

## 5. PROCEDURY STANDARDOWE

### 5.1. Wiadomości ogólne o procedurach standardowych

Procedurami standardowymi nazywamy te procedury, których ciała w postaci programów w języku wewnętrznym maszyny są stałe umieszczone w translatorze. Dlatego procedur standardowych nie deklaruje się, a mimo to można je wywoływać. Lista procedur standardowych zależy od budowy translatora. Jedną i tą samą maszyną może mieć więcej niż jeden translator i w zależności od tego, którego z nich użyjemy, może zmienić się również lista procedur standardowych. Poniżej podamy listę procedur standardowych w przypadku użycia translatora o nazwę GIER ALGOL III BUF dla maszyn typu GIER. W języku ALGOL procedury te nie występują z wyjątkiem funkcji standardowych o których była już mowa w paragrafie 1.15.

Translator GIER ALGOL III BUF powstał jako modyfikacja translatora GIER ALGOL III przez dodanie do tego ostatnich procedur standardowych

from buf, from car, to buf, to car.\*

Jeśli zatem korzystamy z translatora GIER ALGOL III, nie możemy z powyższych czterech procedur korzystać. Przed przystąpieniem do obliczeń na maszynie należy sprawdzić, z jakim translatozem będziemy mieć do czynienia.

Do podanej niżej listy dołączyliśmy również drumplace, choć nie jest to procedura lecz zmienna standardowa, ponieważ nazwa jej jest podobnie zastrzeżona jak nazwy procedur standardowych, a musi być omówiona łącznie z procedurami wspólnej pracy z bębniem.

Wszystkie procedury standardowe można podzielić następująco:

9 funkcji standardowych:	abs	
	sign	
	entier	
	sqr	
	sin	
	cos	
	arctan	
	ln	
	exp	
11 procedur wyjścia:	output	write
	outtext	writetext
	outsp	
	outcr	writecr
	outclear	
	outsum	
	outchar	writechar

\* Powyższa modyfikacja została dokonana w Polsce przez A. Salwickiego na Uniwersytecie Warszawskim.

```

9 procedur wejścia:      input
                          inone      typein
                          kbon
                          inchar     typechar
                          setchar
                          char
                          lyn

2 procedury wejściowo-wyjściowe:  outcopy i writecopy
2 procedury współpracy z bębniem:  to drum i from drum
1 zmienna standardowa dla współpracy
  z bębniem i buforem:              drumplace
2 procedury współpracy z buforem:  to buf i from buf
2 procedury współpracy z karuzelą: to car i from car
5 procedur dla fragmentów w kodzie wewnętrznym maszyny:

    pack
    split
    gier
    gierproc
    gierdrum

```

Ponieważ funkcje standardowe zostały już omówione w paragrafie 1.15, w rozdziale niniejszym zajmiemy się pozostałymi niefunkcyjnymi procedurami standardowymi. Wszystkie one nie służą do obliczania funkcji, ale wykonanie niektórych daje jednak wartość bądź to liczbową, bądź logiczną i dlatego wywołujemy je analogicznie do procedur funkcyjnych. Takie procedury standardowe (ale tylko standardowe) będziemy nazywali *pseudofunkcyjnymi*.

Do procedur pseudofunkcyjnych należą:

```

inone, kbon, typein, inchar, typechar, char, lyn,
from drum, to drum, from buf, to buf, from car, to car,
pack, split, gier, gierproc, gierdrum.

```

Wywołanie pozostałych procedur ma postać odrębnych instrukcji.

Niektóre z procedur pseudofunkcyjnych, a mianowicie

```

from drum, to drum, from buf, to buf, from car, to car,
pack, split, gier, gierproc, gierdrum

```

można wywoływać zarówno wewnątrz wyrażeń, jak też w postaci odrębnych instrukcji, jak na przykład

```
a:= c - to drum(A);
```

czy po prostu

```
to drum(A);
```

W następnych paragrafach będziemy kolejno poznawać niefunkcyjne procedury standardowe. Przed tym jednak zapoznamy się jeszcze z tzw. łańcuchami, ponieważ mają one zastosowanie w niektórych z tych procedur, oraz z urządzeniami zewnętrznymi maszyny GIER.

## 5.2. Łańcuchy

Określenie łańcuchów podane w niniejszym paragrafie będzie określeniem przyjętym w GIER-ALGOLu. Różni się ono nieco

od klasycznego określenia algolowskiego, co jednak dla programowania nie ma zasadniczego znaczenia.

Ł a ń c u c h a m i (strings; czytaj: stryngs) nazywamy t e k s t y (proper strings; czytaj: proper stryngs) i w z o r c e (layouts; czytaj: lejauts).

T e k s t e m nazywamy dowolny ciąg znaków nie zawierający < i >. W szczególnym przypadku możemy mieć do czynienia z t e k s t e m p u s t y m, tzn. ciągiem nie zawierającym żadnego znaku, a mimo to uważanym za tekst. W tekstach symbol „ ” jest równoznaczny z odstępem (SPACE) i jest drukowany jako odstęp, natomiast powrót karetki wraz z przejściem do nowego wiersza (CARriage RETurn) nie ma specjalnego symbolu, a w tekstach jest sygnalizowany samym wykonaniem przejścia do nowego wiersza z powrotem karetki. Łańcuchy stanowią jedyny przypadek, gdy odstępy i powroty karetki nie są ignorowane przez translator. Translator rozróżnia ten przypadek po tym, że – jak zobaczymy później – teksty zawsze będą występować w nawiasach < < i >, a wzorce w nawiasach < i >.

Oto przykłady tekstów:

Wyniki,,:

TABLICA,,,A1,,,A2,,,A3

Parametr a b c

W z o r c e m nazywamy specjalny ciąg znaków służący do określenia sposobu drukowania liczb. Składa się on z trzech kolejno po sobie następujących części: wzorca znaku, wzorca dziesiętnego i wzorca cechy.

W z o r z e c z n a k u może mieć postać następującą

- albo + albo ±

gdzie znak „±” pisze się jako podkreślony plus. Wzorzec znaku „-” oznacza, że chcemy drukować tylko minusy dla liczb ujemnych, a w miejsce plusów dla liczb dodatnich odstępy. Wzorzec znaku „+” oznacza, że chcemy drukować zarówno plusy jak i minusy. W obu tych przypadkach znak jest przesuwany – gdy jest to możliwe – w prawo tak, że zostaje on wydrukowany bezpośrednio przed pierwszą wydrukowaną cyfrą, gdy we wzorcu nie jest ona poprzedzona ani kropką, ani odstępem, bezpośrednio przed kropką, jeśli kropka poprzedza pierwszą wydrukowaną cyfrę, wreszcie na pozycji oddzielonej jednym odstępem od pierwszej wydrukowanej cyfry, jeśli odstęp ten jest w tym miejscu przewidziany we wzorcu. Takie przesunięcie znaku w prawo następuje, gdy część całkowita liczby drukowanej ma mniej cyfr znaczących niż przewiduje wzorzec. Wzorzec znaku „±” oznacza, że chcemy drukować zarówno plusy, jak i minusy, ale zawsze na pierwszej pozycji, tzn. bez ewentualnego dosuwania znaku w prawo.

Wzorzec znaku może być we wzorcu pominięty. Oznacza to, że znaku w ogóle nie chcemy drukować. Gdyby wtedy pojawiła się do druku liczba ujemna, będzie ona co prawda poprawnie z minusem wydrukowana, ale nastąpi zmiana przewidywanego układu kolumn: znak minus, dla którego nie było zarezerwowanego miejsca, pojawi się jak w przypadku wzorca znaku „-”, ale ca-

ły zapis przesunie się o jedno miejsce w prawo, zajmując o jedno miejsce więcej niż przewiduje wzorzec. Tego rodzaju druk, wykraczający poza przewidziany wzorzec, nazywamy **d r u k i e m a l a r m o w y m**. Należy podkreślić, że druk alarmowy daje poprawną wartość drukowanej liczby, ale psuje układ kolumn.

