

ZAM-41

Język programowania

eol-2

ZAM - 41

Przedmiotem badań jest język
programowania ZAM-41 opisany w
dokumentach technicznych i
zawierający informacje o strukturze i
zasadach działania tego języka.

Język programowania

EOL - 2

dla ZAM - 41

Opis

Instytut Maszyn Matematycznych
Branżowy Ośrodek INTE

Opracował Zespół:

prof. dr Leon Łukaszewicz
mgr Iwona Messner
mgr Jan Walasek
mgr Zdzisława Wrotek

Komitet Redakcyjny: J. Borowiec /red. nacz./, W. Kossakowski, A. Mazurkiewicz, J. Wierzbowski, A. Wiśniewski, W. Wudel /sekr. red./

Adres Redakcji Warszawa, ul. Krzywickiego 34

Pow. w DDM n. 600 egz. pap. offset. kl. III g. 80
zam. 127/71. U-106

S p i s r z e c z y

Słowo wstępne	VII-VIII
1. WSTĘP	1-1
1.1. Przeznaczenie języka EOL	1-1
1.2. Ogólna struktura języka	1-2
1.3. Formalizm opisu składni	1-3
1.4. Wersje języka EOL	1-4
1.5. Notacja wartości zmiennej	1-4
2. SYMBOLE PODSTAWOWE	2-1
2.1. Znaki	2-1
2.1.1. Litery	2-1
2.1.2. Cyfry	2-1
2.1.3. Znaki specjalne	2-1
3. ZMIENNE I ICH WARTOŚCI	3-1
3.1. Wejście	3-1
3.2. Wyjście	3-2
3.3. Wyrażenie	3-2
3.3.1. Słowo	3-2
3.3.2. Liczba	3-2
3.3.3. Adres zapisu	3-3
3.4. Pliki	3-3
3.5. Zmienna logiczna	3-3
3.6. Stos adresów rozkazów	3-4
3.7. Konwencja wartości początkowych	3-4

4. POSTAĆ OGÓLNA ROZKAZÓW	4-1
4.1. Struktura rozkazu	4-1
4.2. Argumenty	4-1
4.2.1. Litera z indeksem	4-2
4.2.1.1. Sposób czytania	4-2
4.2.1.2. Sposób dopisywania	4-3
4.2.1.3. Sposób wykorzystania	4-4
4.2.1.4. Inne litery	4-5
4.2.2. Etykieta	4-5
4.2.3. Symbol klasy znaków	4-5
4.2.4. Tekst	4-6
4.2.5. Całkowita	4-6
4.2.6. Brak argumentu	4-7
4.3. Oznaczoność wyniku	4-7
5. ROZKAZY, DEKLARACJE, KOMENTARZE	5-1
5.1. Przesłania	5-1
5.1.1. Prześlij	5-1
5.1.2. Czytaj	5-3
5.1.3. Pisz	5-4
5.1.4. Kopiuj	5-4
5.1.5. Wstaw	5-6
5.1.6. Pobierz	5-6
5.1.7. Schowaj	5-7
5.1.8. Ustaw	5-7
5.1.9. Zamień	5-7
5.2. Umieszczenie	5-8
5.3. Usunięcie	5-8
5.4. Relacja	5-9
5.4.1. Relacja arytmetyczna	5-9
5.4.2. Relacja tekstowa	5-10
5.4.3. Relacja zapisu	5-12
5.5. Rozkaz sterujący	5-13
5.5.1. Skok	5-13
5.5.2. Skok warunkowy	5-13
5.5.3. Skok zwrotnicowy	5-14

5.5.4.	Skok powrotny	5-16
5.6.	Rozkaz arytmetyczny	5-16
5.7.	Przekształcenie	5-17
5.7.1.	Przekształcenie na słowo	5-18
5.7.2.	Przekształcenie na liczbę	5-18
5.7.3.	Zbijanie	5-19
5.7.4.	Rozbijanie	5-20
5.8.	Szukanie	5-21
5.8.1.	Szukaj	5-21
5.8.2.	Omiń	5-22
5.9.	Zliczanie	5-22
5.9.1.	Znajdź	5-23
5.9.2.	Szukaj i licz	5-23
5.9.3.	Omiń i licz	5-24
5.10.	Różne	5-25
5.10.1.	Cofnij	5-25
5.10.2.	Pakuj	5-25
5.10.3.	Stop	5-25
5.10.4.	Idź	5-25
5.11.	Deklaracja	5-26
5.11.1.	Zwrotnica	5-26
5.11.2.	Robocze	5-27
5.11.3.	Start programu	5-27
5.12.	Komentarz	5-27
6.	PROGRAMY I SEKCJE	6-1
6.1.	Program	6-1
7.	MAKRODEFINICJE	7-1
7.1.	Odwółanie do makrodefinicji	7-4
8.	PRZYKŁADY	8-1
DODATEK A. ZBIÓR ZNAKÓW DLA DANYCH		
DODATEK B. UŻYCIE W PROGRAMIE SEKCJI W MSAS-ie		
DODATEK C. PRZYPISYWANIE URZĄDZEŃ ZMIENNYM WEJŚCIOWYM /WYJŚCIOWYM/		
DODATEK D. SYGNALIZACJA BŁĘDÓW I PRZYCZYNY ZAKOŃCZENIA PROGRAMU		
DODATEK E. LISTA SŁÓW KLUCZOWYCH		
DODATEK F. SKOROWIDZ TERMINÓW		

został wykorzystany w swoim miejscu d' EOL
w tym celu.

Wszystkie wyrażenia w językach komputerowych EOL-1
są wyrażeniaem albo składnikiem wyrażeń
takie jak: nazwy, prototypy, deklaracje, modyfikatory
i zmiennej, skierowane do albo składnikiem wyrażeń
takie jak: nazwy, prototypy, deklaracje, modyfikatory
i zmiennej, skierowane do albo składnikiem wyrażeń

Słowo wstępne

EOL jest stosunkowo prostym językiem, słu-
żącym do przetwarzania symboli, przedstawio-
nych jako ciągi znaków /sznury/. Zakres zasto-
sowań EOL'u jest dość szeroki, aczkolwiek ję-
zyk ten jest przeznaczony w szczególności do
pisania translatorów, czyli programów tłumac-
zących z jednego języka programowania na
drugi. Przykładowo, znaczna część translato-
ra języka COBOL na maszynę ZAM 41 została za-
programowana w EOL'u. Innym przykładem zasto-
sowań EOL'u może być przekształcanie symbo-
liczne wyrażeń arytmetycznych lub logicznych -
z tego typu zastosowań pochodzi nazwa języka -
Expression Oriented Language.

Język EOL był już opracowany w kilku kolej-
nych wersjach.

EOL-1 oraz EOL-2 były opracowane koncepcyjnie w latach 1965/66 w Instytucie Maszyn Matematycznych w Warszawie [1].

EOL-2 został następnie udoskonalony oraz
zrealizowany w roku 1967 na maszynie ZAM 41
przez zespół: Zdzisława Wrotek, Jan Walasek
i Iwona Messner [2]. Ta właśnie wersja

EOL VIII

EOL' u została opisana w niniejszym podręczniku.

EOL-3 opracowany został w roku 1967 na uniwersytecie Illinois [3] i zrealizowany w roku 1968 na maszynach IBM 7094 oraz IBM 360. Stał on się punktem wyjścia dla opracowania języka NUCLEOL [4] oraz prac nad EOL-4 traktowanych jako wspólne przedsięwzięcie Uniwersytetu Illinois oraz Uniwersytetu Colorado.

Nie wątpię, że przedstawiona tutaj wersja EOL-2 znajdzie szersze grono użytkowników w naszym kraju.

Leon Łukaszewicz

Warszawa, 10.12.1969 r.

1. WSTĘP

1.1. Przeznaczenie języka EOL

EOL jest prostym językiem do manipulacji symbolami.

EOL ma na celu ułatwienie pisania programów w zakresie przykładowo następującej problematyki:

- a/ Tłumaczenie z jednego formalnego języka na drugi, w szczególności w zakresie języków programowania.
- b/ Przekształcanie wyrażeń arytmetycznych lub logicznych, na przykład symboliczne różniczkowanie funkcji.
- c/ Ułatwienie wymiany informacji pomiędzy operatorem a maszyną.

Przy projektowaniu języka EOL starano się uzyskać język o stosunkowo prostej budowie, łatwy do nauczenia się oraz realizacji.

W języku EOL zastosowano wiele idei użytych wcześniej w innych językach do manipulacji symbolami, a w szczególności w językach IPL-V oraz COMIT.

1.2. Ogólna struktura języka

W języku EOL wszystkie programy oraz dane wejściowe i wyjściowe mają postać ciągów, złożonych ze znaków pisarskich. W szczególności każdy program w języku EOL może stanowić dane wejściowe lub być wynikiem działania innego programu w tym języku.

Programy w języku EOL składają się z rozkazów i deklaracji.

Wykonanie programu polega na kolejnym wykonywaniu rozkazów zawartych w tym programie. Z budowy programu wynika, który z jego rozkazów ma być wykonany jako pierwszy /patrz 5.11.3/. Po zakończeniu wykonania każdego rozkazu, o ile treść jego nie mówi wyraźnie inaczej, wykonany zostaje rozkaz występujący w najbliższej kolejności jego wypisania w programie, czyli tak zwanej sekwencji normalnej. Wykonanie każdego rozkazu powoduje z zasady wykonanie prostej czynności jak zmiana wartości zmiennej lub zmiana normalnej sekwencji wykonania rozkazów.

Program w języku EOL składa się z ciągu sekcji. Wewnątrz sekcji rozkazy i deklaracje mogą być grupowane w zależne od siebie lub niezależne procedury. Struktura procedur języka EOL wzorowana jest na PL/1.

Ponadto w języku EOL przewidziana jest możliwość deklarowania makrodefinicji, pozwalających na automatyczną modyfikację programu przed jego wykonaniem /patrz 7/.

1.3. Formalizm opisu składni

W niniejszej publikacji opis składni programu i danych sporządzono, wzorując się na notacji zaproponowanej przez Backus'a i zastosowanej w opisie języka ALGOL.

Notację tę uzupełniono następującymi konwencjami:

$\{a|b\}$ oznacza "dokładnie jeden z symboli a, b"

$[a|b]$ oznacza "żaden lub jeden z symboli a, b"

Powyższe reguły uogólnia się w sposób naturalny na dowolną ilość symboli.

$\{a\}..$ oznacza "dowolny ciąg niepusty symboli, z których każdy jest określony przez a"

$[a]..$ oznacza "dowolny, ewentualnie pusty, ciąg symboli, z których każdy jest określony przez a".

Przykłady

$\{A|B\}$ może oznaczać A

$\{A|B\}..$ " " ABA

$[A|B]..$ " " ciąg pusty

W niniejszej publikacji dla zaznaczenia dosłowności stosować będziemy symbol cudzysłów np. "..."

1.4. Wersje języka EOL

Istnieją dwie wersje językowe EOL-u - polska i angielska. Program może się składać z ciągu sekcji napisanych zarówno w wersji polskiej jak i angielskiej. W obrębie jednej sekcji należy stosować wyłącznie słowa kluczowe polskie bądź też angielskie.

W niniejszym opracowaniu podano postać syntaktyczną poszczególnych jednostek w obu wersjach.

1.5. Notacja wartości zmiennej

W przykładach podanych w niniejszej publikacji celem ilustracji wartości zmennych stosować będziemy następującą notację pomocniczą:

< nazwa zmiennej > : < wartość zmiennej >

Do opisu wartości zmiennej oprócz symboli podstawowych używane są znaczniki.

<znacznik> ::= ${}^{\wedge} | {}^{\circ} | {}^{\wedge} | {}^{\vee} | \{$

gdzie:

- - znacznik słowa,
- \circ - " liczby,
- \wedge - " adresu,
- \vee - " zapisu,
- \uparrow - " wskazówkii.

Pojęcia: słowo, liczba, adres, zapis i wskazówka zostały opisane w rozdziale 3.

Przykłady:

I1 : ABCD

I3 :

E4 : "A1" = "3
P4 : ^"BC"D°9f"-R

Należy podkreślić, że zapisy powyższego typu mają znaczenie wyłącznie pomocnicze i nie mają żadnego wpływu na przebieg wykonania programu lub strukturę danych.

Wprowadzenie do EOL - Języka komputerowego opartego o kod ASCII. Wprowadzenie do EOL - Języka komputerowego opartego o kod ASCII. Wprowadzenie do EOL - Języka komputerowego opartego o kod ASCII. Wprowadzenie do EOL - Języka komputerowego opartego o kod ASCII.

2. SYMBOLE PODSTAWOWE

Język EOL zbudowany jest z następujących symboli podstawowych:

2.1. Znaki

`<zna&kgt; ::= <litera>|<cyfra>|<zna&specjalny>`

2.1.1. Litery

`<litera> ::= A|B....|Z`

2.1.2. Cyfry

`<cyfra> ::= 0|1|2.....|9`

2.1.3. Znaki specjalne

`<zna&specjalny> ::= _|+|;|-|,|:|'|λ`

Znaczenie symboli:

a/λ - oznacza przejście do nowej linii,
b/_ - " spację.

Symboli spacji /_ / i przecinka /,/ mogą być zastąpione przez dowolną niepustą kombinację spacji i przecinków.

Każdy znak ";" , ":" i "λ" może być otoczony z obu stron przez dowolną kombinację spacji i przecinków.

Należy zauważyć, że zbiór znaków języka EOL jest węższy niż zbiór znaków dla danych. Pełny zbiór znaków dla danych zawarty jest w Dodatku A.

3. ZMIENNE I ICH WARTOŚCI

W programach operujemy na ograniczonej ilości zmiennych, wybranych z następującego zbioru:

16	zmiennych wejścia	I ₁ , I ₂ , ..., I ₁₆ ,
16	"	wyjścia Q ₁ , Q ₂ , ..., Q ₁₆ ,
32	"	wyrażeń E ₁ , E ₂ , ..., E ₃₂ ,
32	"	plików P ₁ , P ₂ , ..., P ₃₂ ,
1	zmienna logiczna H,	
1	"	stosu adresów rozkazów /SAR/.

W trakcie wykonywania programu każdej z powyższych zmiennych przypisywane są różne kolejne wartości, których struktura opisana jest poniżej.

Sposób przypisania zmiennym wejścia lub wyjścia urządzeń wejścia lub wyjścia podano w Dodatku C.

3.1. Wejście

<wartość wejścia> ::= <ciąg znaków>
 <ciąg znaków> ::= [<zak>] . .

Przykłady

I1 :JAN - KOWALSKI

I3 :A = B*(C+D) - λ

3.2. Wyjście

< wartość wyjścia > ::= < ciąg znaków >

Przykłady

Q2 :X = 3.1415; END

3.3. Wyrażenie

< wartość wyrażenia > ::= [< składnik >] . .

< składnik > ::= < słowo > | < liczba > | < adres zapisu >

3.3.1. Słowo

< słowo > ::= {znak} . .

Przykłady

ABC

X=A+B - λ;

38

Słowa służą na ogół do zapisu tekstów.

3.3.2. Liczba

< liczba > ::= °< całkowita >

< całkowita > ::= [+ | -] {< cyfra >} . .

Liczby przedstawiają wielkości wyrażone w układzie dziesiętnym. Bezwzględna wartość liczby nie może być większa niż 8388607.

Przykłady

°36 °-48 °0006

3.3.3. Adres zapisu

$\langle \text{adres zapisu} \rangle ::= \{\langle \text{znak} \rangle\} . .$

Adresem zapisu jest pewien symbol, określający jednoznacznie położenie tego zapisu w pliku.

Przykłady

Wyrażenia mogą przyjmować przykładowo następujące wartości:

E1 : $\overline{-X1} = \overline{-A} * (\overline{-B} + \overline{-GAMMA})$

E7 : $\overline{-Z} + \overline{-Y} = \overline{-X}$

E16 : ${}^{\circ} 13^{\wedge} \text{POB} - \lambda$

3.4. Pliki

$\langle \text{wartość pliku} \rangle ::= [\langle \text{zapis} \rangle] . . | [\langle \text{zapis} \rangle] . .$
 $\langle \text{zapis} \rangle ::= \langle \text{wartość wyrażenia} \rangle$

Przykłady

P1 : \uparrow

P3 : $\vee \overline{-A} \overline{-BC} {}^{\circ} 38 \uparrow$

P6 : $\vee a \vee b \vee c \vee d \vee e$

gdzie a,b,... oznaczają zapisy.

