a formuła wywoławcza jak następuje:

Wynik = nazwa (lista argumentów)

Podprogram taki kończy się następująco:

nazwa ( ) = < wyrażenie arytmetyczne w ogólności >
WROC

np. PODPROGRAM : TRA (A, B, C)

C = C + 150·25
Z = A/B
TRA ( ) = Z + C
WROC

Uwaga !

W przykładzie Nr 10 przedstawia się sposób użycia podprogramu.

8.2.12. STRUKTURA (I, J,...,M) : lista

gdzie: I,J,..., M zmienne całkowite dodatnie lub wyrażenia arytmetyczne.

Deklaracja jest powiązana z deklaracją BLOK i rozkazem CZYTAJ przy wprowadzaniu bloków liczbowych do komputera.

Zmienia ona i określa dynamicznie zakres pola /liczbę miejsc/ bloków wymienionych na liście.

np. BLOK (49) : A
STRUKTURA (N) : A
CZYTAJ : A

Ten fragment programu realizuje wczytanie (N + 1) liczb do komórek pamięci przeznaczonych na blok A.

W przypadku nie zadeklarowania struktury N maszyna wczytywałaby zbiór 50-cio liczbowy o nazwie A.

8.2.13. CALKOWITE : lista

Deklaracja pisana na początku rozdziału programu dokonująca podziału zmiennych i bloków występujących w programie na ułamkowe i całkowite. Ten zasadniczy podział spotyka się we wszystkich autokodach. Zmienne i bloki nie występujące na liście potraktowane będą automatycznie jako ułamkowe i maszyna zarezerwuje dla nich komórki długie, natomiast dla zmiennych i bloków na liście komórki krótkie

np.:

CALKOWITE : N, WZ, *A, ZH

oznacza rezerwację komórek krótkich dla N, WZ, ZH i bloku A.

8.2.14. KONIEC : n

Deklaracja jest umieszczona na końcu programu. Liczba n /naturalna/ oznacza nazwę pierwszego rozdziału programu. Jeżeli program rozpoczął się od rozdziału zero /ROZDZIAL : 0/ wówczas na końcu programu umieszczamy deklarację KONIEC : 0; jeżeli od ROZDZIAL : 15 to piszemy KONIEC: 15 itd.
W przypadku programu jednorozdziałowego wystarczy napisać deklarację KONIEC zakończoną nową linią.

8.2.15. KONIEC ROZDZIALU

Deklarację umieszcza się po końcu każdego rozdziału, jeżeli dalej występują dalsze rozdziały.
Dla programu jednorozdziałowego deklaracji tej nie umieszcza się. Deklaracja ta jest pomijana przez translator.

Uwaga !
Pomiędzy poszczególnymi rozdziałami programu nie może występować żadna informacja z wyjątkiem komentarza.

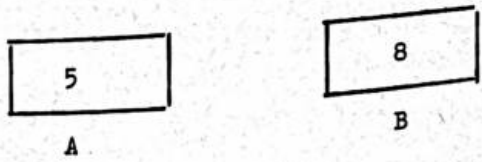
8.2.16. Formuła arytmetyczna

Fundamentalna formuła aktualna dla wszystkich maszyn cyfrowych i wszystkich autokodów.

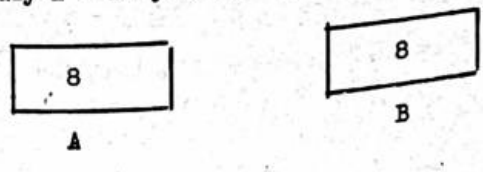
A = B /"A staje się B"/

co oznacza : weź wartość z komórki B i wstaw do komórki A; ale wartość B w komórce B pozostaje, natomiast to co było w komórce A zostaje zniszczone /"wymazane"/

np. mamy dwie komórki, w komórce A tkwi wartość 5, natomiast w komórce B tkwi wartość 8



po wykonaniu formuły A = B sytuacja wygląda jak następuje:



Mówiąc w sposób uproszczony:

- to co jest z prawej strony "przechodzi" na lewą stronę a nigdy odwr
  nie.
  Tak więc wyrażenie A = B + C powoduje obliczenie prawej strony tzn. pr
  branie z komórek B i C wartości w nich tkwiących, zsumowanie i umieszc
  nie w komórce A /przerzucenie na lewą stronę ! ! !/, zaś w komórkach B
  C wartości pozostają.

Z rozkazu tego wypływa konsekwentnie druga fundamentalna formuła

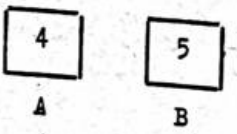
A = A + B

absurdalna w matematyce, podstawowa w programowaniu.

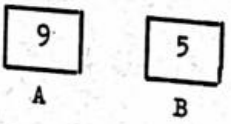
Oznacza ona co następuje:

pozostaw w komórce A to co w niej tkwi i dołóż zawartość z komórki B
lecz w B zawartość pozostanie

np. W komórce A tkwi wartość 4, natomiast w komórce B wartość 5 jak na rys
sunku



po wykonaniu A = A + B sytuacja będzie wyglądała następująco:



Czytelniku !

Staraj się zrozumieć konsekwencje tych rozkazów, bez których wprowadzenie maszyn cyfrowych do użytku byłoby niemożliwe. Ten rozkaz i opisana dalej "pętla" stanowią podstawę programowania w ogóle.

8.2.17. POWTORZ OD α : I = J (K) L

gdzie: I, J, K, L liczby lub zmienne całkowite lub ułamkowe, lecz wszystkie jednego typu.

Rozkaz powoduje wielokrotne powtórzenie odcinka programu zwanego zasięgiem tego rozkazu przy zmieniającej się wartości I. Wartość ta zmienia się od wartości J do wartości L w "skoku" co K. Na przykładzie wyjaśnimy sens tego podstawowego działania.

Zadaniem naszym jest zsumować trzy liczby znajdujące się w bloku A na trzech pierwszych miejscach.

15	35	62	
A (0)	A (1)	A (2)	

Komórkę, w której winna znaleźć się suma tych wyrazów nazwijmy S /zmienna S/

W takim przypadku napiszemy rozkaz pętli w założeniu, że jest to fragment jakiegoś programu

1) S = 0
* 2) S = S + A (K)

POWTORZ OD 2 : K = 0 (1) 2

Otóż te dwa ostatnie rozkazy tworzą pętlę. Co te rozkazy zrealizują ? Analizujemy po kolei wszystkie czynności maszyny w tym cyklu rozkazów.

a/ w komórce S znalazła się wartość zero /przejście z prawej strony na lewą wartości zero/

b/ maszyna podstawia za zmienną K wartość = 0

c/ wówczas przy pierwszym "nawrocie" w komórce S znajdzie się wartość 15, gdyż maszyna pozostawiła w komórce S wartość zero i dodała pierwszy element bloku A to jest A (0) gdyż K = 0

d/ przy drugim nawrocie maszyna dołożyła do wartości K wartość 1 tzw. krok wartość znajdująca się w nawiasie 0 1 2 i działając wg rozkazu 2 dołożyła do komórki S wartość 35 gdyż K = 1 /w tej fazie w komórce S znajduje się już wartość 50/.

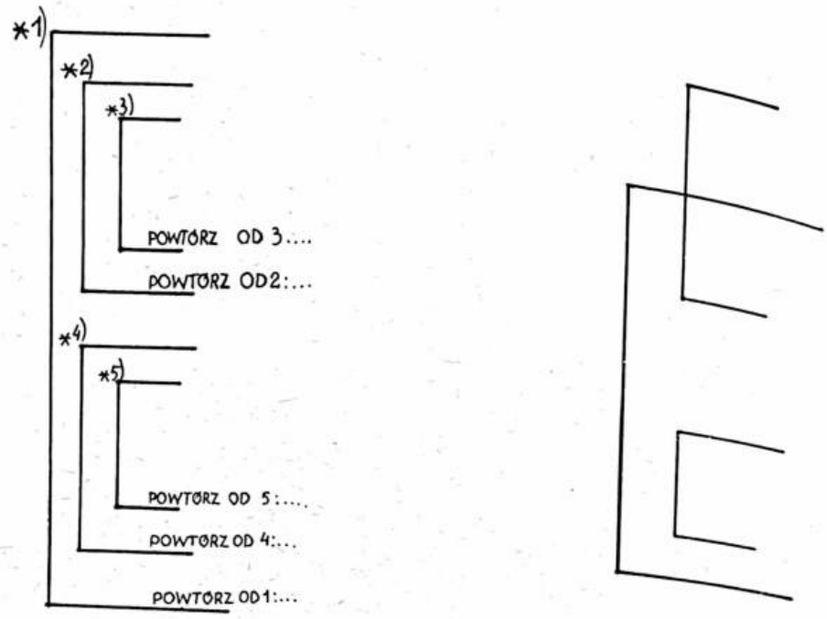
e/ przy ostatnim nawrocie maszyna dołożyła do K wartość 1 /a więc K = 2/ i pobierając z pamięci ostatnią wartość z bloku A tzn. A (2) dołożyła do komórki S, w której ostatecznie znalazła się wartość 112.
Tym sposobem zadanie zostało wykonane.

<u>Reguły</u> - rozpoczęcie pętli sygnalizowane jest gwiazdką stojącą przed jej pierwszym rozkazem

- zakres pętli może obejmować dowolną liczbę rozkazów
- pętlę można było wykonywać od tyłu tzn. POWTORZ OD 2 : K = 2 (1) 0
- pętle nie mogą się przecinać, tzn. że ich zakresy nie mogą się zazębiać.

Na przykładach pokażemy jej wszechstronne zastosowanie.

Graficzne przedstawienie konstrukcji pętli



Pętlę można również zapisać w następującej postaci.

np. *) A (I) = B (I) + 2

POWTORZ : I = 0 (1) N

tzn. bez numeru etykiety.

8.2.18. **SKOCZ DO α**

Rozkaz sterujący powodujący przejście do rozkazu o numerze α. Zamiast ; meru można napisać NASTĘPNY, a wówczas maszyna przejdzie do wykonywan: rozkaz wypisanego w następnej kolejności

np. 1) L = Z* H
SKOCZ DO 3
2) L = 12 x Z* H
3) ZKB = PWK (X)

- rozkaz SKOCZ DO 3 spowodował /po obliczeniu L/ ominięcie wykonywania rozkazu etykiecie 2.

8.2.19. **SKOCZ WEDŁUG I : α, β, ..., ω**

gdzie: I zmienna lub wyrażenie arytmetyczne o wartości całkowitej dodat niej.

Rozkaz ten powoduje przejście do rozkazu na liście o numerze określonym I podstawie wartości I /I liczone od zera/

Każdy numer etykiety na liście $\alpha, \beta, \ldots \omega$ / można zastąpić słowem NASTEPNY

np. 1) I = 0
2) SKOCZ WEDŁUG I : 3,4,5
3) Z20 = (63/A) X (Z20 + BW)
SKOCZ DO 5
4) XY = H
5) IM = ENT (X)

- I = 0 spowodowało przejście do rozkazu o numerze 3;

np. 1) I = 2
2) SKOCZ WEDŁUG I : 10,13,21 - rozkaz 2) powoduje przejście do trzeciego rozkazu na liście, to jest do nr 21 (liczymy rozkazy na liście od zera); tak więc

nr 0 - rozkaz 10
1 - " 13
2 - " 21
itd.

10) HJ = Z - W

13) W = ZOP - R

21) NK = NK + 2

8.2.20. IDZ DO ROZDZIAŁU : numer

Rozkaz powoduje "przejście" maszyny do rozdziału o zadanym numerze. Wynikiem tego rozkazu rozdział o zadanym numerze zostanie wczytany z pamięci zewnętrznej do pamięci wewnętrznej i nastąpi przejście do wykonywania pierwszego rozkazu w tym rozdziale.

np. IDZ DO ROZDZIALU : 22
IDZ DO ROZDZIALU : 13 itd.

8.2.21. GDY A > B : α, INACZEJ

Gdy podana nierówność jest spełniona maszyna przechodzi do wykonywania rozkazu o numerze α, w przypadku przeciwnym do numeru β

W programie wygląda to jak następuje:

```
1 A = 5
2 B = 3
3 GDY A > B : 4, INACZEJ 5
```

fragment programu

4) Z = (H - B) * 3
SKOCZ DO 6
5) Z = (H - B) * 4
6) W = Z

fragment programu

W przypadku ogólnym A i B będą zmiennymi, których wartości zostaną określone w programie w poprzednich operacjach. Tutaj dla zrozumienia podano ich wartość tuż nad rozkazem nr 3.

ETYKIETY α i β można zastąpić słowem NASTEPNY

W powyższym układzie maszyna przejdzie do wykonania rozkazu nr 4; gdyby B $\geqslant$ A maszyna przeszłaby do wykonywania rozkazu nr 5.

Uwaga ! Gdyby nie było rozkazu SKOCZ DO 6, wówczas maszyna przeszłaby do wykonywania automatycznie rozkazu nr 5, co przekreślałoby sens stosowania tej właśnie <u>logicznej</u> alternatywy /albo albo/.

8.2.22. | GDY A = B : α, INACZEJ β |

Rozkaz powyższy ma identyczny sens co poprzedni z tym, że przejście do rozkazu o numerze α wymaga równości A i B.

8.2.23. | GDY KLUCZ I : α INACZEJ β |

gdzie: I liczba lub zmienna, lub wyrażenie arytmetyczne

Z monitora zadajemy maszynie tzw. klucz, który służy jako wartość sterująca w programie

np. - "włączony" jest klucz NR 5

18) GDY KLUCZ 5 : 19, INACZEJ 22
.....
.....
19) ML = KN - 1
_____
_____
22) WK = 6 x F/2

w tym przypadku maszyna przejdzie do wykonywania rozkazu Nr 19.

Rejestr kluczy jest rejestrem fikcyjnym.

jestr ten zajmuje 1 słowo krótkie w Systemie Operacyjnym /adres PAO = 3/. mer pozycji słowa, w której znajduje się bit określa numer klucza, który st zadany

.

	0															1	0						0
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16							23

klucz Nr 16

rzucenie problemu /patrz pkt 7.3.2/ automatycznie zeruje rejestr kluczy.

lucz można zadawać z monitora lub deklarować w czołówce języka SAKO, bądź a pomocą wstawki w języku M-SAS.

lucz z monitora zadajemy jak następuje:

wciskamy klawisze "zgłoszenie wyłączenia" i "zgłoszenie operatora" przy monitorze

piszemy na monitorze następującą dyrektywę:

0 - W - NR klucza - 1 *

wynikiem tej czynności jest potwierdzenie zadania tego klucza w postaci automatycznego wydruku na tabulogramie monitora:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

tutaj przykładowo klucz 16
/na 17 pozycji bit 1/.

asowanie uprzednio zadanego klucza odbywa się również z monitora za pomo- ą następującej dyrektywy

0 - W - Nr klucza - 0 *

onitor automatycznie wypisuje potwierdzenie kasowania.

.2.24. | STOP β |

ozkaz powoduje zatrzymanie się maszyny i po ponownym uruchomieniu rozpo- zęcie pracy od rozkazu o numerze /etykiecie/ β ; względnie powoduje zawie- zenie maszyny, jeżeli w czołówce SAKO występuje zdanie LIKWIDUJ STOP. Po- owne uruchomienie /odwieszenie/ maszyny odbywa się za pomocą dyrektywy adanej z monitora

0 - S *

np. 28) STOP 30
29) IDZ DO ROZDZIAŁU : 5
30) IDZ DO ROZDZIAŁU : 8
KONIEC ROZDZIAŁU

Widzimy, że w takim układzie EMC zatrzyma się na rozkazie nr 28, a następnie wystartuje /rozpocznie pracę/ od rozkazu nr 30, tzn. "pójdzie" do rozdziału nr 8. Nr β można zastąpić słowem NASTEPNY.

8.2.25. **CZYTAJ : lista**

Rozkaz powoduje wczytanie układu liczb z urządzenia wejściowego tzw. czytnika do pamięci wewnętrznej maszyny. Układ tych liczb jest zaprogramowany.

<u>Dane muszą rozpoczynać się od znaku cyfr !</u>

- lista - składa się z nazw zmiennych i nazw bloków z ewentualnie określoną strukturą

przykład 1 : zaprogramować wczytanie zmiennych całkowitych X, Y, Z

CZYTAJ : X, Y, Z

Przykładowa postać powyższych danych wprowadzonych do maszyny:

5
6
7 } zmienne proste perforowane w kolumnie

przykład 2 : zaprogramować czytanie

a/ zmiennych całkowitych X, Y, Z
b/ " ułamkowych WZ, ZH, BC
CZYTAJ : X, Y, Z, WZ, ZH, BC

Przykładowa postać tych danych wprowadzanych do maszyny:

5
6
7 } liczby całkowite bez kropki pozycyjnej

20.53
- 68.25
- 35.503 } liczby ułamkowe z kropką pozycyjną

przykład 3 : zaprogramować wczytanie bloku A o 5-ciu elementach całkowitych
CZYTAJ : * A /obowiązkowo znak gwiazdki/

- postać danych wprowadzonych do maszyny:

15 7 8 50 1203
*

Dane dla bloku można pisać zarówno w kolumnie, jak i w wierszu rozdzielając elementy co najmniej jedną spacją lub nową linią i kończąc znakiem gwiazdki w nowym wierszu. Po gwiazdce (*) perforujemy jeszcze znak linii. Jeden rozkaz może powodować czytanie kilku bloków liczbowych, jak np.:

CZYTAJ : * A, * ZP, * WH, ... itd.

lub czytanie bloków zmiennych i zmiennych pojedynczych w różnej kolejności np.

CZYTAJ : ZW, * H, PLW, * B6Y, ... itd.

Rozkaz CZYTAJ pomija dowolne teksty umieszczone pomiędzy danymi. Ułatwia to programistom orientację przy odczytywaniu i kontroli licznych zbiorów liczbowych.

8.2.26. **CZYTAJ WIERSZ : lista**

Rozkaz powoduje wczytanie wierszy, których nazwy tworzą listę. Przez wiersz należy rozumieć dowolny ciąg znaków alfanumerycznych. Długość wiersza nie powinna być większa niż 120 znaków ze względu na możliwości drukarki. W przypadku większej długości maszyna dokona automatycznego przerzutu pozostałych znaków do następnego wiersza;

np.: CZYTAJ WIERSZ : TYT
OBLICZANIE MACIERZY ROWNAN

lub

CZYTAJ WIERSZ : W1, W2
WARSZAWA 1972
MGR INŻ. W. ZBOROWSKI

Przy stosowaniu tego rozkazu należy zarezerwować odpowiednią liczbę słów krótkich $\leqslant$ 120 w postaci bloku o nazwie wiersza np. BLOK (45): TYT, co oznacza, że długość wiersza TYT nie przekroczy 46 znaków alfanumerycznych.

<u>Wiersz kończy się znakiem nowej linii</u>.

8.2.27. **CZYTAJ OKTALNIE : lista**

Rozkaz powoduje wczytanie słów bulowskich długich lub krótkich w postaci zmiennych prostych i bloków z jednoczesnym ich przypisaniem kolejnym zmiennym na liście.

Przy perforowaniu danych oktalnych nie mogą występować teksty

np.: CZYTAJ OKTALNIE : A, B

np. 00024674 ⟶ krótkie

0002473107124562 ⟶długie

8.2.28. | DRUKUJ (I) : lista |

Rozkaz powoduje kolejne drukowanie układu liczb całkowitych wymienionych na liście z kolejnego urządzenia wyjściowego maszyny, gdzie I oznacza maksymalną liczbę miejsc przeznaczoną na wydruk w aktualnie drukowanym wierszu.

Rozkaz ten nie powoduje wydruku znaku +, lecz rezerwuje dla niego jedno miejsce.

Wydruk może nastąpić bezpośrednio za pomocą drukarki wierszowej lub za pomocą taśmy perforowanej, którą trzeba jeszcze odczytać na dalekopisie.

np. : DRUKUJ (5) : Z1

oznacza wydrukowanie liczby całkowitej Z1 /zawartość komórki pamięci maszyny pod tą nazwą/, na którą przeznacza się 5 miejsc + 1 miejsce na jej znak, a więc:

| | | | - | 2 | 3 | 2 |

- jeżeli liczba cyfr nie przekracza 5, wówczas maszyna wydrukuje jak na rysunku obok

Jeżeli liczba cyfr przekroczy 5 wówczas liczba wydrukowana jest prawidłowo ale zostanie naruszona rozplanowana programem postać wydruku.

8.2.29.

DRUKUJ (I, J) : lista
DRUKUJ (I. J) : lista

rozkazy równorzędne

Rozkaz powoduje kolejne wypisanie w aktualnie drukowanym wierszu liczb będących wartościami zmiennych podanych na liście:

I, J - liczba miejsc zarezerwowanych dla wypisania wartości zmiennej, przy czym:

I - dla części całkowitej liczby

J - dla części ułamkowej liczby

oraz jedno miejsce na znak i jedno na kropkę pozycyjną.

Przykład:

DRUKUJ (3.4) : B

Rozkaz ten powoduje pobranie z pamięci wartości B i wydrukowanie wg szablonu:

- na część całkowitą maszyna rezerwuje 3 miejsca + miejsca na znak,
- na część ułamkową maszyna rezerwuje 4 miejsca + 1 miejsce na kropkę pozycyjną,
- drukowanie rozpoczyna się od skraju marginesu stałego arkusza tabulogramu.

margines stały	+	3	3	3	.	3	3	7	8
		+	2	2	.	5	3		

Suma zajętych miejsc:

$$1 + I + 1 + J$$

co w naszym przypadku wynosi:

$$1 + 3 + 1 + 4 = 9$$

Jeśli liczba cyfr części całkowitej liczby drukowanej będzie mniejsza od zarezerwowanych I = 3 np. liczba 22,53 wówczas wydruk będzie wyglądał jak na rysunku wyżej. Natomiast w przypadku drukowania części całkowitej większej niż przewiduje rezerwa I, maszyna wydrukuje liczbę w zmiennym przecinku.