**W z o r z e c d z i e s i ę t n y** składa się z liter d, z których każda reprezentuje jedną cyfrę znaczącą lub początkowe zera nieznaczące, oraz ewentualnie pewnej liczby zer pisanych zawsze tylko na końcu wzorca dziesiętnego, których rolę omówimy później. Jeśli wzorzec dziesiętny ma reprezentować liczbę niecałkowitą, pomiędzy powyższymi symbolami albo przed nimi umieszczamy ponadto w odpowiednim miejscu kropkę. Pierwszą literę d (ale tylko pierwszą) w przypadku, gdy nie jest ona po kropce, można zastąpić literą n i wtedy właściwe ułamki dziesiętne są drukowane z zerem przed kropką, a liczba zero przewidziana wzorcem jako całkowita - jako 0. Gdy pierwszą literą we wzorcu dziesiętnym w przypadku, gdy nie jest ona po kropce, jest d, pojedyncze zero przed kropką zostaje zastąpione odstępem, podobnie jak zero (ale nieopatrzone znakiem) oznaczające liczbę całkowitą 0, z tym, że na miejsce tego odstępu może w pewnych przypadkach być przesunięty znak, jak to było omówione dla wzorca znaku. Jeśli jest przewidziany druk znaku "+", liczba 0 jest zawsze drukowana, niezależnie od tego, czy występuje ona według wzorca jako całkowita czy rzeczywista, i niezależnie od tego, czy pierwszą literą wzorca dziesiętnego jest n czy d.

Jeżeli część całkowita liczby, którą mamy wydrukować, jest mniejsza od przewidzianej danym wzorcem, nadliczbowe zarezerwowane miejsca są wypełniane odstępami. Jeśli np. wzorzec podaje

+ddd.dd

a drukowana liczba jest co do modułu zawarta między 1 i 10, to znak zostaje przesunięty o dwa miejsca w prawo, a na dwu pierwszych pozycjach będą odstępami. Nie dotyczy to oczywiście liczb drukowanych jako zmiennoprzecinkowe, ponieważ istnieje wtedy możliwość dopasowania mantysy do danego wzorca dziesiętnego poprzez odpowiednią zmianę cechy.

Jeżeli liczba, którą mamy wydrukować, jest większa od przewidzianej danym wzorcem, zostaje ona poprawnie wydrukowana, ale z odpowiednio dopisaną cechą, co powoduje wykroczenie poza przewidziany wzorzec, czyli tzw. druk alarmowy.

Kropka jest w liczbach dziesiętnych drukowana na stałej pozycji przewidzianej wzorcem, chyba że mamy do czynienia z drukiem alarmowym, który może spowodować przesunięcie kropki.

Jak już było powiedziane, na końcu wzorca dziesiętnego można umieścić pewną ilość zer. Ich działanie zależy od tego, czy drukowana liczba jest zerem, czy nie. Dla liczby zero, zera występujące na końcu wzorca dziesiętnego mają to samo znaczenie, co litery d. Dla liczb różnych od zera, miejsca zarezerwowane we wzorcu przez zera są wykorzystywane w przypadku, gdy liczba drukowana jest mniejsza od przewidzianej wzorcem, na druk dodatkowych cyfr znaczących z tym jednakże warunkiem, że liczba wydrukowanych cyfr znaczących nie może przekroczyć liczby miejsc zarezerwowanych we wzorcu literami d i n. Miejsca zarezerwowane we wzorcu przez zera i niewykorzystane do druku cyfr znaczących są w druku wypełniane od-

stępami, gdy są po kropce i po ostatniej cyfrze znaczącej albo przed kropką i przed pierwszą cyfrą znaczącą, a zerami, gdy są po kropce i przed pierwszą cyfrą znaczącą albo przed kropką i po ostatniej cyfrze znaczącej. Jeśli na skutek powyższych reguł po kropce mają być drukowane tylko odstępy, kropka również zostaje zastąpiona odstępem.

Przed drukiem liczba drukowana jest zawsze prawidłowo zaokrąglona do ilości cyfr znaczących przewidzianych wzorcem.

Na przykład liczba 567.8 będzie według wzorca ddd wydrukowana jako 568, według wzorca dddd.0 jako 567.8, a według wzorca dddd.000 jako 567.80. Ta sama liczba według wzorca d.d będzie wydrukowana jako 5.7, co stanowi druk alarmowy, gdyż dla zachowania poprawnej wartości liczby wyłączyliśmy się z przewidywanego wzorca (dopisanie nieprzewidzianej cechy).

Liczne inne przykłady znajdzie czytelnik na końcu niniejszego podręcznika pod tytułem WZORCE DRUKU. Podano tam druk według najróżnorodniejszych wzorców dla liczby 0, ośmiu liczb typu real i czterech liczb typu integer.

Wzorzec c e c h y składa się z dziesiątki algolowskiej, wzorca znaku cechy i liter d reprezentujących poszczególne cyfry cechy. Wzorzec znaku cechy jest podawany na tych samych zasadach co wzorzec znaku mantysy czyli znaku całej liczby, z tym jednak, że oba te wzorce znaków nie muszą być jednakowe, tzn. wzorzec znaku cechy może się różnić od wzorca znaku mantysy.

Gdy cecha liczby drukowanej w postaci zmiennoprzecinkowej jest równa zeru, to cecha ta nie jest drukowana, jak również nie jest drukowana dziesiątka algolowska, a na ich miejsce przychodzą odstępy.

Dziesiątka algolowska jest drukowana zawsze bezpośrednio przed pozycją przeznaczoną na znak cechy, a gdy na ten znak nie rezerwuje się miejsca, bezpośrednio przed pierwszą cyfrą cechy. Jeśli cecha drukowanej liczby ma mniej cyfr niż przewiduje wzorzec cechy, są możliwe dwa przypadki:

- jeżeli wzorcem znaku cechy jest  $\pm$ , to dziesiątka algolowska pozostaje na swym miejscu, tak samo jak i znak cechy, a na niewykorzystane pozycje cechy wchodzi odstępy, z tym że pozostałe cyfry cechy są drukowane na swoich pozycjach;

- jeżeli wzorcem znaku cechy nie jest  $\pm$ , to dziesiątka algolowska zostaje przesunięta w prawo o liczbę miejsc niewykorzystanych dla cechy i tyleż odstępów pojawia się dodatkowo między mantysą a dziesiątką algolowską.

Nie istnieje możliwość drukowania dziesiątki algolowskiej bez drukowania po niej cechy a przed nią mantysy. Jest to warunek silniejszy od obowiązującego w ogóle przy zapisach liczb zmiennoprzecinkowych w ALGOLu, ponieważ żąda się dodatkowo obowiązkowego druku mantysy. To dodatkowe ograniczenie występuje w GIER-ALGOLu tylko dla druku liczb. Nie będzie ono obowiązywało na przykład przy wprowadzaniu danych.

Gdy nie przewidujemy druku liczb zmiennoprzecinkowych, wzorzec cechy może być we wzorcu pominięty. Natomiast w każdym wzorcu musi być zawsze wzorzec dziesiętny.

Jeżeli wzorzec zawiera wzorzec cechy, a na końcu wzorca dziesiętnego występuje m zer, powoduje to, że drukowana cecha będzie podzielna przez  $m+1$ , co jest uzyskiwane w ten sposób, że mantysa zostaje ewentualnie przesunięta odpowied-

nio w prawo na miejsca zarezerwowane we wzorcu przez dodatkowe zera, według reguł opisanych wyżej dla wzorca dziesiętnego. Należy tu zauważyć, że dodatkowe zera we wzorcu nie reprezentują dodatkowych cyfr znaczących, których liczba jest dyktowana wyłącznie przez litery  $d$  i  $n$ , a tylko rezerwują miejsca na ewentualne przesunięcia mantysy. Należy tu jeszcze zapamiętać, że wspomniany warunek podzielności cechy przez  $m+1$  nie dotyczy alarmowego druku cechy, gdy we wzorcu nie przewidziano wzorca cechy.