3.5. Zmienna logiczna

$\langle \text{wartość logiczna} \rangle ::= + | -$

Przykłady

H : +

H : -

3.6. Stos adresów rozkazów

<wartość stosu adresów> ::= [<adres rozkazu>].
<adres rozkazu> ::= ^{<zakres>} . .

Stos adresów rozkazów służy do zapamiętywania adresów rozkazów typu WYKONAJ /CALL/, które jednoznacznie wyznaczają pozycję tych rozkazów w programie.

Adresy te mogą być następnie wykorzystane przez rozkaz WRÓĆ /RETURN/.

P r z y k l a d

SAR : ^38^63^18

3.7. Konwencja wartości początkowych

Przyjmuje się, że w momencie rozpoczęcia wykonywania programu zmienna H ma wartość "+", natomiast wszystkie pozostałe zmienne są wyzerowane, to znaczy odpowiadające im wartości są ciągami pustymi.

4. POSTAĆ OGÓLNA ROZKAZÓW

4.1. Struktura rozkazu

Struktura każdego rozkazu jest następująca:

`<operator>[<pierwszy argument> [, <drugi argument>[, <trzeci argument>]]]`

Operator jest słowem kluczowym w postaci ciągu dużych liter np. MOVE, CZYTAJ, WYKONAJ.

Postać i znaczenie różnego typu argumentów opisane są poniżej.

4.2. Argumenty

Argumenty w rozkazach mają jedną z następujących postaci:

Litera indeksowana

Etykieta

Symbol klasy znaków

Tekst

Całkowita

W niektórych rozkazach pominięcie ostatniego argumentu może mieć sprecyzowane znaczenie.

Z każdą z podanych powyżej postaci argumentu związane jest jego znaczenie opisane poniżej.

4.2.1. Litera z indeksem

Argument tego typu ma postać

```
<litera indeksowana> ::= <litera argumentu>
<indeks>
<litera argumentu> ::= A|B|C|D|E|I|N|P|T|Y|Z|K
<indeks> ::= 1|2...|32
```

Argument w postaci <litera indeksowana> służy do określenia zmiennej oraz wskazuje na sposób operowania wartością tej zmiennej.

Wielkość <litera argumentu> może określać:

- a/ sposób czytania wartości zmiennej,
- b/ sposób dopisywania do wartości zmiennej,
- c/ sposób wykorzystania wartości zmiennej.

Ponadto w niektórych rozkazach występują litery posiadające jeszcze inne znaczenie.

Wielkość <indeks> jest rzeczywistym indeksem zmiennej. Dla przejrzystości opisu rozkazów wprowadzamy następujące symbole dla oznaczenia indeksów:

```
<n> ::= <indeks>
<m> ::= <indeks>
<k> ::= <indeks>
```

4.2.1.1. Sposób czytania

Sposób czytania określony jest przez poszczególne litery następująco:

- A czytanie kolejnych składników początkowych z wyrażenia E z jednoczesnym usuaniem tych składników,
- B czytanie kolejnych składników początkowych z wyrażenia E bez ich usuwania,
- I czytanie kolejnych znaków z wejścia I, z jednoczesnym ich usuwaniem,
- C czytanie kolejnych zapisów znajdujących się bezpośrednio za wskazówką w pliku P z jednoczesnym usuwaniem każdego przeczytanego zapisu,
- D czytanie kolejnych zapisów znajdujących się bezpośrednio za wskazówką w pliku P połączone z jednoczesnym przeskokiem wskazówki o każdy przeczytany zapis. Przeczytane zapisy nie są usuwane z pliku P.

4.2.1.2. Sposób dopisywania

Sposób dopisywania określony jest przez poszczególne litery następująco:

- A dopisywanie na początek E ciągu składników z odwróceniem ich pierwotnej kolejności,
- B dopisywanie na początek E ciągu składników z zachowaniem ich pierwotnej kolejności,
- I dopisywanie na koniec E ciągu składników z odwróceniem ich pierwotnej kolejności,
- Z dopisywanie na koniec E ciągu składników z zachowaniem ich pierwotnej kolejności,

- Q dopisywanie na końcu Q kolejnych składników wyrażenia E,
- C dopisywanie ciągu zapisów bezpośrednio za wskazówką ↑ w pliku P wraz z jednoczesnym odwróceniem pierwotnego porządku tych zapisów,
- D dopisywanie ciągu zapisów bezpośrednio za wskazówką ↑ w pliku P bez zmiany porządku tych zapisów. Jednocześnie wskazówka ↑ umieszczona zostaje bezpośrednio za ostatnim elementem dopisanego ciągu.

4.2.1.3. Sposób wykorzystania

Przez w y k o r z y s t a n i e składnika rozumiemy jego użycie jako argumentu porównania, argumentu w działaniach arytmetycznych itp.

- A wykorzystanie pierwszego składnika E, a następnie usunięcie tego składnika z wyrażenia,
- B wykorzystanie pierwszego składnika E bez jego usunięcia z wyrażenia,
- T porównanie z dowolnym składnikiem E bez jego usunięcia. Porównanie uznaje się za spełnione, gdy jest spełnione w stosunku do chociaż jednego składnika E,
- E porównanie z całym wyrażeniem E bez zmiany wartości tego wyrażenia lub użycie całego wyrażenia,
- I porównanie z pierwszym znakiem wejścia I bez jego usunięcia.

4.2.1.4. Inne litery

Podane niżej litery mają następujące znaczenie:

- N argument określa wyrażenie, którego pierwszy składnik należy interpretować jako liczbę,
- K argument określa wyrażenie, do którego jako pierwszy składnik dopisuje się liczbę wskazującą położenie elementu,
- P argument odnosi się do pliku.

4.2.2. Etykieta

`<etykieta> ::= <identyfikator>`

`<identyfikator> ::= <litera>[<litera>|<cyfra>]. .`

Długość identyfikatora nie może przekraczać 60 znaków. Etykiety stosowane są w programach do oznaczania rozkazów, zwrotnic i procedur.

4.2.3. Symbol klasy znaków

W wielu rozkazach zachodzi potrzeba określenia klasy znaków.

`<klasa> ::= {L|D|B|R}... .`

Znaczenie powyższych symboli jest następujące:

- L oznacza duże litery,
- D " cyfry,
- B " znak "_",
- R " pozostałe znaki.

Klasy określone przez powyższe symbole są rozłączne, a jednocześnie wyczerpują wszystkie stosowane znaki.

4.2.4. T e k s t

`<tekst> ::= <ciąg znaków>`

Ze względu na budowę tekstów niektóre znaki mają reprezentację dwuznakową a mianowicie:

- * L oznacza symbol nowej linii
- * , " " ! /apostrof/
- * * " " *

Przyjęto również dwuznakową reprezentację dla nietypowych symboli kodu wewnętrznego. Szczegóły na ten temat zawiera Dodatek A. Długość tekstu nie może przekraczać 60 znaków. Sekcja programu może zawierać co najwyżej 128 różnych tekstów więcej niż jednoznakowych.

4.2.5. C a ɼ k o w i t a

Liczby całkowite są używane w tekstach, jako argumenty rozkazów arytmetycznych, rozkazów porównania i w rozkazie UMIESĆ.

Na równi z liczbą całkowitą w podanych rozkazach może wystąpić wielkość POLE /SPACE/.

Wielkość ta określa ilość bloków pamięci bębnowej zajętych przez pliki. Długość bloku wynosi 128 słów.

Sekcja programu nie może zawierać więcej niż 128 różnych liczb, których wartość bezwzględna przekracza 15.

4.2.6. Brak argumentu

W niektórych rozkazach brak ostatniego argumentu posiada ścisłe sprecyzowane znaczenie /Patrz 5.1.1./.

4.3. Oznaczoność wyniku

W opisie działania wielu rozkazów przyjmuje się, że argumenty spełniają pewne założenia.

Przykładowo:

W rozkazach arytmetycznych zakłada się, że wielkości określone przez oba argumenty rozkazu są liczbami.

W rozkazach zawierających porównania zakłada się, że obie porównywane wielkości są tego samego typu, to znaczy są to dwa słowa, dwie liczby lub dwa adresy zapisów.

W przypadku, gdy jakikolwiek warunek założony w opisie rozkazu nie jest spełniony, wynik działania tego rozkazu jest nieokreślony.

5. ROZKAZY, DEKLARACJE, KOMENTARZE

```
<rozkaz> ::= <przesłanie> | <umieszczenie> |
             <usunięcie> | <relacja> |
             <rozkaz sterujący> |
             <rozkaz arytmetyczny> |
             <przekształcenie> | <szukanie> |
             <zliczanie> | <różne>
```

5.1. Przesłania

```
<przesłanie> ::= <prześlij> | <czytaj> | <pisz> |
                  <kopiuj> | <wstaw> | <pobierz> |
                  <schowaj> | <ustaw> | <zamień>
```

5.1.1. Prześlij

```
<prześlij> ::= {PRZEŚLIJ | MOVE} - {A|B} <n>,  

                  {A|B|Y|Z} <m>, <test przesłania>  

<test przesłania> ::= <ilość> | <klasa> | <tekst>  

                  {B|T} <k>  

<ilość> ::= <całkowita> | N <k>
```

Przenies kolejne składniki z E <n> do E <m>.

Pierwszy argument określa sposób pobierania składników z E <n>. Drugi argument określa sposób dopisywania tych składników do E <m>.

Trzeci argument określa moment zakończenia przeniesień. W szczególności składowa <ilość> określa ilość przéniesionych składników.

Pozostałe składowe trzeciego argumentu określają najbliższy składnik, który nie zostaje przeniesiony i na którym kroki przeniesienia zostają zakończone. Składnik ten jest określony w myśl reguł podanych w rozdziale 4.

Brak argumentu oznacza, że należy przenieść wszystkie składniki.

Jeżeli wszystkie składniki z $E <n>$ zostaną przeniesione do $E <m>$ zanim nastąpi moment zakończenia przeniesień określony przez trzeci argument, to wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

P r z y k l a d y

Załóżmy, że

$E1 : \neg XA \neg 1 = \neg 18^{\circ} 3^-;$

$E3 : \neg ALFA __$

$E8 : \neg ; \neg =$

$H : +$

Wykonanie wypisanych poniżej rozkazów PRZEŚLIJ powoduje w stosunku do powyższych wartości następujące przekształcenia:

PRZEŚLIJ A1,A3,2	$E1 : \neg = \neg 18^{\circ} 3^-;$
PRZEŚLIJ A1,A3,'='	$E3 : \neg 1 \neg XA \neg ALFA __$
PRZEŚLIJ A1,A3, R	
PRZEŚLIJ A1,A3,T8	
PRZEŚLIJ B1,B3, 2	$E3 : \neg XA \neg 1 \neg ALFA __$
PRZEŚLIJ B3,B3	$E3 : \neg ALFA __ \neg ALFA __$
PRZEŚLIJ A3,Y3, R	$H : +$ $E3 : \neg __ \neg ALFA$ $H : -$

5.1.2. Czytaj

$\langle \text{czytaj} \rangle ::= \{\text{CZYT AJ} \mid \text{READ}\}_I \langle n \rangle ,$
 $\{\text{A} \mid \text{B} \mid \text{Y} \mid \text{Z}\} \langle m \rangle [, \langle \text{test przesłania} \rangle]$

Pobieraj kolejne znaki początkowe z $I \langle n \rangle$, utwórz z nich jedno słowo i dopisz do $E \langle m \rangle$.

Znaki pobierane z $I \langle n \rangle$ są jednocześnie z tego ciągu usuwane. Porządek znaków w utworzonym słowie jest zgodny z porządkiem ich pobierania.

Drugi argument określa sposób dopisania utworzonego słowa do $E \langle m \rangle$.

Trzeci argument lub jego brak jest określony podobnie jak trzeci argument w rozkazie $\langle \text{prześlij} \rangle$ z tą różnicą, że odnosi się nie do kolejnych składników wyrażenia, a do kolejnych znaków w ciągu $I \langle n \rangle$.

W przypadku gdy trzeci argument rozkazu określa ciąg znaków, to uwzględniony jest jedynie pierwszy znak.

Przykłady

Niech

$I1 : X1=28; \lambda __$
 $E3 : \neg PQ$
 $H: +$

Wypisane poniżej rozkazy powodują:

CZYT AJ I1,A3,5

CZYT AJ I1,Y3,D

$I1 : ; \lambda __$

$E3 : \neg X1=28 \neg PQ$

$I1 : 1=28; \lambda __$

$E3 : \neg PQ \neg X$

CZYT AJ I1,A3,'.' I1 :
 E3 : - X1=28;λ - PQ
 H : -

5.1.3. P i s z

$\langle \text{pisz} \rangle ::= \{\text{PISZ} | \text{WRITE}\} \cup \{\text{A} | \text{B}\} \langle n \rangle, \text{Q} \langle m \rangle [, \langle \text{test przesłania} \rangle]$

Pobieraj kolejne słowa z E $\langle n \rangle$ i dopisuj je na końcu Q $\langle m \rangle$ jako następujące po sobie ciągi znaków.

Sposób pobierania składników określony jest przez pierwszy argument. Znaczenie trzeciego argumentu lub jego brak jest takie samo jak w rozkazie $\langle \text{prześlij} \rangle$.

P r z y k l a d y

Niech

E7 : - X1 = - 13 - λ
 Q2 : WYNIK -

Wykonanie poniższych rozkazów powoduje:

PISZ A7 ,Q2

E7 :

Q2 : WYNIK.X1=13_λ

PISZ B7,Q2,1

E7 : - X1 = - 13 - λ

Q2 : WYNIK.X1

5.1.4. K o p i u j

$\langle \text{kopiuj} \rangle ::= \{\text{KOPIUJ} | \text{COPY}\} \cup \{\text{C} | \text{D}\} \langle n \rangle, \{\text{C} | \text{D}\} \langle m \rangle [, \langle \text{test kopiowania} \rangle]$
 $\langle \text{test kopiowania} \rangle ::= \langle \text{ilosc} \rangle | \langle \text{tekst} \rangle | \{\text{B} | \text{T} | \text{E}\} \langle k \rangle$

Przenieś kolejne zapisy z P <n> do P <m>.

Pierwszy argument określa sposób pobierania zapisów z P <n>. Drugi argument określa sposób dopisywania tych zapisów do P <m>. Trzeci argument określa moment zakończenia przeniesień. W szczególności składowa <ilość> oznacza ilość przeniesionych zapisów.

Pozostałe składowe trzeciego argumentu określają najbliższy zapis, który nie zostaje przeniesiony i na którym wykonywanie rozkazu zostaje zakończone.

Zapis ten jest określony w myśl reguł podanych w rozdziale 4.

Jeśli trzeci argument jest wymieniony i wszystkie zapisy z P <n> zostaną przeniesione do P <m> zanim nastąpi moment zakończenia przeniesień określony przez trzeci argument, to wykonanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

Brak trzeciego argumentu oznacza przeniesienie do P <m> wszystkich zapisów występujących w P <n> z pozostawieniem wartości H bez zmiany.

Przykład

Przyjmijmy, że

P₁ : $\forall a \nmid \forall b \forall c \forall d$

P₃ : $\forall r \nmid \forall s \forall t$

H : +

gdzie a,b,c,... przedstawiają pewne zapisy. Podane poniżej rozkazy powodują następujące przekształcenia:

KOPIUJ D1,D3,2	P1 : $\forall a \forall b \forall c \exists \forall d$ P3 : $\forall r \forall b \forall c \exists \forall s \forall t$ H : +
KOPIUJ C1,C3,4	P1 : $\forall a \exists$ P3 : $\forall r \exists \forall d \forall c \forall b \forall s \forall t$ H : -

5.1.5. W s t a w

$\langle wstaw \rangle ::= \{ WSTAW | PUT \} \cup \{ A | B \} \langle n \rangle, \{ C | D \} \langle m \rangle$

Pobierz kolejne składniki z E $\langle n \rangle$, utwórz z nich jeden zapis i umieść w P $\langle m \rangle$. Pierwszy argument określa sposób pobierania składników z E $\langle n \rangle$. Drugi argument określa sposób dopisania zapisu do P $\langle m \rangle$.

W przypadku, gdy wyrażenie E $\langle n \rangle$ jest puste, wynikiem działania rozkazu jest jedynie przypisanie H wartości " - ".

5.1.6. P o b i e r z

$\langle pobierz \rangle ::= \{ POBIERZ | GET \} \cup \{ C | D \} \langle n \rangle, \{ A | B | Y | Z \} \langle m \rangle$

Pobierz z P $\langle n \rangle$ zapis i jego składniki dopisz do E $\langle m \rangle$, w sposób określony przez drugi argument.

Pierwszy argument określa sposób pobrania zapisu z P $\langle n \rangle$. W przypadku, gdy wskazówka znajduje się na końcu pliku wynikiem działania rozkazu jest jedynie przypisanie H wartości " - ".