Na liście może znajdować się większa liczba zmiennych, np.:

DRUKUJ (3.4) : AZ, WH, ...

gdzie AZ = 33.82 WH = -168.333

wydruk tego rozkazu będzie wyglądał jak następuje:

margines stały	+	3	3	.	8	2	0	0	-	1	6	8	.	3	3	3	0

$$(1 + 3 + 1 + 4) \times 2 = 18 \text{ miejsc}$$

8.2.30. DRUKUJ (I, J, K) : lista

Rozkaz powoduje drukowanie liczb rzeczywistych w zmiennym przecinku, gdzie I, J symbole o znaczeniu jak wyżej, natomiast K oznacza liczbę cyfr znaczących przy czym $K \leqslant I + J$

np. DRUKUJ (2,3,5) : A gdzie A = 5.1257

wówczas otrzymamy wydruk 51.257 E - 01

8.2.31. **DRUKUJ WIERSZ : Lista**

Rozkaz powoduje wydrukowanie uprzednio wczytanych wierszy (ciągów znaków), których nazwy znajdują się na liście (patrz CZYTAJ WIERSZ : lista).

np. DRUKUJ WIERSZ : TYT

komputer wydrukuje:

OBLICZANIE MACIERZY ROWNAN.

8.2.32. **DRUKUJ OKTALNIE : lista**

Rozkaz powoduje wydruk oktalny słów bulowskich (w systemie ósemkowym) znajdujących się na liście

np. A, B zmienne bulowskie :

A - krótka

B - długa

o następujących wartościach.

A [00024673

B [7152034643215760

rozkaz DRUKUJ OKTALNIE: A,B

spowoduje wydruk zmiennych A, B w następującej postaci

$\underbrace{00.02.46.73}_{\text{zmienna A}}$ $\underbrace{71.52.03.46.43.21.57.60}_{\text{zmienna B}}$

8.2.33. **SPACJA n** gdzie n liczba naturalna > 0

Rozkaz powoduje pozostawienie spacji /odstępu/ w aktualnie drukowanym wierszu, np.:

DRUKUJ (2) : W /W = - 28/

SPACJA (3)

DRUKUJ (3) : H /H = - 32/

wydruk w tabulogramie

W miejscu n może znajdować się zmienna prosta całkowita dodatnia, która może przybierać różne wartości w trakcie liczenia programu, a tym samym powodować różne ilości spacji przy wydruku. Rozkaz SPACJA 1 można zastąpić słowem SPACJA.

8.2.34. | TEKST :
treść tekstu |

Rozkaz ten powoduje wydruk dowolnego tekstu, np.:

```
TEKST :
TEGO JUŻ ZA WIELE
```

spowoduje wydruk całego hasła począwszy od marginesu stałego; nie będą uwzględnione spacje przed pierwszym znakiem alfanumerycznym tekstu.

8.2.35. | TEKST WIERSZY n : |

n - liczba naturalna równa liczbie wierszy do wydruku znajdujących się pod tym rozkazem.

```
np. TEKST WIERSZY 3 :
    ZA DUŻO TYCH ROZKAZÓW (OKOŁO 25)
    JAK SIE TEGO NAUCZYC
    TO JUZ NUDNE.
```

Rozkaz powoduje wydruk tych haseł w tym samym porządku poczynając od marginesu stałego.

8.2.36. | Komentarz |

Jest to dowolny ciąg znaków (przeważnie tekst objaśniający) alfanumerycznych, tzn. składający się z cyfr i znaków alfabetycznych, który umieszczamy w dowolnym miejscu programu pomiędzy rozkazami w celu łatwiejszego czytania całości. Tekst napisany przez nas musi rozpoczynać się od symbolu K) lub : a wówczas jest przez maszynę ignorowany do najbliższego znaku karetka + linia.

Komentarz ma znaczenie tylko pomocnicze dla programisty.

8.2.37. | LINIA n | n liczba naturalna $\geqslant 0$

Rozkaz powoduje przejście do wypisywania danych wyjściowych od nowego wiersza, począwszy od lewej skrajnej pozycji tabulogramu.

Mówiąc lapidarnie rozkaz ten powoduje wysuw papieru o n wierszy i pisanie od stałego marginesu tabulogramu.

Na przykład przy cyklu rozkazów jak niżej maszyna przejdzie po wydrukowaniu zmiennej ZW1 do drukowania zmiennej ZAP od początku nowego wiersza

```
DRUKUJ  (5.3)  :  ZW1
LINIA
DRUKUJ  (2.3)  :  ZAP
```

Gdyby przy rozkazie LINIA figurowała liczba naturalna, np.LINIA 10,to drukowanie rozpoczęłoby się od 11-go wiersza. Rozkaz LINIA 0 (linia zero) powoduje jedynie powrót tzw. karetki (wózka) do początku wiersza.

Rozkaz LINIA 1 może być zastąpiony słowem LINIA. Poza tym n może być również zmienną prostą całkowitą dodatnią.

8.2.38. | STRONA |

Rozkaz powodujący automatyczne ustawienie drukarki na początku strony arkusza wydawniczego. Na jednej stronie papieru mieści się 66 wierszy. Działanie tego rozkazu jest jednak uwarunkowane ustawieniem początku strony na drukarce przed rozpoczęciem pracy na maszynie, co może uczynić na nasze żądanie operator.

8.2.39. | JEZYK SAS : lista
JEZYK SAKO |

Pierwsza z deklaracji rozpoczyna cykl rozkazów w języku M-SAS (MAKROSAS - System Adresów Symbolicznych), a druga występuje na zakończenie tego cyklu. Język M-SAS jest językiem symbolicznym używanym przez programistów w celu optymalnego wykorzystania możliwości komputera i znajduje zastosowanie przede wszystkim w opracowaniach programów bibliotecznych i przy przetwarzaniu danych.

Opis tego języka znajdzie czytelnik w opracowaniach Instytutu Maszyn Matematycznych "ZAM-41 Oprogramowanie, część 2".

Tak więc powyższe deklaracje służą do zrobienia "wstawki" w języku M-SAS w programie napisanym w języku SAKO.

Na liście znajdują się "sakowskie" nazwy zmiennych z programu, które będą występować we wstawce w języku M-SAS. Deklaracja zmiennych na liście dotyczy wszystkich wstawek występujących po niej.

Język M-SAS do wstawek do programu w języku SAKO jest podzbiorem języka M-SAS. Niżej podano zasadnicze cechy tego podzbioru (dla programistów znających M-SAS).

1/ Przy etykiecie mogą być:

a/ liczby całkowite krótkie lub długie,

b/ zmiennoprzecinkowe (flopowskie),

c/ oktalne krótkie lub długie.

<u>Nie można używać czynników skalujących !</u>

2/ Po rozkazie tylko <u>adres</u> (a nie wartość), w makrorozkazach obowiązuje ta sama reguła.

3/ Etykieta o maksymalnej długości do 4 znaków. Etykieta jest nazwą miejsca w którym ta etykieta występuje.

4/ Nazwa funkcji jest nazwą funkcji bibliotecznej SAS.

5/ Długość tekstu ograniczona do 4 znaków. Każdy tekst zajmuje 1 słowo maszyny. Znaki są podawane poczynając od lewej strony słowa.

6/ Przy wykonaniu rozkazu adres przyjmowany jest zawsze w B - modyfikacji (<u>dotyczy funkcji i bloków</u> .

7/ Najpierw wykonuje się P-modyfikacja (pośrednie adresowanie),a następnie B - modyfikacja <u>dotyczy funkcji i bloków</u> .

KOD WEWNĘTRZNY MASZYNY ZAM-41

ZNAK	WARTOŚĆ ZNAKU W KODZIE WEWNĘTRZ.	
	DZIESIĘTNA	OKTALNA
1	2	3
BLANK	0	0
SPACJA	1	1
.	2	2
(	3	3
+	4	4
*	5	5
)	6	6
;	7	7
-	8	10
/	9	11
,	10	12
:	11	13
'(APOSTROF)	12	14
=	13	15
[	14	16
]	15	17
<	16	20
>	17	21
>	18	22
ea	19	23
ml MAŁA LITERA	20	24
A	21	25
B	22	26
C	23	27
D	24	30
E	25	31
F	26	32
G	27	33
H	28	34
I	29	35
J	30	36
K	31	37

1	2	3
L	32	40
M	33	41
N	34	42
O	35	43
P	36	44
Q	37	45
R	38	46
S	39	47
T	40	50
U	41	51
V	42	52
W	43	53
X	44	54
Y	45	55
Z	46	56
dl DUŻA LITERA	47	57
0	48	60
1	49	61
2	50	62
3	51	63
4	52	64
5	53	65
6	54	66
7	55	67
8	56	70
9	57	71
LF	58	72
◇	59	73
NOWA STRONA	60	74
$ub#	61	75
CR	62	76
SPECJALNY	63	77

8.2.40.

CZYTAJ Z BĘBNA OD I : lista PISZ NA BEBEN OD I : lista

I - liczba naturalna lub zmienna prosta całkowita dodatnia.

Rozkazy współpracy z bębnem służą do przepisania z pamięci operacyjnej do pamięci bębnowej poszczególnych zmiennych prostych i bloków i ponownego wczytania ich do pamięci operacyjnej.

Konieczność użycia tych rozkazów jest spowodowana ograniczonością pamięci operacyjnej. W przypadku wczytywania bardzo dużych zbiorów liczbowych wykorzystywanych później w programie, liczba komórek pamięci operacyjnej może być tak uszczuplona, że na dany rozdział z konieczności obejmujący rozwiązanie przynajmniej fragmentu zagadnienia zabraknie już miejsca (komórek pamięci). Zdarzają się nawet przypadki, że wczytywane informacje liczbowe zajęłyby całą pojemność pamięci operacyjnej.

Użycie powyższych rozkazów pozwala na wykonanie szeregu korzystnych operacji programowych, z których najczęściej spotykane opisujemy niżej.

Należy pamiętać, że rozkazy te powodują przepisywanie komórek krótkich, co wpływa na podawanie adresów w rozkazie CZYTAJ Z BEBNA.

Jednostką przesyłania jest jedno słowo krótkie.
Np. W pamięci operacyjnej znajduje się blok A zawierający 5000 elementów. Komórki tego bloku pragniemy w tym samym rozdziale wykorzystać do zupełnie innych operacji, ale liczb bloku nie możemy przecież "zgubić".

Używając przeto rozkazu PISZ NA BEBEN OD 100 : * A spowodowaliśmy automatyczne przepisanie na bęben od nr komórki bębna 100 wszystkich elementów bloku A.

Wartości elementów bloku w pamięci operacyjnej (w aktualnym rozdziale) pozostają, ale w przypadku ich likwidacji zostają w stanie nienaruszonym na bębnie. W każdym miejscu programu możemy spowodować ponowne ich przepisanie do pamięci operacyjnej.

Przepisanie ostatniego tysiąca elementów do pamięci operacyjnej, jeżeli był to blok liczb całkowitych, spowoduje rozkaz:

CZYTAJ Z BĘBNA OD 4100 : * B

W przypadku, gdy był to blok liczb rzeczywistych, gdzie chodzi o podwójną długość komórek, rozkaz będzie miał postać:

CZYTAJ Z BEBNA OD 8100 : * B

W przypadku, gdy chodziłoby o odczytanie z bębna dowolnego elementu, np. A (726), zostałby użyty następujący rozkaz:

dla bloku o elementach całkowitych

CZYTAJ Z BEBNA OD 826 : A (I)

dla bloku o elementach ułamkowych

CZYTAJ Z BEBNA OD 1553 : A (I)

Oczywiście powyższą operację można zaprogramować inaczej, chcąc w maksymalnym stopniu pozostawić do dyspozycji pamięć operacyjną PAO, a mianowicie wczytywać sukcesywnie do pamięci operacyjnej po 100 elementów i sukcesywnie odsyłać na bęben.

Podajemy fragment takiego programu dla bloku liczb całkowitych o 5-ciu tysiącach elementów.

```
BLOK (99) : A
*11) CZYTAJ : * A
     PISZ NA BEBEN OD K : * A
 12) POWTORZ OD 11 : K = 100 (100) 5000
 13) ———
```

. Przykłady programów w języku SAKO

zytelnik przystępując do studiowania przykładów powinien mieć już ogólne ojęcie o funkcji rozkazów, czyli zapoznać się z rozdziałem nr 8.

rzykłady 1, 2 i 3 zawierają wszystkie najważniejsze elementy programowa- ia i całkowite ich zrozumienie jest absolutną koniecznością. Studiowanie alszych przykładów bez zrozumienia w.w. nie da rezultatów.

zytelnik winien sobie ponadto uświadomić kolejność opracowania programu, tórą można wyrazic jak następuje:

- deklaracja rozdziału (ROZDZIAŁ : n);
- dokonanie podziału zmiennych występujących w programie na całkowite i rzeczywiste (ułamkowe), przez napisanie deklaracji CAŁKOWITE : np.CAŁKOWITE: AZ, WH, Z1W2, * C, * W1 itd. (zmienne i bloki niezadeklarowane jako całwite zostaną przez komputer potraktowane jako wartości rzeczywiste);
- rezerwacja w pamięci maszyny miejsc na zmienne pojedyncze (np. BLOK (0): A1, Z, WH, PJ, itd/ i zmienne indeksowane tzn. bloki liczbowe np. BLOK (499) : A, BLOK (999): A, Z, W1, H1 itd ;
- przygotowanie do komputera danych liczbowych, które zostaną wprowadzone za pomocą rozkazów wejścia CZYTAJ;
- zapisanie problemu w autokodzie,
- przygotowanie wyprowadzenia wyników za pomocą rozkazów wyjścia DRUKUJ.

 większości przypadków program opracowuje się na symbolach, natomiast da- e przygotowane na osobnej taśmie są już liczbami, które program wprowadza utomatycznie na miejsce symboli. Jeszcze raz ogarnijmy myślą podstawowe asady programowania.

- program to wierne odbicie sieci działań;
- nie może być w programie rozkazów nie przewidzianych w autokodzie;
- kolejność czynności (wykonywania rozkazów) jest rozstrzygnięta przez zadę, że maszyna wykonuje każdy rozkaz następny z kolei jeżeli brak rozkazu sterującego kierującego ją do rozkazu o innym numerze;
  Właśnie w tym celu stawia się przy rozkazach numery - (etykiety);
- maszyna nie wie NIC (!) poza rozkazami zawartymi w programie;nie zna ani jednej liczby czy parametru, jeżeli go jej nie podamy i nie zapewnimy mu miejsca w programie;
- zaczynając od STARTU maszyna musi dojść do STOPU idąc nie zawsze jedną drogą (ścieżką logiczną).

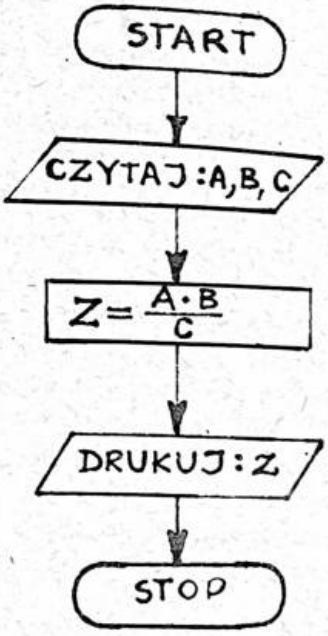
Przy pierwszym czytaniu można opuścić czołówki rozpoczynając program od eklaracji ROZDZIAL: .../

PRZYKŁAD NR. 1

Zaprogramować obliczenie wartości Z na podstawie danych A, B i C związanych następującą zależnością:

$$Z = \frac{A \cdot B}{C} \qquad (C \neq 0)$$

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 1.
translacja
rozdzial:0
blok(0):a,b,c,z
1)czytaj:a,b,c
z=a^b/c
linia 5
tekst:
z=
drukuj(4.5):z
stop 1
koniec:0
```

```
dane
2.1535
1.54
100.25
```

```
----------------------------------------------------------------------
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST 0(C) - 5.5.72(126)/23
SUPERVISOR = SVC    PAD = 15872 , BEB = 42384

PRZYKLAD NR 1



Z=   +0.03308


**********************************************************************
    PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO 63 LR= 1232.
    POZOSTALO CZASU: 7999999.
**********************************************************************
**********************************************************************
```

PRZYKŁAD NR 1 - KOMENTARZ

Najprostszy przykład obliczeniowy, ale zawiera w sobie podstawowe elementy programowe aktualne dla każdych obliczeń, tj.:

- informacje wejściowe i ich wprowadzenie
  CZYTAJ : A, B, C ;
- przetworzenie tych informacji;
  Z = A ◊ B/C

  Uwaga ! z prawej strony równania nastąpiło mnożenie i dzielenie, a wynik został przerzucony do komórki Z - z lewej strony.
- informacje wyjściowe i ich wyprowadzenie
  DRUKUJ (4.1) : Z

Taki schemat pracy komputera zawsze obowiązuje

ROZDZIAŁ : 0 i KONIEC : 0 deklaracje, które "obejmują" program jak gdyby wyznaczają jego zakres czy też rozmiar .

BLOK (0) : A, B, C, Z ; wczytanie tej deklaracji spowoduje zarezerwowanie czterech słów długich w pamięci maszyny zarówno na dane A, B, C jak i na wynik Z.

Rezerwacja objęła słowa długie dlatego, że nie było deklaracji CAŁKOWITE, a w związku z tym wszystkie zmienne są potraktowane jako rzeczywiste /ułamkowe/.

STRONA i LINIA 5 rozkazy wyjścia spowodowały tylko odpowiednie ustawienie papieru na drukarce wierszowej, to znaczy od początku strony i wysunięcie papieru o długości odpowiadającej 5-ciu wierszom.

TEKST : rozkaz wyjścia spowodował wydrukowanie tego, co jest pod nim napisane w jednym wierszu, to znaczy Z = ; ten rozkaz nie "wnika" w treść tekstu i mógłby wydrukować wszystko co napiszemy np. DO DIABŁA Z KOMPUTEREM.

DRUKUJ (4.1) : Z rozkaz wyjścia powoduje wydobycie z komórki pamięci przeznaczonej na Z jej liczbowej wartości i wydrukowanie. Z chwilą,gdy maszyna przeczytała ten program, to w momencie przeczytania deklaracji BLOK wyznaczyła po prostu adresy komórek pamięci dla zmiennych A, B, C, Z. Gdy przeczytała DRUKUJ sięgnęła po prostu pod odpowiedni adres.

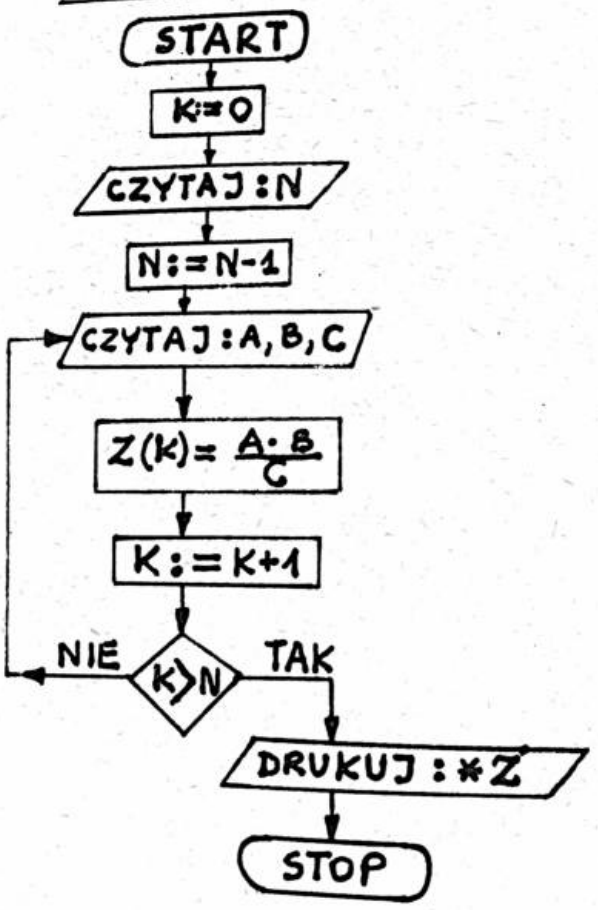
STOP 1 rozkaz sterujący spowodował zatrzymanie się maszyny i ustawienie się na rozkazie o etykiecie Nr 1 /CZYTAJ/.
Po ponownym uruchomieniu wystartuje już z tego miejsca, a więc znowu zacznie czytać następną grupę liczb A, B, C itd. Jeżeli budzą twoje wątpliwości cyfry w rozkazie DRUKUJ (4.1): Z to wróć koniecznie do pkt. 8.2.29; są to po prostu ilości miejsc zarezerwowane na drukowaną liczbę - cztery miejsca dla części całkowitej i jedno dla części ułamkowej.

PRZYKŁAD NR. 2

Zaprogramować obliczenie N wartości Z_K na podstawie danych A_K, B_K i C_K związanych następującą zależnością:

$$Z_K = \frac{A_K \cdot B_K}{C_K} \qquad (C_K \neq 0)$$

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 2.
chowaj slowniki
post mortem:60
translacja
rozdzial:0
calkowite:n,k
blok(0):n,a,b,c
blok(999):z
111a)k=0
czytaj:n
n=n-1
17)czytaj:a,b,c
z(k)=a0b/c
k=k+1
gdy k]n:11,inaczej 17
11)strona
tekst wierszy 1:
w a r t o s c i   z b i o r u   z
linia 3
*1)drukuj(5.3):z(k)
linia
powtorz od 1:k=0(1)n
stop 111a
koniec:0


dane
3
4.1
5
3.1
1.2
5.02
3.5
1
2.5
3
```

```
----------------------------------------------------------------------
ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST 0(C) - 4.5.72(125)/60
SUPERVISOR = SVC   PAO = 15872   BEB = 42384

PRZYKLAD NR 2


W A R T O S C I   Z B I O R U   Z

    +6.613
    +1.721
    +0.833

========
    PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOP 111A W ROZDZIALE 0 LR= 1273.
    POZOSTALO CZASU: 8000000.
========

    SLOWNIK ETYKIET.

  111A  1205    17  1213    11  1235     1  1251

    POST MORTEM.

    N 15871          2        A 15869 +1.000
    B 15867 +2.500            C 15865 +3.000
    Z 13865 +6.613     +1.721     +0.833     +0.000
            +0.000     +0.000     +0.000     +0.000     +0.000     +0.000
            +0.000     +0.000     +0.000     +0.000     +0.000
    K 13864          2
========
```

PRZYKŁAD NR 2 - KOMENTARZ

W przykładzie istnieje N trzyelementowych liczb. Wstawione do wzoru Z (K) nadadzą zmiennej indeksowanej Z wartości Z (0), Z (1), Z (2), Z (3),.....Z (N-1), a więc powstanie blok liczbowy Z, który możemy sobie wyobrazić jak np.:

Z (K)	Z (0)	Z (1)	Z (2)	Z (3)	Z (4)	Z (5)	.....	Z(N-1)
K =	0	1	2	3	4	5	itd	N-1

gdzie K jest indeksem czyli kolejnym numerem liczby w bloku

W tym przypadku komputer musi:

- wczytać grupę liczb A, B, C i obliczyć Z (0) = $\frac{A \cdot B}{C}$
- wczytać następną grupę liczb A, B, C i obliczyć Z (1) = $\frac{A \cdot B}{C}$ itd

aż do momentu wyczerpania wszystkich grup.