Jeżeli cecha drukowanej liczby jest większa od przewidzianej wzorcem, liczba zostaje wydrukowana poprawnie, ale w postaci druku alarmowego, tzn. przez zajęcie przez cechę większej liczby miejsc dziesiętnych niż przewiduje wzorzec, co powoduje wyłamanie się z przewidzianego układu kolumn.

Należy zapamiętać, że przypadek, gdy liczba drukowana jest mniejsza od przewidzianej wzorcem, nie powoduje druku alarmowego, a jedynie co najwyżej zmniejszenie ilości wydrukowanych cyfr znaczących. Wyjątek stanowi przypadek, gdy wzorzec przewiduje druk cechy bez znaku, a cecha wypada ujemna. W mantysie nie następuje wtedy zmniejszenie ilości cyfr znaczących, a za to pojawia się druk alarmowy cechy: znak minus pojawia się na miejscu przewidzianym dla pierwszej cyfry cechy i w ten sposób zapis cyfr cechy przesuwają się o jedno miejsce w prawo. Ale w przypadku cechy ujemnej nie następuje nigdy powiększenie przez druk alarmowy ilości cyfr przewidzianych we wzorcu dla cechy.

Do wzorca można wprowadzać odstęp, co w druku powoduje odstęp między odpowiednimi cyframi, używając znaku  $\_$  lub robiąc odstęp w samym wzorcu. Jest to jednak dozwolone tylko we wzorcu dziesiętnym i tylko między symbolami  $d$ ,  $n$ ,  $0$ . Wynika stąd, że wzorzec nie może rozpoczynać się ani kończyć odstępem, nie może zawierać odstepu obok kropki czy dziesiętnej algolowskiej, jak również nie może zawierać dwu odstępów obok siebie. Należy tu wyraźnie podkreślić, że są to ograniczenia dla wzorca, a nie dla liczb drukowanych.

A oto przykłady wzorców:

- 1)  $+ddd\ ddd.d$
- 2)  $-dd.d_n-d$
- 3)  $ndd.dd0\ 000$
- 4)  $-n.ddd_d+d$
- 5)  $+ddd.000_n-dd$
- 6)  $.dddd_d d$

Poniżej podajemy przykłady druku liczb według powyższych wzorców, z tym że pod kreską poziomą są przykłady druku alarmowego:



$$\begin{array}{r}
 1) \quad +123\,456.8 \\
 -12\,345.7 \\
 +1\,234.6 \\
 -123.5 \\
 +12.3 \\
 -1.2 \\
 +.1 \\
 +.0 \\
 \hline
 +123\,456.8_{10^2} \\
 -123\,456.8_{10^{13}}
 \end{array}$$

$$\begin{array}{r}
 2) \quad -87.6_{10^{-2}} \\
 87.6_{10^{-1}} \\
 -87.6 \\
 87.6_{10^1} \\
 -87.6_{10^2} \\
 87.6_{10^3} \\
 87.6_{10^{-9}} \\
 8.8_{10^{-9}} \\
 .9_{10^{-9}} \\
 .0 \\
 \hline
 87.6_{10^{12}} \\
 -87.6_{10^{117}}
 \end{array}$$

$$\begin{array}{r}
 3) \quad 123.45 \\
 12.345 \\
 1.234\,5 \\
 0.123\,45 \\
 0.012\,345 \\
 0.001\,235 \\
 0.000\,123 \\
 0.000\,012 \\
 0.000\,001 \\
 0.000\,000 \\
 \hline
 -123.45 \\
 -1.234\,5 \\
 -0.000\,012 \\
 \hline
 123.45_{10^5} \\
 -123.45_{10^2}
 \end{array}$$

$$\begin{array}{r}
 .4) \quad -1.2345_{10^+2} \\
 1.2345_{10^+1} \\
 -1.2345 \\
 1.2345_{10^{-1}} \\
 -1.2345_{10^{-2}} \\
 1.2345_{10^{-9}} \\
 0.1235_{10^{-9}} \\
 0.0123_{10^{-9}} \\
 0.0012_{10^{-9}} \\
 0.0001_{10^{-9}} \\
 0.0000 \\
 \hline
 -1.2345_{10^+11} \\
 1.2345_{10^+124}
 \end{array}$$

5)	+123	$10^{12}$
	- 12.3	$10^{12}$
	+ 1.23	$10^{12}$
	- .123	$10^{12}$
	+123	$10^8$
	- 12.3	$10^8$
	+ 1.23	$10^8$
	- .123	$10^8$
	+123	$10^4$
	- 12.3	$10^4$
	+ 1.23	$10^4$
	- .123	$10^4$
	+123	
	- 12.3	
	+ 1.23	
	- .123	
	+123	$10^{-4}$
	- 12.3	$10^{-4}$
	+ 1.23	$10^{-4}$
	- .123	$10^{-4}$
	- .123	$10^{-28}$
	- .123	$10^{-96}$
	- .012	$10^{-96}$
	- .001	$10^{-96}$
	+ .000	
	-123	$10^{100}$
	123	$10^{127}$

6)	.87654	$10^9$
	.87654	$10^8$
	.87654	$10^7$
	.87654	$10^6$
	.87654	$10^5$
	.87654	$10^3$
	.87654	$10^2$
	.87654	$10^1$
	.87654	
	.87654	$10^{-1}$
	.87654	$10^{-2}$
	.87654	$10^{-3}$
	.87654	$10^{-4}$
	.87654	$10^{-5}$
	.87654	$10^{-6}$
	.87654	$10^{-7}$
	.87654	$10^{-8}$
	.87654	$10^{-9}$
	.08765	$10^{-9}$
	.00877	$10^{-9}$
	.00088	$10^{-9}$
	.00009	$10^{-9}$
	.00000	
	.87654	$10^{10}$
	.87654	$10^{11}$
	-.87654	$10^8$
	-.00088	$10^{-9}$



Dla wzorców istnieją następujące ograniczenia:

- liczba liter d i n we wzorcu dziesiętnym musi być nie większa od 15,
- liczba symboli n, d, 0 na lewo od kropki musi być nie większa od 15,
- liczba symboli d, 0 na prawo od kropki we wzorcu dziesiętnym musi być nie większa od 15,
- liczba liter d we wzorcu cechy musi być nie większa od 7,
- odstępy mogą być tylko w takich pozycjach, które są poprzedzone w danym wzorcu mniej niż 20 symbolami n, d, 0 i kropką.

Ograniczenia powyższe są tak niekrępujące, że nie mają praktycznego znaczenia.

Liczne przykłady druku znajdzie czytelnik na końcu podręcznika pod wspomnianym już tytułem WZORCE DRUKU.

### 5.3. Zadanie

Napisać wzorce, według których zostały wydrukowane liczby w następujących pięciu przypadkach:

$$1) \begin{array}{|l} +.58 \ 477 \\ -.00 \ 296 \\ -.00 \ 000 \\ +.13 \ 400 \end{array}$$

$$2) \begin{array}{|l} 9876.5 \\ 53.112 \\ 0.1684 \\ 0.0031 \end{array}$$

$$3) \begin{array}{|l} + \ 1.33 \\ -279 \quad 10^+ 9 \\ + \ 68.8 \quad 10^{-12} \\ -784 \quad 10^{-3} \end{array}$$

$$4) \begin{array}{|l} 43 \ 21 \ 00 \quad 10^{+5} \\ - \ 85 \ 63 \\ -1 \ 28.6 \quad 10^{-5} \\ 0.79 \end{array}$$

$$5) \begin{array}{|l} 223 \ 000 \quad 10^7 \\ 7 \ 650 \\ .013 \quad 10^{-14} \\ 4.38 \quad 10^{21} \end{array}$$

5.4. Zadanie

Jak będą wydrukowane liczby

567  
 -567.00  
 -.003679  
 0  
 1388.35649  
 -69.7<sub>10</sub><sup>+4</sup>

według wzorców

+d  
 -d.d  
 d.dd  
+dddddd  
 -nd.ddd  
 +d.d<sub>10</sub>ddd  
 -d.d<sub>10</sub>+ddd  
 +d.d<sub>10</sub>+ddd  
 -ddd.d0000  
 +d.dd00<sub>10</sub>+d  
+d<sub>10</sub>dd.d<sub>10</sub>0<sub>10</sub>+d

5.5. Wyrażenia łańcuchowe

Prostym wyrażeniem łańcuchowym nazywamy:

- tekst ujęty w nawiasy  $\langle \langle i \rangle \rangle$ ,
- wzorzec ujęty w nawiasy  $\langle i \rangle$ ,
- parametr formalny kategorii string jakiejś procedury, będący nazwą wyrażenia łańcuchowego,
- dowolne wyrażenie łańcuchowe ujęte w nawiasy okrągłe.