5.1.7. Schowaj

$\langle \text{schowaj} \rangle ::= \{\text{SCHOWAJ} | \text{SAVE}\} __ P \langle n \rangle,$
 $\{\text{A} | \text{B} | \text{Y} | \text{Z}\} \langle m \rangle$

Adres zapisu w pliku $P \langle n \rangle$, który występuje bezpośrednio przed wskazówką dopisz do wyrażenia $E \langle m \rangle$ w sposób określony przez drugi argument rozkazu.

5.1.8. Ustaw

$\langle \text{ustaw} \rangle ::= \{\text{USTAW} | \text{RESTORE}\} __ \{\text{A} | \text{B}\} \langle n \rangle, P \langle m \rangle$

Pobierz z $E \langle n \rangle$ adres zapisu w sposób określony przez pierwszy argument, a następnie wskazówkę w $P \langle m \rangle$ umieść bezpośrednio po zapisie wskazanym przez ten adres. Jeśli zapis ten został uprzednio usunięty z pliku, to wynik tego rozkazu jest nieokreślony.

5.1.9. Zamień

$\langle \text{zamień} \rangle ::= \{\text{ZAMIEN} | \text{EXCHANGE}\} __ \{E \langle n \rangle, E \langle m \rangle | P \langle n \rangle, P \langle m \rangle\}$

Zamień wzajemnie pomiędzy sobą wartości wyrażeń lub plików, wskazanych przez dwa argumenty rozkazu.

Przykład

Niech

$E1 := X1^Q$
 $E2 := X2^3$

Po wykonaniu poniższego rozkazu mamy

ZAMIEN	E1, E2	E1 := X2^3
		E2 := X1^Q

5.2. Umieszczenie

```

<umieszczenie> ::= <umieść w wyrażeniu>|
    <umieść w pliku>|<umieść na wyjściu>
<umieść w wyrażeniu> ::= {UMIEŚĆ | SET}_
    {<całkowita>|<tekst>}, {A|B|Y|Z} <n>
<umieść w pliku> ::= {UMIEŚĆ | SET}
    {<całkowita><tekst>}, {C|D} <n>
<umieść na wyjściu> ::= {UMIEŚĆ | SET}<tekst>,
    Q <n>

```

Wielkość określoną przez pierwszy argument dopisz do zmiennej określonej przez drugi argument. Argument ten określa jednocześnie sposób dopisania podanej wielkości. Wielkość **<całkowita>** określa składnik wyrażenia, który jest liczbą o podanej wartości.

Jeżeli wskazaną zmienną jest plik, to podana wielkość dopisana jest jako odrębny zapis, składający się z jednego składnika.

Przykłady

Niech
 E8 : "ALFA" —
 Q3 : X1 =

Wypisane niżej rozkazy powodują

UMIEŚĆ 3846,B8	E8 : "3846 ALFA" —
UMIEŚĆ 'BETA', Y8	E8 : "ALFA" — BETA
UMIEŚĆ '*L', Q3	Q3 : X1=λ

5.3. Usunięcie

```

<usunięcie> ::= <usuń z wyrażenia>|
    <usuń z wejścia>|<usuń z pliku>
<usuń z wyrażenia> ::= {USUŃ | CLEAR}_
    A<n>
    [, <test przesłania>]

```

<usuń z wejścia> ::= {USUŃ|CLEAR} \sqcup I<n> [,
 <test przesłania>]

<usuń z pliku> ::= {SKREŚL|DELETE} \sqcup C <n> [,
 <test kopiowania>]

Usuń z wejścia, wyrażenia lub pliku, określonego przez pierwszy argument, elementy w ilości określonej przez drugi argument. Znaczenie drugiego argumentu jest takie samo jak odpowiednio w rozkazach <prześlij>, <czytaj> lub <kopiuj>.

P r z y k l a d y

Niech

E1 : $\neg XA = \neg 18$

I3 : $\lambda \ldots \text{SKIP}$

P7 : $v_a \uparrow v_b v_c$

Wypisane poniżej rozkazy powodują

USUŃ A1, 1

E1 : $\neg = \neg 18$

USUŃ I3, 1

I3 : $\ldots \text{SKIP}$

SKREŚL C7

P7 : $v_a \uparrow$

5.4. Relacja

<relacja> ::= <relacja arytmetyczna> |
 <relacja tekstowa> |
 <relacja zapisu>

5.4.1. Relacja arytmetyczna

<relacja arytmetyczna> ::=
 <relacja arytmetyczna równości> |
 <relacja arytmetyczna uporządkowania>

$\langle \text{relacja arytmetyczna równości} \rangle ::=$
 $\{ \text{RW} | \text{EQ} \} \cup \{ \text{A} | \text{B} | \text{T} \} \langle n \rangle, \{ \langle \text{całkowita} \rangle | N \langle m \rangle \}$

$\langle \text{relacja arytmetyczna uporządkowania} \rangle ::=$
 $\{ \{ \text{WK} | \text{MN} \} | \{ \text{GT} | \text{LT} \} \} \cup B \langle n \rangle, \{ \langle \text{całkowita} \rangle | B \langle n \rangle \}$

Jeśli pomiędzy liczbami wskazanymi przez pierwszy i drugi argument zachodzi relacja wskazana przez operator rozkazu, to wartość zmiennej H pozostaw bez zmiany. W przeciwnym przypadku zmiennej H nadaj wartość " - ".

W przypadku użycia pierwszego argumentu w postaci A <n> usunięcie argumentu ma miejsce tylko wtedy, gdy relacja jest spełniona.

Znaczenie operatorów relacji arytmetycznych jest następujące:

RW /EQ/ - równy
 WK /GT/ - większy
 MN /LT/ - mniejszy

P r z y k ł a d y

Niech

E8 : °8°3

E9 :

RW A8,8

E8: °3

H : bez zmian

RW A8,10

E8: bez zmian

H : -

WK B9,1

wynik nieokreślony

5.4.2. Relacja tekstowa

$\langle \text{relacja tekstowa} \rangle ::=$
 $\langle \text{relacja tekstowa równości} \rangle |$
 $\langle \text{relacja tekstowa uporządkowania} \rangle$

$\langle \text{relacja tekstowa r\ównosci} \rangle ::=$

$\{\text{RW}|\text{EQ}\} \cup \{\text{A}|\text{B}|\text{T}|I\} \langle n \rangle, \langle \text{tekst} \rangle | \langle \text{klasa} \rangle | \{\text{B}|\text{T}\}^m \}$

$\langle \text{relacja tekstowa uporządkowania} \rangle ::=$

$\{\{\text{PP}|\text{NS}\} | \{\text{PR} | \text{FL}\}\} \cup \text{B} \langle n \rangle, \{ \langle \text{tekst} \rangle | \text{B} \langle n \rangle \}$

Jeśli pomiędzy słowami wskazanymi przez pierwszy i drugi argument rozkazu zachodzi relacja wskazana przez operator rozkazu, to wartość zmiennej H pozostaw bez zmiany. W przeciwnym przypadku umiennej H nadaj wartość " - ".

W przypadku użycia pierwszego argumentu w postaci A <n> usunięcie argumentu ma miejsce tylko wtedy, gdy relacja jest spełniona.

Znaczenie operatorów relacji tekstowych jest następujące:

RW /EQ/ - równy

PP /PR/ - poprzedza,

NS /FL/ - następuje.

W przypadku, gdy pierwszy argument ma postać I <n> w porównaniach uwzględniany jest tylko pierwszy znak drugiego argumentu. W tekście relacji r\ówności jako argumenty mogą występować adresy zapisów.

Mówimy, że dwa słowa są r\ówne, jeśli są one identyczne. Mówimy, że słowo X poprzedza słowo Y, jeśli przy porównaniu tych słów kolejno znak po znaku przy pierwszej parze różnych znaków znak słowa X jest wcześniejszy od odpowiadającego mu znaku słowa Y.

Przyjmujemy przy tym, że brak znaku jest wcześniejszy od jakiegokolwiek znaku.

Mówimy, że słowo X następuje po słowie Y, jeśli słowo Y poprzedza słowo X. Uporządkowanie znaków: patrz Dodatek A.

P r z y k l a d y

Niech

E1 : $\neg A^\circ BETA^{\circ 5}$

E2 : $\neg Q^\circ A^\circ 3$

I7: ALFA

Wykonanie następujących rozkazów powoduje

RW B1, 'A'

H : bez zmian

RW I7, B1

E1: $\neg A^\circ BETA^{\circ 5}$

H : bez zmian

5.4.3. R e l a c j a z a p i s u

$\langle \text{relacja zapisu} \rangle ::= \{ RZ | ER \} \cup P \langle n \rangle ,$
 $\{ \langle \text{całkowita} \rangle | \langle \text{tekst} \rangle |$
 $\{ B | E | N | T \} \langle m \rangle \}$

Rozkaz powoduje porównanie pierwszego składnika lub całego zapisu z wartością określoną przez drugi argument. Znaczenie liter 2-go argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

W przypadku równości wartość H pozostaw bez zmian. W przeciwnym przypadku nadaj H wartość " - ".

P r z y k l a d

Niech

P1: $\forall a \forall b \nmid \forall \neg AB^\circ TOM$

E3: $\neg ROM \neg AB$

Wykonanie następującego rozkazu spowoduje:

RZ P1, T3
H :bez zmian

5.5. Rozkaz sterujący

`<rozkaz sterujący> ::= <skok> | <skok warunkowy> |
<skok zwrotnicowy> |
<skok powrotny>`

5.5.1. Skok

`<skok> ::= {{SKOCZ | WYKONAJ}} |
{GOTO | CALL}} | <etykieta>`

Przerwij normalną sekwencję rozkazów i przejdź do wykonania rozkazu oznaczonego etykietą. W przypadku rozkazu z operatorem WYKONAJ /CALL/ adres tego rozkazu umieść na wierzchołku stosu SAR.

5.5.2. Skok warunkowy

`<skok warunkowy> ::= {{SKOMI | SKOPL | WYMI |
WYPL}} | {GOMI | GOPL |
CAMI | CAPL}} | <etykieta>`

Jeśli spełnione są warunki wskazane przez operator rozkazu, to przejdź do rozkazu oznaczonego etykietą. W przypadku rozkazów z operatorem WYMI /CAMI/ i WYPL /CAPL/ adres tego rozkazu umieść na wierzchołku stosu SAR. W każdym przypadku zmiennej H nadaj wartość " + ".

Poszczególne operatory rozkazu wskazują następujące warunki:

SKOPL /GOPL/, WYPL /CAPL/	wartość H jest równa "+"
SKOMI /GOMI/, WYMI /CAMI/	wartość H jest równa "-"

Przykład

Jeśli H jest równe "-" to rozkaz WYMI ALFA powoduje umieszczenie adresu tego rozkazu na wierzchołku stosu SAR i przejście do miejsca programu oznaczonego etykietą ALFA oraz zmianę wartości H na "+".

5.5.3. Skok zwrotnicowy

```
<skok zwrotnicowy> ::= {{SKONA|SKOIN|WYNA|
WYIN}|{GONA|GOIN|
CANA|CAIN}}.{A|B}<n>,
<nazwa zwrotnicy>
```

Przerwij normalną sekwencję rozkazów i przejdź do rozkazu, którego etykieta określona jest przez pierwszy argument rozkazu oraz znajduje się na liście etykiet zwrotnicy o nazwie podanej jako drugi argument.

W przypadku operatorów WYNA /CANA/ oraz WYIN /CAIN/ adres rozkazu umieść na wierzchołku stosu SAR.

Typ zwrotnicy oraz związany z tym sposób wyboru etykiety określone są przez operator rozkazu w sposób następujący:

SKONA /GONA/, WYNA /CANA/ oznaczają, że rozkaz odnosi się do zwrotnicy typu NAZWA i wskazany składnik wyrażenia E <n> zawiera słowo równe wybranej etykiecie bądź tekstowi przypisanemu etykiecie na liście etykiet zwrotnicy.

SKOIN /GOIN/, WYIN /CAIN/ oznaczają, że rozkaz odnosi się do zwrotnicy typu INDEKS i wskazany składnik wyrażenia E <n> jest indeksem wybranej etykiety na liście zwrotnicy. Pierwsza etykieta na liście ma indeks równy jeden.

W przypadku, gdy w zwrotnicy typu NAZWA brak jest wskazanej etykiety bądź tekstu przypisanego etykiecie, wówczas sekwencja rozkazów nie zostaje przerwana, a wartość zmiennej SAR pozostaje bez zmian. Ponadto w przypadku operatora WYNA /CANA/ zmiennej H zostaje przypisana wartość " - ".

W przypadku, gdy pierwszy argument ma postać A <n> jego usunięcie ma miejsce tylko wtedy, gdy szukanie zostanie uwieńczone sukcesem. W przypadku, gdy w zwrotnicy typu INDEKS brak jest wskazanej etykiety, wówczas wynik działania rozkazu jest nieokreślony.

P r z y k l a d y

Niech będą zadeklarowane zwrotnice:

LX : INDEKS LA, LB, LC

KEY : NAZWA FOR, IF, BEGIN, END

Wartości E1 oraz E3 wynoszą:

E1 : "2-A"

E3 : "BEGIN-REAL-X";

Wówczas rozkaz

SKOIN B1, LX

Jest równoważny rozkazowi

SKOCZ LB

Podobnie rozkaz
 SKONA B3, KEY
 jest równoważny rozkazowi
 SKOCZ BEGIN

5.5.4. Skok powrotny

$\langle \text{skok powrotny} \rangle ::= \{\text{WRÓĆ} | \text{RETURN}\}$

Przerwij normalną sekwencję rozkazów i przejdź do rozkazu następnego w stosunku do rozkazu, którego adres umieszczony jest na wierzchołku stosu SAR. Jednocześnie usuń ten adres ze stosu.

W przypadku, gdy stos SAR jest pusty, znaczenie rozkazu WRÓĆ /RETURN/ jest nieokreślone.

5.6. Rozkaz arytmetyczny

$\langle \text{rozkaz arytmetyczny} \rangle ::= \{\{\text{DODAJ} | \text{ODEJMIJ} | \text{MNÓŻ} | \text{DZIEL}\} | \{\text{ADD} | \text{SUB} | \text{MULT} | \text{DIV}\}\}$
 $\quad \neg A \langle n \rangle , \langle \text{całkowita} \rangle$
 $\quad | B \langle m \rangle\}$

Wykonaj działanie arytmetyczne określone przez operator na liczbach określonych przez argumenty rozkazu i otrzymany wynik umieść na początku wyrażenia E $\langle n \rangle$ z równoczesnym usunięciem 1-go argumentu.

Znaczenie poszczególnych operatorów jest następujące:

DODAJ /ADD/ Dodaj do siebie dwie liczby, określone przez dwa argumenty rozkazu, a

następnie otrzymaną sumę umieść na początku wyrażenia E < n > .

MNÓŻ /MULT/ Pomnóż przez siebie dwie liczby, określone przez dwa argumenty rozkazu, a następnie otrzymany iloczyn umieść na początku wyrażenia E < n > .

ODEJMIJ /SUB/ Od liczby określonej przez pierwszy argument rozkazu odejmij liczbę określoną przez drugi jego argument, a następnie otrzymaną różnicę umieść na początku wyrażenia E < n > .

DZIEL /DIV/ Liczbę określoną przez pierwszy argument rozkazu podziel przez liczbę określoną przez drugi jego argument. Resztę z tego dzielenia umieść jako początkowy składnik E < n >, a całkowitą część ilorazu jako drugi składnik tego wyrażenia. Znak reszty jest zawsze równy znakowi dzielnej. W przypadku, gdy liczba określona przez drugi argument jest równa zeru wynik dzielenia jest nieokreślony.

P r z y k ł a d y

Niech

E1 : °9^-A

E3 : °-11^-B

Wykonanie poniższych rozkazów powoduje:

DODAJ A1, B3

E1 : ° -2^-A

DZIEL A1, B3

E1 : ° 9 ° 0^-A

5.7. Przekształcenie

<przekształcenie> ::= <przekształcenie na słowo>
 <przekształcenie na liczbę> | <zbijanie>
 <rozbijanie>

5.7.1. Przekształcenie na słowo

$\langle \text{przekształcenie na słowo} \rangle ::= \{\text{SŁOWO|WORD}\}$
 $\quad \cup \{A|B\} \langle n \rangle$

Liczbe stanowiącą początkowy składnik E $\langle n \rangle$ zamień na słowo stanowiące zapis tej liczby w układzie dziesiętnym i otrzymany rezultat umieść na początku E $\langle n \rangle$.