Dlatego zmniejszono wartość N o 1 ?
Gdyż w SAKO liczymy liczbę elementów w bloku od zera. W jakim celu wprowadzamy wielkość K i zwiększamy ją o 1 ?

Otóż jest to miara liczby "obrotów" maszyny przy czytaniu kolejnych grup. W bloku decyzyjnym ◇ sprawdzamy, czy odbyło się już czytanie ostatniej grupy N - 1 liczb, czy też maszyna ma powrócić do rozkazu CZYTAJ i wczytać następną grupę A, B, C.

Czytelniku !

Nauczsię sprawdzać i rozumieć program ! Oto szkolarska, ale wypróbowana metoda. Wyobrażasz sobie, że sam jesteś maszyną i nie rozumiesz kompletnie nic z wyjątkiem znaków umownych graficznych - listy rozkazów i deklaracji (taka sytuacja w rzeczywistości istnieje). Idąc schematem blokowym (po ścieżkach logicznych), odnotowując na kartce papieru wartości liczbowe powstające w trakcie pracy komputera dla uproszczonych danych, dojdziesz od STARTU do STOPU. W ten sposób zbadasz logikę schematów. To samo uczyń czytając kolejne rozkazy programu.

Nie wstydź się tej metody. Bardzo dobrzy programiści przeważnie w ten sposób dokonują sprawdzenia i też popełniają błędy, a tych którzy o tym dużo opowiadają (szczególnie na zjazdach poświęconych ETO) zapytaj mimochodem, ile programów w życiu uruchomili i gdzie można z nich natychmiast skorzystać. Bądź przy tym delikatny i nie pytaj o czas ich uruchamiania, choć byłoby to często humorystyczne.

Tak więc rozkazy programu spowodują:

- wczytanie N (ilość grup liczbowych);
- zmniejszenie N o 1;

- wczytanie do pamięci liczb A, B, C;
- obliczenie pierwszego wyrazu bloku czyli Z(0);
- zwiększenie K o 1;
- zbadanie czy K>N; czy już zostały wczytane wszystkie grupy liczbowe ;
- jeżeli nie, to komputer powróci do rozkazu czytania, ale drugi wynik "włoży" już w następną komórkę Z (K + 1), gdyż taka wartość tkwi w komórce K aktualnie;
- maszyna dokona tyle "obrotów" ile wynosi N-1 ;

Uwaga !

Ostatnią grupę rozkazów stanowi pętla drukowania patrz pkt. 8.2.17 .W tym przypadku maszyna nie "spocznie", aż nie wydrukuje wszystkich wartości bloku Z (K); ten rozkaz powoduje po prostu wykonywanie tyle razy rozkazu o etykiecie 1 DRUKUJ (5.3) : Z (K) ile razy jest to przewidziane w POWTORZ OD 1 : K = 0 (1) N. Ale za każdym razem zmienia K, a więc wyciąga z bloku następną liczbę, tj. Z (1),Z (2)..... itd aż do Z (N).

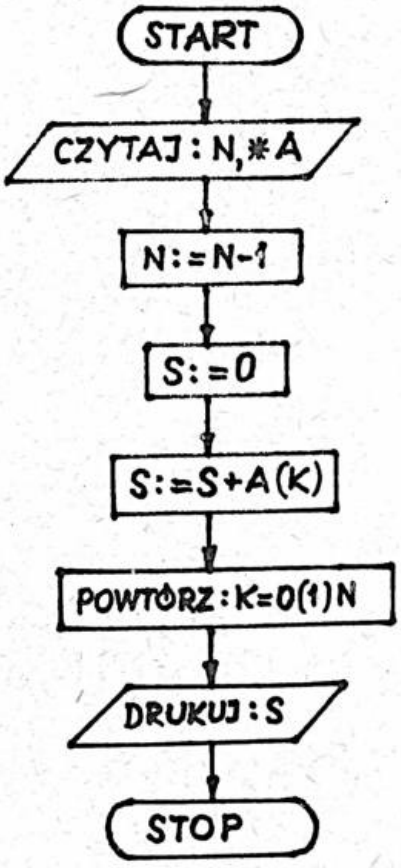
Myśl kategorią programisty, tzn. pamiętaj co tkwi aktualnie w komórkach pamięci i co to powoduje w konsekwencji.

K = 0 oznacza po prostu wyzerowanie komórki, gdyż wartość zero z prawej strony przeszła na lewą stronę wyrażenia (patrz pkt 8.2.16).

PRZYKŁAD NR. 3

Zaprogramować obliczanie sumy zbioru liczb dodatnich o nazwie A i maksymalnej liczności 2000.

Sieć działań:



```
p:sako,program;
wy:0=pdw(dx,ddx1);
t:przyklad nr 3.
translacja
rozdzial:11
calkowite:n,k
blok(1999):a
5a)czytaj:n
n=n-1
struktura(n):a
czytaj:*a
s=0
*1)s=s+a(k)
powtorz od 1:k=0(1)n
linia 5
tekst:
s u m a  e l e m e n t o w  w  b l o k u =
drukuj(10.6):s
stop 5a
koniec:11
```

```
dane
10
1 5 7 8.5 5 3 8.1
1000.1 9.5 10.7
*
```

```
-----------------------------------------------------------
ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST 0(C) - 4,5,72(125)/61
SUPERVISOR = SVC   PAO = 15872   BEB = 42384

PRZYKLAD NR 3



S U M A   E L E M E N T O W   W   B L O K U =     +1057,900000
```

PRZYKŁAD NR 3 - KOMENTARZ

W przykładzie tym widzimy użycie deklaracji STRUKTURA i rozkazu sterującego PĘTLA.

Znaczenie STRUKTURY

Dla bloku A zarezerwowano 2000 miejsc, tzn. że taki może być jego rozmiar maksymalny (liczba 1999 jest właściwa, gdyż liczbę elementów bloku liczymy od zera).

Użytkownik programu niekoniecznie musi sumować za każdym razem 2000 liczb, lecz przy każdorazowym użyciu programu może występować inna ilość liczb.

Otóż STRUKTURA reguluje tę ilość. Liczba zawarta w nawiasie np. STRUKTURA N gdy N = 4 określa, że w tym przypadku wystąpi w programie tylko 5-cio elementowy blok 4 + 1 . Ponieważ za każdym razem wczytujemy liczbę N; za jej pomocą określamy ilość miejsc w bloku A, którą będzie zajęta w danym przypadku liczenia.

Gdyby nie było tej deklaracji przed rozkazem CZYTAJ : * A, komputer "usiłowałby" wczytywać przy każdym liczeniu 2000 liczb, gdyż na taką ilość został zadeklarowany blok A deklaracją BLOK (1999) : A.

Tak więc STRUKTURA dynamicznie reguluje liczbę miejsc wykorzystywanych w danym bloku. Deklaracja ta musi oczywiście występować przed rozkazem wejścia CZYTAJ : * A, gdyż najpierw aktualny rozmiar bloku musi być określony, a później dopiero może nastąpić wczytanie liczb.

W naszym przypadku wprowadzamy dane w postaci 10 liczb, a więc N = 10.

Po raz pierwszy użyta jest pętla do obliczenia sumy elementów bloku. Obejmuje ona jedynie dwa rozkazy tzn. * 1) S = S + A (K)

POWTORZ OD 1 : K = 0 (1) N

Opisujemy tutaj jej działanie pomimo, że w punkcie 8.2.17 ten rozkaz sterujący został szczegółowo opisany.

- Przed tym rozkazem została ustalona wartość N (w naszym przypadku 9);
- Komputer ustala zakres pętli (w tym przypadku tylko te dwa rozkazy);
- Komputer dokonuje "pierwszego obrotu" tzn. na K podstawia wartość zero i wraca do pierwszego rozkazu;
- Sumuje wartość będącą w komórce S (a więc zero, gdyż powyżej pętli umieszczono działanie S = 0) i wartość A (0) czyli 1.
  Następnie sumę umieszcza w tej samej komórce S;
- Za K przyjmuje wartość 1 (reguluje to liczba w nawiasie 0(1) N - tzw. krok pętli), wraca do pierwszego rozkazu, sumuje wartość tkwiącą aktualnie w komórce S czyli 1, dokłada wartość A (1) czyli 5 i umieszcza wszystko w komórce S, a więc w komórce S tkwi już wartość 1 + 5 = 6 i wówczas przechodzi do następnego "obrotu".

Takich obrotów wykona maszyna N, a więc 10, gdyż K zmienia się od 0 co 1 do 9 tzn. aż wszystkie liczby bloku A umieści w komórce S. Gdyby przed pętlą nie było działania S = o, wówczas maszyna przy pierwszym obrocie pozostawiłaby w komórce S jakąś liczbę, która mogła być tam umieszczona w poprzednim liczeniu zupełnie innym programem.

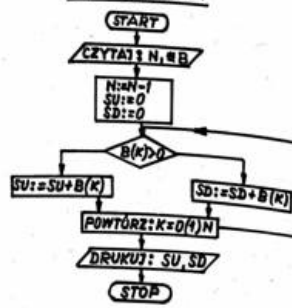
Czytelniku !

Jeżeli nie zrozumiałeś pojęcia pętli, nie czytaj dalej, tylko wróć do jej opisu ogólnego pkt. 8.2.17.

PRZYKŁAD NR. 4

Zaprogramować obliczanie sumy liczb dodatnich i ujemnych zbioru liczb o nazwie B, którego liczność jest mniejsza lub równa 1500.

SIEĆ DZIAŁAŃ




```
p:sako,program;
wy:0=pdw(dx,ddx1);
t:przyklad nr 4.
opisz program
nazwa:przyk4
translacja
rozdzial:3
calkowite:k,n
blok(1499):b
10)czytaj:n
struktura(n-1):b
czytaj:*b
n=n-1
su=0
sd=0
*1p)gdy b(k)]0:5,inaczej 3
5)sd=sd+b(k)
skocz do 77
3)su=su+b(k)
77)powtorz od 1p:(=0(1)n
linia 5
tekst:
s u m a   l i c z b   d o d a t n i c h =
drukuj(10.5):sd
linia
tekst:
s u m a   l i c z b   u j e m n y c h =
drukuj(10.5):su
linia 3
stop 10
koniec:3

dane
7
-5.2 2.5 -100.1 7125.4 -7.3
100.5314 -6.542
*
```

```
-----------------------------------------------------------------

ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST 0(C) - 4.5.72(125)/58

SUPERVISOR = SVC    PA0 = 15872    BEB = 42384

PRZYKLAD NR 4


OPIS    ROZDZIALU:      3

DLUGOSC ROZDZIALU=    929
DLUGOSC POLA ZMIENNYCH=  3007
POLE WOLNE ZACZYNA SIE   OD 2134
POLE ZMIENNYCH ZACZYNA SIE   OD12865
POLE WOLNE=    10731


    PROGRAM SAKO

NAZWA   PRZYK4
LICZBA ROZDZIALOW    1
NA BEBNIE
WOLNY BEBEN41052
DLUGOSC PROGRAMU  929
DLUGOSC SEKCJI START  403
MAX ADRES PA015872
WYMAGANY BEBEN    0


S U M A   L I C Z B   D O D A T N I C H =      +7228.43140
S U M A   L I C Z B   U J E M N Y C H =       -119.14200
```

PRZYKŁAD NR 4 - KOMENTARZ

Pod etykietą 1P wprowadzono rozkaz sterujący:

GDY B(K) > 0 : 5, INACZEJ 3

W tym miejscu komputer bada znak kolejnej liczby bloku, a następnie w zależności od wyniku umieszcza ją w komórkach przeznaczonych na liczby dodatnie lub ujemne. Badanie to odbywa się w pętli, gdyż maszyna musi zbadać każdą z liczb bloku. Pętla o zakresie od 1P do 77 spełnia tę funkcję całkowicie.

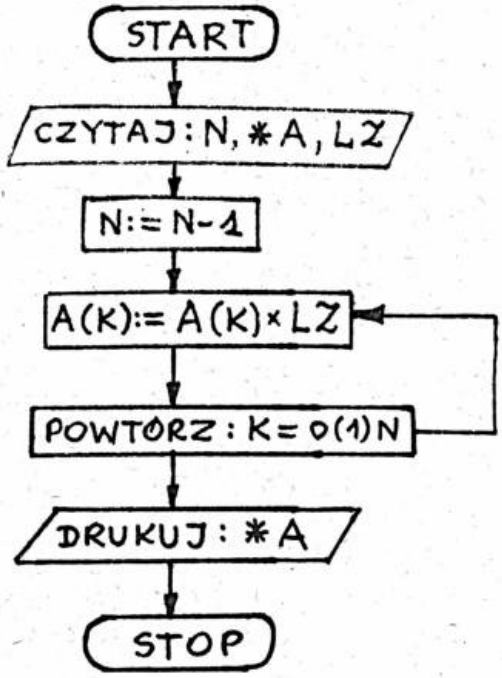
Rozkazy poniżej pętli to już wydruk wyników wraz z tekstami. Widzimy tutaj podstawową funkcję bloku decyzyjnego, za pomocą którego dokonujemy "rozesłania" liczb wg ich znaku na miejsca dla nich przygotowane.

Czytelnik winien również zwrócić uwagę na rozkaz sterujący SKOCZ DO 77 umieszczony pod rozkazem o etykiecie Nr 5. Gdyby nie było tego rozkazu maszyna po wykonaniu rozkazu 5, a więc w przypadku gdy B(K) > 0, wykonywałaby rozkaz o etykiecie Nr 3, gdyż taka jest naturalna kolejność wykonywania rozkazów w komputerze. Wówczas w komórce SU przeznaczonej dla liczb ujemnych umieściłaby również dodatnią liczbę B(K), co byłoby sprzeczne z zadaniem. Tak więc rozkaz sterujący SKOCZ DO 77 powoduje zmianę naturalnej kolejności wykonywania rozkazów. Możemy sobie uzmysłowić co oznacza popularny, choć niezbyt fortunny termin, że "program steruje sam sobą"; to po prostu rozkazy sterujące determinują kolejność wykonania rozkazów w nim samym.

PRZYKŁAD NR. 5

Napisać program realizujący mnożenie każdej liczby ze zbioru liczb o nazwie A przez zadaną wartość liczby LZ. Liczność zbioru A jest mniejsza lub równa 100.

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 5.
chowaj slowniki
post mortem:55
translacja
rozdzial:7
calkowite:n,k
blok(99):a
7)czytaj:n
n=n-1
struktura(n):a
czytaj:lz,*a
*5)a(k)=a(k)◊lz
powtorz od 5:k=0(1)n
linia 5
*5a)drukuj(10.3):a(k)
linia
powtorz od 5a:k=0(1)n
stop 7
koniec:7
```

```
dane
6
5.5
1.5 3 5.1 8.3 9.2 7.1
*
```

```
----------------------------------------------------------------
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST OCC) - 5.5.72(126)/24
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKŁAD NR 5


        +8.250
       +16.500
       +28.050
       +45.650
       +50.600
       +39.050


========
   PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOP 7 W ROZDZIALE 7 LR= 1264.
   POZOSTALO CZASU: 7999999.

========


   SLOWNIK ETYKIET.

   7  1205     5  1220     SA  1242


   POST MORTEM.


   A 15672 +8.250     +1.650E+1 +2.805E+1 +4.565E+1
           +5.060E+1 +3.905E+1 +0.000     +0.000     +0.000
           +0.000     +0.000     +0.000     +0.000     +0.000
           +0.000
   N 15671          5      LZ 15668 +5.500
   K 15667          5
========
```

PRZYKŁAD NR 5 - KOMENTARZ

Treścią programu jest przemnożenie każdej liczby bloku przez dowolną liczbę IZ.

Zwracamy szczególną uwagę na fakt, że nie został w ogóle utworzony nowy blok wynikowy, w którego komórkach znajdowałyby się elementy bloku już przemnożone przez podaną liczbę.

Otóż jest to możliwe dzięki zasadzie podstawiania opisanej w pkt.8.2.16.

Rozkaz nr 5 powoduje przemnożenie zawartości komórki A (K) przez liczbę IZ i umieszczenie nowej już wartości w tej samej komórce (pod tym samym adresem). Program wykonuje tę czynność w pętli, a więc każdy z wyrazów bloku jest przemnożony i wstawiony na powrót w tę samą komórkę, w której tkwił poprzednio.

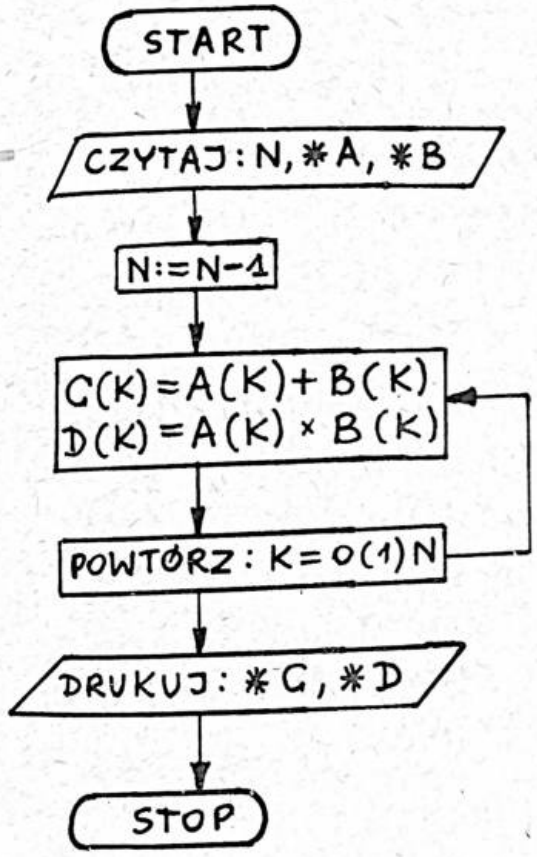
Drukowanie wyników odbywa się również za pomocą pętli. Dla K zmieniającego się od 0 do N (wartość N została wyżej zmniejszona o jeden) wydrukowuje się element bloku A (K).

Rozkaz LINIA powoduje przesunięcie papieru drukarki o jeden wiersz po każdym wydrukowaniu jednego elementu. Tak więc wydruk wynikowego bloku liczb następuje w postaci kolumny.

PRZYKŁAD NR. 6

Napisać program realizujący dodawanie i mnożenie odpowiadających sobie elementów zbioru liczb A i B, których liczność jest mniejsza lub równa 1000. Wyniki tworzą zbiory liczbowe C i D.

Sieć działań



```
p:sako,program;
wy:0=pdw(dx,ddx1);
t:przyklad nr 6.
opisz program
nazwa:przyk6
translacja
rozdzial:0
calkowite:n,k
blok(999):a,b,c,d
11)czytaj:n
n=n-1
struktura(n):a,b
czytaj:*a,*b
strona
linia 3
*1)c(k)=a(k)+b(k)
d(k)=a(k)◊b(k)
drukuj(10.3):c(k),d(k)
linia
powtorz od 1:k=0(1)n
stop 11
koniec:0
```

```
dane
4
1.7 21.3 71.5 31.5
*
2.5 3.1 100.1 7.37
*
```

```
-----------------------------------------------------------------
ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST 0(C) - 4,5,72(125)/63
SUPERVISOR = SVC   PAO = 15872   BEB = 42384

PRZYKLAD NR 6


OPIS     ROZDZIALU:        0

DLUGOŚĆ ROZDZIALU=   885
DLUGOŚĆ POLA ZMIENNYCH=   8005
POLE WOLNE ZACZYNA SIE   OD 2090
POLE ZMIENNYCH ZACZYNA SIE   OD 7867
POLE WOLNE=      5777


    PROGRAM SAKO

NAZWA   PRZYK6
LICZBA ROZDZIAŁOW     1
NA BĘBNIE
WOLNY BĘBEN41096
DLUGOŚĆ PROGRAMU  885
DLUGOŚĆ SEKCJI START  403
MAX ADRES PAO15872
WYMAGANY BEBEN     0