Wyrażenie łańcuchowe może być proste albo alternatywne. Alternatywne wyrażenie łańcuchowe ma - zgodnie z określeniem podanym w paragrafie 2.4 - następującą budowę:

warunek "jeśli"  
proste wyrażenie łańcuchowe  
else  
wyrażenie łańcuchowe

z tym, że oba wyrażenia: proste wyrażenie łańcuchowe występujące po warunku "jeśli" i wyrażenie łańcuchowe po else muszą mieć wartości w postaci tekstów albo obie wartości w postaci wzorców, ujętych w odpowiednie nawiasy.

Przykłady wyrażen łańcuchowych:

- 1)  $\{ \langle Q_{1.1} \rangle \}$
- 2)  $\text{if } a \geq 0 \text{ then } \{ \langle \text{wynik}_{1.1.1} \rangle \} \text{ else } w$
- 3)  $\{ \langle W_{1.1} \text{ tym zdaniu zrobiono}$   
powrot karetki  $\rangle \}$
- 4)  $\text{if } x < a \text{ then } \{ \langle -\text{ndd.dd} \rangle \} \text{ else } \{ \langle -\text{ndd.d00}_m + d \rangle \}$
- 5)  $\text{if } x > 0 \wedge x < 1 \text{ then } f1 \text{ else if } x > 5 \text{ then } \{ \langle +\text{dd.d00} \rangle \} \text{ else } f2$

W przykładzie pierwszym mamy do czynienia z tekstem o postaci

$Q =$

gdzie po obu stronach znaku równości zrobiono po jednym odstęp.

W przykładzie drugim wyrażenie jest alternatywne, a jego wartością jest tekst. W przypadku, gdy  $a \geq 0$ , wyrażenie to daje tekst

Wynik :

gdzie przed znakiem ":" jest jeden odstęp, a po znaku ":" dwa odstępy, natomiast w przypadku  $a < 0$  dane wyrażenie daje tekst, którego nazwą jest  $w$ . Nazwa ta jest w programie wykorzystywana jako parametr kategorii string dla jakiejś procedury.

W przykładzie trzecim mamy znowu tekst, który rozpoczyna się w aktualnym ustawieniu karetki maszyny do pisanie czy flexowritera, a po wyrazie "zrobiono" następuje powrót karetki i przejście do nowego wiersza, po czym tekst biegnie dalej.

W przykładzie czwartym mamy do czynienia z alternatywną postacią wzorca: jeżeli  $x < a$ , to całe wyrażenie jest równoznaczne ze wzorcem danym przez

$\{ \langle -\text{ndd.dd} \rangle \}$

w przeciwnym przypadku ze wzorcem

$\{ \langle -\text{ndd.d00}_m + d \rangle \}$

W przykładzie piątym mamy do czynienia z trzema możliwymi wzorcami:

gdy  $x > 0$  i  $x < 1$ :      wzorzec dany przez wyrażenie o nazwie f1,  
 gdy  $x > 5$ :      wzorzec dany przez wyrażenie  $\{+dd.d00\}$ ,  
 gdy  $x \leq 0$  albo  $1 \leq x \leq 5$ :      wzorzec dany przez wyrażenie o nazwie f2.

### 5.6. Zadanie

Napisać wyrażenie łańcuchowe dla przypadku, gdy mamy do czynienia z następującymi trzema wariantami tekstu, w zależności od znaku pewnej zmiennej z:

dla  $z < 0$ :

wyraz "Funkcja" oddalony od początku wiersza o 15 odstępów,

dla  $z = 0$ :

trzy kolejne zera oddalone od siebie o 2 odstępy, z tym że pierwsze zero jest drukowane w aktualnym ustawieniu karetki,

dla  $z > 0$ :

najpierw 13 odstępów licząc od początku wiersza, potem tekst XYZ, potem 6 odstępów, potem YZX, potem 6 odstępów, potem ZXY, potem powrót karetki, 14 odstępów i wreszcie cyfra 5.

### 5.7. Urządzenia zewnętrzne maszyny GIER

Standardowe wyposażenie maszyny GIER składa się z:

- czytnika dziurkowanej (czyli perforowanej) taśmy papierowej,
- maszyny do pisania,
- perforatora taśmy papierowej,
- bębna magnetycznego,
- bufora.

Wszystkie wymienione urządzenia są podłączone bezpośrednio do maszyny cyfrowej. Niektóre z maszyn GIER mają jeszcze tzw. karuzelę, podłączoną bezpośrednio do bufora.

Ponadto do wyposażenia maszyny GIER należy zazwyczaj kilka flexowriterów (czytaj: fleksorajterów), pracujących niezależnie od maszyny, najczęściej w innych nawet pomieszczeniach. Omówimy kolejno wymienione urządzenia.

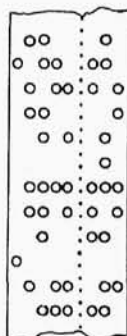
Czytnik jest urządzeniem, które "czyta" dziurkowaną taśmę papierową. Czytanie to polega na przetwarzaniu układu dziurek na układ sygnałów elektrycznych przesyłanych do maszyny cyfrowej.

Taśma papierowa zawiera układy dziurek zwane kodami, w rzędkach poprzecznych względem taśmy, z tym że jeden rząd odpowiada jednemu jakiemuś symbolowi (na przykład literze lub cyfrze). Dziurki na taśmie układają się również w linie podłużne względem taśmy, zwane kanałami. Stosu-

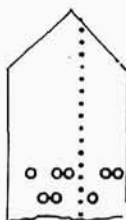
je się taśmy 5,6,7 lub 8-kanalowe. Dalekopisy pracujące w Polsce w telekomunikacji są przystosowane wyłącznie do taśmy 5-kanalowej. Maszyny typu GIER pracują z reguły z taśmą 8-kanalową i do takiej się tylko ograniczymy.

Środkiem taśmy papierowej biegnie linia małych dziurek, służących do przesuwania taśmy a nie zaliczanych do kodów. Ta linia dziurek prowadzących nie leży dokładnie w środku taśmy lecz nieco z boku, aby przez umowę, na przykład że linia dziurek prowadzących ma leżeć po prawej stronie taśmy, uniknąć odwracania taśmy na drugą stronę, co mogłoby wprowadzić dwuznaczność odczytywania kodów.

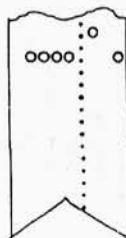
Oto jak wygląda odcinek 8-kanalowej taśmy papierowej:



Aby określić kierunek przesuwania taśmy, należy zawsze oznaczyć jej początek i koniec. Jednym ze sposobów często używanych jest obcinanie początku taśmy w kształt



a końca w kształt



Maszy na do p i s a n i a podłączona bezpośrednio do maszyny cyfrowej służy do:

- przesyłania maszynie cyfrowej sygnałów przez operatora,
- automatycznego druku sygnałów przez maszynę cyfrową,

- druku wyników obliczeń lub tekstów według programu wykonywanego przez maszynę cyfrową,
- wprowadzania danych do maszyny.

Operator przesyła maszynie cyfrowej sygnały przyciskając odpowiednie klawisze na klawiaturze maszyny do pisania, która nieznacznie tylko różni się od klawiatur zwykłych maszyn do pisania. Maszyna cyfrowa uruchamia maszynę do pisania sygnałami elektrycznymi. Należy unikać masowego druku wyników obliczeń na maszynie do pisania jak również wprowadzania danych poprzez maszynę do pisania, gdyż są to czynności czasochłonne, okupujące kosztowny czas pracy maszyny cyfrowej. Normalnie do wprowadzania danych powinien służyć czytnik, czytający z taśmy papierowej zarówno programy jak i dane liczbowe, a do wyprowadzania wyników perforator dziurkujący wyniki na taśmie papierowej, którą później - już niezależnie od maszyny cyfrowej - można odczytywać na flexowriterach i wyniki drukować na papierze.