Pierwsza cyfra słowa jest różna od zera z wyjątkiem przypadku, gdy liczba równa się 0, a w przypadku liczby dodatniej pominięty jest znak " + ".

Znaczenie liter pierwszego argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

Przykłady

Niech

E3 : ${}^{\circ}6{}^{\circ}7$

E5 : ${}^{\circ}-11{}^{\circ}A$

E6 : ${}^{-}BA$

E10:

Wypisane poniżej rozkazy powodują:

SŁOWO A3

E3 : ${}^{-}6{}^{\circ}7$

SŁOWO B5

E5 : ${}^{\circ}-11{}^{\circ}-11{}^{\circ}A$

SŁOWO A6

wynik nieokreślony

SŁOWO B10

wynik nieokreślony

5.7.2. Przekształcenie na liczbę

$\langle \text{przekształcenie na liczbę} \rangle ::= \{\text{LICZBA|NUMBER}\} \cup \{A|B\} \langle n \rangle$

Słowo stanowiące początkowy składnik E <n> zamień na liczbę stanowiącą wartość tego słowa w układzie dziesiętnym i otrzymany rezultat umieść na początku E <n>.

Znaczenie liter pierwszego argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

P r z y k l a d y

Niech

E3 : -11-A

E5 : -07-B

Wykonanie poniższych rozkazów powoduje

LICZBA A3

E3 : °11-A

LICZBA B5

E5 : °-7--07-B

5.7.3. Z b i j a n i e

`<zbijanie> ::= {ZBIJ|COMPRESS} {A|B} <n> [,
 <test przesłania>]`

Początkowe składniki E <n> ułóż kolejno jeden za drugim i powstały w ten sposób jeden składnik umieść na początku E <n>. Pierwszy argument rozkazu określa sposób pobierania składników z E <n>. Drugi argument lub jego brak określa ostatni z pobranych składników, analogicznie jak w rozkazie <prześlij>.

Jeśli wszystkie składniki E <n> zostaną pobrane zanim warunek określony przez drugi argument zostanie spełniony, to wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana zostaje wartość " - ".

Przykłady

Niech

E2 : $\neg \text{ALFA} = \neg X + \neg 3$ H : +

Wykonanie poniższych rozkazów powoduje

ZBIJ A2, B

E2 : $\neg \text{ALFA} = X + 3$

ZBIJ A2, ;

E2 : $\neg \text{ALFA} = X + 3$ H : -

5.7.4. Rozbijanie

 $\langle \text{rozbijanie} \rangle ::= \{\text{ROZBIJ|SPLIT}\} \cup \{A|B\} \langle n \rangle [,$
 $\langle \text{test przesłania} \rangle]$

Słowo stanowiące pierwszy składnik E⟨n⟩ zamień na ciąg złożony z kolejnych znaków początkowych tego słowa i umieść go na początku E⟨n⟩.

Drugi argument rozkazu lub jego brak ma takie samo znaczenie w stosunku do kolejnych znaków początkowego słowa w E⟨n⟩ jak wyrażenie ⟨test przesłania⟩ lub jego brak w rozkazie ⟨czytaj⟩.

W przypadku gdy całe pierwsze słowo w E⟨n⟩ zostanie rozbite przed spełnieniem warunku określonego przez drugi argument rozkazu, wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

Przykłady

Niech

E1 : $\neg AB = 3 \neg C$

Wykonanie poniższych rozkazów powoduje

ROZBIJ A1,1

E1 : $\neg A \neg C$

ROZBIJ B1,1

E1 : $\neg A \neg AB = 3 \neg C$

ROZBIJ A1, B

E1 : $\neg A \neg B = \neg 3 \neg C$ H : -

5.8. Szukanie

$\langle \text{szukanie} \rangle ::= \langle \text{szukaj} \rangle | \langle \text{omiń} \rangle$

5.8.1. S z u k a j

$\langle \text{szukaj} \rangle ::= \{\text{SZUKAJ|SEARCH}\}_{-} \{A|B\} \langle n \rangle,$
 $\langle \text{nazwa zwrotnicy} \rangle$

Jeśli na liście zwrotnicy typu NAZWA o nazwie określonej przez ostatni argument rozkazu znajduje się etykieta / bądź tekst przypisany etykiecie/ określona przez pierwszy argument tego rozkazu, to wartość H pozostaw bez zmiany. W przeciwnym przypadku zmiennej H nadaj wartość "-".

W przypadku gdy pierwszy argument ma postać $A \langle n \rangle$ usunięcie następuje tylko wtedy, gdy szukanie zostaje uwieńczone sukcesem.

P r z y k l a d y

Niech dana będzie zwrotnica:

KLUCZ :NAZWA IF, PLUS: '+', MINUS: '-', BEGIN

oraz wartość zmiennych

E3 : -+ -A

E4 : ^MARMUR^BAR

Wypisane poniżej rozkazy powodują:

SZUKAJ A3, KLUCZ

E3 : -A

H : bez zmian

SZUKAJ A4, KLUCZ

E4 : bez zmian

H : -

5.8.2. O m i n

`<omín> ::= OMÍN | SHIFT ↦ P<n>
[,<test kopiowania>]`

Przesuwaj wskazówkę w pliku $P<n>$ do przo-
du aż do momentu określonego przez argument
`<test kopiowania>`. Jeżeli argument ten nie
występuje w rozkazie, przesuwaj wskazówkę aż
do końca pliku.

Argument `<test kopiowania>` określa moment
zatrzymania się wskazówki w ten sam sposób jak
ma to miejsce przy czytaniu pliku bez usuwania
zapisów w rozkazie `<kopiuj>`. Również w tych
samych okolicznościach zmiennej H zostaje na-
dana wartość `"-"`.

P r z y k l a d y

Przyjmijmy, że

$E_3 : \neg ALFA \neg BETA$

$E_{23} : \neg S$

$P_1 : \neg a \neg b \neg c \neg d \neg e$

gdzie $d : \neg ALFA \circ 3$

natomiast początkowe składniki w zapisach b
oraz c są różne od $\neg ALFA$. Wówczas wykonanie
następujących rozkazów powoduje:

$OMÍN P_1, B_3$

$P_1 : \neg a \neg b \neg c \neg d \neg e$

$OMÍN P_1, 10$

$P_1 : \neg a \neg b \neg d \neg e$

$H : -$

5.9. Zliczanie

`<zliczanie> ::= <zajdz> | <szukaj i licz> |
<omín i licz>`

5.9.1. Z n a j d ź

$\langle \text{znajdź} \rangle ::= \{\text{ZNAJDŹ} \mid \text{FIND}\} \cup \{A \mid B\} \langle n \rangle, K \langle m \rangle$
 $[, \langle \text{test przesłania} \rangle]$

Jeśli w wyrażeniu określonym przez pierwszy argument rozkazu znajduje się składnik określony przez trzeci argument rozkazu, to wartość H pozostaw bez zmiany, a na początku wyrażenia $E \langle m \rangle$ umieść liczbę określającą ilość składników $E \langle n \rangle$ poprzedzających ten składnik.

Pierwszemu składnikowi tego wyrażenia odpowiada liczba jeden. Jeśli takiego składnika brak, to jedynie zmiennej H nadaj wartość "-",

W przypadku braku trzeciego argumentu na początku $E \langle m \rangle$ umieszczona zostaje liczba wskazująca pełną ilość składników w $E \langle n \rangle$.

P r z y k l a d y

Przyjmijmy, że

$E_3 : \neg AX \neg AY \neg BX \neg CZ$

$E_4 : \neg Q$

Wypisane poniżej rozkazy powodują:

ZNAJDŹ A₃, K₄, 'BX'

E₃ : $\neg BX \neg CZ$

E₄ : ${}^o 2 \neg Q$

ZNAJDŹ B₃, K₃

E₃ : ${}^o 4 \neg AX \neg AY \neg BX \neg CZ$

5.9.2. S z u k a j i l i c z

$\langle \text{szukaj i licz} \rangle ::= \{\text{SZL} \mid \text{SEC}\} \cup \{A \mid B\} \langle n \rangle, K \langle m \rangle$
 $\langle \text{nazwa zwrotnicy} \rangle$

Wykonaj operację opisaną w 5.8.1. /Szukaj/. W przypadku znalezienia określonej wartości

na liście zwrotnicy umieść liczbę określającą jej położenie na liście na początku wyrażenia określonego przez drugi argument. Pierwszej wartości na liście zwrotnicy odpowiada liczba 1,

Przykład

Niech dana będzie zwrotnica

KLUCZ :NAZWA IF, FOR, BEGIN, END
oraz wartość zmiennej

E3 : "FOR" A

Poniższy rozkaz powoduje

SZL B3, K3, KLUCZ

E3 : "2" FOR" A

H : bez zmian

5.9.3. Ominięcie i licz

$\langle \text{ominięcie i licz} \rangle ::= \{\text{OML} \mid \text{SHIC}\} \cup P\langle n \rangle, K\langle m \rangle$
 $[, \langle \text{test kopiowania} \rangle]$

Wykonaj operację opisaną w 5.8.2. /Omini/.
Liczbe określającą ilość omieniętych zapisów umieść na początku wyrażenia określonego przez drugi argument.

Przykład

Przyjmijmy, że

E3 : "ALFA" BETA

E23 : "S

P1 : "a" "b" "c" "d" "e"

gdzie

d : "ALFA" 3

i pierwsze składniki zapisów b i c są różne od "ALFA"

Poniższy rozkaz powoduje

OML P1, K23, B3 E3 : "ALFA" BETA
 E23: °2-S
 P1 : 'a'b'c' 'd' 'e'

5.10. Różne

<różne> ::= <cofnij> | <pakuj> | <stop> | <idź>

5.10.1. C o f n i j

<cofnij> ::= {COFNIJ | RESET} _ P <n>

Wskazówkę w pliku P<n> ustaw przed po-
czątkowym zapisem w pliku.

5.10.2. P a k u j

<pakuj> ::= {PAKUJ | PACK}

Rozkaz pozwala na odzyskanie obszarów pa-
mięci zajętych przez zapisy usunięte uprzed-
nio przez program. Rozkaz powoduje ustawienie
wskazówek we wszystkich plikach przed ich po-
czątkowe zapisy.

Wszystkie adresy wskazówek zapamiętano
uprzednio stają się nieaktualne.

5.10.3. S t o p

<stop> ::= {STOP | STOP}

Zatrzymaj wykonywanie programu.

5.10.4. I d ź

<idź> ::= {IDŹ | GOSEC} _ <nazwa sekcji>
 <nazwa sekcji> ::= <identyfikator>

Przejdź do wykonania sekcji o podanej nazwie. Sekcja ta może być napisana w języku EOL lub MSAS. W przypadku przejścia do sekcji napisanej w języku EOL, wyrażenia i SAR zostają wyzerowane, wartość pozostałych zmiennych jest zachowywana.

Informacje niezbędne do pisania programów mieszanych MSAS-EOL zawiera Dodatek B.

5.11. Deklaracja

`<deklaracja> ::= <zwotnica> | <robocze> | <start>`

5.11.1. Z w r o t n i c a

`<zwotnica> ::= <zwotnica indeksowa> |
 <zwotnica nazwowa>
 <zwotnica indeksowa> ::= {<nazwa zwrotnicy>}
 .. { INDEKS | INDEX } ..
 <lista zwrotnicy indeksowej>
 <lista zwrotnicy indeksowej> ::= <etykieta> [,
 [-λ] <etykieta>] ..
 <nazwa zwrotnicy> ::= <etykieta>
 <zwotnica nazwowa> ::= {<nazwa zwrotnicy>}
 .. { NAZWA | NAME } ..
 <lista zwrotnicy nazwowej>
 <lista zwrotnicy nazwowej> ::= <etykieta>
 [: <tekst>] [, [-λ] <etykieta> [: <tekst>]] ..`

Ze zwrotnicy indeksowej korzysta się za pomocą rozkazów SKOIN /GOIN/, WIIN /CAIN/, zaś ze zwrotnicy nazwowej za pomocą rozkazów SKONA /GONA/ i WYNA /CANA/. Rozkazy te zostały opisane w 5.5.3.

5.11.2. R o b o c z e

**<robocze> ::= { ROBOCZE | WORKSPACE } _
 <całkowita>**

Deklaracja ta określa ilość bloków pamięci zarezerwowanej na wyrażenia i SAR. Wielkość bloku wynosi 128 słów.

Brak deklaracji równoważny jest zadeklarowaniu maksymalnej ilości pamięci.

5.11.3. S t a r t p r o g r a m u

<start> :: { START | START }

Deklaracja ta poprzedza rozkaz, który w sekcji programu ma być wykonany jako pierwszy. Brak deklaracji **<start>** w sekcji równoważny jest umieszczeniu tej deklaracji przed pierwszym rozkazem sekcji.

5.12. Komentarz

<komentarz> ::= /* dowolny ciąg znaków, w którym para znaków "*/" nie występuje */

Komentarz nie powoduje wykonania żadnej czynności, a służy jedynie do włączenia odpowiednich objaśnień do programu.

W zapisie tego kodu wykorzystano dwie sekcje. W sekcji M-SAS znajdują się instrukcje sterujące, a w sekcji KONS znajdują się instrukcje wykonywane.

Wykonanie programu może odbyć się za pomocą komputera lub za pomocą komputera i drukarki. Wykonanie programu za pomocą komputera jest łatwiejsze, ale nie jest tak dokładne, jak wykonanie za pomocą drukarki. Drukarka jest dokładniejsza, ale jest bardziej kosztowna.

6. PROGRAMY I SEKCJE

6.1. Program

```
<program> ::= {<sekcja>|<sekcja M-SAS>}
               . . . KONP ↳ <nazwa sekcji>
```

Nazwa sekcji po słowie KONP wyznacza sekcję, od której rozpoczyna się wykonywanie programu.

```
<sekcja> ::= { SEKCJA | SECTION } <nazwa sekcji>
               { λ | ; } {<zdanie zewnętrzne>
               { λ | ; } } . . . { KONS | ENDS }

<nazwa sekcji> ::= <identyfikator>
<zdanie zewnętrzne> ::= <zdanie>
<zdanie> ::= [<etykieta> :] . . . <rozkaz>
               <deklaracja>|<procedura>|
               <komentarz>

<procedura> ::= {<nazwa procedury>} . .
               { PROC | PROC } { λ | ; } {<zdanie>
               { λ | ; } } . . . { KONIEC | END } { λ | ; }

<nazwa procedury> ::= <etykieta>
```

Definicja procedury jest rekursywna, gdyż w skład jednej procedury może wchodzić inna procedura.

Zakresem etykiety umieszczonej we wnętrzu procedury P jest cała zawartość procedury P pomniejszona o zawartość tych procedur, w których wnętrzu ta sama etykieta występuje ponownie.

Zakresem etykiet zdań zewnętrznych /tzn. etykiet tych rozkazów oraz nazw tych procedur i zwrotnic, które są zdaniemami zewnętrznymi/ jest cała sekcja również pomniejszona o zawartość procedur, w których wnętrzu ta sama etykieta występuje ponownie. Zakresem nazwy sekcji jest cały program.

P r z y k l a d

Przykład różnych zakresów etykiet przedstawiony jest w poniższym programie, złożonym z dwóch procedur zewnętrznych o nazwie A oraz P.

Zakres nazwy procedury zewnętrznej A nie przenosi się do wnętrza procedury B, gdyż we wnętrzu tym umieszczona jest inna etykieta A. Dlatego też rozkaz WYKONAJ A odnosi się do procedury A zawartej w procedurze B.

Zakres nazwy procedury zewnętrznej P rozciąga się na całą zawartość procedur A oraz P, gdyż nazwa ta nigdzie nie jest zadeklarowana powtórnie. Dlatego też oba wypisane powyżej rozkazy WYKONAJ P odnoszą się do tej samej procedury P.

Rozkaz SKOMI B powoduje ewentualne przejście do rozkazu SKOCZ A, umieszczonego w procedurze P.