        +4.200          +4.250
       +24.400         +66.030
      +171.600       +7157.150
       +38.870        +232.155
```

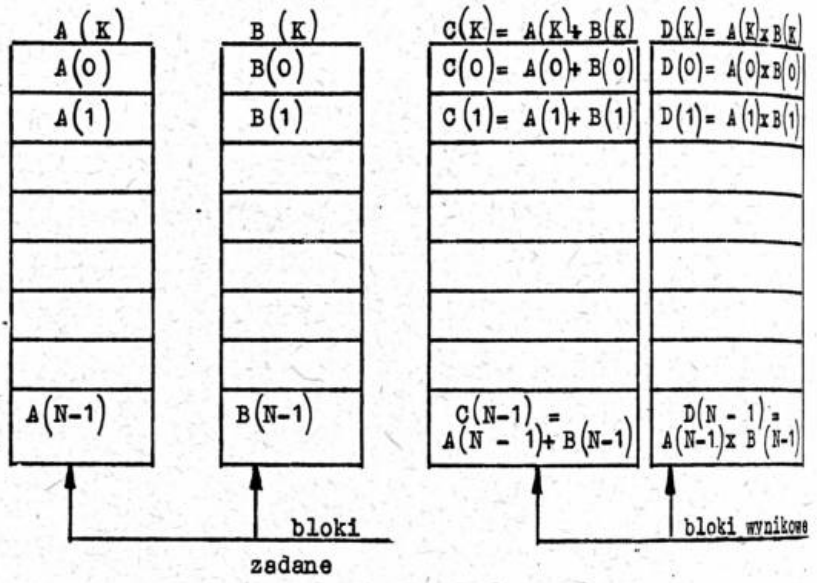
PRZYKŁAD NR 6 - KOMENTARZ

Program pokazuje rozszerzone zastosowanie pętli. Czytelnik zapoznał się już z pojęciem zakresu pętli, to jest cyklem rozkazów, który znajduje się pomiędzy początkiem pętli (etykieta z gwiazdką) a jej końcem, to znaczy rozkazem rozpoczynającym się od słowa POWTORZ ...

Otóż w zakresie pętli może znajdować się dowolna liczba rozkazów a niekiedy prawie cały program, co zdarza się najczęściej w programach tablicujących.

Wydruk bloków wynikowych powstaje tutaj w dwu kolumnach również przy użyciu pętli.

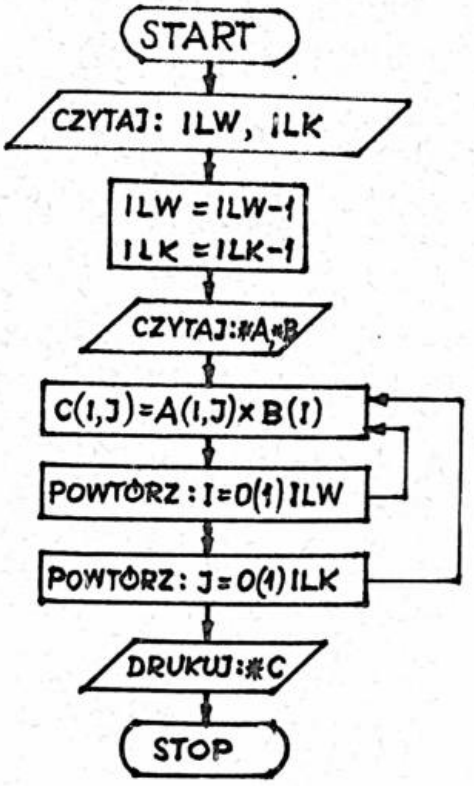
Dla ułatwienia podaje się również graficzny obraz idei dodawania i mnożenia dwu bloków, to znaczy taką postać w jakiej początkujący programista może sobie wyobrażać bloki liczbowe.



PRZYKŁAD NR. 7

Dana jest macierz A o maksymalnych wymiarach 50×50 i wektor B, którego wymiar jest zgodny z ilością wierszy macierzy A.
Napisać program realizujący mnożenie odpowiadających sobie elementów każdej kolumny macierzy i wektora B.
W wyniku powstaje macierz C.

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 7.
chowaj slowniki
post mortem:60
nazwa:przyk7
translacja
rozdzial:0
calkowite:i,j,ilw,ilk
blok(49,49):a,c
blok(49):b
1ab)czytaj:ilw,ilk
ilw=ilw-1
ilk=ilk-1
struktura(ilw,ilk):a
struktura(ilw):b
czytaj:*a,*b
**1)c(i,j)=a(i,j)*b(i)
powtorz od 1:i=0(1)ilw
powtorz od 1:j=0(1)ilk
strona
linia 3
**2)rob=c(i,j)
drukuj(10.3):rob
linia
powtorz od 2:i=0(1)ilw
linia 2
powtorz od 2:j=0(1)ilk
stop 1ab
koniec:0


dane
3
4
10 15.2 3.12 21.27
1.5 4.3 1 51.3
2.3 1.3 7.1 153.2
*
5 3 7.1
*
```

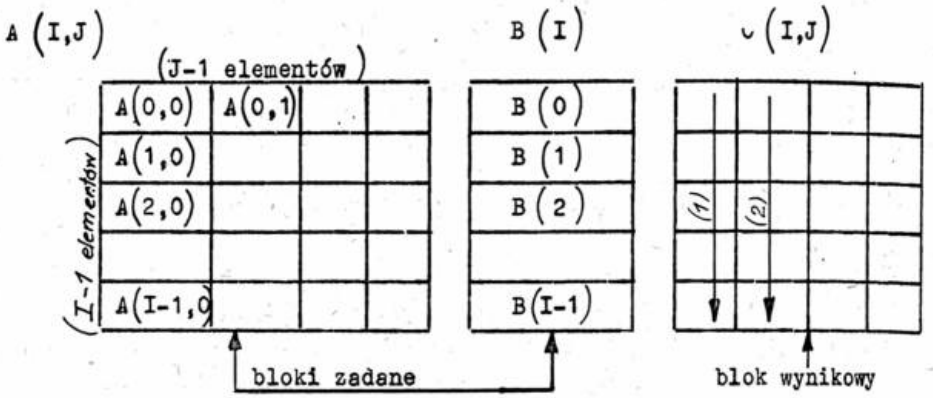
```
----------------------------------------------------------------------
ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST O(C) - 4,5,72(125)/64
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKLAD NR 7.

      +50.000
       +4.500
      +16.330


      +76.000
      +12.900
       +9.230


      +15.600
       +3.000
      +50.410


     +106.350
     +153.900
    +1087.720
```

```
========
   PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOP 1AB W ROZDZIALE 0 LR= 1322.
   POZOSTALO CZASU: 7999999.

========


   SLOWNIK ETYKIET.

  1AB  1205     1  1247     2  1284


   POST MORTEM.


   A 10872 +1.000E+1 +1.520E+1 +3.120     +2.127E+1
           +1.500    +4.300    +1.000     +5.130E+1 +2.300    +1.300
           +7.100    +1.532E+2 +0.000     +0.000    +0.000
   C  5872 +5.000E+1 +7.600E+1 +1.560E+1  +1.064E+2
           +0.000    +0.000    +0.000     +0.000    +0.000    +0.000
           +0.000    +0.000    +0.000     +0.000    +0.000
   B  5772 +5.000    +3.000    +7.100     +0.000
           +0.000    +0.000    +0.000     +0.000    +0.000    +0.000
           +0.000    +0.000    +0.000     +0.000    +0.000
  ILW  5771        2    ILK  5770        3
    J  5765        3      I  5764        2
  ROB  5760 +1.088E+3
========
```

PRZYKŁAD NR 7 - KOMENTARZ

W przykładzie podajemy zastosowanie pętli dwukrotnej. Za pomocą zmiany indeksów I i J i kolejnych "obrotów" maszyna dokonuje przemnożenia odpowiednich wyrazów bloków liczbowych przez siebie.

Poniżej podajemy graficzny obraz idei tego działania:



Kolejność obliczania poszczególnych wyrazów bloku wynikowego

C (I,J) w naszym przypadku będzie:

C (0,0), C(1,0), C(2,0) ... C (I-1,0)

C (0,1), C(1,1), C(2,1) ... C (I-1,1)

.....

(patrz rysunek powyżej)

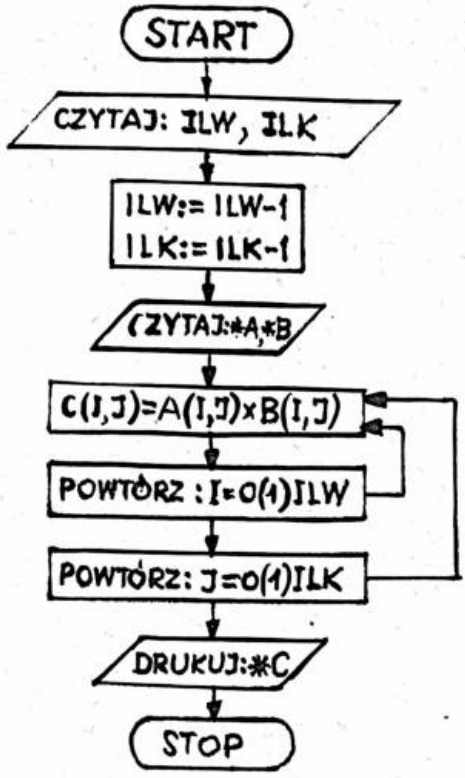
To znaczy, że pętla podwójna działa wg następujących reguł:

- komputer przyjmuje wartość J = 1 (zewnętrzny rozkaz powtórz);
- dla tej wartości J zmieni wartość I od 0 co 1 do ILW dokonując za każdym razem obliczenia C (I,J), a więc dokonując obliczeń pierwszej kolumny;
- komputer przyjmuje wartość J = J + 1 tzn. powiększa wartość J o 1 (wg rozkazu zewnętrznego);
- dla tej wartości J = 1 zmieni wartość I od 0 co 1 do ILW, a więc obliczy 2-gą kolumnę itd.

PRZYKŁAD NR. 8

Dane są dwie macierze A i B o takich samych wymiarach (maksymalny wymiar 40x40)
Napisać program realizujący mnożenie odpowiadających sobie elementów tych macierzy.
Macierz C będzie wynikiem.

Sieć działań



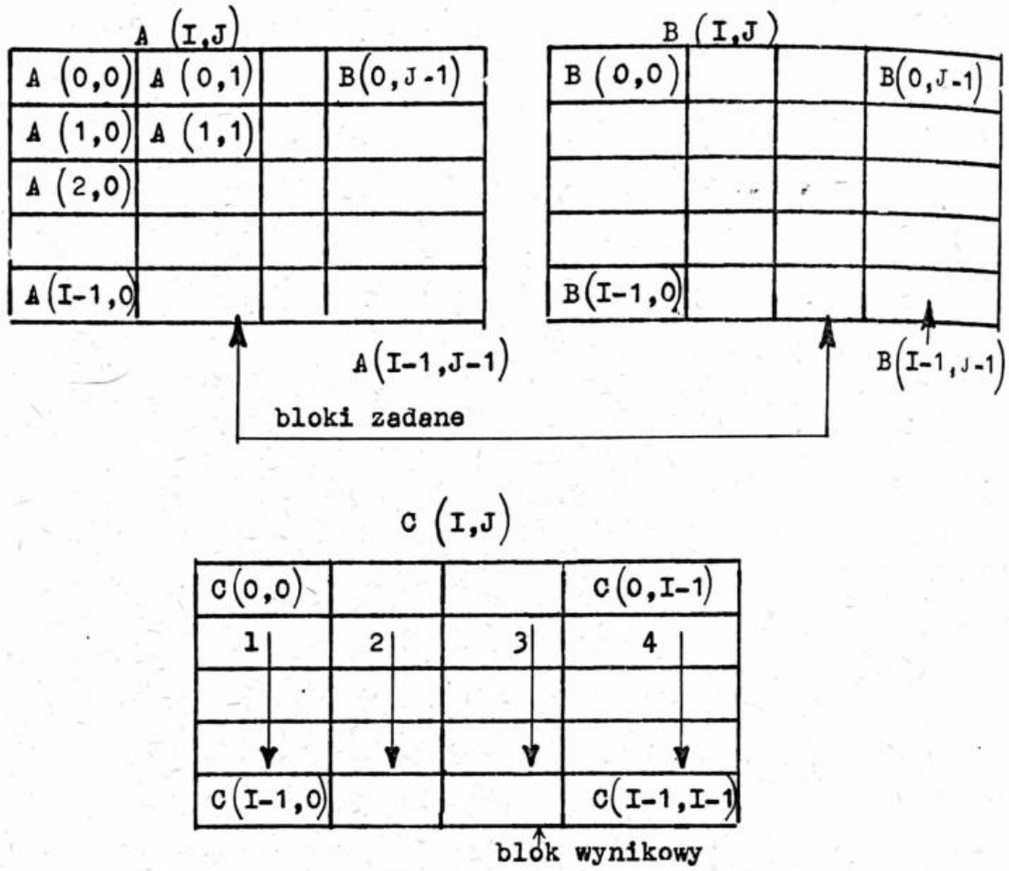
```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 8.
opisz program
chowaj slowniki
post mortem:60
nazwa:przyk8
translacja
rozdzial:333
calkowite:ilw,ilk,i,j
blok(40,40):a,b,c
blok(0):ilw,ilk
7a)czytaj:ilw,ilk
ilw=ilw-1
ilk=ilk-1
struktura(ilw,ilk):a,b
czytaj:*a,*b
**2)c(i,j)=a(i,j)*b(i,j)
powtorz od 2:i=0(1)ilw
powtorz od 2:j=0(1)ilk
linia
**33)rob=c(i,j)
drukuj(7.5):rob
linia
powtorz od 33:i=0(1)ilw
linia 2
powtorz od 33:j=0(1)ilk
stop 7a
koniec:333


dane
3
2
1.1 10.1
3 2.1
10.75 3.1
*
2.1 4.7
101.33 1.2
2.1 3.5
*
```

```
----------------------------------------------------------------
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST 0(C) - 4.5.72(125)/29
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKLAD NR 8
OPIS      ROZDZIALU:     333
DLUGOSC ROZDZIALU=     932
DLUGOSC POLA ZMIENNYCH= 10098
POLE WOLNE ZACZYNA SIE    OD 2137
POLE ZMIENNYCH ZACZYNA SIE    OD 5774
POLE WOLNE=      3637


   PROGRAM SAKO

NAZWA    PRZYK8
LICZBA ROZDZIALOW     2
NA BEBNIE
WOLNY BEBEN40977
DLUGOSC PROGRAMU 1004
DLUGOSC SEKCJI START   403
MAX ADRES PA015872
WYMAGANY BEBEN     0

    +2.31000
   +303.99000
   +22.57500


   +47.47000
    +2.52000
   +10.85000


========
   PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOP ZA W ROZDZIALE 333 LR= 1323.
   POZOSTALO CZASU: 7999999.

========


   SLOWNIK ETYKIET.

  7A  1205     2  1247     33  1285


   POST MORTEM.


   A 12510 +1.100     +1.010E+1 +3.000     +2.100
           +1.075E+1 +3.100     +0.000     +0.000     +0.000     +0.000
           +0.000     +0.000     +0.000     +0.000     +0.000
   B  9148 +2.100     +4.700     +1.013E+2 +1.200
           +2.100     +3.500     +0.000     +0.000     +0.000     +0.000
           +0.000     +0.000     +0.000     +0.000     +0.000
   C  5786 +2.310     +4.747E+1 +0.000     +0.000
           +0.000     +0.000     +0.000     +0.000     +0.000     +0.000
           +0.000     +0.000     +0.000     +0.000     +0.000
 ILW  5785          2     ILK  5784          1
   J  5779          1       I  5778          2
 ROB  5774 +1.085E+1
========
```

PRZYKŁAD NR 8 - KOMENTARZ

W niniejszym przykładzie podwójna pętla obejmuje przemnażanie elementów dwu bloków dwuwymiarowych.

Poniżej podajemy graficzny obraz idei tego zadania.



Zasada obliczania nie zmienia się w porównaniu z poprzednim zadaniem.

Dla każdej zmiany J nastąpi ILW zmian I. W związku z tym obliczania poszczególnych elementów bloku nastąpi kolumnami w kolejności jak niżej.

Obliczanie elementów bloku "C" mogłoby również następować wierszami, ale wówczas odpowiedni fragment programu wyglądałby jak następuje:

```
C(I,J) = A(I,J) ◊ B(I,J)
POWTORZ : J = 0 (1) I L K
POWTORZ : I = 0 (1) I L W
```

to znaczy dla zmian I o jeden nastąpi ILK+1 zmian J.

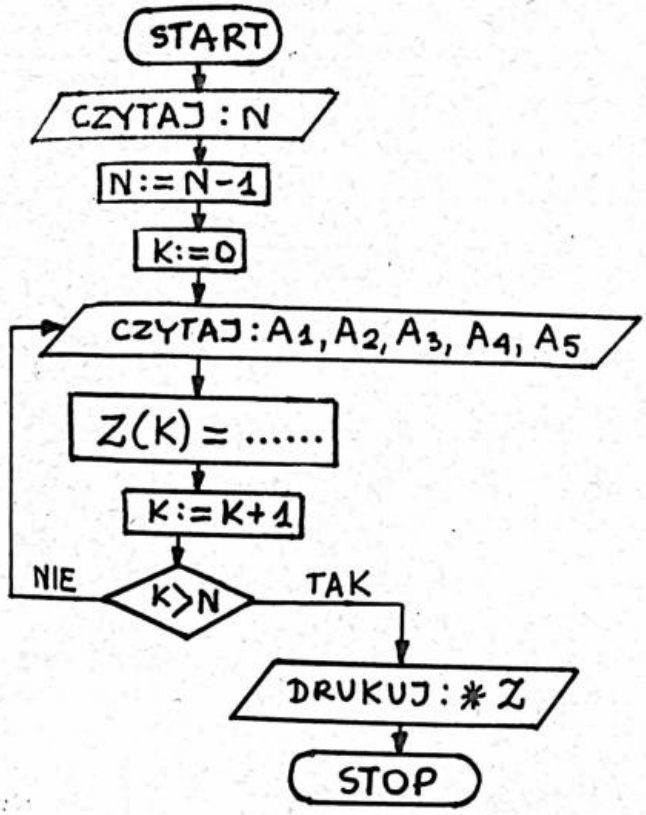
Pętle mogą być również n-krotne, a zmiany indeksów będą rozpoczynały się zawsze od rozkazu zewnętrznego.

PRZYKŁAD NR. 9

Napisać program realizujący obliczenia N wartości Z na podstawie danych: A_1, A_2, A_3, A_4, A_5 z następującej zależności:

$$Z = \sqrt[3]{\left(\frac{A_1 \cdot A_2}{A_4}\right)^{A_3} \sqrt{\left|\frac{A_1^{A_5}}{A_3}\right|^{A_4}}}$$

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 9.
chowaj slowniki
post mortem:60
translacja
rozdzial:0
calkowite:k,n,*tyt
blok(0):n,a1,a2,a3,a4,a5
blok(999):z
blok(119):tyt
1st)czytaj:n
n=n-1
*5)czytaj:a1,a2,a3,a4,a5
z(k)=pwk(abs(exp(a5◊ln(a1))/a3)◊a4)
z(k)=z(k)◊exp(a3◊ln(a1◊a2/a4))
z(k)=pws(z(k))
powtorz od 5:k=0(1)n
strona
linia 5
czytaj wiersz:tyt
drukuj wiersz:tyt
linia 2
*1)gdy mod(k,3)=0:nastepny,inaczej 3b
linia
3b)drukuj(7.5):z(k)
powtorz od 1:k=0(1)n
stop 1st
koniec:0

dane
4
10
11.1
12
15.5
20
2.5
3.1
15.8
1
2.5
7
8.4
10
12
3.1
7.3
5.2
2.4
1.5
4
z b i o r   w a r t o s c i   z
```

```
----------------------------------------------------------------
ZAM 41 - SO 141 BCP(Q) 213/203 - ZEST O(C) - 4.5,72(125)/65
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKLAD NR 9


Z B I O R   W A R T O S C I   Z

+5913213.11047   +44636.25901     +562,96898
   +46.14414


========
   PRZYCZYNA ZAKOŃCZENIA PROGRAMU SAKO  STOP 1ST W ROZDZIALE 0 LR= 1308.
   POZOSTALO CZASU: 7999999.

========

   SLOWNIK ETYKIET.

 1ST  1205     5  1211     1  1278    38  1290


   POST MORTEM.