Maszynę do pisania wykorzystuje się jednak często do druku wyników pośrednich, kontrolnych, zwłaszcza gdy od tych wyników zależy dalsze sterowanie programem.

**P e r f o r a t o r** jest urządzeniem dziurkującym taśmę papierową według kodów przesyłanych sygnałami elektrycznymi z maszyny cyfrowej. Pracuje kilkadziesiąt razy szybciej od maszyny do pisania (150 i więcej rządków na sekundę, gdy maszyna do pisania 8-10 symboli na sekundę) i dlatego jest normalnym urządzeniem dla wyprowadzenia wyników obliczeń z maszyny cyfrowej. Późniejsze przedrukowywanie taśmy na flexowriterach nie jest już tak kosztowne jak druk na maszynie do pisania, ponieważ czas pracy flexowritera kosztuje znikomo mało w porównaniu do czasu pracy całej maszyny cyfrowej. Poza tym stosuje się równoległy przedruk kilku taśm papierowych na kilku flexowriterach, co przyspiesza otrzymanie wyników w formie druku na papierze.

Aby maszyna cyfrowa mogła wykonywać programy obliczeniowe, musi mieć możliwość "pamiętania" liczb czy instrukcji (które w maszynie również mają postać liczb). Odpowiednią część maszyny cyfrowej nazywana jest **p a m i ę c i ą** maszyny. Maszyna GIER ma kilka rodzajów pamięci. Przede wszystkim ma bardzo szybką tzw. **p a m i ę ć o p e r a c y j n ą** (ferrytową) o pojemności około 1000 słów (tzn. liczb). Ponadto ma tzw. **b ę b ę n m a g n e t y c z n y**, który będziemy nazywali krótko bębniem, o pojemności 12 800 słów, z których zazwyczaj około 5 800 jest zarezerwowanych dla przechowywania translatora i programu przetłumaczonego na język wewnętrzny maszyny. Wolnych dla programującego jest około 7000 miejsc na bębnie. O ile współpraca z bębniem w zakresie translatora jest całkowicie zautomatyzowana i programujący może się nią w ogóle nie interesować, o tyle współpraca z bębniem w zakresie miejsc swobodnych nie jest całkowicie automatyczna i odbywa się za pośrednictwem procedur standardowych to drum i from drum oraz zmiennej standardowej drumplace, o czym będzie mowa w jednym z późniejszych paragrafów. Z punktu widzenia operatora bęben magnetyczny stanowi jakby dodatkowe urządzenie wejściowo-wyjściowe.

Programy w zakresie dotychczas omówionym, pozwalają - ze względu na ograniczoność pamięci operacyjnej - na operowanie równoczesne około 600-700 zmiennymi. Nie jest to dużo, gdy mamy do czynienia z rachunkami macierzowymi. Gdy musimy użyć

większej liczby zmiennych korzystamy właśnie z pamięci dodatkowych: bębna, buforu czy karuzeli.

B u f o r może być używany jako osobny rodzaj pamięci albo jako pamięć przejściowa dla pamięci karuzelowej. Pamięć buforowa jest pamięcią ferrytową, tak jak pamięć operacyjna, i znacznie, bo trzydzieści razy szybszą od pamięci bębnowej. Szybkość pobierania liczb z pamięci buforowej lub przesyłania na bufor jest około 60 000 liczb na sekundę wobec 2 000 liczb na sekundę w przypadku pamięci bębnowej. Wobec powyższego, gdy bufor nie jest zajęty jako pamięć przejściowa do karuzeli, zaleca się korzystanie z buforu przed korzystaniem z bębna. Współpraca z buforem odbywa się za pośrednictwem procedur standardowych to buf i from buf oraz zmiennej standardowej drumplace, o czym będzie mowa w jednym z paragrafów późniejszych. Pamięć buforowa ma pojemność 4096 liczb.

Oprócz pamięci operacyjnej, bębna magnetycznego i pamięci buforowej niektóre z maszyn GIER posiadają jeszcze pamięć na taśmach magnetycznych, tzw. k a r u z e l e. Karuzela posiada na swym obwodzie 64 szpule taśmy magnetycznej. Każda szpula zawiera 16 bloków o pojemności 512 liczb każdy. Wynika stąd, że pojemność całej pamięci karuzelowej jest 524 288 liczb.

Pamięć karuzelowa ma bezpośrednią komunikację tylko z pamięcią buforową za pośrednictwem procedur standardowych to car i from car, o czym będzie mowa w jednym z paragrafów późniejszych.

Czas przepisywania z karuzeli na bufor lub odwrotnie jest długi ze względu na konieczność mechanicznego obrotu całej karuzeli i mechanicznego przewijania taśmy magnetycznej ze szpuli. Czas ten wynosi około  $1+0.1 \times n$  sekund, gdzie  $n$  jest liczbą przepisywanych bloków, po czym tyleż mniej więcej czasu potrzeba na powrotne przewinięcie taśmy magnetycznej, z tym, że ta ostatnia część pracy karuzeli odbywa się już równolegle z dalszą pracą maszyny cyfrowej. Należy o tym pamiętać, aby w programie - o ile to możliwe - wykorzystywać czas powrotnego przewijania taśmy magnetycznej na obliczenia, tzn. nie dawać instrukcji przepisywania zbyt gęsto.

F l e x o w r i t e r jest urządzeniem bardzo podobnym do elektrycznej maszyny do pisania czy dalekopisu. Jego klawiatura niewiele różni się od klawiatury zwykłej maszyny do pisania. Można wykonywać na nim następujące czynności:

- pisać na papierze jak na maszynie do pisania,
- równolegle z pisaniem na papierze perforować taśmę papierową, dziurkując w niej kody wypisywanych symboli,
- pisać automatycznie na papierze treść czytaną równolegle również automatycznie z perforowanej taśmy papierowej,
- reperforować taśmę papierową, tzn. sporządzać jej kopię poprzez równoległe czytanie jednej taśmy i perforowanie drugiej.

Na flexowriterze przygotowuje się program dla maszyny cyfrowej przepisując go z rękopisu i kodując na taśmie papierowej. Analogicznie przygotowuje się taśmy papierowe z wydrukowanymi danymi liczbowymi. Na flexowriterze poprawia się również błędy na taśmie papierowej poprzez automatyczną reperforację poprawnych części taśmy i perforację miejsc poprawianych drogą wypisywania ich od nowa na klawiaturze.

Zarówno flexowriter, jak i maszyna do pisania mają urządzenie zwane tabulatorem. Najpierw ręcznie ustawia się tzw.



koniki tabulatora na wyróżnionych miejscach wiersza. Gdy później przyciśniemy klawisz z napisem TAB, karetką zrobi tyle odstępów, aż punkt druku znajdzie się w najbliższym wyróżnionym miejscu w prawo arkusza papieru. Gdy flexowriter drukuje automatycznie, przeskok karetki następuje, gdy flexowriter czytając perforowaną taśmę papierową napotyka kod symbolu TAB. Gdy maszyna do pisania drukuje automatycznie, przeskok karetki następuje, gdy maszyna cyfrowa prześle maszynie do pisania układ sygnałów elektrycznych odpowiadający kodowi symbolu TAB. Tabulator jest używany głównie do druku tablic liczbowych, w których liczby są wypisywane w z góry określonych kolumnach. Wtedy pojawienie się symbolu TAB powoduje przeskok karetki umożliwiając druk następnej liczby w odpowiedniej kolumnie.

Na zakończenie niniejszego paragrafu warto zwrócić uwagę, że taśma papierowa stanowi również pewien rodzaj pamięci dla maszyny cyfrowej. Treść bowiem, którą chcemy zapamiętać, można wyperforować na taśmie, a w żądanym momencie wczytać z powrotem do maszyny cyfrowej. Jest to jednak pamięć bardzo powolna i niepewna, gdyż urządzenia mechaniczne takie jak perforator, czy czytnik, powodują bez porównania więcej przekłamań niż urządzenia elektroniczne.