A : PROC

B : PROC

A : PROC

WYKONAJ A

WYKONAJ P

KONIEC

KONIEC

P : PROC

B : SKOCZ A

WYKONAJ P

SKOMI B

KONIEC

7. MAKRODEFINICJE

```

<sekcja źródłowa> ::= <sekcja> |
                           <sekcja modyfikowana>

<sekcja modyfikowana> ::= {SEKCJA | SECTION} ..
                           <nazwa sekcji> {λ|; } <lista makrodefini-
                           cji> {λ|; }   <treść sekcji modyfikowanej>

<lista makrodefinicji> ::= {<makrodefinicja>} . .

<makrodefinicja> ::= <nazwa makrodefinicji> ..
                           <lista parametrów> [λ]
                           (<treść makrodefinicji>)

<nazwa makrodefinicji> ::= <identyfikator>

<lista parametrów> ::= <parametr> [{,|λ}
                           <parametr>] . .

<parametr> ::= <identyfikator>

<treść makrodefinicji> ::= .
                           [<element makrodefinicji>] . .

<element makrodefinicji> ::= <identyfikator> |
                           <parametr> | <identyfikator>' <parametr> |
                           <calkowita> | <tekst> | _ | , | : | ; | λ |
                           <odwołanie wewnętrzne> | <komentarz>

<odwołanie wewnętrzne> ::= *
                           * {<nazwa makrodefinicji> | <parametr>}
                           [ _ <argument wewnętrzny> [, ,
                           <argument wewnętrzny>] .. ] { λ|; }

```

```

<argument wewnętrzny> ::= <identyfikator> |
    <parametr> | <identyfikator> ' <parametr> |
    <całkowita> | <tekst>

<treść sekcji modyfikowanej> ::= {<element sekcji modyfikowanej>} ..

<element sekcji modyfikowanej> ::= <identyfikator> | <całkowita> | - | |
    <komentarz> | <tekst> | : | ; | λ | <odwołanie>

<odwołanie> ::= * <nazwa makrodefinicji>
    [ - <argument> [, <argument> ] ..] { λ ; }

<argument> ::= <identyfikator> | <całkowita> |
    <tekst>

```

Makrodefinicje pozwalają na modyfikację sekcji programu przed wykonaniem.

Każda makrodefinicja jest określeniem odcinka programu, w którym mogą występować parametry, lecz tylko te, które zostały wymienione na liście parametrów danej makrodefinicji. Odcinek ten zostaje włączony do programu w tych punktach, gdzie nastąpiło odwołanie do danej makrodefinicji.

W wyniku takich modyfikacji <sekcja modyfikowana> przekształca się w <sekcję>, która jest następnie wykonywana.

P r z y k l a d y

- a. WYPISZ PERFORATOR
(PISZ A1, Q'PERFORATOR)

Nazwą powyższej makrodefinicji jest WYPISZ. Makrodefinicja jest jednoparametrowa. Parametrem jest PERFORATOR.

- b. ABC LA, TEKST, ARG
 (UMIESĆ TEKST, A'ARG
 SKOMI LA)

Nazwą powyższej makrodefinicji jest ABC.
 Makrodefinicja jest trójparametrowa. Parametrami jej są LA, TEKST, ARG.

- c. QX TOM, BX, L
 (USUŃ I'L, D
 *ABC EX, 'STOP' ,5
 SKOCZ TOM)

Nazwą powyższej makrodefinicji jest QX.

Makrodefinicja jest trójparametrowa. Jej parametrami są TOM, EX, L. Makrodefinicja ta zawiera odwołanie do makrodefinicji o nazwie ABC.

- d. PUSTE ()

Powyższa makrodefinicja nosi nazwę PUSTE.
 Makrodefinicja ta jest bezparametrowa. Jej treść jest pusta. Możliwość stosowania makrodefinicji o pustej treści umożliwia np. odwołania do pewnych makrodefinicji na etapie uruchamiania programu, a po uruchomieniu zamianę treści tych makrodefinicji na puste, co pozwala na uniknięcie usuwania zbędnych odwołań z treści sekcji modyfikowanej.

- e. POMOCNICZA A, B ()

Powyższa makrodefinicja nosi nazwę POMOCNICZA. Jej parametrami są A i B. Treść tej makrodefinicji jest pusta.

7.1. Odwołanie do makrodefinicji

Włączenie makrodefinicji do programu ma miejsce tylko wtedy, gdy spełnione są następujące warunki:

- Makrodefinicja o danej nazwie została zdefiniowana na liście makrodefinicji danej sekcji modyfikowanej.
- Ilość argumentów odwołania jest identyczna z ilością parametrów makrodefinicji.

W momencie odwołania jest ustanawiana odpowiedniość między parametrami i argumentami w ten sposób, że każdemu parametrowi z listy parametrów makrodefinicji odpowiada argument zajmujący tę samą pozycję na liście argumentów.

Po ustanowieniu odpowiedniości następuje włączenie makrodefinicji. Włączenie makrodefinicji polega na zastąpieniu odwołania do makrodefinicji treścią tejże makrodefinicji z równoczesnym podstawieniem argumentów w miejscach odpowiadających im parametrów. Jeśli odwołanie kończy się średnikiem, to po włączeniu treści makrodefinicji średnik jest usuwany.

W przypadku, gdy element makrodefinicji ma postać <identyfikator>!<parametr>, przy podstawianiu usuwany jest apostrof.

Jeśli odwołanie do makrodefinicji powoduje włączenie takiej treści makrodefinicji, w której znajduje się ponowne odwołanie do makrodefinicji, to wykonywane są następujące czynności:

- W przypadku, gdy nazwa makrodefinicji jest parametrem pierwotnej makrodefini-

nicji, to zostaje ona zastąpiona odpowiednim argumentem.

- b. Parametry pierwotnej makrodefinicji, które występują w odwołaniu, zostaną zamienione na odpowiadające im argumenty nadrzednego odwołania.
- c. Zmodyfikowane zgodnie z p. a i b odwołanie zostaje zastąpione treścią odpowiadającej mu makrodefinicji zgodnie z omówionymi wcześniej regułami.

Z powyższego wynika, że włączanie makrodefinicji powoduje przechodzenie argumentów z zewnętrznych odwołań do wewnętrznych. Makrodefinicje muszą być tak zdefiniowane, żeby po ich włączeniu <sekcja modyfikowana> została przekształcona w poprawną syntaktycznie <sekcję>.

Przykłady

- a/ Przytoczymy przykład sekcji modyfikowanej o nazwie

STARTOWA i uproszczony schemat jej przekształcenia w <sekcję>. Przykładem argumentu przenoszonego z zewnętrznego odwołania do wewnętrznego jest tutaj GAMMA.

SEKCJA STARTOWA

```
(ABC LA, TEKST, ARG
(UMIEŚĆ TEKST, A'ARG
SKOMI LA)
QX TOM, EX, L
(USUŃ J'L, D
*ABC EX, 'STOP', 5
SKOCZ TOM))
```

ALFA : PROC
* QX BETA, GAMMA, 1
GAMMA: STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

Pierwszy etap modyfikacji daje w wyniku:

SEKCJA STARTOWA

ALFA : PROC
USUŃ I1, D
* ABC GAMMA, 'STOP' ,5
SKOCZ BETA

GAMMA: STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

Drugi etap modyfikacji daje w wyniku < sekcję > :

SEKCJA STARTOWA

ALFA : PROC
USUŃ I1, D
UMIESĆ 'STOP', A5
SKOCZ GAMMA
SKOCZ BETA

GAMMA : STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

b/ Poniżej przytoczymy przykład sekcji modyfikowanej o nazwie PIERWSZA, w której

makrodefinicja P1 zawiera odwołanie do makrodefinicji, której nazwa jest parametrem formalnym ARA.

Schemat przekształcenia tej < sekcji modyfikowanej > na < sekcję > podano niżej.

SEKCJA PIERWSZA

```
(P1 B1, ARA, C
    (PRZEŚLIJ A'B1, B'C
        *ARA, 3)
P2 M
    (PISZ A'M, Q1)
P3 T
    (CZYTAJ I1, B'T, L))
```

START : PROC

```
*P1 4, P3, 7
    UMIESĆ 'ALFABET', Q1
*P1 2, P2, 8
    KONIEC
    KONS
```

Pierwszy etap modyfikacji daje w wyniku:

SEKCJA PIERWSZA

```
START : PROC
    PRZEŚLIJ A4, B7
    *P3, 3
    UMIESĆ 'ALFABET', Q1
    PRZEŚLIJ A2, B8
    *P2, 3
    KONIEC
    KONS
```

Następny etap modyfikacji daje w wyniku
<sekcję>:

EOL 7-8

SEKCJA PIERWSZA

START : PROC

PRZEŚLIJ A4, B7

CZYTAJ I1, B3, L

UMIEŚĆ 'ALFABET', Q1

PRZEŚLIJ A2, B8

PISZ A3, Q1

KONIEC

KONS

Wydrukowany program jest skierowany do jednostki wyjściowej, której zadaniem jest wypisanie kolejnych znaków wejścia. Wysłanie kolejnego znaku wejścia powoduje zmianę stanu jednostki wejścia na stan oznaczony kodem znaku. Wysłanie kolejnego znaku wejścia powoduje zmianę stanu jednostki wejścia na stan oznaczony kodem znaku. Wysłanie kolejnego znaku wejścia powoduje zmianę stanu jednostki wejścia na stan oznaczony kodem znaku. Wysłanie kolejnego znaku wejścia powoduje zmianę stanu jednostki wejścia na stan oznaczony kodem znaku. Wysłanie kolejnego znaku wejścia powoduje zmianę stanu jednostki wejścia na stan oznaczony kodem znaku.

8. PRZYKŁADY

Niżej przytoczono cztery przykłady programów w języku EOL-2. Opis każdego przykładu składa się z następujących części: opis wejścia, opis wyjścia, tekst programu, objaśnienie programu, przykładowe dane i wyniki.

Numery wierszy wypisane z lewej strony tekstu programu nie należą do programu. Zostały one wprowadzone w celu ułatwienia objaśnienia programu.

Przykład 1

Zadaniem programu SORTOWANIE jest wprowadzenie kolejnych słów tekstu, posortowanie tych słów w kolejności alfabetycznej i wypisanie posortowanych słów.

Wejście. Wejście programu stanowi dowolny tekst zakończony słowem XXX. Słowem tekstu jest dowolny ciąg literowo-cyfrowych symboli.

Separatorem tekstu jest dowolny ciąg symboli różnych od liter i cyfr.

W y j s c i e. Na wyjściu program wypisuje ciąg posortowanych słów, przy czym każde słowo jest wypisywane w oddzielnym wierszu. Separatory tekstu wejściowego są pomijane.

Plik jest przeglądany kolejno zapis po zapisie. Przed pobraniem kolejnego zapisu zapamiętuje się jego położenie w wyrażeniu E2. Przegląd odbywa się aż do natrafienia na koniec pliku lub na zapis zawierający słowo późniejsze niż bieżące.

Program

(1) SEKCJA SORTOWANIE

- (2) CZYTAJ: COFNIJ P1 /*WCZYZTAJ SLOWO*/
 - (3) USUN I1,LD
 - (4) CZYTAJ I1,A1,RB
 - (5) RW A1,'XXXX'
 - (6) SKOPL WYPISZ
- (7) TEST: SCHOWAJ P1,A2 /*WPISANIE SLOWA DO PLIKU P1*/
 - (8) POBIERZ D1,B2
 - (9) SKOMI TEST2
 - (10) PP B1,B2
 - (11) SKOPL TEST1
 - (12) USUN A2
 - (13) SKOCZ TEST
- (14) TEST1: USUN A2,1
 - (15) USTAW A2,P1
- (16) TEST2: WSTAW A1,D1
 - (17) SKOCZ CZYTAJ
- (18) WYPISZ: UMIESC 'L',Q1 /*WYPISANIE SLOWA/
 - (19) POBIERZ C1,A2
 - (20) PISZ A2,Q1
 - (21) SKOPL WYPISZ
- (22) STOP
- (23) KONS

O b j a s n i e n i e p r o g r a m u

- (2) Ustawienie wskazówki na początku pliku P1
- (3) - (6) Wczytanie kolejnego słowa i sprawdzenie czy jest to słowo końcowe
- (7) - (13) Odszukanie miejsca wpisania słowa do pliku
- (14) - (17) Wpisanie słowa do pliku
- (18) - (22) Wypisanie posortowanych słów w oddzielných wierszach

P r z y k l a d o w e d a n e -

PRZED DAWNYMI, BARDZO DAWNYMI CZASY LUDZIE
 MIESZKALI W JASKINIACH.
 MIAST WOWCZAS NIE BYŁO. NIE BYŁO ZADNYCH SKLEPOW.
 NIGDZIE NIE SPRZEDAWANO
 CIASTEK ANI CUKIERKOW. TO BYŁY BARDZO
 ZLE CZASY.

-XXXX-

W y n i k i

ANI
 BARDZO
 BARDZO
 BYŁO
 BYŁO
 BYŁY
 CIASTEK
 CUKIERKOW
 CZASY
 CZASY
 DAWNYMI
 DAWNYMI
 JASKINIACH
 LUDZIE
 MIAST
 MIESZKALI
 NIE
 NIE
 NIE
 NIGDZIE
 PRZED
 SKLEPOW
 SPRZEDAWANO
 TO
 W
 WOWCZAS
 ZADNYCH
 ZLE

Przykład 2

Zadaniem przykładu DRUKI jest wypełnianie formularzy i wypisywanie wypełnionych formularzy na wyjście.

Wejście. Dane wejściowe są dwóch typów:

a/ Lista formularzy

Każdy formularz poprzedzony jest [. Bezpośrednio po [występuje nazwa formularza, a za nią po przecinku ciąg parametrów formularza zakończony]. Parametrem formularza jest albo dowolny ciąg znaków różnych od] i =, albo = cyfra.

Formalny opis formularza zdefiniowany jest poniżej:

```
<formularz> ::= [= <nazwa formularza> ,  
                      {<parametr formularza>} .. ]
```

Znaki [= i] oznaczają [i] występujące w danych do programu

```
<nazwa formularza> ::= <dowolny ciąg znaków  
                           różnych od , i ;>  
<parametr formularza> ::= <tekst>|= <cyfra>  
<tekst> ::= <dowolny ciąg znaków różnych od  
                ] i = >
```

b/ Lista danych do formularzy

Dane do formularza poprzedzone są *, po której występuje nazwa formularza a dalej po przecinku ciąg argumentów aktualnych oddzielonych przecinkami i zakończony; .

Argumentem aktualnym jest ciąg dowolnych znaków różnych od ; i , .

Formalny opis danych do formularza jest następujący:

```
< dane do formularza > ::= =* < nazwa formularza >
                                [, < argument aktualny > ] ...;
< argument aktualny > ::= < dowolny ciąg znaków różnych od ;
                                i , >
```

W y j ś c i e. Wczytanie kolejnych danych do formularza powoduje wyszukanie odpowiedniego formularza, a następnie wypisanie na wyjście trzech znaków nowej linii, a po nich wypełnionej zgodnie z danymi treści formularza. Przy wypełnianiu formularza parametr w postaci ciągu znaków przepisywany jest bezpośrednio na wyjście, natomiast w miejsce parametru postaci =cyfra wypisywany jest odpowiedni argument z listy argumentów.

P r o g r a m

(1) SEKCJA DRUKI

- (2) UMIESC '**[,B5
- (3) UMIESC ';;',B6
- (4) ROZBIJ A5
- (5) ROZBIJ A6
- (6) CZYTAJ:USUN 11,T5
- (7) SKOMI KONIEC
- (8) RW 11,'['
- (9) USUN 11,1
- (10) SKOMI CZYTAJ1

- (11) CZYTAJ2:CZYT AJ 11,B1,' ,'
 (12) USUN 11,1
 (13) CZYT AJ 11,Z1,']'
 (14) SKOMI BLAD
 (15) WSTAW A1,D1
 (16) SKOCZ CZYT AJ
- (17) CZYT AJ1:CZYT AJ 11,Z2,T6
 /*WCZYTANIE DANYCH DO FORMULARZA*/
 (18) SKOMI BLAD
 (19) RW 11,' ,'
 (20) SKOMI TWORZ
 (21) USUN 11,1
 (22) SKOCZ CZYT AJ1
- (23) TWORZ: PRzeslij B2,B7
 /*AKTUALIZACJA FORMULARZY*/
 (24) COFNIJ P1
 (25) OMIN P1,B2
 (26) SKOMI BLAD1
 (27) UMIESC 'L*L*L',Q1
 (28) POBIERZ D1,B1
 (29) USUN A1,1
 (30) ROZBIJ A1
 (31) USUN A2,1
- (32) T1: PRzeslij A1,A1,1
 /*CZY JEST JESZCZE NIE OPRACOWANA
 CZESC FORMULARZA*/
 (33) SKOPL TWORZ3
- (34) T: OMIN P1
 (35) USUN A2
 (36) USUN A7
 (37) SKOCZ CZYT AJ
- (38) TWORZ3:RW A1,' ='
 (39) SKOPL TWORZ1
 (40) PISZ A1,Q1,' ='
 (41) SKOMI T
- (42) TWORZ1:USUN A1,D
 (43) LICZBA A1
 (44) PRzeslij B2,A4,N1
 (45) SKOMI BLAD1
 (46) PISZ B4,Q1,1

(47) USIJN A4
 (48) USIJN A1, 1
 (49) SKOCZ T1

(50) KONIEC:STOP

(51) BLAD: UMIESC '*LBRAK ; LUB] W DANYCH', Q1
 (52) STOP

(53) BLAD1: UMIESC '*LNIE POPRAWNE DANE*L-', Q1

(54) PISZ A7, Q1

(55) SKOCZ T

(56) KONS

(57) KONP DRIUKI

O b j a s n i e n i e p r o g r a m u

(2) - (5) Załadowanie do E5 znaków początków formularzy i danych, do E6 separatora pozycji w danych i znaku końca danych.