   N 15871          3      A1 15869 +7.300
  A2 15867 +5.200          A3 15865 +2.400
  A4 15863 +1.500          A5 15861 +4.000
   Z 13861 +5.915E+6 +4.464E+4 +5.630E+2 +4.614E+1
           +0.000    +0.000    +0.000    +0.000    +0.000    +0.000
           +0.000    +0.000    +0.000    +0.000    +0.000
  TYT 13741          46         1         22        1
                     29         1         55        1         38
                      1         1         43        1         21
   K 13740          3
========
```

PRZYKŁAD NR 9 - KOMENTARZ

W niniejszym przykładzie na skutek N-krotnego wczytywania grup liczbowych A1, ... A5 powstanie blok Z składający się z N wartości wynikających z kolejnego obliczenia złożonego wyrażenia arytmetycznego. Sieć działań nie powinna już budzić wątpliwości.

Kolejność wykonywania poszczególnych operacji jest następująca:

- wczytanie pierwszej grupy wartości A1, ..., A5;
- powiększenie wskaźnika K o 1;
- obliczenie wartości Z(K);
- zbadanie, czy wszystkie zbiory zostały wczytane;
- jeżeli NIE, wówczas następuje czytanie następnego zbioru;
- jeżeli TAK, wówczas następuje przejście do wydruku.

Program ten ma na celu pokazanie różnicy form zapisu, do których przywykliśmy i zapisu w autokodzie.

Czytelnik winien zwrócić uwagę na podstawowy rozkaz opisany w pkt.8.2.16. Jeden wzór jest rozbity na kilka części za pomocą zapisu jak na przykładzie niżej

$$Z = (A \cdot B + C) / D \quad \text{co można zapisać}$$

$$Z = A \cdot B$$
$$Z = Z + C$$
$$Z = Z / D$$

Najpierw realizowane są wyrażenia z prawej strony znaku równości. W każdym kolejnym wyrażeniu następuje operacja na poprzedniej wartości Z i po wykonaniu umieszczenie wyniku ponownie w tej samej komórce.

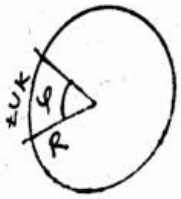
Tak rozdzielamy nawet bardzo długie i skomplikowane wyrażenia. Poza tym przypominamy, że zapisanie wyrażenia

A^B odbywa się za pomocą wzoru

EXP (B ◊ LN(A))

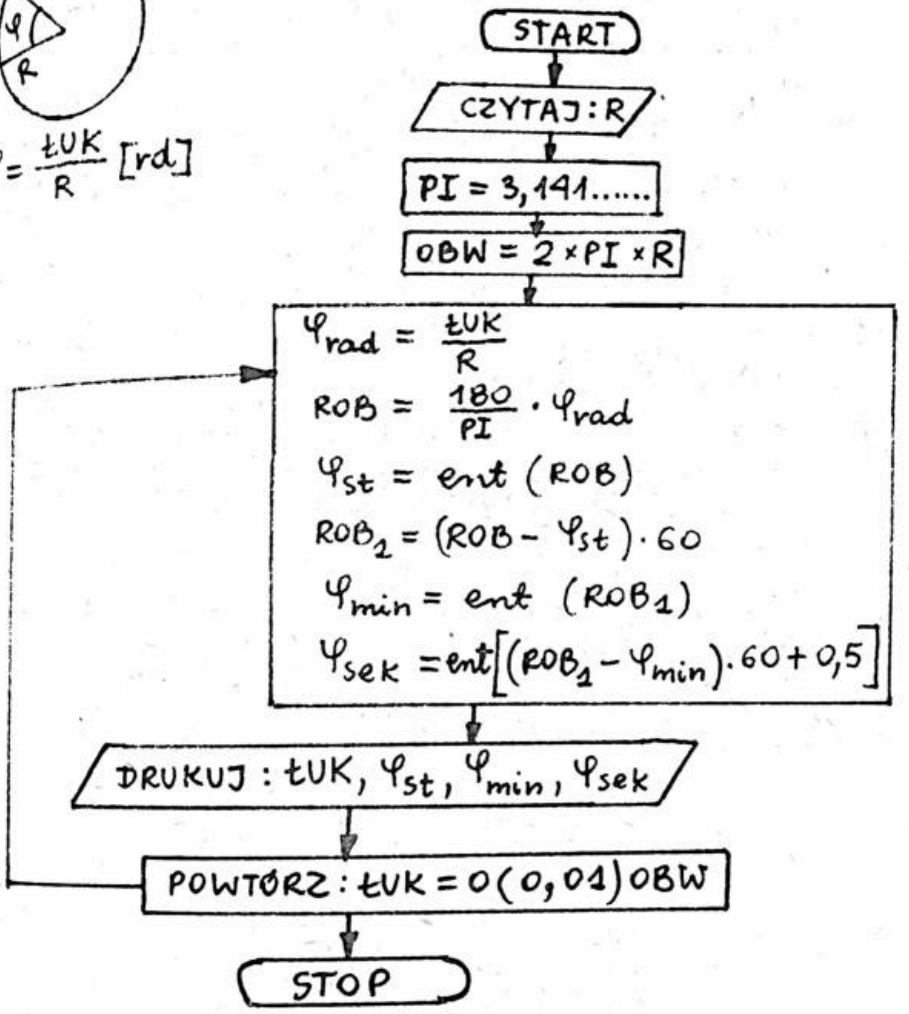
PRZYKŁAD NR. 10

Stablicować wartości kąta (w stopniach, minutach i sekundach) w zależności od długości łuku na okręgu o promieniu R [m] przyjmując skok tablicowania równy 1 m.



$$\varphi = \frac{ŁUK}{R} \; [rd]$$

Sieć działań



```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 10.
chowaj slowniki
post mortem:59
translacja
rozdzial:123
calkowite:alfa,beta,gama
blok(0):r,pi,obw,fi,luk
blok(0):alfa,beta,gama
100)strona
linia 2
tekst wierszy 1:
t a b l i c o w a n i e  f u n k c j i  k a t =f(luku)
linia 3
pi=3.141592654
czytaj:r
obw=2◊pi◊r
tekst wierszy 1:
        luk[m]     stopnie    minuty     sekundy
*1)fi=luk/r
(alfa,beta,gama)=rad(fi)
drukuj(10.2):luk
drukuj(10):alfa,beta,gama
linia
powtorz od 1:luk=0.(1.)obw
stop 100
podprogram:(st,min,sek)=rad(beta)
calkowite:st,min,sek
btst=(180/3.141592654)◊beta
st=ent(btst)
btm=(btst-st)◊60
min=ent(btm)
sek=ent((btm-min)◊60+0.5)
wroc
koniec:123
```

```
dane
2.5
```

```
----------------------------------------------------------------
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST 0(0) - 5.5.72(126)/25
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKLAD NR 10
T A B L I C O W A N I E   F U N K C J I   K A T =F(LUKU)

      LUK(M)      STOPNIE     MINUTY      SEKUNDY
      +0.00          0           0           0
      +1.00         22          55           6
      +2.00         45          50          12
      +3.00         68          45          18
      +4.00         91          40          24
      +5.00        114          35          30
      +6.00        137          30          36
      +7.00        160          25          41
      +8.00        183          20          47
      +9.00        206          15          53
     +10.00        229          10          59
     +11.00        252           6           5
     +12.00        275           1          11
     +13.00        297          56          17
     +14.00        320          51          23
     +15.00        343          46          29
     +16.00        366          41          35


=======
   PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOP 100 W ROZDZIALE 123 LR= 1302.
   POZOSTALO CZASU: 7999998.

=======


   SLOWNIK ETYKIET.

 100  1205      1   1252
PODPROGRAM *RAD   1303


   POST MORTEM.


   R 15870 +2.500          PI 15868 +3.142
 OBW 15866 +1.571E+1       FI 15864 +6.400
 LUK 15862 +1.600E+1    ALFA 15861       366
BETA 15860        41    GAMA 15859        35

PODPROGRAM *RAD
BETA  1305  26214 6710886  +6.400
  ST  1307    366    366
 MIN  1309     41     41
 SEK  1311     35     35
BTST 15857 +3.667E+2    BTM 15852 +4.158E+1
=======
```

PRZYKŁAD NR 10 - KOMENTARZ

W przykładzie tym zastosowano po raz pierwszy podprogram. Jest on wywoływany tyle razy, ile jest "obrotów" w pętli.

Rozkaz wywołania (uwaga na formę zapisu) znajduje się pod rozkazem o etykiecie NR 1. Z lewej strony formuły operacyjnej znajduje się lista wyników, które zostaną w tym programie obliczone, z prawej zaś strony nazwa podprogramu /RAD/ i lista argumentów.

W tym miejscu Czytelnik winien uświadomić sobie w pełni formę zapisu i sposób działania podprogramu (pkt. 8.2.11).

Cykl jego rozkazów oblicza ST, MIN, SEK w zależności od wartości BETA.

W formule operacyjnej symbole ALFA, BETA, GAMMA oznaczają komórki, w których zostaną umieszczone wartości ST, MIN, SEK. Zmienna BETA w podprogramie będzie przybierać wartości FI nadawane w formule operacyjnej.

Mówiąc schematycznie formuła operacyjna powoduje w komputerze:

- skok do podprogramu
- obliczenie wartości ST, MIN, SEK
- powrót do miejsca w programie, z którego nastąpił wyskok i nadanie wartości ST dla ALFA, MIN dla BETA i SEK dla GAMMA.

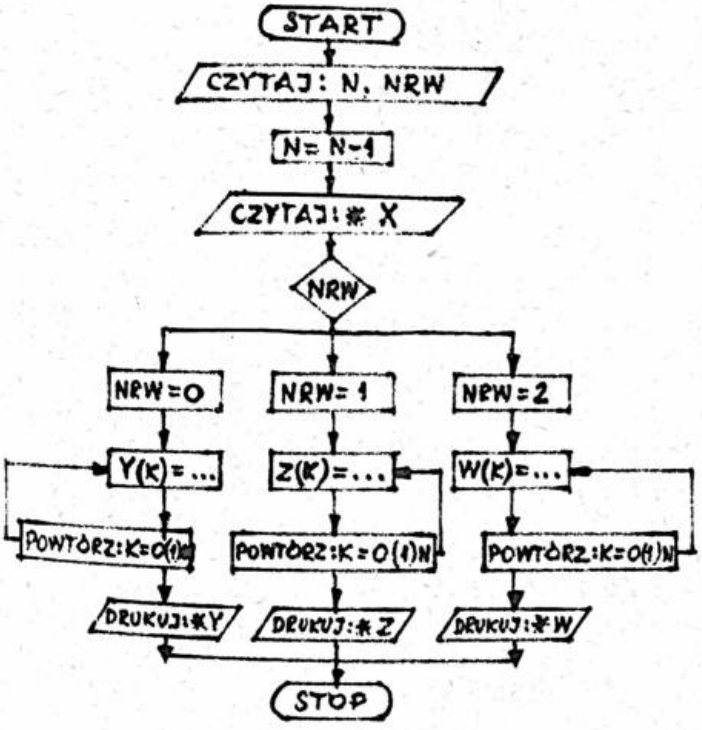
PRZYKŁAD NR. 11

Napisać program dla wyłącznego tablicowania trzech funkcji:

$$y = x^5 - 4x^3$$
$$z = 2x^8 - 16x^2 - 3$$
$$w = 16x^7 - 4x^3 + 28$$

Zbiór wartości x dla których przeprowadza się tablicowanie składa się maksymalnie ze 100. liczb.

Sieć działań



```
p:sako,program,program,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 11.
translacja
rozdzial:0
calkowite:nrw,k,n
blok(99):x,y,z,w
blok(0):nrw,k,n
1)czytaj:n
n=n-1
struktura(n):x
2)czytaj:*x
21)czytaj:nrw
*3)skocz wedlug nrw:4,5,6
4)y(k)=(x(k)*5)-40(x(k)*3)
skocz do 7
5)z(k)=20(x(k)*8)-160x(k)0x(k)-3
skocz do 7
6)w(k)=160(x(k)*7)-40(x(k)*3)+28
7)powtorz od 3:k=0(1)n
linia 2
8)skocz wedlug nrw:9,10,11
9)tekst wierszy 2:
f u n k c j a  y
----------------
linia 2
*91)drukuj(8.3):x(k),y(k)
linia
powtorz od 91:k=0(1)n
stop 1
10)tekst wierszy 2:
f u n k c j a  z
----------------
linia 2
*101)drukuj(8.3):x(k),z(k)
linia
powtorz od 101:k=0(1)n
stop 1
11)tekst wierszy 2:
f u n k c j a  w
----------------
linia 2
*111)drukuj(8.3):x(k),w(k)
linia
powtorz od 111:k=0(1)n
stop 1
koniec:0
dane
10
1 3 5 7 8 9 10 11 6 20
*
0
10
1 3 5 7 8 9 10 11 6 20
*
1
10
1 3 5 7 8 9 10 11 6 20
*
2
```

```
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST 0(C) - 5.5.72(126)/26
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKŁAD NR 11

F U N K C J A   Y

      +1.000          -3.000
      +3.000        +135.000
      +5.000       +2625.000
      +7.000      +15435.000
      +8.000      +30720.000
      +9.000      +56133.000
     +10.000      +96000.000
     +11.000     +155727.000
      +6.000       +6912.000
     +20.000    +3168000.000

F U N K C J A   Z

      +1.000         -17.000
      +3.000      +12975.000
      +5.000     +780847.000
      +7.000+11528815.000
      +8.000+33553405.000
      +9.000+86092143.000
     +10.000+19999839.700E+1
     +11.000+42871582.300E+1
      +6.000 +3358653.000
     +20.000+51199993.597E+3

F U N K C J A   W

      +1.000         +40.000
      +3.000      +34912.000
      +5.000    +1249528.000
      +7.000+13175344.000
      +8.000+33552412.000
      +9.000+76524616.000
     +10.000+15999602.800E+1
     +11.000+31178944.000E+1
      +6.000 +4478140.000
     +20.000+20479968.028E+3


PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO 63 LR= 1421.
POZOSTAŁO CZASU: 7999998.
```

PRZYKŁAD NR 11 - KOMENTARZ

W programie pokazano użycie rozkazu SKOCZ WEDŁUG. Jest on użyty dwukrotnie. W zależności od wartości NRW jest realizowana odpowiednia ścieżka logiczna programu. Ta wartość w rozkazie sterującym kieruje obliczenie za pierwszym razem do rozkazu o etykiecie 4 lub 5 względnie 6, a za drugim razem do etykiety 9 lub 10, względnie 11.

Widzimy tutaj również rolę tzw. liczb sterujących, w naszym przypadku NRW, za pomocą których program "pracuje" na różnych ścieżkach logicznych.

Oczywiście zmienna NRW musi posiadać określoną wartość jeszcze przed rozkazem sterującym.

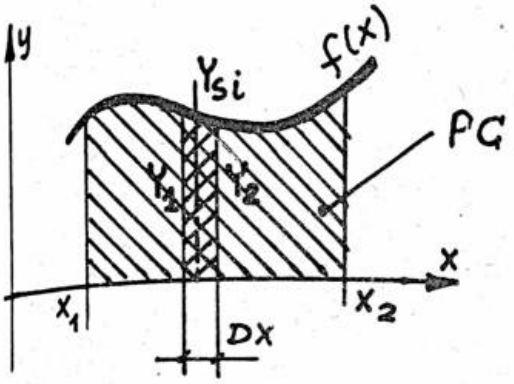
Dla NRW = 0	maszyna	przejdzie	do rozkazu	o numerze	4
" NRW = 1	"	"	"	"	5
" NRW = 2	"	"	"	"	6

a później

dla NRW = 0	"	"	"	"	9
" NRW = 1	"	"	"	"	10
" NRW = 2	"	"	"	"	11

PRZYKŁAD NR. 12

Zaprogramować całkowanie funkcji $f(x) = x^{10} - \frac{1}{2}x^3$ w przedziale (x_1, x_2) oraz określić odległość środka ciężkości pola PC od osi x.



$$DX = \frac{X_2 - X_1}{IP}$$

$$Y_{si} = \frac{Y_1 + Y_2}{2}$$

$$PCP_i = DX \cdot Y_{si}$$

$$PC = \sum_{i=1}^{IP} PCP_i$$

$$MS = \sum_{i=1}^{IP} PCP_i \left(Y_{si}/2\right)$$ moment statyczny

$$PSC = \frac{MS}{PC}$$ środek ciężkości

Tym razem pozostawiamy napisanie sieci działań Czytelnikowi.

```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przyklad nr 12.
chowaj slowniki
post mortem:62
nazwa:przy12
translacja
rozdzial:0
calkowite:k,ip
blok(0):x1,x2,ip,k,n,dx,y1,y2,ys
blok(0):pc,xr1,xr2,pcp,msp,ms,psc
1)czytaj:x2,x1,ip
k)obliczenie pola
2)dx=(x2-x1)/ip
pc=0
k=1
ms=0
3)xr1=x1
4)y1=xr1*10-0.50xr1*3
xr2=xr1+dx
5)y2=xr2*10-0.50xr2*3
6)ys=(y1+y2)/2
pcp=ys◊dx
msp=pcp◊(ys/2)
7)pc=pc+pcp
ms=ms+msp
k=k+1
y1=y2
8)xr2=k◊dx+xr1
gdy xr2]x2:10,inaczej 5
k)obliczanie srodka ciezkosci pola
10)psc=ms/pc
linia 3
tekst:
pole calkowite=
drukuj(10.5):pc
linia 2
tekst:
srodek ciezkosci=
drukuj(10.5):psc
stop 1
koniec:0

dane
5
1
100
```

```
ZAM 41 - SO 141 BCP(A) 233/573 - ZEST 0(C) - 5.5.72(126)/22
SUPERVISOR = SVC    PAO = 15872    BEB = 42384

PRZYKŁAD NR 12



POLE CAŁKOWITE=    +4441446.32395

SRODEK CIEZKOSCI=    +2557365.58507



========
   PRZYCZYNA ZAKONCZENIA PROGRAMU SAKO  STOp 1 W ROZDZIALE 0 LR= 1329.
   POZOSTAŁO CZASU: 7999998.

========


    SŁOWNIK ETYKIET.

    1  1205     2  1214     3  1229     4  1231     5  1246
    6  1258     7  1269     8  1280    10  1290

    POST MORTEM.

   X1 15870 +1.000          X2 15868 +5.000
   IP 15867       100        K 15866       101
    N 15864 +0.000         DX 15862 +0.400E-1
   Y1 15860 +9.766E+6      Y2 15858 +9.766E+6
   YS 15856 +9.389E+6      PC 15854 +4.441E+6
  XR1 15852 +1.000        XR2 15850 +5.040
  PCP 15848 +3.755E+5     MSP 15846 +1.763E+12
   MS 15844 +1.136E+13    PSC 15842 +2.557E+6
========
```

PRZYKŁAD NR 12 - KOMENTARZ

Program ma na celu pokazanie prostoty całkowania funkcji, nawet w przypadku gdy nie znamy wzoru skończonego całki. Wystarczy stwierdzić jedynie jej ciągłość w przedziale nas interesującym i metodą sumowania pól określamy w pętli jej wartość.

Należy zwrócić uwagę, że dokładność tego obliczenia zależy od przyjętej wielkości DX, którą można obrać dowolnie małą regulując tzw. krok pętli.

To samo odnosi się do obliczania środków ciężkości, momentów bezwładności tych pól itd.

Przy dużej szybkości operacyjnej współczesnych komputerów liczba obrotów pętli nie odgrywa zasadniczej roli.

PRZYKŁAD NR 13 - KOMENTARZ

Przykład ten nie ma sprecyzowanego tematu.
Zastosowano w nim wszystkie deklaracje, rozkazy oraz formuły bulowskie, które nie występowały w dotychczasowych przykładach, w celu pokazania ich działania (patrz wyniki z maszyny) i zapisu. Czytelniku ! jeśli prześledzisz ten program i zrozumiesz go oznacza to, że opanowałeś b. dobrze zasady programowania w języku SAKO.

```
p:sako,program,zako;
wy:0=pdw(dx,ddx1);
t:przykład nr 13.
chowaj słowniki
post mortem:55
translacja
rozdział:0
całkowite:rob
całkowite:*b,c,d,e,f,g,j,h
tablica(4):wsp1
1.2 1.5 3 5.1 2
*
tablica oktalna(4):b
00000003 00000004 00000005 12300456 00000777
*
blok(4):zb
blok(0):c,d,e,f,g,h,j
1)linia 2
czytaj oktalnie:c
czytaj:*zb
k)negacja
d[-c
k)suma bulowska
e[c+d
k)przesuniecie cykliczne w prawo
f[c*(-2)
k)przesuniecie cykliczne w lewo
g[c*2
k)iloczyn bulowski
h[c◊e
*)drukuj oktalnie:b(j)
spacja 2
powtorz:j=0(1)4
linia 2
*)zb(j)=zb(j)◊wsp1(j)
powtorz:j=0(1)4
drukuj oktalnie:c
spacja
drukuj oktalnie:d
spacja
drukuj oktalnie:e
spacja
drukuj oktalnie:f
spacja
drukuj oktalnie:g
spacja
drukuj oktalnie:h
linia 2
*5)drukuj(5,3):zb(j)
spacja 3
drukuj(1,5,6):zb(j)
linia
powtorz od 5:j=0(1)4
pisz na beben od 0:*zb
idz do rozdziału:2
koniec rozdziału
```

```
rozdzial:2
calkowite:*b,i,adr,j,*e
blok(4):wsp1
blok(4):b
blok(4):zb
blok(3,5):e
zb(=i)=0
e(=i,=j)=0
e(=i,0)=4
e(=i,5)=6
e(2,=j)=3
czytaj z bebna od 0:*zb
linia
*)drukuj(4,5,6):zb(i)
spacja
powtorz:i=0(1)4
linia
**5)drukuj oktalnie:e(i,j)
spacja 2
powtorz od 5:i=0(1)3
linia
powtorz od 5:j=0(1)5
stop nastepny
idz do rozdzialu:0
koniec:0


43576102
3 5 7 8 9
*
```

```
----------------------------------------------------------------

ZAM 41 - SO 141 BCJ(Q) 237/392 - ZEST 0(C) - 10.5.72(131)/1

SUPERVISOR = SVC   PAO = 15872   BEB = 42384


PRZYKLAD NR 13

00.00.00.03  00.00.00.04  00.00.00.05  12.30 04.56  00.00.07.77

43.57.61.02 34.20.16.75 77.77.77.77 50.73.74.20 16.77.04.12 43.57.61.02

    +3.600    +3.60000
    +7.500    +7.50000
   +21.000    +2.10000E+1
   +40.800    +4.08000E+1
   +18.000    +1.80000E+1

   +3.60000        +7.50000        +21.00000        +40.80000        +18.00000
00.00.00.04  00.00.00.04  00.00.00.03  00.00.00.04
00.00.00.00  00.00.00.00  00.00.00.03  00.00.00.00
00.00.00.00  00.00.00.00  00.00.00.03  00.00.00.00
00.00.00.00  00.00.00.00  00.00.00.03  00.00.00.00
00.00.00.00  00.00.00.00  00.00.00.03  00.00.00.00
00.00.00.06  00.00.00.06  00.00.00.03  00.00.00.06



========
    PRZYCZYNA ZAKOŃCZENIA PROGRAMU SAKO  STOP NASTEPNY W ROZDZIALE 2 LB= 1325.
    POZOSTALO CZASU: 7999998.

========


     SLOWNIK ETYKIET.

     5. 1295


     POST MORTEM.