### 5.8. Drukowane symbole i ich kody

Zarówno maszyna do pisania, jak flexowriter czy dalekopis drukują znaki poprzez uderzenie odpowiedniego młoteczka o papier. Na każdym młoteczku są umieszczone 2 czcionki: wyższa i niższa. To, czy młoteczek uderza o papier wyższą czy niższą czcionką, zależy od ustawienia urządzenia drukującego. Ustawienia tego dokonuje się poprzez przyciśnięcie klawisza LOWER CASE (niższy przypadek; czytaj: loer kejs) lub UPPER CASE (wyższy przypadek; czytaj: apper kejs). Po przyciśnięciu klawisza LOWER CASE, urządzenie będzie drukowało znaki dolne aż do chwili, gdy zostanie przyciśnięty klawisz UPPER CASE. Wtedy urządzenie zacznie drukować znaki górne aż nie zostanie przyciśnięty klawisz LOWER CASE itd. Tak jest w przypadku, gdy druk następuje na skutek pisania ręcznego na klawiaturze. Ponieważ flexowritery i dalekopisy są tak skonstruowane, że przy pisaniu ręcznym na klawiaturze oprócz druku na papierze można otrzymać równolegle na taśmie papierowej kolejne kody wypisywanych znaków, tzn. układy dziurek charakterystyczne dla danego znaku, przez przyciśnięcie klawisza LOWER CASE czy UPPER CASE można również otrzymać wyperforowanie rzędka dziurek dla każdego z nich charakterystycznego, tzn. wyperforowanie ich kodów. W przypadku druku automatycznego ustawianie urządzenia na drukowanie niższymi czy wyższymi czcionkami dokonuje się poprzez sygnalizowanie kodu LOWER CASE lub UPPER CASE np. przez pojawienie się charakterystycznego rzędka dziurek na taśmie papierowej w czasie przedrukowywania jej przez flexowriter czy dalekopis.

Same symbole LOWER CASE i UPPER CASE należą do grupy symboli typograficznych i kontrolnych, które są jednakowo interpretowane w przypadku LOWER CASE, jak i UPPER CASE. Do tej grupy należą jeszcze symbole:

SPACE (czytaj: spejs) czyli odstęp,  
 CAR RET (czytaj: ker rit), czyli powrót karetki z równoczesnym przejściem do nowego wiersza,  
 STOP CODE (czytaj: stop koud) używany w procedurze wyjściowej outsum,  
 PUNCH OFF (czytaj: pancz of), czyli symbol powodujący wyłączenie urządzenia perforującego taśmę,  
 PUNCH ON (czytaj: pancz on), czyli symbol powodujący włączenie urządzenia perforującego taśmę,  
 END CODE (czytaj: end koud), czyli symbol powodujący zatrzymanie maszyny i wydrukowanie na maszynie do pisania wyrazu "pause", po czym uruchomienie maszyny następuje przez napisanie jakiegokolwiek znaku na maszynie do pisania,  
 CLEAR CODE (czytaj: kliar koud) używany w procedurze wyjściowej outclear,  
 SUM CODE (czytaj: sam koud) używany w procedurze wyjściowej outsum,  
 TAPE FEED (czytaj: tejp fid) powodujący jedynie posuw taśmy papierowej na czytniku o 1 rządę dalej,  
 TAB (czytaj: teb), czyli symbol uruchamiający tabulator, urządzenie analogiczne do znanego w maszynach do pisania, a służące do bezpośredniego przeskoku karetki do najbliższej wyróżnionej kolumny,  
 RED RIBBON (czytaj: red ribn) symbol używany tylko dla maszyny do pisania, powodujący jej nastawienie na druk czerwoną wstążką,  
 BLACK RIBBON (czytaj: blak ribn) symbol używany tylko dla maszyny do pisania, powodujący jej nastawienie na druk czarną wstążką.

A oto lista symboli i ich kodów:

LOWER CASE	UPPER CASE	Kod	LOWER CASE	UPPER CASE	Kod
a	A	00 0	l	L	0 00
b	B	00 0	m	M	0 0 0
c	C	000 00	n	N	0 0 0
d	D	00 0	o	O	0 00
e	E	000 0 0	p	P	0 0 000
f	F	000 00	q	Q	0 00
g	G	00 000	r	R	0 0 0
h	H	00 0	s	S	00 0
i	I	0000 0	t	T	0 00
j	J	0 0 0	u	U	00 0
k	K	0 0 0	v	V	0 0 0

w	W	o .oo	5	;	o o o
x	X	oo .ooo	6	[	o .oo
y	Y	ooo	7	]	.ooo
z	Z	o o o	8	(	o
æ	Æ	ooo	9	)	oo o
ø	Ø	o oo oo	,	»	ooo oo
0	Λ	o	.	:	oo o oo
1	V		-	+	o
2	x		<	>	oo o
3	/	o oo	-		o oo
4	=	o	ã	Ä	o o o

Symbole    i    nie powodują przesuwu karetki. Litery    i    występują tylko na maszynie do pisania.

A oto symbole typograficzne i kontrolne:

LOWER CASE	oooo. o
UPPER CASE	oooo.o
SPACE	o
CAR RET	o
STOP CODE	o .oo
PUNCH OFF	o o.ooo
PUNCH ON	o o.o
END CODE	oo.o
CLEAR CODE	o o.o
SUM CODE	oo o.o o
TAPE FEED	oooo.ooo
TAB	ooo.oo
RED RIBBON	ooo.o o
BLACK RIBBON	oo o.oo

Z wymienionych symboli następujące będziemy nazywać symbolami niewłaściwymi: LOWER CASE, UPPER CASE, PUNCH OFF, PUNCH ON, ale tylko w przypadku, gdy jest na-

wiasem zamykającym w stosunku do poprzednio otwartego przez PUNCH OFF, END CODE, CLEAR CODE, TAPE FEED, SUM CODE. Pozostałe symbole będziemy nazywać symbolami właściwymi.

Ponieważ symbole  $\_ 1$  nie powodują przesuwu karetki, można z ich pomocą tworzyć następujące symbole złożone:

$\uparrow \_ \pm \leftarrow \rightarrow \vdots \leq \geq$

Na flexowriterze z podanych poprzednio symboli nie ma:

$\grave{a}$  i  $\grave{A}$ , END CODE, CLEAR CODE, SUM CODE,  
RED RIBBON, BLACK RIBBON,

natomiast klawiatura jest zaopatrzona jeszcze w cztery klawisze:

START READ, STOP READ, PUNCH ADRES, AUX CODE.

(Czytaj: start rid, stop rid, pancz adres, oks koud).

Przyciśnięcie klawisza START READ uruchamia czytanie taśmy papierowej, przyciśnięcie STOP READ zatrzymuje czytanie. Przyciśnięcie klawiszy PUNCH ADRES lub AUX CODE jest możliwe tylko równocześnie z jakimś innym klawiszem, co powoduje wydziurkowanie dodatkowych dziurek:

$\begin{array}{c} \circ \\ \vdots \\ \circ \end{array}$	przez PUNCH ADRES,
$\begin{array}{c} \circ \\ \circ \circ \\ \circ \end{array}$	przez AUX CODE

Kod END CODE brakujący na flexowriterze można uzyskać na taśmie papierowej przyciskając równocześnie klawisze AUX CODE i SPACE. Podobnie kod CLEAR CODE można uzyskać przyciskając równocześnie AUX CODE i O, kod SUM CODE przyciskając AUX CODE i A (albo a), kod  $\grave{A}$  (albo  $\grave{a}$ ) przyciskając AUX CODE i 1 (albo  $\vee$ ). Należy jednak wtedy pamiętać, że tak złożone kody będą jedynie wyperforowane na taśmie, a nie będą miały odrębnych znaków w równoległym druku na papierze.