(6) - (10) Usunięcie znaków poprzedzających początek formularza lub danych.
 Jeśli nie zostanie napotkany znak * lub [to skok do etykiety KONIEC. Jeśli usuwanie zakończyło się na znaku * następuje skok do czytania danych.

(11) - (16) Wczytanie formularza do wyrażenia E1 i przesłanie go do pliku P1, a następnie skok do czytania.

(17) - (23) Wczytanie danych do E2 z równoczesnym usunięciem oddzielających je przecinków.

Przesłanie danych do E7 w celu późniejszej ewentualnej sygnalizacji błędu.

- (24) - (26) Szukanie w pliku P1 formularza mającego nazwę taką jak aktualnie opracowywane dane. W przypadku nie znalezienia takiego - skok do sygnalizacji błędu.
- (27) - (31) Wysłanie na wyjście trzech znaków nowej linii, pobranie formularza do wyrażenia E1, rozbicie go na pojedyncze składniki oraz usunięcie nazwy z wyrażeń E1 i E2.
- (32) - (37) Sprawdzenie czy formularz ma jeszcze jakieś składniki, jeśli tak to przejście do opracowywania ich, jeśli nie - ustawienie wskazówki na końcu pliku P1, wyczyszczenie wyrażeń E2 i E7 i przejście do ponownego czytania.
- (38) - (49) Sprawdzenie typu bieżącego parametru formularza. Jeśli parametr jest tekstem, to wypisanie go na wyjściu, jeśli jest typu=liczba - to wybranie z E2 odpowiadającego mu argumentu aktualnego i wypisanie go na wyjście. Jeśli argument aktualny nie zostanie znaleziony - skok do błędu.
- (50) - (55) Zakończenie działania programu.
Sygnalizacja błędu:
1/ w przypadku braku ; lub] w danych - zasygnalizowanie błędu i zakończenie działania programu;

2/ w pozostałych przypadkach -
zasygnalizowanie błędu i
przejście do opracowywania
ewentualnych danych.

Przykładowe dane

[WEZWANIE, OB=1=2 PROSZE ZGŁOSIC SIE =3=4]
[ZYCZENIA, WSZYSTKIEGO DOBREGO=1=2=3=4]

*ZYCZENIA, DROGIEJ, HANI,
ZYCZA,
KOLEDZY;

*WEZWANIE, JAN, KOWALSKI,
DO WRN W WARSZAWIE,
W DNIIU 5.10.1970R. GODZ.12.00.;

*WEZWANIE, ADAM,
DO WRN W WARSZAWIE,
W DNIIU 7.10BR.;

*ZYCZENIA,
Z OKAZJI, 25-LECIA PRACY ZAWODOWEJ,
ZYCZY DYREKCJA I WSPOLPRACOWNICY,

; *WEZWANIE, ANNA, KOWALSKA,
W MHD,-SKLEP NR.2.
;

Wyniki

WSZYSTKIEGO DOBREGO DROGIEJ HANI
ZYCZA
KOLEDZY

OB JAN KOWALSKI PROSZE ZGŁOSIC SIE
DO WRN W WARSZAWIE
W DNIIU 5.10.1970R. GODZ.12.00.

OB ADAM
DO WRN W WARSZAWIE PROSZE ZGŁOSIC SIE
W DNIIU 7.10BR.
NIEPOPRAWNE DANE
-WEZWANIE ADAM
DO WRN W WARSZAWIE
W DNIIU 7.10BR.

Z WSZYSTKIEGO DOBREGO
Z OKAZJI 25-LECIA PRACY ZAWODOWEJ
ZYCZY DYREKCJA I WSPOLPRACOWNICY

OB ANNA KOWALSKA PROSZE ZGLOSIC SIE
W MHD-SKLEP NR.2.

Przykład 3

Zadaniem tego programu jest obliczanie wartości prostych formuł arytmetycznych.

Wejściem programu jest lista formuł arytmetycznych złożonych z liczb całkowitych, nawiasów i znaków czterech działań arytmetycznych.

Każda z formuł znajduje się w oddzielnym wierszu. Znaki spacji i powrotu karetki są nieznaczące. Nie mogą jednak występować pomiędzy cyframi całkowitej. Wiersz, którego pierwszym symbolem znaczącym jest ◊ /romb/ określa koniec danych.

Wyjściem programu wypisuje formułę, znak równości i wartości formuły. Jeżeli formuła jest błędna, to zamiast wyniku program podaje miejsce, w którym wystąpił błąd.

Program

- (1) SEKCJA ARYT
- (2) /*DRUKOWANIE TYTULU*/
- (3) UMIESC 'WARTOSCI FORMUL*L',Q1
- (4) /*WCZYTANIE WIERSZA*/

- (5) Wczyt: USUN I1, RDL
 (6) RW I1, '*L'
 (7) SKOPL USPOCZ
 (8) RW I1, '*G'
 (9) SKOMI KLASA
- (10) USPOCZ: USUN I1, 1
 (11) SKOCZ Wczyt
- (12) KLASA: RW I1, D
 (13) SKOMI ROZNE
 (14) CZYTAJ I1, Z1, BLR
 (15) SKOCZ USWewn
- (16) ROZNE: CZYTAJ I1, Z1, 1
- (17) USWewn: USUN I1, LRD
 (18) RW I1, '*G'
 (19) SKOMI LTEST
 (20) USUN I1, 1
 (21) SKOCZ USWewn
- (22) LTEST: RW I1, '*L'
 (23) SKOMI KLASA
- (24) /*BADANIE CZY KONIEC DANYCH*/
- (25) RW A1, '0'
 (26) SKOPL. KODAN
- (27) /*LICZENIE I DRUKOWANIE WARTOSCI*/
- (28) PRZESLIJ B1, B2
 (29) UMIESC '*L', Z1
 (30) WYKONAJ FORMULA
 (31) RW A1, '*L'
 (32) SKOMI BLAD
 (33) PISZ A2, Q1
 (34) UMIESC '=', Q1
 (35) SLOWO A3
 (36) PISZ A3, Q1
 (37) UMIESC '*L', Q1
 (38) SKOCZ Wczyt
- (39) /*LICZENIE FORMULY*/
- (40) FORMULA: PROC
 (41) WYKONAJ TERM
- (42) OPERATOR: NAZWA PLUS: '+', MINUS: '-'
- (43) W1: SKONA A1, OPERATOR
 (44) WROC

(45) PLUS: WYKONAJ TERM
 (46) PRZESLIJ A3,B4,1
 (47) DODAJ A3,B4
 (48) USUN A4,1
 (49) SKOCZ W1

(50) MINUS: WYKONAJ TERM
 (51) PRZESLIJ A3,B4,1
 (52) ODEJMIIJ A3,B4
 (53) USUN A4,1
 (54) SKOCZ W1

(55) KONIEC

(56) /*LICZENIE TERMU*/

(57) TERM: PROC
 (58) WYKONAJ ARGUMENT

(59) OPERATOR: NAZWA MN:'**',DZ:'/'

(60) T1: SKONA A1,OPERATOR
 (61) WROC

(62) MN: WYKONAJ ARGUMENT
 (63) PRZESLIJ A3,B4,1
 (64) MNOZ A3,B4
 (65) USUN A4,1
 (66) SKOCZ T1

(67) DZ: WYKONAJ ARGUMENT
 (68) PRZESLIJ A3,B4,1
 (69) DZIEL A3,B4
 (70) USUN A4,1
 (71) USUN A3,1
 (72) SKOCZ T1

(73) KONIEC

(74) /*LICZENIE ARGUMENTU*/

(75) ARGUMENT: PROC
 (76) RW B1,D
 (77) SKOPL CAŁKOWITA
 (78) RW A1,'('

SKOMI BLAD

WYKONAJ FORMULA

RW A1,')'

SKOMI BLAD

WROC

- (84) CAŁKOWITA:LICZBA A1
 (85) PRZESLIJ A1,B3,1
 (86) WROC
 (87) KONIEC
- (88) /*SYGNALIZACJA BLEDU*/
- (89) BLAD: PROC
 (90) USUN A3
 (91) UMIESC '*LW FORMULE :*L ',Q1
 (92) PISZ A2,Q1
 (93) RW A1,'*L'
 (94) SKOPL BL1
 (95) UMIESC '*LBLAD OD MIEJSCA : ',Q1
 (96) PISZ A1,Q1
 (97) UMIESC '*L',Q1
 (98) SKOCZ WCZYT
- (99) BL1: UMIESC '*LZLY KONIEC*L',Q1
 (100) SKOCZ WCZYT
- (101) KONIEC
- (102) /*KONIEC DANYCH*/
- (103) KODAN: STOP
- (104) KONS
 KONP ARYT

O b j a ś n i e n i e p r o g r a m u

- (3) - Wypisanie tytułu wyników.
- (5) - (11) Usuwanie początkowych spacji, znaków powrotu karetki i nowej linii.
- (12) - (21) Ta część programu wczytuje kolejny wiersz do wyrażenia E1 w ten sposób, że liczby całkowite bez znaku i znaki klasy R tworzą odzielne składniki. Wewnętrzne spacje i znaki powrotu karetki są usuwane. Np. wiersz

$12 \leftarrow + \leftarrow (43 - 5 * 5)$

zostanie wczytany następująco:

E1 : $-12 - + (-43 - -5 - * -5 -)$

- (25) - (26) Badanie czy nastąpił koniec danych i ewentualne przejście do zakończenia.
- (28) Zapamiętanie kopii formuły w wyrażeniu E2.
- (29) Umieszczenie specjalnego składnika dla łatwiejszego określenia końca formuły.
- (30) - (38) Składnia naszej formuły zapisana formalnie jest następująca:
- ```

<formuła arytmetyczna> ::= <term>
[{+|-} <term>] ..
<term> ::= <argument> [{*|/}
 <argument>] ..
<argument> ::= <całkowita nie-
 znakowana> |
 (<formuła arytmetyczna>)

```

Poszczególnym pojęciom metajęzycznym odpowiadają podprogramy:

FORMULA, TERM i ARGUMENT.

Odwołanie do podprogramu FORMULA powoduje usunięcie tych wszystkich składników, które tworzą formułę na początku wyrażenia E1. Wartość tej formuły przesyłana jest przez podprogram na początek E3. Jeżeli wiersz danych był poprawny, to w E1 powinien pozostać jedynie składnik umieszczony tam rozkazem (29). Jeśli

tak jest, następuje drukowanie wyniku. W przeciwnym razie program przechodzi do sygnalizacji błędu.

- (40) - (55) W podprogramie FORMULA wykonuje się najpierw podprogram TERM. Usuwa on początkowy term z E1 przesyłając przy tym wartość termu na początek E3. Z kolei bada się w pętli, czy na początku E1 znajduje się składnik '+' lub '-' . Jeśli tak, to ponownie wykonuje się podprogram TERM, a dwie liczby z początku E3 zastępuje się odpowiednio sumą lub różnicą tych liczb. Jeśli w kolejnym przejściu pętli nie ma owych składników, następuje powrót z podprogramu.
- (57) - (73) Podprogram TERM działa analogicznie jak podprogram FORMULA.
- (75) - (87) Gdy wywoływany jest podprogram ARGUMENT na początku E1 powinien znajdować się tekstowy składnik złożony z cyfr lub z jednego znaku - nawiasu otwierającego. W pierwszym przypadku składnik ten jest zamieniany na liczbę i przesyłany na początek E3. W drugim przypadku korzysta się z podprogramu FORMULA usuwając przy tym odpowiednie nawiasy. W razie wystąpienia niezgodności ze składnią sygnalizowany jest błąd.

(89) - (101) Ta część programu działa wówczas, gdy zostanie wykryty błąd.

(103) Zakończenie programu.

### P r z y k l a d o w e   d a n e

2+2  
 $45 - (23-22/(45-44)*((12345-12344)+1))$   
 $2*2*2*2*2*2*2*2*2*2*2*2*2*2*2$   
 $3*3*3*3*3*3*3*3*3$   
 $((((5*10-3)*10+24)*10+9)+78$   
 $5+5+5+5+5+5+5+5+5+5+5+5$   
 $0-9-9-9-9-9-9-9-9-9-9-9-9$   
 $1000000-1$   
 $6-(5*5)(7-4/2)$   
 $8192/2/2/2/2/2$   
 $1000000/6$   
 $0$

### W y n i k i

#### WARTOSCI FORMUL.

2+2=4  
 $45 - (23-22/(45-44)*((12345-12344)+1)) = 66$   
 $2*2*2*2*2*2*2*2*2*2*2*2*2 = 16384$   
 $3*3*3*3*3*3*3*3*3 = 19683$

#### W FORMULE :

$((((5*10-3)*10+24)*10+9)+78$

ZLY KONIEC

$5+5+5+5+5+5+5+5+5+5+5+5 = 65$

$0-9-9-9-9-9-9-9-9-9-9-9 = -108$

$1000000-1 = 999999$

#### W FORMULE :

$6-(5*5)(7-4/2)$

BLAD OD MIEJSCA :  $(7-4/2)$

$8192/2/2/2/2/2 = 256$

$1000000/6 = 166666$

### Przykład 4

Zadaniem tego przykładu jest wygenerowanie 10-ciu liczb pseudolosowych z przedziału 0 - 1000 i wypisanie ich na wyjście.

W e j ś c i e. Program nie wczytuje z wejścia żadnych danych.

W y j s c i e. Na wyjściu program wypisuje 10 wygenerowanych liczb pseudolosowych, przy czym każda liczba wypisywana jest w odzielnym wierszu.

### P r o g r a m

```

(1) SEKCJA DRIJK
(2) ROBOCZE 1
(3) UMIESC 0, B2
(4) /*REZERWOWANIE SKLADNIKA NA LICZBE LOSOWA*/
(5) UMIESC 'XXXX', B1
(6) A: IDZ LOS
(7) UMIESC '*L', Q1
(8) SLOWO B1
(9) PISZ A1, Q1, 1
(10) DODAJ A2, 1
(11) RW B2, 10
(12) SKOMI A
(13) STOP
(14) KONS

(15) SEST LOS
(16) UMM S/*GENEROWANIE LICZBY LOSOWEJ*/
(17) MNM S1
(18) PMM S
(19) MNM ZAKRES
(20) /*PRZESLANIE LICZBY LOSOWEJ DO 1-GO
(21) SKLADNIKA WYRAZENIA*/
(22) UMB 7607+2
(23) NZB
(24) PZA 1+
(25) UMA 7428*
(26) DOA +5
(27) PZA 7428*
(28) SKO 742.9*
(29) S:=14631461
(30) S1:=14631461
(31) ZAKRES :+1000
(32) KONS
(33) KONP DRIJK

```

### O b j a ś n i e n i a p r o g r a m u

- (2) Rezerwacja pamięci roboczej.  
 W przypadku, gdy część programu stanowią sekcje stałe w SAS-ie,

brak tej deklaracji może spowodować zniszczenie tych sekcji.

- (3) Ustawienie licznika liczb.
- (5) Rezerwacja składnika w wyrażeniu E1 na liczbę losową.
- (6) Przejście do sekcji w SAS-ie w celu wygenerowania liczby losowej.
- (7) - (12) Wydrukowanie liczby i sprawdzenie czy wydrukowano 10 liczb.
- (16) - (19) Generowanie liczby losowej.
- (22) - (24) Przesłanie wygenerowanej liczby losowej do pierwszego składnika wyrażenia E1.
- (25) - (28) Powrót z sekcji SAS do rozkazu oznaczonego numerem (7) w sekcji EOL.

#### W y n i k i

520  
255  
636  
199  
362  
803  
332  
768  
310  
917

DODATEK A  
ZBIÓR ZNAKÓW DLA DANYCH

Uporządkowanie znaków dla danych programu w języku EOL i tekstów zawartych w programie jest zgodne z Normą Zakładową Kodu Wewnętrzniego /KW-ZAM/.