  WSP1 15862 +3.600    +7.500    +2.100E+1 +4.080E+1
             +1.800E+1
     B 15857     3928330   -2342672   -8388607    7406525
                -982082
    ZB 15847 +3.600    +7.500    +2.100E+1 +4.080E+1
             +1.800E+1
     E 15823           4         0         0         0
                       0         6         4         0         0
                       0         0         6         3         3
                       3
     I 15822        3      J 15821        5
========
```

10. Praktyczne wskazówki dla pracowników inżynieryjno-technicznych korzystających z E T O

Rozdział ten celowo umieszczamy na końcu podręcznika. Czytelnik po zapoznaniu się z poprzednim materiałem rozumie już technikę programowania i na wiele pytań z tej dziedziny sam udzieli wyczerpującej odpowiedzi.

W przypadku istnienia w biurze projektowym lub innej jednostce gospodarczej specjalisty, komórki, czy też pracowni zajmującej się zastosowaniami E T O, wszelkie zagadnienia z tym związane winny być rozwiązywane przez pracujących tam specjalistów.

Jednakże może się zdarzyć, ze pracownik będzie pozbawiony możliwości korzystania z pomocy fachowej i musi samodzielnie uzyskać rozwiązanie za pomocą komputera.

W takiej sytuacji pierwszą czynnością jest zaznajomienie się z listą ośrodków obliczeń pracujących aktualnie w kraju.

Dostarczenie na żądanie pracownika listy ośrodków obliczeń i spisu programów znajdujących się w ich dyspozycji jest obowiązkiem komórki lub wydziału informacji danej instytucji

Na ogół każdy ośrodek obliczeń ma dosyć ściśle sprecyzowany branżowy profil działania. Jednakże bardzo często w ośrodkach nie zajmujących się branżą nas interesującą znajdują się programy aktualnie nam potrzebne.

Pierwszym kryterium przydatności ośrodka jest jego oprogramowanie tzn. zasób biblioteki gotowych programów

Komputer nieoprogramowany nie przedstawia wielkiej wartości, gdyż wykorzystywanie go do naszych celów będzie wymagało opracowania programów co jest procesem czasochłonnym, trudnym i kosztownym.

Toteż przy nawiązywaniu współpracy z ośrodkiem obliczeń należy przede wszystkim poznać co następuje:

- charakter branżowy ośrodka
- typ maszyny i rodzaje autokodów, w których jest oprogramowana
- listę programów i podprogramów

W przypadku, gdy celem naszym jest skorzystanie z programu, którego funkcja odpowiada naszym aktualnym potrzebom, należy w ośrodku obliczeń zażądać dostarczenie opisu programu oraz instrukcji pisania danych. Bardzo często ograniczenia dotyczące programu nie są uwidocznione w opisach informacyjnych i reklamowych i dopiero dokładne zapoznanie się z jego dokumenta-

cją może rozstrzygnąć o jego przydatności w konkretnym przypadku. Instrukcja pisania danych zawiera szczegółowe informacje dotyczące kolejności i znaczenia poszczególnych danych.

Jakiekolwiek odstępstwo od przewidzianego reżimu nie jest możliwe, gdyż komputer po prostu nie wczyta danych.

Zarówno przygotowanie nośników informacji jak i dokonanie obliczeń należy całkowicie do ośrodka, którego obowiązkiem jest dostarczenie gotowych tabulogramów wynikowych.

Weryfikacja otrzymanych wyników nie jest potrzebna, gdyż ośrodek obliczeń względnie osoba, która podjęła się dokonania obliczeń na komputerze ponosi za nie pełną odpowiedzialność.

Jeżeli natomiast czytelnik sam napisał dane do programu, to oczywiście ponosi także pełną odpowiedzialność za ich poprawność.

Komputer nie wykrywa błędów merytorycznych ani nie odróżni błędnej informacji liczbowej w sensie wielkości np. gdy użytkownik wpisał wartość 5, zamiast 15.

Gdy pisanie danych za pomocą instrukcji jest również pracochłonne lub dla zamawiającego zbyt trudne można i tę czynność zlecić ośrodkowi obliczeń, co ma często miejsce przy obliczeniach statycznych i wytrzymałościowych, dla których dostarcza się schematy obliczeniowe.

Inaczej wygląda sprawa gdy zagadnienie opracowywane przez nas wymaga sporządzenia nowego programu. Taką decyzję możemy podjąć przede wszystkim wtedy, gdy przyszły program będzie miał walor powtarzalności tzn. wielokrotnego wykorzystywania.

Jeżeli znamy jakiś autokod w stopniu dostatecznym, możemy zdecydować się na samodzielne uruchomienie programu na komputerze. W przypadku nieznajomości autokodu i gdy w grę wchodzi również krótki termin opracowania należy wykonanie programu zlecić ośrodkowi obliczeń, gdyż czasokres opracowania przy najlepszych chęciach będzie na pewno zbyt długi a wynik nie zawsze zadowalający. Przy pewnych predyspozycjach psychicznych i średnich uzdolnieniach nauczenie się przez fachowca dowolnej branży technicznej nowego autokodu w stopniu pozwalającym na swobodne i ekonomiczne nim operowanie trwa:

- w przypadku uczenia się pierwszego w życiu autokodu
  8 miesięcy do 1 roku
- w przypadku uczenia się dalszych autokodów
  4 - 6 miesięcy.

Jeżeli nie programujemy sami należy wówczas przygotować dla ośrodka algorytm zagadnienia wraz z graficzną siecią działań. Ta ostatnia wybitnie skró-

ci czas opracowania. W ośrodku program będzie opracowywany prawdopodobnie przez matematyka /rzadko przez fachowca branżowego/ toteż przygotowanie zagadnienia musi być szczegółowe. Jeżeli programujemy sami należy dokładnie zapoznać się z biblioteką podprogramów.

Czytelnik może spodziewać się, że dla większości maszyn współczesnych istnieje załączona do nich bogata biblioteka podprogramów matematycznych,które można dołączyć do nowo tworzonych programów, co znacznie skraca i ułatwia ich opracowanie. Szczególnie dotyczy to programów technicznych. Każdy podprogram posiada dokumentację, w której zawarte są wszelkie informacje dotyczące sposobów dołączania go do programu. Ponadto w systemie komputera znajdują się również "zapisane" podprogramy, które można "przywoływać" do programu w trakcie liczenia za pomocą odpowiednich instrukcji wywoławczych. Instrukcje te znajdują się w opisach systemów, które również są dostępne w ośrodkach obliczeń.

Zamawiając czas pracy maszyny musimy być przygotowani na fakt,że nawet najbardziej reklamowane komputery ulegają dość często awariom i wymagają skomplikowanych napraw, co może zdarzyć się akurat w czasie przeznaczonym dla nas i przekreślić napięty termin naszej pracy, czego na ogół nie rozumieją laicy, którzy niestety mają prawo nas oceniać. Do dnia dzisiejszego do wielu odpowiedzialnych pracowników różnych szczebli dotarły tylko sensacyjne szczegóły o ogromnej szybkości komputerów i rzekomo ich cudownych nieograniczonych możliwościach.

Prawdą jest, że obliczenia na komputerze trwają bardzo krótko,o czym głośno w popularnych publikacjach, ale że jest to wynikiem długiej i żmudnej pracy przygotowawczej, wymagającej dodatkowych i wysokich kwalifikacji,dowiadujemy się znacznie rzadziej.

Przede wszystkim od pracy przygotowawczej i w znacznie mniejszym stopniu od typu komputera zależy ostateczny sukces obliczeniowy, który całe przedsięwzięcie powinien uczynić opłacalnym. Mówimy o tym nie dlatego, aby w czymkolwiek uszczuplić walory tego najsprawniejszego narzędzia jakie stworzył człowiek XX wieku, za pomocą którego pracujemy zresztą od lat, lecz dlatego aby czytelnik zdawał sobie sprawę z trudności jakie napotka w tej dziedzinie i potrafił je umiejętnie ominąć.

11. SPIS BŁĘDÓW SYGNALIZOWANYCH PRZEZ TRANSLATOR JĘZYKA SAKO /błędy formalne/

Nr	Znaczenie
1	Ciąg znaków pobrany przez translator został potraktowany jako wiersz programu i zawierający więcej niż 128 znaków.
2	Wiersz pobrany przez translator nie został rozpoznany jako zdanie języka SAKO.
3	W procesie translacji danych zadania TABLICA ...natrafiono na informacje nienależące do odpowiedniej klasy danych.
4	Liczba występująca w języku SAKO nie jest poprawna.
5	Nie rozpoznano końca programu.
6	Brak taśmy magnetycznej przydzielonej dla realizacji zdania BIBLIOTEKA.
7	Zdanie rozpoznane jako TEKST WIERSZY posiada niepoprawny parametr.
8	Zdanie rozpoznane jako ROZDZIAŁ posiada niepoprawny parametr.
9	Wielkość rozpoznana jako liczba oktalna zawiera więcej niż 16 cyfr lub posiada cyfry 8 lub 9.
10	Translator natrafił na zdanie POWTORZ dla którego nie wystąpiła gwiazdka.
11	Zdanie rozpoznane jako TABLICA jest niepoprawnie zbudowane.
12	Pobrana informacja nie jest nazwą części operacyjnej rozkazu we wstawce języka SAS.
13	Pobrana informacja jest niepoprawną częścią adresową rozkazu we wstawce języka SAS.
14	We wstawce j. SAS pobrany wiersz nie został przez translator rozpoznany.
15	Informacja rozpoznana jako tekst we wstawce języka SAS jest zbyt długa lub zawiera niedozwolone znaki.
16	Rozpoznana informacja nie jest poprawną makrooperacją.
17	Tekst pobrany z taśmy magnetycznej zawiera niepoprawną nazwę strony.
18	Na taśmie bibliotecznej tekst o zadanej nazwie nie został odnaleziony.
19	Błąd odczytu taśmy magnetycznej biblioteki.
20	Błędna składnia zdania.
21	Zbyt dużo miejsc zajmuje zdanie BLOK lub TABLICA.
22	Zabrakło translatorowi miejsc w pamięci operacyjnej na listy identyfikatorów użytych w zdaniach: BLOK, STRUKTURA, CAŁKOWITE, POWTORZ

Nr	Znaczenie
23	Argument jest całkowity, a powinien być ułamkowy.
24	Niewłaściwa ilość indeksów zmiennej indeksowanej.
25	Wystąpiło w programie więcej gwiazdek niż odpowiadających im rozkazów POWTORZ.
26	Ilość elementów TABLICY nie jest zgodna z ilością zadeklarowaną.
29	Deklaracja JEZYK SAKO w tekście SAKO.
30	Argument jest ułamkowy, a powinien być całkowity.
31	Parametry POWTORZ są niejednorodne.
32	WROC w programie głównym.
33	Niezadeklarowany BLOK lub STRUKTURA.
34	Zdanie typu F( ) = wystąpiło w programie głównym.
38	Ułamkowa liczba przesunięć w wyrażeniu bulowskim.
39	Brak nawiasu zamykającego w wyrażeniu arytmetycznym lub bulowskim.
47	Błędna składnia makrooperacji.
48	Brak makrooperacji w bibliotece funkcji j.M.SAS.
50	Za mało argumentów w makrooperacji.
54	W bibliotece funkcji M.SAS brak żądanej funkcji.
57	Błędna lista parametrów zdania JEZYK SAS
46,48, 51,52, 53,55	Błędna funkcja w bibliotece M.SAS lub błąd maszyny.
71	Użyto zbyt dużo zmiennych całkowitych.
72	Brak miejsca w translatorze na listę identyfikatorów i listę stałych.
75	Powtórna deklaracja etykiety.
76	Powtórna deklaracja podprogramu.
77	Identyfikator deklarowany jako tablica już wystąpił.
78	Identyfikator deklarowany jako blok już wystąpił.
92	Brak pamięci dla translatora przy dołączaniu funkcji języka.
96	Brak miejsca na liście stałych w procesie adresacji.
97	Brak etykiety.
98	Brak deklaracji BLOK lub PODPROGRAM

Nr	Znaczenie
99	Program wynikowy nie mieści się w PAO.
89	Program zawiera więcej rozdziałów niż przewiduje translator.
91	Przepełniono taśmę magnetyczną z programami binarnymi.
100	Brak pamięci bębnowej dla dalszej translacji.
200	Rozpoznany napis nie jest poprawnym zdaniem języka czołówki.
201	Obraz wiersza na tabulogramie nie odpowiada informacjom umieszczonym na taśmie.
202	Niezidentyfikowany tekst po końcu zdania języka czołówki SAKO.
203	Błędny parametr w zdaniu języka czołówki.
204	Koniec taśmy (nadmiar wejścia) w tekście czołówki.
205	Brak NL po zdaniu końca czołówki.
206	Numer klucza większy od 23 w zdaniu KLUCZE.
210	Za mały wolny bęben lub program nie mieści się na bębnie.
211	Za mała pamięć operacyjna dla programu.
215	Błędna suma kontrolna metryki bloku na tm (informacja ostrzegawcza nie liczy się jako błąd).
218	Błędna nazwa rozdziału lub słownika w pobierz.
219	Numer słownika lub rozdziału podany w POBIERZ nie został odnaleziony.
221	KONIEC w czołówce, a poprzednio nie działał translator SAKO.
223	Przy minimalizacji bębna wykonano zwiększenie obszaru (typ translacji jak w przypadku 215).
224	Wymagania na pamięć bębnową nie mogą być spełnione.
225	Próba określenia niniejszej pamięci operacyjnej w programie czołówki zakończonym zdaniem KONIEC.
226	Rozdział nie mieści się na zarezerwowanym bębnie (typ informacji jak w 215).
227	Błędne parametry ZAPISZ/POBIERZ.
228	Brak TM w zestawie dla ZAPISZ/POBIERZ.
229	Nie otwarto TM dla ZAPISZ/POBIERZ/ BIBL.
230	Próba zapisu na taśmie otwartej do czytania.
231	Niewłaściwy numer szpuli dla ZAPISZ/POBIERZ/BIBL.
233	Brak nazwy zapisywanego lub pobieranego programu
	Błędy o numerach: 207, 208, 212-214, 216, 217, 220, 222, 232, 234 oznaczają błędy taśmy magnetycznej lub błędy operatora (np. założenie niewłaściwej taśmy).

Numery:
73, 74, 90, 101, 79-84, 40-44, 93-95, 209 oznaczają błędy maszyny lub translatora.

Błędy sygnalizowane są w sposób następujący:

Pozycja pierwsza zawiera nazwę etykiety lub słowo POCZ, po której wystąpił błąd.

Słowo POCZ występuje gdy etykieta jeszcze nie wystąpiła. Pozycja druga jest numerem wiersza względem wymienionej etykiety, w którym wystąpił błąd, wiersze są numerowane od zera. Pozycja pierwsza po słowie BLAD jest numerem błędu.

Pozycja druga po słowie BLAD zawiera słowo CZOL lub GLOW lub nazwy podprogramu.

CZOL - błąd wystąpił w czołówce języka SAKO
GLOW - błąd wystąpił w programie głównym
nazwa p.p. - błąd wystąpił w podanym podprogramie.

Powyższa operacja jest wypełniona tylko przy błędzie występującym pierwszy raz w danej części programu danego rozdziału.

12. SPIS BŁĘDÓW SYGNALIZOWANYCH PRZEZ FUNKCJE JĘZYKA I PROGRAMY OBSŁUGUJĄCE PROGRAM WYNIKOWY

Nazwa błędu	Znaczenie
ZZ	Dzielenie przez zero zmiennoprzecinkowe
ZP	Wystąpił podmiar zmiennoprzecinkowy
ZN	Wystąpił nadmiar zmiennoprzecinkowy
PA	Brak pamięci dla rozdziału, zbyt duży rozdział
RB	Błędny numer rozdziału
RN	Brak rozdziału
PC	Drukuj argument nieznormalizowany
BD	Błędne parametry zdania DRUKUJ
NC	Koniec danych dla CZYTAJ WIERSZ (nadmiar wejścia)
NU	Koniec danych dla CZYTAJ OKTALNIE
EX	Błędny argument funkcji EXP
LN	Błędny argument funkcji LN
BK	Nadmiar wejścia przy CZYTAJ
BC	Błędne dane dla CZYTAJ
IL	Błędna ilość elementów bloku
NG	Gwiazdka zamiast pojedynczej liczby w CZYTAJ
DO	Zła długość liczby oktalnej
ZP	W liczbie oktalnej cyfry 8 lub 9
DK	Nadmiar potęgi całkowitej
UW	Ujemny wykładnik potęgi całkowitej lub ułamkowej o wykładniku całkowitym
AC	Błąd argumentu funkcji ACS lub ASN
TG	Błąd argumentu funkcji TG
SE	Błąd argumentu funkcji SEL
DL	Błąd argumentu funkcji DOL
OD	Błąd argumentu funkcji ODL

13. PRZYCZYNY ZAKOŃCZENIA PROGRAMU UŻYTKOWEGO

nr przyczyny	
504 do 506 i 511	Skok do jednego z miejsc od 4 do 30 (gdy zawartość tych miejsc nie była zmieniana)
507	Próba wykonania rozkazu P31 (symboliczne WE-WY) o niezidentyfikowanym w czołówce dekoderze (dekoder pusty)
508	Próba wykonania rozkazu P31 (symboliczne WE-WY) przy niedołączonym podprogramie
509	Próba wyczyszczenia wejścia lub niedołączonego wyjścia
510	Próba wykonania rozkazu P31 (symboliczne WE-WY) przy niewłaściwie w czołówce zdefiniowanym dekoderze
512	Jedna z następujących przyczyn: 1) nielegalna część operacyjna 2) za duża krotność pośredniego adresowania 3) SDY NIETYPOWE z błędną częścią adresową 4) nadmiar wykładnika lub dzielenie przez 0 w operacji zmiennoprzecinkowej 5) próba wykonania rozkazu P16, P17 ... P30 dla nie ustawionej zawartości miejsca PAO o adresach odpowiednio 16, 17 ... 30.
513	Wyczerpany czas
514	Program wyrzucony przez operatora
515	Błędne parametry SDY DEFINIUJ PRZESŁANIE lub SDY WYMIEN PROGRAM (nr urządzenia, adres PAO lub ilość)
516	Próba wykonania SDY INICJUJ PRZESŁANIE, gdy przesłanie nie było uprzednio zdefiniowane
517	Próba dołączenia symbolicznego WE-WY niezdefiniowanego w czołówce
518	Próba dołączenia symbolicznego WE-WY, gdy wyczerpano wolne numery przesłań
519	Próba dołączenia symbolicznego WE-WY, gdy błędnie zdefiniowano podprogram lub dekoder
520	Próba dołączenia symb. WE-WY, gdy podprogram z dekoderem nie mieszczą się w PAO

14. PRZYKŁAD ALGORYTMU

Temat

"Wyznaczanie naprężeń głównych za pomocą koła Mohra"

Zadanie polega na tablicowaniu wartości naprężeń normalnych σ i stycznych τ przy wyjściowych naprężeniach σ min i σ max zadawanych każdorazowo. Tak więc tablica wynikowa jest funkcją σ min i σ max oraz kąta α wyrażonego w radianach.

Operacje matematyczne polegają jedynie na korzystaniu z podanych wzorów.

14.1. Parametry /dane/ wejściowe

Lp.	Symbol	Znaczenie	Rodzaj liczby całkow.	Rodzaj liczby ułamk.	Jednostka	Orientacyjny zakres wartości
1	NR OB.	Numer obliczenia	+		-	NR OB $\leq$ 10000
2	σ max	Naprężenie normalne maksym.		+	kG/cm^2	$3600 \leq \sigma$ max 4200
3	σ min	Naprężenie normalne minimalne		+	kG/cm^2	$3600 \leq \sigma$ min 4200

14.2. Parametry /dane/ wyjściowe /obliczone w programie/

Lp.	Symbol	Znaczenie	Rodzaj liczby całkow.	Rodzaj liczby ułamk.	Jednostka	Orientacyjny zakres wartości
1	σ_1	Naprężenie normalne większe		+	kG/cm^2	$\frac{\sigma_{max}}{2} \leq \sigma_1 \leq \sigma_{max}$
2	σ_2	Naprężenie normalne mniejsze		+	kG/cm^2	$\sigma_{min} \leq \sigma_2 \leq \frac{\sigma_{max}}{2}$
3	τ	Naprężenie styczne		+	kG/cm^2	$0 \leq \tau \leq 1000$

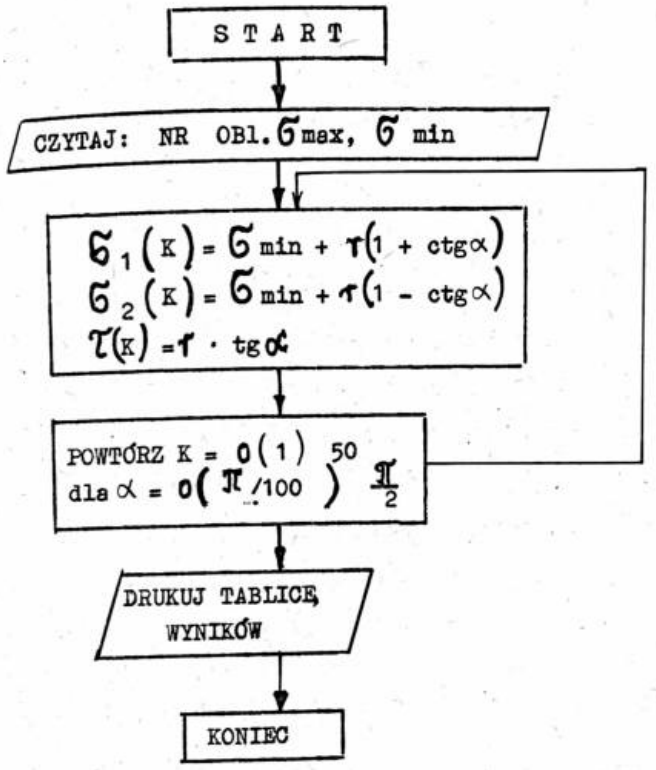
14.3. Wzory obliczeniowe

$\tau = (\sigma_{max} - \sigma_{min})/2$ parametr pomocniczy drukowany w kG/cm^2.

α zmienia się od 0 do $\pi/2$ co $\pi/100$ - kąt nachylenia płaszczyzny ukośnej wyrażony w radianach

$$\sigma_1 = \sigma_{min} + \tau(1 + ctg\alpha)$$
$$\sigma_2 = \sigma_{min} + \tau(1 - ctg\alpha)$$
$$\tau = \tau \cdot tg\alpha$$

4.4. Schemat blokowy /ideowy/



Uwaga !

Dla każdej zmiany K o jeden następuje zmiana kąta o $\pi/100$.

14.5. Forma wyników

Nr obliczenia: .....

Naprężenie zadane σ_{max} = ... σ_{min} = ... (kG/cm^2)

Lp.	Kąt α	σ_1	σ_2	τ
1	0	.....	.....	.....
2	$(\pi/100)$x 1	.....	.....	.....
3	$(\pi/100)$x 2	.....	.....	.....
.	$(\pi/100)$x 3	.....	.....	.....
.	$(\pi/100)$x 4			
.				
.				
.				
.				
5 1	$(\pi/100)$x50	.....	.....	.....

14.6. Wymagania dodatkowe dotyczące programu

- program winien mieć możliwość tablicowania ciągłego przy zadanej dowolnej ilości grup σ_{max} i σ_{min},
- program winien mieć możliwość zmiany "skoku" kąta α o dowolną część π, który to skok możnaby wczytywać (w przypadkach wyjątkowych) w postaci dodatkowej liczby w danych.

  W przypadku braku tej liczby program winien tablicować dla zmiany kąta α od 0 co $\pi/100$.

DODATEK

PRZYKŁADY ZASTOSOWAŃ KOMPUTERÓW DO OBLICZEŃ INŻYNIERSKICH

PRZYKŁAD A

OBLICZENIA INSTALACJI CENTRALNEGO OGRZEWANIA WODNEGO DWURUROWEGO

PROGRAM NA EMC OBEJMUJE NASTĘPUJĄCE OBLICZENIA:

- obliczenie średnic przewodów
- " kryz dławiących (regulacja hydrauliczna)
- dobór grzejników
- zestawienie ilości ciepła
- " " grzejników wg typów
- " " rur wg średnic
- " pojemności zładu

PRZYKŁAD ZAWIERA:

1 - Rozwinięcie instalacji (z zaznaczeniem n-rów grzejników i działek)

2 - Dane wypełniane przez projektanta instalacji

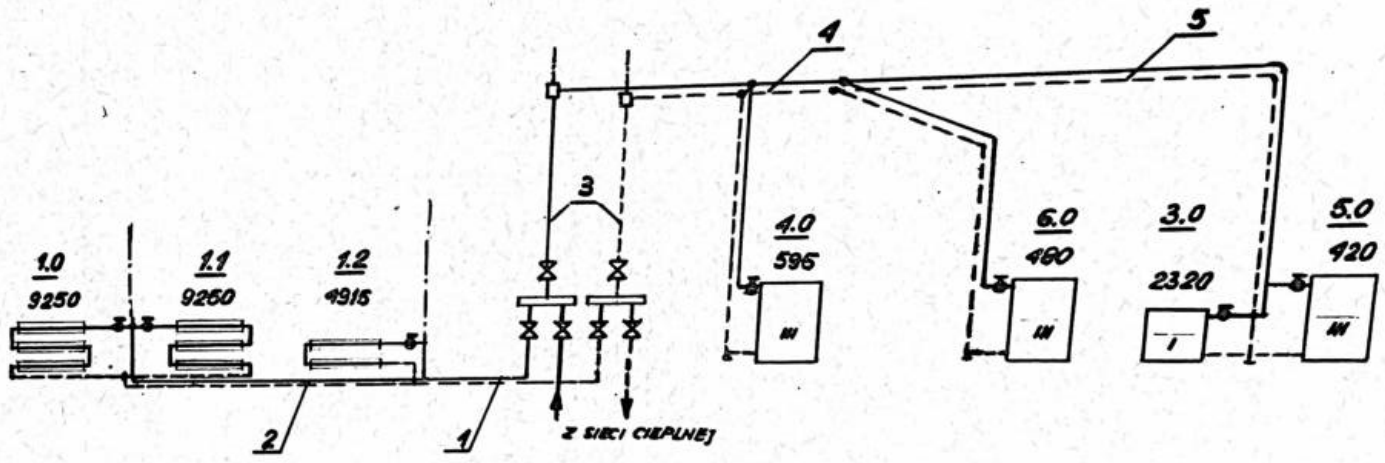
2a - dane ogólne
2b - opis grzejników
2c - opis działek

3 - Dane przepisane na dalekopisie

4 - Wyniki obliczeń uzyskane z komputera:

4a - średnice przewodów i kryz dławiących
4b - dobór grzejników
4c - zestawienia:

całk. ilości ciepła
długości przewodów wg średnic
ilości grzejników wg typów
pojemność zładu instalacji



1. ROZWINIĘCIE INSTALACJI C.O. DO PRZYKŁADU

2.

?????

2a/ DANE DLA INSTALACJI NR. -37512

t_z = 95

dt = 25

h_{dysp} = 500

ilość grzejników = 7

D_{min} = 1

D_{max} = 5

2b/ OPIS GRZEJNIKÓW: -37512

po literze				z		g				f		t		(r) r	n
nr g.	rzędna	włącz z	włącz p	zyski 1	zyski 2	typ rozgał. z	typ rozgał. p	Q kcal/h	lg	ilość		tw	typ	(r) r	
1.0	-0,5	-	2g	-	-	-	1	9250	7f	8	2t	15	6n		
1.1	-0,5	-	2g	-	-	-	1	9250	7f	2	2t	15	6n		
1.2	-0,5	-	1g	-	-	-	4	4915	2f	4	2t	15	6n		
4.0	-0,5	-	3g	-	-	-	4	595	6f	2	2t	15	3n		
6.0	-0,5	-	4g	-	-	-	4	480	8f	6	2t	15	3n		
3.0	-0,5	-	5g	-	-	-	1	2320	2f	4	2t	18	1n		
5.0	-0,5	-	5g	-	-	-	1	420	2f	2	2t	15	3	koniec	

2c/ OPIS DZIAŁEK -37512

Nr działki	Nr dz. wcześn.	Typ rozg.	Ilości			Ld [m]	Nr (r) r	Uwagi
1i	0	6	2s	10	2	5 n		
2i	1	2	0	8	2	25 n		
3i	0	6	2s	6	0	18 n		
4i	3	2	0	0	2	4 n		
5i	4	2	0	2	2	6 koniec		
			?????					

3.