Każdemu z wymienionych symboli jest przyporządkowana pewna liczba. Fakt ten jest wykorzystany w niektórych procedurach standardowych, o czym będzie mowa później. Liczbę przyporządkowaną danemu symbolowi można odczytać z jego kodu wyperforowanego na taśmie papierowej w następujący sposób:

- ujmujemy taśmę w ten sposób, by linia małych dziurek służących do przesuwania taśmy była położona na prawo od środka taśmy,

- przyjmujemy numerację kanałów na taśmie od 1 do 8 od prawej strony ku lewej (w ten sposób kanały 1, 2, 3 są położone na prawo od linii dziurek prowadzących, a kanały 4, 5, 6, 7, 8 na lewo),

- przyporządkowujemy poszczególnym kanałom liczby w następujący sposób: kanałowi 1 liczbę 1, kanałowi 2 liczbę 2, kanałowi 3 liczbę 4, kanałowi 4 liczbę 8, kanałowi 5 liczbę 0, kanałowi 6 liczbę 16, kanałowi 7 liczbę 32 i kanałowi 8 liczbę 64,

- liczbę przyporządkowaną danemu rządowi dziurek (tzn. danemu przez ten rząd symbolowi) otrzymujemy, sumując liczby przyporządkowane tym kanałom, w których w danym rzędzie występują dziurki.

Na przykład kropka na przyporządkowany kod

$$\begin{vmatrix} & \cdot & \\ \infty & 0 & \cdot & \infty \\ & \cdot & \end{vmatrix}$$

Dziurki występują tutaj w kanałach 1, 2, 4, 6, 7. Sumujemy

$$1+2+8+16+32 = 59.$$

Zatem kropka "." odpowiada liczba 59.

Należy zauważyć, że kanałowi 5 została przyporządkowana liczba 0. Wynika to stąd, że kanał ten jest wykorzystywany jedynie w celach kontrolnych; umieszczamy w tym kanale dziurkę lub nie w taki sposób, by liczba dziurek w rzędzie była zawsze nieparzysta, gdyż ułatwia to kontrolę.

Aby nie trzeba było za każdym razem obliczać liczby przyporządkowanej danemu symbolowi, podajemy tabelkę liczb, odpowiadających im kodów i odpowiadających im symboli, z tym, że nie wszystkim podanym liczbom i kodom odpowiadają symbole. Jak widać z poniższej tabelki, liczbom 10, 15, 26, 42, 45, 46, 47 oraz liczbom od 65 do 127 włącznie nie odpowiada żaden z używanych symboli, chociaż liczby te mają swoje kody.

Kody odpowiadające symbolom niewłaściwym oraz kod odpowiadający liczbie 127 będziemy nazywać kodami nie właściwymi. Wszystkie inne kody - niezależnie od tego, czy odpowiadają jakimś symbolom czy nie - będziemy nazywać kodami właściwymi.

L i c z b a	K o d	S y m b o l	
		Lower Case	Upper Case
0	o . . . . .	SPACE	
1	. . . . . o	1	V
2	. . . . . o	2	x
3	o . . . . . oo	3	/
4	. . . . . o	4	=
5	o . o . o	5	;
6	o . . . . . oo	6	[
7	. . . . . .ooo	7	]
8	o . . . . .	8	(
9	oo . . . . . o	9	)
10	oo . . . . . o		
11	. . . . . oo		
12	oo . . . . . o	STOP CODE	
		END CODE	

13	0 0 0
14	0 0 0
15	0 0 0 0 0
16	0 0 0 0 0
17	0 0 0 0 0
18	0 0 0 0 0
19	0 0 0 0 0
20	0 0 0 0 0
21	0 0 0 0 0
22	0 0 0 0 0
23	0 0 0 0 0
24	0 0 0 0 0
25	0 0 0 0 0
26	0 0 0 0 0
27	0 0 0 0 0
28	0 0 0 0 0
29	0 0 0 0 0
30	0 0 0 0 0
31	0 0 0 0 0
32	0 0 0 0 0
33	0 0 0 0 0
34	0 0 0 0 0
35	0 0 0 0 0
36	0 0 0 0 0
37	0 0 0 0 0
38	0 0 0 0 0
39	0 0 0 0 0
40	0 0 0 0 0
41	0 0 0 0 0
42	0 0 0 0 0
43	0 0 0 0 0
44	0 0 0 0 0
45	0 0 0 0 0
46	0 0 0 0 0
47	0 0 0 0 0
48	0 0 0 0 0
49	0 0 0 0 0
50	0 0 0 0 0
51	0 0 0 0 0

ä	Ä
-	
0	^
<	>
s	S
t	T
u	U
v	V
w	W
x	X
y	Y
z	Z

, 0  
CLEAR CODE

RED RIBBON

TAB

PUNCH OFF

-	+
j	J
k	K
l	L
m	M
n	N
o	O
p	P
q	Q
r	R

ø Ø  
PUNCH ON

m	E
a	A
b	B
c	C

52	00	:	0
53	000	:	0 0
54	000	:	00
55	00	:	000
56	00	:	0
57	0000	:	0
58	0000	:	0
59	00	:	0 00
60	0000	:	0
61	00	:	0 0 0
62	00	:	0 00
63	0000	:	000
64	0	:	
65	0	:	0
66	0	:	0
67	0	:	00
68	0	:	0
69	0	:	0 0
70	0	:	00
71	0	:	0 000
72	0	:	00
73	0	:	0
74	0	:	0
75	0	:	00 00
76	0	:	0 0
77	0	:	00 0 0
78	0	:	00 00
79	0	:	0 000
80	0	:	00
81	0	:	0
82	0	:	0
83	0	:	00 00
84	0	:	0
85	0	:	00 0 0
86	0	:	00 00
87	0	:	0 000
88	0	:	0 0
89	0	:	000 0
90	0	:	000 0
91	0	:	0 00

d	D
e	E
f	F
g	G
h	H
i	I
LOWER CASE	
.	:
UPPER CASE	
SUM CODE	
BLACK RIBBON	
TAPE FEED	
CAR	RET



92	0 000:0
93	0 0 0:0 0
94	0 0 0:00
95	0 000:000
96	00 0
97	00 0
98	00 0
99	00 0 00
100	00 0
101	00 0 0 0
102	00 0 00
103	00 000
104	00 0
105	00 00 0
106	00 00 0
107	00 0 00
108	00 00 0
109	00 0 0 0
110	00 0 00
111	00 00 000
112	000
113	0000 0
114	0000 0
115	000 00
116	0000 0
117	000 0 0
118	000 00
119	0000 000
120	00000
121	000 0 0
122	000 0 0
123	00000 00
124	000 0 0
125	00000 0 0
126	00000 00
127	000 0 000

### 5.9. Procedury wyjścia

Procedury wyjścia dzielą się na 2 grupy: procedury o nazwach rozpoczynających się od "out" i procedury o nazwach rozpoczynających się od "write". Pierwsze z nich powodują perforowanie wyników na 8-kanałowej taśmie papierowej, drugie druk wyników na maszynie do pisania podłączonej bezpośrednio do maszyny cyfrowej. Istnieje możliwość interwencji operatora obsługującego maszynę, powodującej zmiany urządzeń wyjściowych przyporządkowanych procedurom out- i write-. Możliwość ta powstaje, gdy maszyna cyfrowa wydrukuje słowo "run" i zatrzyma się. Jeśli wtedy przyciśniemy jakikolwiek klawisz różny od b, p, w, e, urządzenia wyjściowe będą przyporządkowane normalnie: maszyna do pisania dla procedur write- i perforator dla procedur out-. Jeśli jednak przyciśniemy b, to oba wymienione urządzenia będą działać zarówno dla procedur write- jak też out-. Jeśli przyciśniemy w, zarówno procedury write- jak out- będą uruchamiały tylko maszynę do pisania. Wreszcie przyciśnięcie p powoduje przyporządkowanie procedurom write- i out- tylko perforatora.

W poniższym opisie procedur wyjścia zakładamy normalne przyporządkowanie urządzeń wyjściowych: perforatora dla procedur out- i maszyny do pisania dla procedur write-.