Poniższa tablica zawiera pełny zbiór znaków dla danych.

|              |   |
|--------------|---|
| spacja       | L |
| .            | M |
| (            | N |
| +            | O |
| *            | P |
| )            | Q |
| ;            | R |
| -            | S |
| /            | T |
| ,            | U |
| :            | V |
| ' /apostrof/ | W |
| =            | X |
| [            | Y |
| ]            | Z |

|     |     |
|-----|-----|
| <   | S1  |
| >   | 0   |
| ?   | 1   |
| ESC | 2   |
| SO  | 3   |
| A   | 4   |
| B   | 5   |
| C   | 6   |
| D   | 7   |
| E   | 8   |
| F   | 9   |
| G   | LF  |
| H   | ◊   |
| I   | FF  |
| J   | CS  |
| K   | CR  |
|     | DEL |

Sposób przyporządkowania znakom kodu wewnętrznego znaków kodów zewnętrznych zawarty jest w opracowaniu "System operacyjny cz. I, Maszyna Użytkowa Załącznik 1".

Znaki nietypowe kodu wewnętrznego mają w tekstach następującą reprezentację:

| Znak kodu<br>wewnętrznego | Reprezentacja w<br>tekście |
|---------------------------|----------------------------|
| [                         | * U                        |
| ]                         | * V                        |
| <                         | * W                        |
| >                         | * X                        |
| ?                         | * Y                        |
| ESC                       | * Z                        |
| SO                        | * M                        |
| SJ                        | * D                        |

|     |     |
|-----|-----|
| LF  | * L |
| FF  | * E |
| CS  | * F |
| CR  | * G |
| DEL | * S |

W przypadku, gdy kod zewnętrzny posiada jednoznakową reprezentację nietypowego znaku kodu wewnętrznego, można jej użyć zamiast wymienionej wyżej reprezentacji dwuznakowej. Np.: w kodzie M-2 zamiast "\*U" można napisać "[ ", tekst "\*U" będzie w tym przypadku równoważny tekstowi "[ ".



## DODATEK B

## UŻYCIE W PROGRAMIE SEKCJI W MSAS-ie

W tym samym programie mogą występować zarówno sekcje w EOL-u jak i w MSAS-ie. Komunikacja między sekcjami napisanymi w różnych językach możliwa jest zasadniczo tylko poprzez odpowiednie rozkazy przejść między sekcjami. Dalej przedstawiona jest pewna inna możliwość.

Poniżej przedstawiony jest schemat pamięci wewnętrznej w czasie wykonywania programu.

|   |                                                            |
|---|------------------------------------------------------------|
| O | ROBOCZE SYSTEMU OPERACYJNEGO                               |
|   | EGZEKUTOR<br>/program realizujący przejścia międzysekcjne/ |
|   | SEKCJA W EOL-u LUB SEKCJA WYMIENNA W MSAS-ie               |
|   | PAMIĘĆ ROBOCZA<br>/tylko dla sekcji EOL-u/                 |
|   |                                                            |
|   | SEKCJA STAŁA W MSAS-ie                                     |
|   |                                                            |
|   | SEKCJA STAŁA W MSAS-ie                                     |
|   | INTERPRETER<br>/program realizujący operacje EOL-u/        |

Z powyższego schematu wynika, że w przypadku współdziałania sekcji w EOL-u z sekcjami stałymi należy tak określać wielkość pamięci roboczej, aby ta nie nakładała się na sekcje stałe. Wielkość pamięci roboczej jest określana za pomocą deklaracji ROBOCZE /WORKSPACE/.

PPAO oznacza pojemność pamięci operacyjnej danego zestawu.

Jeśli do sekcji stałej w MSAS-ie dokonano przejścia z sekcji w EOL-u za pomocą rozkazu IDZ /GOSEC/, to można powrócić do wykonania następnego rozkazu stosując sekwencję rozkazów:

UMA PPAO - 252\*

DOA +5

PZA PPAO - 252\*

SKO PPAO - 251\*

Takie przejście nie powoduje zerowania wyrażeń ani SAR. Interpreter EOL-u wykorzystuje numery przesyłań 28, 29, 30, 31. Interpreter definiuje swoje przesyłania przy każdym przejściu z MSAS-u do EOL-u za pomocą makrorozkazu IDZ DO SEKCJI.

Natomiast przy przejściu opisanyem powyżej, przesyłania nie są definiowane i wymienionych wyżej numerów nie wolno używać. Wymiana informacji między sekcjami różnojęzycznymi odbywa się zasadniczo poprzez wyrażenia.

#### Budowa wyrażeń

Wyrażenie zapisane jest w pamięci roboczej jako ciąg tzw. ogniw. Ogniwo jest parą kolejnych słów, lecz same ognia nie muszą być w pamięci rozmieszczone kolejno.

Pierwsze słowo ogniska zawiera adres następnego ogniska, drugie zaś informację. Ostatnie ognisko zamiast adresu zawiera liczbę -0. Składnik zapisywany jest w całkowitej ilości ognisk. Ostatnie ognisko składnika zawiera 1 w pozycji znaku słowa zawierającego adres. W słowie informacyjnym mieści się do czterech znaków kodu wewnętrznego lub liczba binarna. Znaki zapisywane są od lewej strony słowa poczynając. Jeśli znaków jest mniej niż cztery, słowo dopełnia się znakami pustymi.

Adres pierwszego ogniska i-tego wyrażenia mieści się w PPA0-73+2i-tym słowie pamięci. W kolejnym słowie mieści się adres ostatniego ogniska tego wyrażenia. Oba adresy mają 1 w pozycji znaku. Jeśli wyrażenie jest puste oba adresy mają wartość -0.

#### Przykład

Niech

E2: -ABCDEF °5 -OKO - +12

Wyrażenie to może być zapisane w pamięci maszyny następująco:

| adres | zawartość | typ zawartości |
|-------|-----------|----------------|
| 1200  | 1350      | binarna        |
| 1201  | ABCD      | znakowa        |
| 1350  | -1402     | binarna        |
| 1351  | EFpp      | znakowa        |
| 1402  | -1500     | binarna        |
| 1403  | +5        | binarna        |
| 1500  | -1450     | binarna        |
| 1501  | OKOp      | znakowa        |

1450 -0 binarna  
 1451 +12p znakowa  
 p- oznacza znak "pusty".

Podany niżej fragment programu ilustruje przekazywanie informacji między sekcjami różnojęzycznymi dla PPAO równej 7680.

SEKCJA TR

- - - - -  
 - - - - -

/\* W PONIŻSZEJ PROCEDURZE PIERWSZY SKŁADNIK  
 WYRAŻENIA E1 ZAWIERAJĄCY LICZBĘ ZMIENNOPRZE-  
 CINKOWĄ W POSTACI DZIESIĘTNEJ ZOSTAJE ZASTA-  
 PIONY DWUOGNIOWYM SKŁADNIKIEM ZAWIERAJĄCYM  
 TĘ LICZBĘ W POSTACI BINARNEJ. PROCEDURA WYKO-  
 RZYSTUJE PRZEJŚCIE DO M-SAS-U\*/.

FLT:PROC

UMIEŚĆ 'XXXXX', A2 /\*REZERWACJA SKŁADNIKA NA  
 WYNIK\*/

IDŹ ZM

USUŃ A1,1

PRZEŚLIJ A2, B1

WRÓĆ

KONIEC

- - - - -  
 - - - - -

KONS

SEST ZM

/\*SEKCJA TA TŁUMACZY LICZBĘ ZMIENNOPRZECINKO-  
 WĄ MIESZCZĄCĄ SIĘ W PIERWSZYM SKŁADNIKU WYRA-  
 ŻENIA E1 NA POSTACЬ BINARNĄ I UMIESZCZA W PIER-  
 WSZYM SKŁADNIKU WYRAŻENIA E2 WYMAZUJĄC POPRZED-  
 NIA ZAWARTOŚĆ\*/.

- - - - -  
 - - - - -

UMA 7428\*

DOA +5

PZA 7428\*

SKO 7429\*

KONS

## DODATEK C

PRZYPISYWANIE URZĄDZEŃ ZMIENNYM WEJŚCIOWYM  
/WYJŚCIOWYM/

Zmienna  $I^<n>/Q^<n>$  przyporządkowana jest  $n-1$ -emu symbolicznemu wejściu /wyjściu/ systemu operacyjnego.

Posługując się zmiennymi  $I^<n>/Q^<n>$  dla  $n > 1$  należy przedtem dołączyć odpowiedni podprogram i dekoder. Dołączenia dokonuje się w MSAS-ie. Określając obszar na podprogram i dekoder należy uwzględnić rozmieszczenie programu głównego, pamięci roboczej i interpretera.



## DODATEK D

**SYGNALIZACJA BŁĘDÓW  
I PRZYCZYNY ZAKOŃCZENIA PROGRAMU**

1. W czasie translacji programu w języku EQL sygnalizowane są następujące błędy:

| Numer błędu | T r e ś c                                                                          |
|-------------|------------------------------------------------------------------------------------|
| 00          | Brak nazwy sekcji                                                                  |
| 01          | Za dużo argumentów                                                                 |
| 02          | Niewłaściwy argument                                                               |
| 03          | Brak testu                                                                         |
| 04          | Niewłaściwy test lub niewłaściwy argument                                          |
| 05          | Niepoprawna struktura blokowa                                                      |
| 06          | Podwójna etykieta                                                                  |
| 07          | Podwójne wystąpienie deklaracji START                                              |
| 08          | Brak nazwy procedury lub nazwy zwrotnicy                                           |
| 09          | Łączna długość programu i zadeklarowanej pamięci roboczej przekracza pojemność PAO |
| 10          | Niepoprawny rozkaz                                                                 |

| Numer błędu | T r e s c                                                   |
|-------------|-------------------------------------------------------------|
| 11          | Przekroczono pojemność tablicy liczb lub napisów            |
| 12          | Niezadeklarowana etykieta                                   |
| 13          | Niezadeklarowana zwrotnica                                  |
| 14          | Użyto niedozwolonego znaku                                  |
| 15          | Niedozwolony znak w tekście                                 |
| 16          | Liczba, tekst lub identyfikator przekracza 60 znaków        |
| 17          | Niepoprawna nazwa lub parametr makrodefinicji               |
| 18          | Identyczne nazwy makrodefinicji                             |
| 19          | Użyto niezadeklarowanego parametru formalnego               |
| 20          | Odwołanie do nieistniejącej makrodefinicji                  |
| 21          | Niezgodna ilość parametrów i argumentów makrodefinicji      |
| 22          | Przekroczono listę odwołań do makrodefinicji                |
| 23          | Łączna długość przedrostka i argumentu przekracza 60 znaków |
| 24          | Niepoprawne zakończenie listy makrodefinicji                |
| 25          | Niepoprawne zakończenie programu                            |
| 26          | Niepoprawne zakończenie odwołania wewnętrznego              |

Informacje o błędach występujących w danej sekcji poprzedzone są tekstem SEKCJA, po którym wypisywana jest nazwa sekcji.

Poszczególne rodzaje błędów sygnalizowane są następująco:

a/ Błąd 00 sygnalizowany jest tekstem  
BLAD 00.

b/ Błędy 12 i 13 sygnalizowane są za pomocą tekstu BLAD, po którym następuje numer błędu, a po nim niezadeklarowana etykieta /12/ lub nazwa niezadeklarowanej zwrotnicy /13/, np.:

BLAD 12 SORT

oznacza, że w sekcji programu nie zadeklarowano etykiety SORT.

c/ Pozostałe błędy są sygnalizowane w następujący sposób: Wypisywane jest słowo BLAD, po którym następuje numer błędu a po nim etykieta i liczony względem tej etykiety numer wiersza, w którym wystąpił błąd. Jeżeli błąd wystąpił przed pojawiением się pierwszej deklaracji etykiety, to zamiast etykiety wypisywane są trzy gwiazdki, a numer wiersza liczony jest względem pierwszego wiersza sekcji nie wliczając makrodefinicji. Np.: BLAD 10 RELACJA 04 oznacza, że w czwartym wierszu względem etykiety RELACJA wystąpił niepoprawny rozkaz<sup>x/</sup>. Błędy w makrodefinicji sygnalizowane są podobnie, z tym że zamiast etykiety wypisywana jest nazwa makrodefinicji, a numer wiersza jest liczony względem wiersza zawierającego tę nazwę.

W przypadku, gdy sekcja programu została napisana w wersji angielskiej zamiast SEKCJA wypisywane jest słowo SECTION, a zamiast słowa BLAD - słowo ERROR.

<sup>x/</sup> Przy lokalizacji błędu należy uwzględnić, że komentarze nie są brane pod uwagę, a zamiast odwołań do makrodefinicji wstawia się do programu treść makrodefinicji.

## 2. Przyczyny zakończenia programu

W programach EOL-u są wykorzystywane pewne numery przyczyn zakończenia dozwolone dla programów użytkowych. Znaczenie tych numerów jest następujące:

- 99 - wykonanie operacji STOP.
- 100 - przepełnienie pamięci roboczej /w pamięci tej zapisywane są wyrażenia i SAR/.
- 101 - próba wykonania operacji plikowej, jeśli kiedykolwiek przedtem w danej sekcji zapisano więcej niż  $n-1$  bloków pamięci roboczej, gdzie  $n$  - ilość zadeklarowanych bloków.
- 102 - przepełnienie pamięci bęбnowej /w pamięci bębnowej zapisywane są pliki/.

|           |        |          |         |
|-----------|--------|----------|---------|
| AHAQ      | AMEN   | OOBY     | BOBY    |
| ATTAQ     | ATTW   | TON      | LIAZBIE |
| BUDHABOKE | BUHAG  | BUHAGROW | BUHOBOK |
| CAMPING   | LINB   | TLINE    | LISBOK  |
| CHIE      | ECLAME | GE       | VER     |

—t, w którym zajmuje się głosowaniem, jest  
przepisem, który na dalej zatwierdza JON, który  
żeby nie zniechęcić sprawu zarządu o głosy  
wyrobacze wykonały zapotrzebowanie JON do zatwierdzenia  
takiego głosowania.

**DODATEK E****LISTA SŁÓW KLUCZOWYCH**

|         |        |         |         |
|---------|--------|---------|---------|
| COFNIJ  | RESET  | RZ      | ER      |
| CZYTAJ  | READ   | SCHOWAJ | SAVE    |
| DODAJ   | ADD    | SEKCJA  | SECTION |
| DZIEL   | DIV    | SKOCZ   | GOTO    |
| IDZ     | GOSEC  | SKOIN   | GOIN    |
| INDEKS  | INDEX  | SKOMI   | GOMI    |
| KONIEC  | END    | SKONA   | GONA    |
| KONP    | KONP   | SKOPL   | GOPL    |
| KONS    | ENDS   | SKREŚL  | DELETE  |
| KOPIUJ  | COPY   | SŁOWO   | WORD    |
| LICZBA  | NUMBER | START   | START   |
| MN      | LT     | STOP    | STOP    |
| MNÓŻ    | MULT   | SZL     | SEC     |
| NS      | FL     | SZUKAJ  | SEARCH  |
| NAZWA   | NAME   | UMIESĆ  | SET     |
| ODEJMIJ | SUB    | USTAW   | RESTORE |
| OML     | SHIC   | USUŃ    | CLEAR   |
| OMIŃ    | SHIFT  | WK      | GT      |
| PAKUJ   | PACK   | WRÓĆ    | RETURN  |
| PISZ    | WRITE  | WSTAW   | PUT     |
| POBIERZ | GET    | WYIN    | CAIN    |
| POLE    | SPACE  | WYKONAJ | CALL    |
| PP      | PR     | WYMI    | CAMI    |

|          |           |        |          |
|----------|-----------|--------|----------|
| PROC     | PROC      | WYNA   | CANA     |
| PRZEŚLIJ | MOVE      | WYPL   | CAPL     |
| ROBOCZE  | WORKSPACE | ZAMIEŃ | EXCHANGE |
| ROZBIJ   | SPLIT     | ZBIJ   | COMPRESS |
| RW       | EQ        | ZNAJDŹ | FIND     |

Przy wprowadzeniu do maszyny program w języku EOL dekodowany jest na postać wewnętrzną zgodnie z regułami użytego dekodera. Tak więc, translator EOL-u przetwarza program przedstawiony w kodzie wewnętrznym. Podczas translacji programu pomijany jest znak: r̄, o ile nie występuje wewnętrz tekstu.