```
*****
dane dla instalacji nr -37512b
95,
25,
500,
7,
1,
5.
1.0,-0.5,2g1,9250,7f8,2t5,6n
1.1,-0.5,2g1,9250,7f2,2t5,6n
1.2,-0.5,1g4,4915,2f4,2t5,6n
4.0,-0.5,3g4,595,6f2,2t15,3n
6.0,-0.5,4g4,480,8f6,2t15,3n
3.0,-0.5,5g1,2320,2f4,2t18,1n
5.0,-0.5,5g1,420,2f2,2t15,3koniec
110,6,2s10,2,5n
211,2,0,8,2,25n
310,6,2s6,0,18n
413,2,0,0,2,4n
514,2,0,2,2,6koniec
*******
```

4. INSTALACJA NR: -37512

BEZ BLEDOW

4a/ NR.INST= -37512 H.DYSP= +500.00 DT= +25.00

```
NRDW   NRD      Q      LD      FI      W      R   DZETA      Z      RI+Z

      OBIEG PRZEZ:      +1.0   H.GRAW=    -5.48   DZ,DP=    2
0     -1   23415     +5.0  +41.85 +0.20  +1.70  +24.1    +45.33    +53.83
                          KRYZA NA DZIALCE   D.KR=  +18.00          +100.76
1     -2   18500    +25.0  +35.95 +0.21  +2.36  +17.5    +37.74    +96.68
   GRZEJ    9250     +7.0  +21.60 +0.29  +8.56  +41.5   +171.67   +231.60
                     T.WZG= +94.3   KRYZA   D.KR= +15.26            +11.63

      OBIEG PRZEZ:      +1.1   H.GRAW=    -5.48   DZ,DP=    2
   GRZEJ    9250     +7.0  +21.60 +0.29  +8.56  +26.5   +109.62   +169.56
                          Z DZIALEK POLICZONYCH WCZESNIEJ=        +251.28
                     T.WZG= +94.3   KRYZA   D.KR= +11.50            +73.68

      OBIEG PRZEZ:      +1.2   H.GRAW=    -5.48   DZ,DP=    1
   GRZEJ    4915     +2.0  +16.10 +0.28 +11.48  +28.5   +107.84   +130.80
                          Z DZIALEK POLICZONYCH WCZESNIEJ=        +154.59
                     T.WZG= +94.9   KRYZA   D.KR=  +6.90           +209.13

      OBIEG PRZEZ:      +4.0   H.GRAW=    -5.48   DZ,DP=    3
0     -3    3815    +18.0  +16.10 +0.21  +7.02  +32.5    +74.09   +200.47
                          KRYZA NA DZIALCE   D.KR=   +6.50          +165.91
   GRZEJ     595     +6.0  +16.10 +0.03  +0.16 +245.5    +13.61    +14.58
                     T.WZG= +93.8   KRYZA   D.KR=  +3.04           +113.55

      OBIEG PRZEZ:      +6.0   H.GRAW=    -5.48   DZ,DP=    4
3     -4    3220     +4.0  +16.10 +0.18  +5.06   +2.5     +4.06    +24.31
   GRZEJ     480     +8.0  +16.10 +0.03  +0.13 +281.5    +10.16    +11.20
                          Z DZIALEK POLICZONYCH WCZESNIEJ=        +366.38
                     T.WZG= +93.5   KRYZA   D.KR=  +2.88            +92.63

      OBIEG PRZEZ:      +3.0   H.GRAW=    -5.48   DZ,DP=    5
4     -5    2740     +6.0  +16.10 +0.15  +3.71   +7.5     +8.82    +31.09
   GRZEJ    2320     +2.0  +16.10 +0.13  +2.70  +31.5    +26.56    +31.96
                          Z DZIALEK POLICZONYCH WCZESNIEJ=        +390.69
                     T.WZG= +92.9   KRYZA   D.KR=  +7.08            +40.78

      OBIEG PRZEZ:      +5.0   H.GRAW=    -5.48   DZ,DP=    5
   GRZEJ     420     +2.0  +16.10 +0.02  +0.11 +261.5     +7.23     +7.45
                          Z DZIALEK POLICZONYCH WCZESNIEJ=        +421.78
                     T.WZG= +92.9   KRYZA   D.KR=  +2.93            +65.29
```

SILNIK V8 N2400-4

LO (MOL/KG)	BETO -	NI -	CSR (M/SEK)
+0.505	+1.04521	+0.71086	+12.297

FI (STOPNIE)	EPSX -	BETA X -	FX (M*2/MOL)	DX/DFI (M/STOWK)	X -	T (KELV)	P (KG/CM*2)	V CYLINDRA (DCM*3)
-130.00	+14.357	+1.00000	+1216.61	+0.000000	+0.00000	+333.00	+0.900	+.1790870E+1
-124.00	+13.818	+1.00000	+1187.31	+0.000000	+0.00000	+338.32	+0.960	+.1718560E+1
-118.00	+13.223	+1.00000	+1154.97	+0.000000	+0.00000	+344.49	+1.021	+.1638748E+1
-112.00	+12.575	+1.00000	+1119.77	+0.000000	+0.00000	+351.61	+1.096	+.1551856E+1
-106.00	+11.878	+1.00000	+1081.91	+0.000000	+0.00000	+359.83	+1.187	+.1458432E+1
-100.00	+11.137	+1.00000	+1041.69	+0.000000	+0.00000	+369.27	+1.299	+.1359175E+1
-94.00	+10.360	+1.00000	+999.45	+0.000000	+0.00000	+380.13	+1.438	+.1254935E+1
-88.00	+9.553	+1.00000	+955.61	+0.000000	+0.00000	+392.61	+1.611	+.1146725E+1
-82.00	+8.725	+1.00000	+910.62	+0.000000	+0.00000	+406.95	+1.828	+.1035710E+1
-76.00	+7.886	+1.00000	+865.04	+0.000000	+0.00000	+423.46	+2.104	+.9232024
-70.00	+7.046	+1.00000	+819.42	+0.000000	+0.00000	+442.50	+2.461	+.8106350
-64.00	+6.218	+1.00000	+774.41	+0.000000	+0.00000	+464.52	+2.928	+.6995396
-58.00	+5.412	+1.00000	+730.64	+0.000000	+0.00000	+490.03	+3.549	+.5915122
-52.00	+4.641	+1.00000	+688.76	+0.000000	+0.00000	+519.59	+4.387	+.4881754
-46.00	+3.917	+1.00000	+649.44	+0.000000	+0.00000	+553.88	+5.541	+.3911397
-40.00	+3.252	+1.00000	+613.31	+0.000000	+0.00000	+593.57	+7.153	+.3019621
-34.00	+2.657	+1.00000	+580.95	+0.000000	+0.00000	+639.16	+9.429	+.2221078
-28.00	+2.141	+1.00000	+552.92	+0.000000	+0.00000	+690.67	+12.645	+.1529130
-27.00	+2.065	+1.00000	+548.70	+0.000000	+0.00000	+699.77	+13.294	+.1425019
-26.00	+1.988	+1.00000	+544.61	+0.000000	+0.00000	+709.00	+13.979	+.1324250
-25.00	+1.915	+1.00000	+540.67	+0.000000	+0.00000	+718.34	+14.700	+.1226869
-24.00	+1.845	+1.00000	+536.86	+0.000000	+0.00000	+727.77	+15.459	+.1132924
-23.00	+1.778	+1.00000	+533.20	+0.000000	+0.00000	+737.29	+16.255	+.1042459
-22.00	+1.713	+1.00000	+529.67	+0.000000	+0.00000	+746.86	+17.090	+.9355179E-1
-21.00	+1.650	+1.00000	+526.29	+0.000000	+0.00000	+756.47	+17.962	+.8721417E-1
-20.00	+1.591	+1.00000	+523.06	+0.000000	+0.00000	+766.09	+18.870	+.7923707E-1
-19.00	+1.534	+1.00000	+519.98	+0.000000	+0.00000	+775.68	+19.814	+.7162432E-1
-18.00	+1.480	+1.00000	+517.04	+0.000000	+0.00000	+785.22	+20.789	+.6437958E-1
-17.00	+1.429	+1.00000	+514.26	+0.000000	+0.00000	+794.66	+21.794	+.5750634E-1
-16.00	+1.380	+1.00000	+511.62	+0.000000	+0.00000	+803.96	+22.823	+.5100791E-1
-15.00	+1.335	+1.00000	+509.14	+0.000000	+0.00000	+813.08	+23.872	+.4488742E-1
-14.00	+1.292	+1.00000	+506.82	+0.000000	+0.00000	+821.96	+24.932	+.3914783E-1
-13.00	+1.252	+1.00000	+504.65	+0.000000	+0.00000	+830.56	+25.997	+.3379191E-1
-12.00	+1.215	+1.00000	+502.63	+0.000000	+0.00000	+838.82	+27.056	+.2882224E-1
-11.00	+1.181	+1.00000	+500.78	+0.000000	+0.00000	+846.67	+28.100	+.2424124E-1
-10.00	+1.150	+1.00000	+499.08	+0.000000	+0.00000	+854.05	+29.115	+.2005111E-1
-9.00	+1.121	+1.00004	+497.52	+0.002500	+0.00100	+866.44	+30.285	+.1625389E-1
-8.00	+1.096	+1.00021	+496.06	+0.005600	+0.00477	+885.04	+31.657	+.1285142E-1
-7.00	+1.073	+1.00061	+494.64	+0.016100	+0.01405	+925.74	+33.818	+.9845330E-2
-6.00	+1.054	+1.00185	+492.98	+0.048000	+0.04232	+1032.51	+38.462	+.7237110E-2
-5.00	+1.038	+1.00498	+490.55	+0.095750	+0.11401	+1229.64	+46.678	+.5028013E-2
-4.00	+1.024	+1.00992	+487.43	+0.115500	+0.22682	+1451.06	+56.083	+.3219100E-2

-3.00	+1.014	+1.01475	+484.54	+0.104300	+0.33727	+1639.25	+64.319	+.1811257E-2
-2.00	+1.006	+1.01902	+482.11	+0.091350	+0.43499	+1795.64	+71.280	+.8051660E-3
-1.00	+1.002	+1.02273	+480.12	+0.077800	+0.51982	+1922.45	+76.936	+.2013159E-3
+0.00	+1.000	+1.02582	+478.60	+0.064000	+0.59052	+2021.69	+81.274	+.0000000
+1.00	+1.002	+1.02833	+477.51	+0.050750	+0.64796	+2095.99	+84.341	+.2013159E-3
+2.00	+1.006	+1.03025	+476.86	+0.036550	+0.69179	+2145.22	+86.095	+.8051660E-3
+3.00	+1.014	+1.03155	+476.65	+0.024550	+0.72161	+2173.68	+86.701	+.1811257E-2
+4.00	+1.024	+1.03249	+476.77	+0.021150	+0.74304	+2194.92	+86.729	+.3219100E-2
+5.00	+1.038	+1.03346	+477.03	+0.023250	+0.76529	+2217.34	+86.557	+.5028013E-2
+6.00	+1.054	+1.03451	+477.41	+0.024600	+0.78935	+2239.83	+86.156	+.7237116E-2
+7.00	+1.073	+1.03560	+477.93	+0.024850	+0.81421	+2260.79	+85.476	+.9845339E-2
+8.00	+1.096	+1.03668	+478.61	+0.024350	+0.83886	+2279.22	+84.497	+.1285142E-1
+9.00	+1.121	+1.03772	+479.45	+0.023400	+0.86280	+2294.61	+83.227	+.1625389E-1
+10.00	+1.150	+1.03872	+480.48	+0.022250	+0.88561	+2306.84	+81.687	+.2005111E-1
+11.00	+1.181	+1.03967	+481.67	+0.021100	+0.90730	+2316.07	+79.916	+.2424124E-1
+12.00	+1.215	+1.04056	+483.04	+0.019750	+0.92776	+2322.19	+77.941	+.2882224E-1
+13.00	+1.252	+1.04140	+484.59	+0.018300	+0.94677	+2325.23	+75.793	+.3379191E-1
+14.00	+1.292	+1.04216	+486.31	+0.016800	+0.96435	+2325.29	+73.506	+.3914783E-1
+15.00	+1.335	+1.04286	+488.22	+0.015100	+0.98032	+2322.23	+71.103	+.4488742E-1
+16.00	+1.380	+1.04348	+490.30	+0.013250	+0.99451	+2316.06	+68.609	+.5100791E-1
+17.00	+1.429	+1.04372	+492.71	+0.011300	+1.00000	+2306.72	+66.030	+.5750634E-1
+22.00	+1.713	+1.04372	+507.48	+0.000000	+1.00000	+2179.48	+52.051	+.9555179E-1
+27.00	+2.063	+1.04372	+525.71	+0.000000	+1.00000	+2058.06	+40.808	+.1425019
+32.00	+2.475	+1.04372	+547.18	+0.000000	+1.00000	+1946.37	+32.163	+.1977969
+37.00	+2.945	+1.04372	+571.63	+0.000000	+1.00000	+1845.59	+25.633	+.2607826
+42.00	+3.467	+1.04372	+598.78	+0.000000	+1.00000	+1755.53	+20.713	+.3307221
+47.00	+4.034	+1.04372	+628.32	+0.000000	+1.00000	+1675.34	+16.986	+.4068077
+52.00	+4.641	+1.04372	+659.91	+0.000000	+1.00000	+1603.99	+14.136	+.4881754
+57.00	+5.281	+1.04372	+693.20	+0.000000	+1.00000	+1540.45	+11.932	+.5739198
+62.00	+5.946	+1.04372	+727.82	+0.000000	+1.00000	+1483.76	+10.207	+.6631097
+67.00	+6.630	+1.04372	+763.42	+0.000000	+1.00000	+1433.08	+8.841	+.7548047
+72.00	+7.325	+1.04372	+799.63	+0.000000	+1.00000	+1387.66	+7.748	+.8480712
+77.00	+8.026	+1.04372	+836.10	+0.000000	+1.00000	+1346.90	+6.864	+.9419980
+82.00	+8.725	+1.04372	+872.48	+0.000000	+1.00000	+1310.24	+6.142	+.1035710E+1
+87.00	+9.416	+1.04372	+908.46	+0.000000	+1.00000	+1277.22	+5.548	+.1128383E+1
+92.00	+10.094	+1.04372	+943.73	+0.000000	+1.00000	+1247.44	+5.053	+.1219252E+1
+97.00	+10.753	+1.04372	+978.04	+0.000000	+1.00000	+1220.56	+4.643	+.1307618E+1
+102.00	+11.389	+1.04372	+1011.13	+0.000000	+1.00000	+1196.27	+4.296	+.1392859E+1
+107.00	+11.997	+1.04372	+1042.80	+0.000000	+1.00000	+1174.33	+4.004	+.1474429E+1
+112.00	+12.575	+1.04372	+1072.86	+0.000000	+1.00000	+1154.50	+3.755	+.1551856E+1
+117.00	+13.118	+1.04372	+1101.15	+0.000000	+1.00000	+1136.58	+3.544	+.1624745E+1
+122.00	+13.626	+1.04372	+1127.56	+0.000000	+1.00000	+1120.40	+3.363	+.1692771E+1
+127.00	+14.095	+1.04372	+1151.98	+0.000000	+1.00000	+1105.80	+3.209	+.1755674E+1
+132.00	+14.524	+1.04372	+1174.34	+0.000000	+1.00000	+1092.67	+3.077	+.1813250E+1
+137.00	+14.913	+1.04372	+1194.56	+0.000000	+1.00000	+1080.87	+2.965	+.1865347E+1

Ž (MSEK)	ALFA Z (STOP.)	FI KOŃCA SPALANIA (STOP.)
+1.31944	+19.000	+17.500

POLE PLUS (KG/CM+2 - DCM+3)	POLE MINUS	SR CISNIENIE INDYKOWANE (KG/CM+2)
+17.376	+0.597	+8.074

4b/

DOBOR GRZEJNIKOW INST. NR= -37512 TW. MAX= +95.00

SYMBOL	QB	Z1	Z2	QN	TSW	TYP	ILOSC	KRYZA
+1.0	9250			9250	+76.8	RF-2	+15.6	+15.3
+1.1	9250			9250	+76.8	RF-2	+15.6	+11.5
+1.2	4915			4915	+77.4	RF-2	+8.2	+6.9
+4.0	595			595	+66.3	S-3	4	+3.0
+6.0	480			480	+66.0	S-3	3	+2.9
+3.0	2320			2320	+62.4	S-1	24	+7.1
+5.0	420			420	+65.4	S-3	3	+2.9

4c/ ZESTAWIENIA KONCOWE NR.INST= -37512

TW.ZAS= +95.00 TW POW= +68.77 GRD. C.

ILOSC OBIEGOW= 7

PRZEPL= +1089.2 KG/GODZ (SKORYGOWANA)

BILANS CIEPLA

Q. GRZEJ= 27230
Q. STRAT= 1342
Q. SUM = 28572

ZESTAWIENIE GRZEJNIKOW:

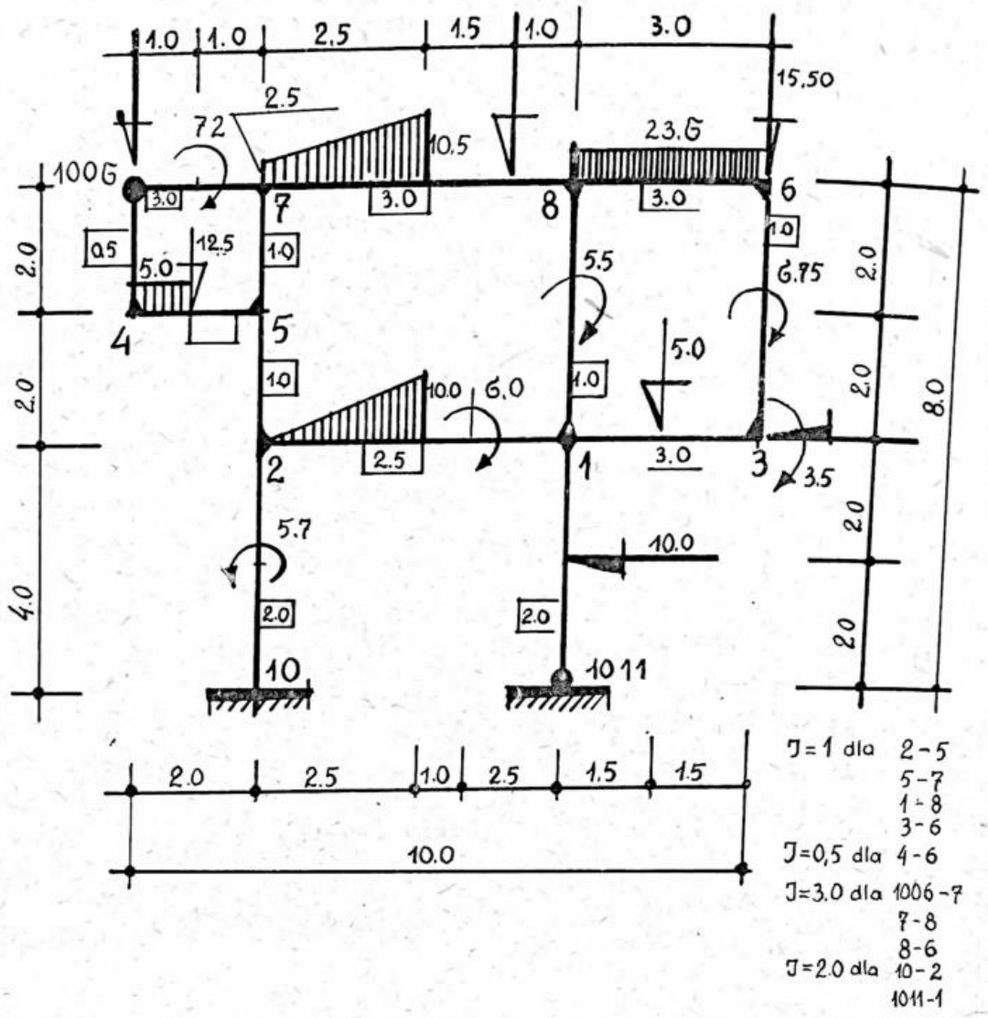
LACZNA DLUGOSC RUR ZEBROWYCH= +39.5
TYP S-1 24
TYP S-3 10

SREDNICE I LACZNE DLUGOSCI PRZEWODOW

1	+16.10	+20.0
2	+16.10	+28.0
3	+21.60	+14.0
5	+35.95	+25.0
6	+41.85	+5.0

POJEMNOSC WODNA ZLADU +240.8 LTR.

B. Obliczenie ustroju prętowego



Schemat statyczny ramy Nr 1

Rama o węzłach sztywnych z wyjątkiem nr 1006 i 1011.

Program oblicza momenty przywęzłowe ram dla dowolnego układu prętowego dla dowolnego obciążenia.

W niniejszym przykładzie podaje się dane oraz wyniki /w postaci oryginalnej/.

Dane do obliczenia ustroju prętowego