Dla wszystkich procedur wyjścia wywołanie ma postać odrębnej instrukcji. Wszystkie instrukcje będące wywołaniami procedur wyjścia będziemy nazywali instrukcjami wyjścia.

Każda z procedur wyjścia powoduje dodawanie do pewnej zmiennej, przechowywanej w pamięci maszyny i zwanej sumą kontrolną, liczb równoważnych znakom perforowanym czy drukowanym (patrz paragraf poprzedni). Suma kontrolna służy do kontrolowania prawidłowości wyprowadzania wyników i ułatwia lokalizację ewentualnych błędów perforacji. Kontrola tego typu jest możliwa tylko w przypadku, gdy wyniki wyperforowane na taśmie papierowej przez procedury wyjścia, są później wczytywane z powrotem do maszyny cyfrowej. Sposób tej kontroli będzie omówiony niżej. Należy zapamiętać, że kontrola powyższa jest poprawna tylko wtedy, gdy kontrolowany odcinek taśmy papierowej nie zawiera żadnego znaku wytworzonego przez procedurę typu write.

### 5.10. Procedura output

Wywołanie procedury output (czytaj: autput) ma postać następującej instrukcji:

output(lista parametrów)

z tym, że

- pierwszym parametrem musi być wyrażenie łańcuchowe, którego wartością jest wzorzec ujęty w nawiasy  $\langle i \rangle$ ,
- pozostałymi parametrami mogą być dowolne wyrażenia arytmetyczne lub dowolne instrukcje wyjścia,
- parametry są oddzielane przecinkami albo ciągiem znaków o następującej budowie:

)nazwa złożona z samych liter:(

stanowiącym rodzaj komentarza (tak jak to było w deklaracjach procedur).

Działanie procedury output jest następujące:

- zostaje obliczona wartość wyrażenia łańcuchowego stanowiącego pierwszy parametr, co w wyniku daje stały wzorzec dla danej procedury,
- parametry wymienione w liście parametrów są rozpatrywane kolejno: jeśli parametrem jest wyrażenie arytmetyczne, jego wartość zostaje obliczona, a następnie wyperforowana na taśmie perforowanej na perforatorze podłączonym bezpośrednio do maszyny, według wzorca danej procedury uprzednio obliczonego jako pierwszy parametr; jeśli parametrem jest procedura wyjścia, zostaje ona wykonana, a następnie przechodzi się do następnego parametru z listy, aż do jej wyczerpania.

Należy zapamiętać, że procedura output zakłada wejście do niej w sytuacji LOWER CASE i w tejże sytuacji następuje zakończenie jej wykonania.

Oto przykłady wywołania procedury output:

- 1) output( $\langle \text{-d.ddd} \rangle$ , x, outsp(3), y)
- 2) output( $\langle \text{+n.dd}_m \text{+d} \rangle$ , sqrt( $\langle \text{a} \rangle^2 + 1$ ), outtext( $\langle \text{<}_1 \text{Z=}_1 \rangle$ ),  $\langle \text{a} \rangle^3 - \langle \text{a} \rangle + 1$ )
- 3) output(if c > d then  $\langle \text{dd.ddd} \rangle$  else  $\langle \text{.ddd} \rangle$ , c $\langle \text{a} \rangle^2 + \langle \text{d} \rangle^2$ )
- 4) output(a,b)
- 5) output(if A then B else C, if D then E else F)
- 6) output( $\langle \text{+d.ddd}_m \text{+d} \rangle$ , X+Y, outsp(4), output( $\langle \text{-ddd} \rangle$ , m), outer, u-sin(v))

W p r z y k ł a d z i e 1 wyrażenie łańcuchowe jest zredukowane do konkretnego wzorca. Według tego wzorca ma być wyperforowana na perforatorze aktualna wartość zmiennej x, potem zostanie wykonana instrukcja outsp(3) i wreszcie całość kończy się wyperforowaniem wartości zmiennej y według tego samego wzorca, co poprzednio x.

P r z y k ł a d 2 tym różni się od poprzedniego, że mają być wyperforowane wartości wyrażen, które najpierw muszą być obliczone. I tak na skutek wywołania danej procedury output zostanie obliczona wartość wyrażenia sqrt( $\langle \text{a} \rangle^2 + 1$ ) dla aktualnej wartości zmiennej a i po obliczeniu wyperforowana według podanego na pierwszym miejscu wzorca. Potem zostanie wykonana procedura outtext ( $\langle \text{<}_1 \text{Z=}_1 \rangle$ ) i całość kończy się

obliczeniem wartości wyrażenia  $a_{3-a+1}$  i wyperforowaniem obliczonej wartości według tegoż co poprzednio wzorca.

W p r z y k ł a d z i e 3 mamy do czynienia z alternatywnym wyrażeniem łańcuchowym. W zależności od tego, czy aktualna wartość zmiennej  $c$  jest większa od aktualnej wartości zmiennej  $d$  czy nie, wartość wyrażenia  $c_{2+d}2$  po jej obliczeniu zostaje wyperforowana według wzorca  $\langle dd.ddd \rangle$  lub  $\langle dddd \rangle$ .

W p r z y k ł a d z i e 4 występują dwie zmienne:  $a$  i  $b$ . Zgodnie z podanymi poprzednio regułami widzimy, że  $a$  musi być parametrem formalnym jakiejś procedury, której ciało zawiera daną instrukcję output. Parametr ten musi być kategorii string i w momencie wywołania procedury na jego miejsce musi być podstawiony parametr aktualny w postaci wyrażenia łańcuchowego, pozwalającego określić konkretny wzorzec jako wartość tego parametru. Zmienna  $b$  występująca w danej instrukcji output jest oczywiście zmienną liczbową, której wartość będzie wyperforowana według wzorca, jaki otrzymamy po podstawieniu na parametr  $a$  parametru aktualnego.

P r z y k ł a d 5 jest podobny do poprzedniego, z tym że występuje tu alternatywa zarówno dla parametru formalnego kategorii string, jak i dla zmiennej liczbowej. Zgodnie z regułami  $A$  i  $D$  muszą być zmiennymi booleowskimi,  $B$  i  $C$  parametrami formalnymi kategorii string, a  $E$  i  $F$  zmiennymi liczbowymi. Gdy obie zmienne  $A$  i  $D$  mają wartość true, cała instrukcja jest równoznaczna z instrukcją output  $(B,E)$ , a więc z instrukcją w postaci rozpatrzonej w poprzednim przykładzie. Gdy  $A$  ma wartość true, a  $D$  false, cała instrukcja jest równoznaczna z instrukcją output  $(B,F)$ . Gdy  $A$  ma wartość false, a  $D$  true, cała instrukcja jest równoznaczna z output  $(C,E)$ . Wreszcie gdy obie zmienne  $A$  i  $D$  mają wartość false, cała instrukcja jest równoznaczna z output  $(C,F)$ .

P r z y k ł a d 6 jest podobny do przykładu 2. Pewne wątpliwości może budzić fakt, że jednym z parametrów danej instrukcji output jest inna instrukcja output i powstaje konkurencja wzorców. Otóż wzorzec  $\langle -ddd \rangle$  podany w wewnętrznej instrukcji output obowiązuje tylko w granicach tej instrukcji, w danym razie tylko dla perforacji wartości zmiennej  $m$ . Na zewnątrz tej instrukcji obowiązuje wzorzec instrukcji zewnętrznej, tj.  $\langle +d.ddd_m+d \rangle$ , zarówno dla perforacji wartości wyrażenia  $X+Y$ , jak i  $u-\sin(v)$ .

Na zakończenie niniejszego paragrafu zwróćmy jeszcze uwagę na fakt, że nie można drukować ani perforować bezpośrednio po sobie dwu liczb, gdyż spowodowałoby to zlanie się ich w jedną całość. Między dwiema drukowanymi czy perforowanymi liczbami muszą być umieszczone jakieś odstępy, przecinki czy teksty rozdzielające. Do tego będą służyły niektóre z dalszych procedur.

### 5.11. Zadanie

Rozwiązać zadanie 3.12-a, traktując zmienną  $X$  jako lokalną i zmieniając podany poprzednio program wykorzystując in-