Jak wynika z tabeli KW-ZAM podanej w Dodatku A znaki: Ć, Ñ, Ð, Š, Ž, Ÿ w kodzie wewnętrznym nie występują. W opisie użyto ich wyłącznie dla przejrzystości zamiast znaków: C, N, O, S, Z, Z.

|        |        |        |        |
|--------|--------|--------|--------|
| 1000   | 1100   | 1200   | 1300   |
| 1010   | 11010  | 12011  | 13001  |
| 1020   | 10010  | 10001  | 00100  |
| 1030   | 10000  | 10000  | 00000  |
| 1040   | 10000  | 10000  | 00000  |
| 105000 | 100000 | 100000 | 000000 |
| 1060   | 00000  | 10000  | 00000  |
| 1070   | 10000  | 00000  | 10000  |
| 1080   | 00000  | 00000  | 00000  |
| 1090   | 00000  | 00000  | 00000  |
| 1100   | 00000  | 00000  | 00000  |
| 1110   | 00000  | 00000  | 00000  |
| 1120   | 00000  | 00000  | 00000  |
| 1130   | 00000  | 00000  | 00000  |
| 1140   | 00000  | 00000  | 00000  |
| 1150   | 00000  | 00000  | 00000  |
| 1160   | 00000  | 00000  | 00000  |
| 1170   | 00000  | 00000  | 00000  |
| 1180   | 00000  | 00000  | 00000  |
| 1190   | 00000  | 00000  | 00000  |
| 11A0   | 00000  | 00000  | 00000  |
| 11B0   | 00000  | 00000  | 00000  |
| 11C0   | 00000  | 00000  | 00000  |
| 11D0   | 00000  | 00000  | 00000  |
| 11E0   | 00000  | 00000  | 00000  |
| 11F0   | 00000  | 00000  | 00000  |
| 1200   | 00000  | 00000  | 00000  |
| 1210   | 00000  | 00000  | 00000  |
| 1220   | 00000  | 00000  | 00000  |
| 1230   | 00000  | 00000  | 00000  |
| 1240   | 00000  | 00000  | 00000  |
| 1250   | 00000  | 00000  | 00000  |
| 1260   | 00000  | 00000  | 00000  |
| 1270   | 00000  | 00000  | 00000  |
| 1280   | 00000  | 00000  | 00000  |
| 1290   | 00000  | 00000  | 00000  |
| 12A0   | 00000  | 00000  | 00000  |
| 12B0   | 00000  | 00000  | 00000  |
| 12C0   | 00000  | 00000  | 00000  |
| 12D0   | 00000  | 00000  | 00000  |
| 12E0   | 00000  | 00000  | 00000  |
| 12F0   | 00000  | 00000  | 00000  |
| 1300   | 00000  | 00000  | 00000  |
| 1310   | 00000  | 00000  | 00000  |
| 1320   | 00000  | 00000  | 00000  |
| 1330   | 00000  | 00000  | 00000  |
| 1340   | 00000  | 00000  | 00000  |
| 1350   | 00000  | 00000  | 00000  |
| 1360   | 00000  | 00000  | 00000  |
| 1370   | 00000  | 00000  | 00000  |
| 1380   | 00000  | 00000  | 00000  |
| 1390   | 00000  | 00000  | 00000  |
| 13A0   | 00000  | 00000  | 00000  |
| 13B0   | 00000  | 00000  | 00000  |
| 13C0   | 00000  | 00000  | 00000  |
| 13D0   | 00000  | 00000  | 00000  |
| 13E0   | 00000  | 00000  | 00000  |
| 13F0   | 00000  | 00000  | 00000  |

## DODATEK F

## SKOROWIDZ TERMINÓW

|                     |          |
|---------------------|----------|
| A                   | 4.2.1.1. |
| A                   | 4.2.1.2. |
| A                   | 4.2.1.3. |
| ADD                 | 5.6.     |
| ADRES ROZKAZU       | 3.6.     |
| ADRES ZAPISU        | 3.3.3.   |
| ARGUMENT            | 4.2.     |
| ARGUMENT            | 7.       |
| ARGUMENT WEWNĘTRZNY | 7.       |
| B                   | 4.2.1.1. |
| B                   | 4.2.1.2. |
| B                   | 4.2.1.3. |
| B                   | 4.2.3.   |
| C                   | 4.2.1.1. |
| C                   | 4.2.1.2. |
| CAIN                | 5.5.3.   |
| CALL                | 5.5.1.   |
| CAŁKOWITA           | 3.3.2.   |
| CAŁKOWITA           | 4.2.5.   |
| CAMI                | 5.5.2.   |
| CANA                | 5.5.3.   |

|                              |          |
|------------------------------|----------|
| CAPL                         | 5.5.2.   |
| CIĄG ZNAKÓW                  | 3.1.     |
| CLEAR                        | 5.3.     |
| COFNIJ                       | 5.10.1.  |
| COMPRESS                     | 5.7.3.   |
| COPY                         | 5.1.4.   |
| CYFRA                        | 2.1.2.   |
| CZYTAJ                       | 5.1.2.   |
| CZYTANIE                     | 4.2.11.  |
| <br>D                        | 4.2.1.1. |
| D                            | 4.2.1.2. |
| D                            | 4.2.3.   |
| DEKLARACJA                   | 5.11.    |
| DELETE                       | 5.3.     |
| DIV                          | 5.6.     |
| DODAJ                        | 5.6.     |
| DOPISYWANIE                  | 4.2.1.2. |
| DZIEL                        | 5.6.     |
| <br>E                        | 4.2.1.3. |
| ELEMENT MAKRODEFINICJI       | 7.       |
| ELEMENT SEKCJI MODYFIKOWANEJ | 7.       |
| END                          | 6.1.     |
| ENDS                         | 6.1.     |
| EQ                           | 5.4.1.   |
| EQ                           | 5.4.2.   |
| ER                           | 5.4.3.   |
| ETYKIETA                     | 4.2.2.   |
| EXCHANGE                     | 5.1.9.   |
| <br>FIND                     | 5.9.1.   |
| FL                           | 5.4.2.   |
| <br>GET                      | 5.1.6.   |
| GOIN                         | 5.5.3.   |
| GOMI                         | 5.5.2.   |
| GONA                         | 5.5.3.   |

|                            |          |
|----------------------------|----------|
| GOPL                       | 5.5.2.   |
| GOSEC                      | 5.10.4.  |
| GOTO                       | 5.5.1.   |
| GT                         | 5.4.1.   |
| I                          | 4.2.1.1. |
| I                          | 4.2.1.3. |
| IDENTYFIKATOR              | 4.2.2.   |
| IDZ                        | 5.10.4.  |
| ILOŚĆ                      | 5.1.1.   |
| INDEKS                     | 4.2.1.   |
| INDEKS                     | 5.11.1.  |
| K                          | 4.2.1.4. |
| KLASA                      | 4.2.3.   |
| KOMENTARZ                  | 5.12.    |
| KONIEC                     | 6.1.     |
| KONP                       | 6.1.     |
| KONS                       | 6.1.     |
| KOPIUJ                     | 5.1.4.   |
| L                          | 4.2.3.   |
| LICZBA                     | 3.3.2.   |
| LICZBA                     | 5.7.2.   |
| LITERA                     | 2.1.1.   |
| LITERA ARGUMENTU           | 4.2.1.   |
| LITERA INDEKSOWANA         | 4.2.1.   |
| LISTA MAKRODEFINICJI       | 7.       |
| LISTA PARAMETRÓW           | 7.       |
| LISTA ZWROTNICY INDEKSOWEJ | 5.11.1.  |
| LISTA ZWROTNICY NAZWOWEJ   | 5.11.1.  |
| LT                         | 5.4.1.   |
| MAKRODEFINICJA             | 7.       |
| MN                         | 5.4.1.   |
| MNOŻ                       | 5.6.     |
| MOVE                       | 5.1.1.   |
| MULT                       | 5.6.     |

|                             |          |
|-----------------------------|----------|
| N                           | 4.2.1.4. |
| NAZWA                       | 5.11.1.  |
| NAZWA MAKRODEFINICJI        | 7.       |
| NAZWA PROCEDURY             | 6.1.     |
| NAZWA SEKCJI                | 5.10.4.  |
| NAZWA SEKCJI                | 6.1.     |
| NAZWA ZWROTNICY             | 5.11.1.  |
| NOTACJA WARTOŚCI ZMIENNEJ   | 1.5.     |
| NOWA LINIA                  | 4.2.4.   |
| NS                          | 5.4.2.   |
| NUMBER                      | 5.7.2.   |
| ODEJMIJ                     | 5.6.     |
| ODWOŁANIE                   | 7.       |
| ODWOŁANIE DO MAKRODEFINICJI | 7.1.     |
| ODWOŁANIE WEWNĘTRZNE        | 7.       |
| OML                         | 5.9.3.   |
| OMIŃ                        | 5.8.2.   |
| OMIŃ I LICZ                 | 5.9.3.   |
| OPERATOR                    | 4.1.     |
| P                           | 4.2.1.4. |
| PACK                        | 5.10.2.  |
| PAKUJ                       | 5.10.2.  |
| PARAMETR                    | 7.       |
| PISZ                        | 5.1.3.   |
| PLIK                        | 3.4.     |
| POBIERZ                     | 5.1.6.   |
| POLE                        | 4.2.5.   |
| PP                          | 5.4.2.   |
| PR                          | 5.4.2.   |
| PROC                        | 6.1.     |
| PROCEDURA                   | 6.1.     |
| PROGRAM                     | 6.1.     |
| PRZEKSZTAŁCENIE             | 5.7.     |
| PRZEKSZTAŁCENIE NA LICZBĘ   | 5.7.2.   |

|                                     |          |
|-------------------------------------|----------|
| PRZEKSZTALCENIE NA SŁOWO            | 5.7.1.   |
| PRZESŁANIE                          | 5.1.     |
| PRZEŚLIJ                            | 5.1.1.   |
| PUT                                 | 5.1.5.   |
| <br>Q                               | 4.2.1.2. |
| <br>R                               | 4.2.3.   |
| READ                                | 5.1.2.   |
| RELACJA                             | 5.4.     |
| RELACJA ARYTMETYCZNA                | 5.4.1.   |
| RELACJA ARYTMETYCZNA RÓWNOŚCI       | 5.4.1.   |
| RELACJA ARYTMETYCZNA UPORZĄDKOWANIA | 5.4.1.   |
| RELACJA TESTOWA                     | 5.4.2.   |
| RELACJA TESTOWA RÓWNOŚCI            | 5.4.2.   |
| RELACJA TESTOWA UPORZĄDKOWANIA      | 5.4.2.   |
| RELACJA ZAPISU                      | 5.4.3.   |
| RESET                               | 5.10.1.  |
| RESTORE                             | 5.1.8.   |
| RETURN                              | 5.5.4.   |
| ROBOCZE                             | 5.11.2.  |
| ROZBIJ                              | 5.7.4.   |
| ROZBIJANIE                          | 5.7.4.   |
| ROZKAZ                              | 4.1.     |
| ROZKAZ                              | 5.       |
| ROZKAZ ARYTMETYCZNY                 | 5.6.     |
| ROZKAZ STERUJĄCY                    | 5.5.     |
| RW                                  | 5.4.1.   |
| RW                                  | 5.4.2.   |
| RZ                                  | 5.4.3.   |
| <br>SAVE                            | 5.1.7.   |
| SCHOWAJ                             | 5.1.7.   |
| SEARCH                              | 5.8.1.   |
| SEC                                 | 5.9.2.   |
| SECTION                             | 6.1.     |
| SECTION                             | 7.       |
| SEKCJA                              | 6.1.     |

|                       |          |
|-----------------------|----------|
| SEKCJA                | 7.       |
| SEKCJA MODYFIKOWANA   | 7.       |
| SEKCJA M-SAS          | 6.1.     |
| SEKCJA M-SAS          | B-1      |
| SEKCJA ŹRÓDŁOWA       | 7.       |
| SET                   | 5.2.     |
| SHIC                  | 5.9.3.   |
| SHIFT                 | 5.8.2.   |
| SKŁADNIA              | 1.3.     |
| SKŁADNIK              | 3.3.     |
| SKOCZ                 | 5.5.1.   |
| SKOIN                 | 5.5.3.   |
| SKOK                  | 5.5.1.   |
| SKOK POWROTNY         | 5.5.4.   |
| SKOK WARUNKOWY        | 5.5.2.   |
| SKOK ZWROTNICY        | 5.5.3.   |
| SKOMI                 | 5.5.2.   |
| SKONA                 | 5.5.3.   |
| SKOPL                 | 5.5.2.   |
| SKREŚL                | 5.3.     |
| SŁOWO                 | 3.3.1.   |
| SŁOWO                 | 5.7.1.   |
| SPACE                 | 4.2.5.   |
| SPLIT                 | 5.7.4.   |
| START                 | 5.11.3.  |
| START PROGRAMU        | 5.11.3.  |
| STOP                  | 5.10.3.  |
| STOS ADRESÓW ROZKAZÓW | 3.6.     |
| SZL                   | 5.9.2.   |
| SZUKAJ                | 5.8.1.   |
| SZUKAJ I LICZ         | 5.9.2.   |
| SZUKANIE              | 5.8.     |
| SUB                   | 5.6.     |
| SYMBOLE PODSTAWOWE    | 2.       |
| T                     | 4.2.1.3. |
| TEKST                 | 4.2.4.   |

|                            |          |
|----------------------------|----------|
| TEST KOPIOWANIA            | 5.1.4.   |
| TEST PRZESŁANIA            | 5.1.1.   |
| TREŚĆ MAKRODEFINICJI       | 7.       |
| TRESC SEKCJI MODYFIKOWANEJ | 7.       |
| <br>UMIESZCZENIE           | 5.2.     |
| UMIEŚĆ                     | 5.2.     |
| UMIEŚĆ NA WYJŚCIU          | 5.2.     |
| UMIEŚĆ W PLIKU             | 5.2.     |
| UMIEŚĆ W WYRAŻENIU         | 5.2.     |
| USTAW                      | 5.1.8.   |
| USUNIĘCIE                  | 5.3.     |
| USUŃ                       | 5.3.     |
| USUŃ Z PLIKU               | 5.3.     |
| USUŃ Z WEJŚCIA             | 5.3.     |
| USUŃ Z WYRAŻENIA           | 5.3.     |
| <br>WARTOŚCI POCZĄTKOWE    | 3.7.     |
| WARTOŚĆ LOGICZNA           | 3.5.     |
| WARTOŚĆ PLIKU              | 3.4.     |
| WARTOŚĆ STOSU ADRESÓW      | 3.6.     |
| WARTOŚĆ WEJŚCIA            | 3.1.     |
| WARTOŚĆ WYJŚCIA            | 3.2.     |
| WARTOŚĆ WYRAŻENIA          | 3.3.     |
| WEJŚCIE                    | 3.1.     |
| WERSJE JĘZYKA EOL          | 1.4.     |
| WK                         | 5.4.1.   |
| WORD                       | 5.7.1.   |
| WORKSPACE                  | 5.11.2.  |
| WRITE                      | 5.1.3.   |
| WRÓĆ                       | 5.5.4.   |
| WSTAW                      | 5.1.5.   |
| WYIN                       | 5.5.3.   |
| WYJŚCIE                    | 3.2.     |
| WYKONAJ                    | 5.5.1.   |
| WYKORZYSTANIE              | 4.2.1.3. |
| WYMI                       | 5.5.2.   |

|                         |          |
|-------------------------|----------|
| WYNA                    | 5.5.2.   |
| WYPL                    | 5.5.2.   |
| WYRAŻENIE               | B-2      |
| WYRAŻENIE               | 3.3.     |
| Y                       | 4.2.1.2. |
| Z                       | 4.2.1.2. |
| ZAMIEN                  | 5.1.9.   |
| ZAPIS                   | 3.4.     |
| ZBIJ                    | 5.7.3.   |
| ZBIJANIE                | 5.7.3.   |
| ZBIÓR ZNAKÓW DLA DANYCH | A-1      |
| ZDANIE                  | 6.1.     |
| ZDANIE ZĘWNETRZNE       | 6.1.     |
| ZLICZANIE               | 5.9.     |
| ZMIENNA                 | 3.       |
| ZMIENNA LOGICZNA        | 3.5.     |
| ZNACZNIK                | 1.5.     |
| ZNAJDZ                  | 5.9.1.   |
| ZNAK                    | 2.1.     |
| ZNAK SPECJALNY          | 2.1.3.   |
| ZWROTNICA               | 5.11.1.  |
| ZWROTNICA INDEKSOWA     | 5.11.1.  |
| ZWROTNICA NAZWOWA       | 5.11.1.  |

## SYMBOLE

|    |        |
|----|--------|
| *  | 4.2.4. |
| ** | 4.2.4. |
| *L | 4.2.4. |
| *' | 4.2.4. |
| ,  | 4.2.4. |
| {  | 1.3.   |
| [  | 1.3.   |
| }  | 1.3.   |
| ]  | 1.3.   |
| λ  | 2.1.   |
| —  | 2.1.   |