```
'DIANA
'PRZYKŁAD OBLICZEŃ INŻYNIERSKICH DO PODRĘCZNIKA PROGR.
8
5
13
7
2
1
1   10   2
2   2   5
3   5   7
4   4   1006
5   1011   1
6   1   8
7   3   6
8   2   1
9   4   5
10   1006   7
11   7   8
12   8   6
13   1   3
X=1   6   7   8   9
X=2   4   5
X=3   2   1   3
Y=4   1006   4
Y=5   9   3
L=4.0   1   5   6   7
L=2.0   3   9   4   10   2
L=3.0   12   13
L=5.0   8   11
J=1.0   2   3   6   7
J=2.0   1   5   9
J=0.5   4
J=3.0   10   11   12   13
J=2.5   8
SCHEMAT 1
WEZLY
M=-3.5   4
M=3.5   3
P=-15.5   1
P=-25.5   3
P=8.8   4
P=15.5   5]
PRETY
T   0.0   2.5   2.5   10.5   11
T   0.0   0.0   2.5   10.0   8
T   0.0   5.0   1.0   5.0   9
R   23.6   12
P   4.0   25.5   11
P   1.5   5.0   13
P   1.0   12.5   9
P   1.0   18.55   2
P   2.0   -10.0   5
M   1.0   7.2   10
M   3.5   6.0   8
M   2.0   5.5   6
M   2.0   6.75   7
M   2.0   -5.75   1]
+++++
```

B) Obliczenie ustroju prętowego. (wyniki)

PRZYKŁAD OBLICZEN INZYNIERSKICH DO PODRECZNIKA PROGR.

WYNIKI MOMENTOW PRZYWEZLOWYCH I REAKCJI PODPOROWY

SCHEMAT OBCIAZENIA NR 1

M[10, 2]=	33.19743;	R[10, 2]=	-14.080
M[2, 10]=	28.87170;	R[2, 10]=	14.080
M[2, 5]=	-19.51730;	R[2, 5]=	23.719
M[5, 2]=	-9.37126;	R[5, 2]=	-5.169
M[5, 7]=	1.41669;	R[5, 7]=	-3.087
M[7, 5]=	4.75722;	R[7, 5]=	3.087
M[4,1006]=	-16.51247;	R[4,1006]=	8.256
M[1006, 4]=	.00000;	R[1006, 4]=	-8.256
M[1011, 1]=	.00000;	R[1011, 1]=	-10.114
M[1,1011]=	20.45593;	R[1,1011]=	.114
M[1, 8]=	8.61338;	R[1, 8]=	-5.018
M[8, 1]=	5.95952;	R[8, 1]=	5.018
M[3, 6]=	17.18812;	R[3, 6]=	-7.395
M[6, 3]=	5.64116;	R[6, 3]=	7.395
M[2, 1]=	-9.35440;	R[2, 1]=	6.756
M[1, 2]=	11.24257;	R[1, 2]=	5.744
M[4, 5]=	13.01247;	R[4, 5]=	-.484
M[5, 4]=	7.95456;	R[5, 4]=	17.984
M[1006, 7]=	.00000;	R[1006, 7]=	-8.316
M[7,1006]=	9.43297;	R[7,1006]=	8.316
M[7, 8]=	-14.19018;	R[7, 8]=	14.816
M[8, 7]=	22.38303;	R[8, 7]=	26.934
M[8, 6]=	-28.34255;	R[8, 6]=	46.728
M[6, 8]=	-5.64116;	R[6, 8]=	24.072
M[1, 3]=	-40.31188;	R[1, 3]=	20.500
M[3, 1]=	-13.68812;	R[3, 1]=	-15.500

SUMY KONTROLNE MOMENTOW PRZYWEZLOWYCH

WĘZEŁ	SUMA
1	.0000
2	-.0000
3	3.5000
4	-3.5000
5	.0000
6	.0000
7	.0000
8	.0000

SUMY KONTROLNE SIL NA PRZESUWACH

PRZESUW	SUMA
1	15.5000
2	-.0000
3	25.5000
4	-8.8000
5	-15.5000

PRZYKŁAD C

Przebieg ciśnień i temperatur w silniku wysokoprężnym

Analityczna metoda przewidywania wielkości ciśnienia i temperatury w cylindrze silnika wysokoprężnego bez doładowania w zależności od kąta obrotu wału korbowego została zaprogramowana na maszynę cyfrową ZAM-41 w języku SAKO. Algorytm przedstawiony jest w kwartalniku "Silniki spalinowe" 1/71 /B.Jung "Algorytm obliczeń przebiegu ciśnień i temperatur w silniku wysokoprężnym bez doładowania"/.

Program niniejszy ma zastosowanie zarówno na etapie projektowania silnika jak i przy produkcji seryjnej. Niżej zamieszczono przykładowe dane i wyniki /z drukarki wierszowej/. Nośnikiem danych jest taśma perforowana w kodzie M2 lub Ferrantiego.

Czas liczenia jednego wariantu łącznie z translacją wynosi około 4 minut, /program ten w wersji na maszynę ZAM-2 Gamma liczył 1 wariant 1 ÷ 2 godz. w zależności od wielkości przyjętych przedziałów kątowych do obliczeń/.

SILNIK V8 N2400-4 tytuł identyfikujący wariant obliczeń

wskaźnik K=1 zależność $dx/d\varphi = f(\varphi)$ stablicowana

1 liczba wartości ciepła właściwego przy stałej objętości dla powietrza i spalin przy $\alpha = 1$

25 liczba wartości $dx/d\varphi$

30 wskaźnik L = 1 wprowadzany do maszyny kąt opóźnienia zapłonu

1 liczba cylindrów

8 τ=2 czterosuw

2 ciśnienie w cylindrze w DZP $[kg/cm^2]$

1.15 prędkość obrotowa wału korbowego podzielona przez 100 [obr/min]

24 kąt opóźnienia zapłonu [stop]

19 zakładany procent spalonego paliwa na końcu fazy spalania

99.95 średnica tłoka [m]

.14 skok tłoka [m]

.135 rzeczywisty stopień sprężenia

14.5 kąt obrotu wału korbowego na początku procesu sprężania φ_a[stop]

-130 kąt kończący obliczenia φ_b[stop]

134 stosunek długości korby do korbowodu

.2721 niższa wartość opałowa paliwa podzielona przez 100 [kcal/kg]

101.5 ciśnienie odpowiadające kątowi φ_a $[kG/cm^2]$

0.9 temperatura odpowiadająca kątowi φ_a $[^oK]$

333 średnia temperatura ścianek cylindra $[^oK]$

423 współczynnik nadmiaru powietrza α

1.5 współczynnik pozostałości spalin γ_r

0.034 θ kąt początku wtrysku [stop]

-28 udział ciężarowy wodoru w paliwie

0.1365 udział ciężarowy węgla w paliwie

0.8625 udział ciężarowy siarki w paliwie

0.006 udział ciężarowy tlenu w paliwie

0.004 przedział temperatury dla której tablicowane są wartości ciepła właściwego powietrza i spalin dla $\alpha = 1$

100 temperatura od której tablicowane są wartości ciepła właściwego powietrza i spalin przy $\alpha = 1$ $[^oK]$

273

0.0015 0.0035 0.0077 0.0245 0.0715 0.120 0.111 0.0976 0.0851 0.0705 0.0575
0.044 0.0291 0.020 0.0223 0.0242 0.025 0.0247 0.024 0.0228 0.0217 0.0205
0.019 0.0176 0.016 0.0142 0.0123 0.0103 0.007 0.0058 0.0032 zbiór wartości $dx/d\varphi$

?

4.958 5.004 5.102 5.243 5.406 5.571 5.729 5.872 5.999 6.112 6.210 6.297
6.374 6.443 6.504 6.558 6.607 6.653 6.695 6.734 6.769 6.803 6.834 6.864
6.892 6.918 zbiór wartości ciepła właściwego przy stałej objętości dla powietrza

?

5.298 5.47 5.658 5.866 6.085 6.302 6.509 6.7 6.874 7.03 7.169 7.293 7.403
7.5 7.586 7.663 7.731 7.794 7.849 7.899 7.943 7.984 8.021 8.021
8.054 8.084 8.111 zbiór wartości ciepła właściwego dla spalin ($\alpha = 1$)

?

16.5	geometryczny stopień sprężania
6	przedział kątowy obliczeń dla fazy sprężania
1	przedział kątowy obliczeń dla fazy spalania
5	przedział kątowy obliczeń dla fazy rozprężania
0.9	współczynnik do wyznaczania ciśnienia podczas suwu ssania

PRZYKŁAD D

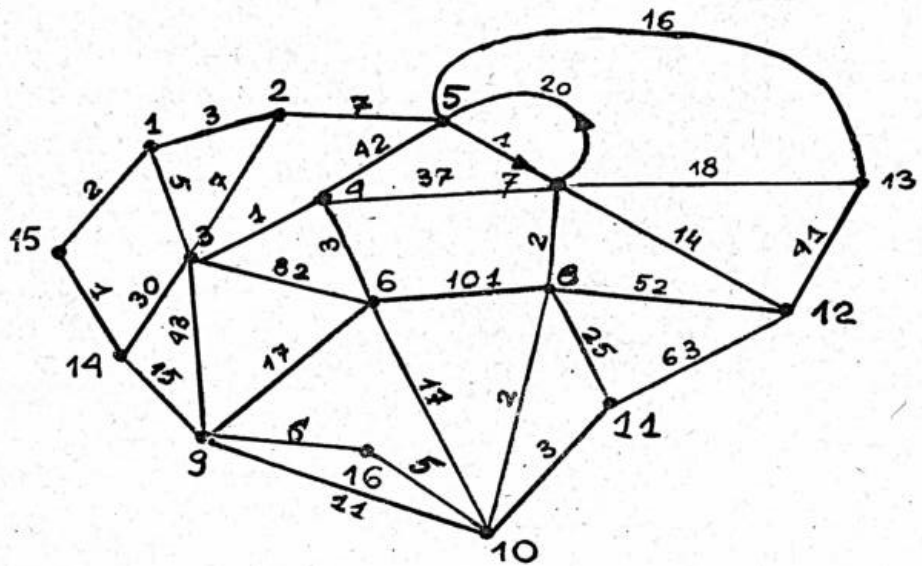
Najkrótsze drogi w sieci od jednego węzła do wszystkich pozostałych /drzewo ekonomiczne/

Problem ten został zaprogramowany w języku FORTRAN na komputer amerykański CDC-3170.

Algorytm jest oparty na metodzie Dantziga /G.B.Dantzig, "On the shortest route through a network", Management Science, January 1960/, który należy do grupy wielkich indywidualności w USA z dziedziny badań operacyjnych. Program ten może mieć duże zastosowanie w różnych dziedzinach techniki tzn. wszędzie tam, gdzie problem jest związany z siecią /grafem zorientowanym/. Każdy "odcinek" sieci /łuk w matematycznej nomenklaturze/ jest opisany przez numer węzła początku i końca "odcinka" i parametr charakteryzujący ten "odcinek" /np. różnica ciśnień, czas, długość, różnica temperatur, koszt itp./. Komputer wyszukuje drogi "najkrótsze" tzn. minimalizujące sumy parametrów charakteryzujących "odcinki" sieci z zadanego węzła do wszystkich pozostałych.

Czas liczenia drzewa ekonomicznego od węzła nr 1 przykładowej sieci załączonej poniżej wynosi łącznie z translacją programu 15 sekund.

Wyniki wyprowadzane są na drukarkę wierszową, dane wprowadzane są na kartach perforowanych.



Liczba węzłów 16
Liczba łuków 62
Nr węzła od którego liczone jest drzewo 1

PRZYKŁADOWA SIEĆ

PRZYKŁADOWE DANE

| | | |

7	13	18
13	7	18
8	11	25
11	8	25
8	10	2
10	8	2
7	8	2
8	7	2
10	11	3
11	10	3
6	8	101
8	6	101
6	10	17
10	6	17
9	10	11
10	9	11
9	6	17
6	9	17
3	6	82
6	3	82

4	6	3
6	4	3
4	7	37
7	4	37
4	5	42
5	4	42
10	16	5
16	10	5
9	16	6
16	9	6
3	9	43
9	3	43
3	4	1
4	3	1
2	3	7
3	2	7
3	14	30
14	3	30
9	14	15
14	9	15

15	14	1
14	15	1
1	3	5
3	1	5

D R Z E W O O D W Ę Z Ł A N R. 1

DROGA OD WĘZŁA NR. 1 DO WĘZŁA NR. 16

16
9
14
15

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI= 24

DROGA OD WĘZŁA NR. 1 DO WĘZŁA NR. 10

10
6
4
3
1

SUMA PARAMETRU CHARAKTERYZUJĄCEGO LUKI= 26

DROGA OD WĘZŁA NR. 1 DO WĘZŁA NR. 11

11
10
6
4
3
1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI= 29

DROGA OD WĘZŁA NR. 1 DO WĘZŁA NR. 8

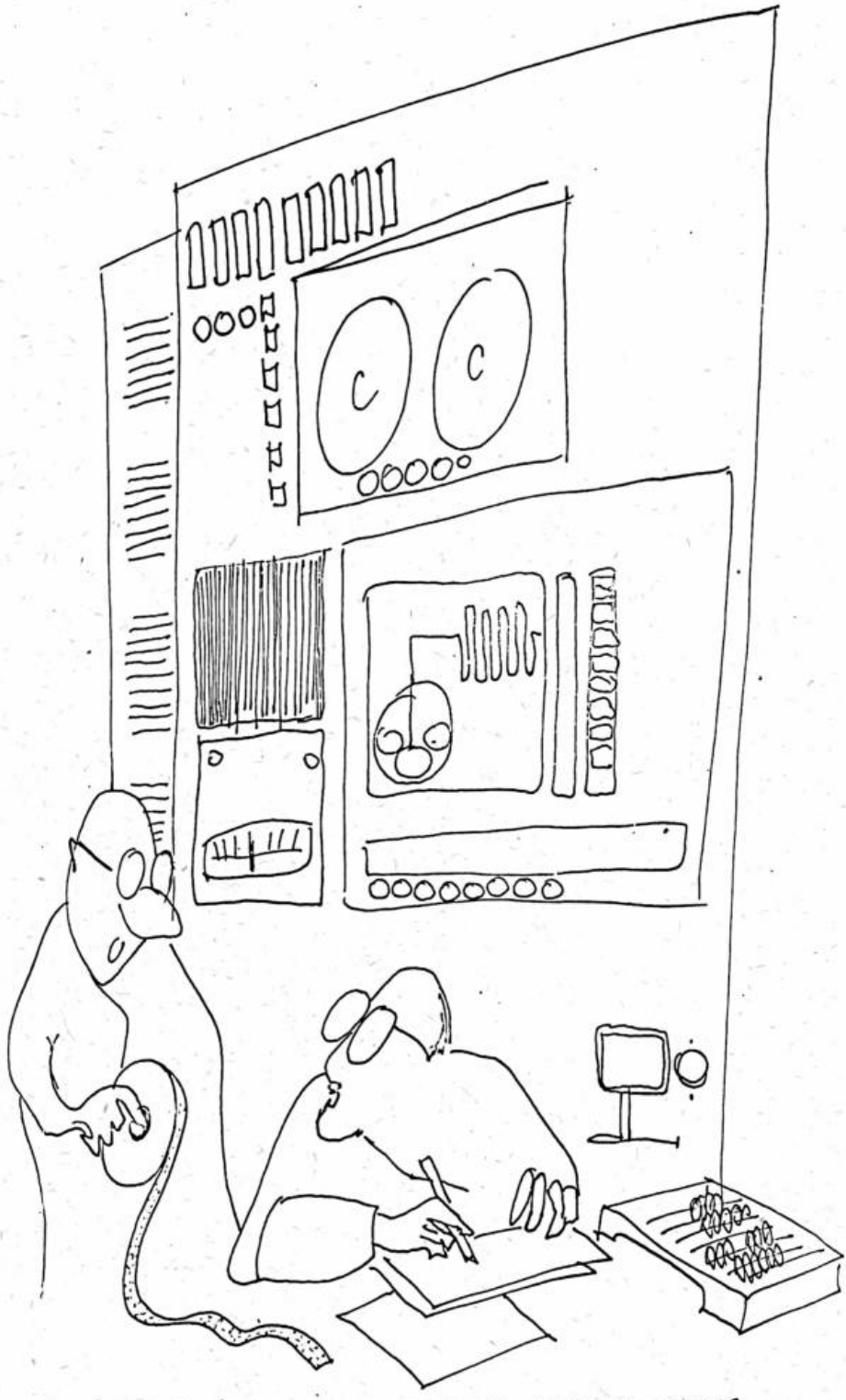
8
7
5
2
1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI= 13

DROGA OD WĘZŁA NR. 1 DO WĘZŁA NR. 12

12
7
5
2
1

```
SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI=          6

            DROGA OD WEZLA  NR.     1    DO WEZLA  NR.     5


 5
 2
 1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI=         10

            DROGA OD WEZLA  NR.     1    DO WEZLA  NR.     3


 3
 1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI=          5

            DROGA OD WEZLA  NR.     1    DO WEZLA  NR.     2


 2
 1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI=          3

            DROGA OD WEZLA  NR.     1    DO WEZLA  NR.    15


 15
  1

SUMA PARAMETRU CHARAKTERYZUJACEGO LUKI=          2



  STOP    01234
```



– SPRAWDŹCIE, PANIE KOLEGO, CZY TO PUDŁO SIĘ NIE POMYLIŁO. (rys. Lengren)

BIBLIOTEKA NARODOWA BN

WOSI "Wspólna Sprawa", PZMP Warszawa,ul. Puławska 108
Zlec. 453/72, nakł. 500+20

20.III.73