

ODPOWIEDZI ZADAŃ

Zadanie 1,5

Dozwolone: 1,2,5,9,10,12,14,16,17,18,19,20.

Niedozwolone: 3 po kropce nie ma cyfry,
 4 cecha jest zmienną,
 6 użyto znaku \pm , który nie jest dopuszczalny,
 7 użyto przecinka zamiast kropki,
 8 po kropce nie ma cyfry a jest dziesiątka
 algolowska,
 11 cechę ujęto w nawiasy,
 13 cecha nie jest całkowita,
 15 użyto zapisu potęgi z tradycyjnego języka
 matematycznego, powinno być $-_{10}2$,
 21 użyto nawiasów,
 22 użyto znaku \times ,
 23 użyto dwu kropek,
 24 po dziesiątce algolowskiej nie następuje ce-
 cha.

Zadanie 1,6

1)	$2.38571896_{10}8$	$_{10}-2$	$_{10}0$
	$1.5833_{10}4$	$-5.3_{10}-5$	$_{10}-1$
	$-3.21_{10}0$	$+6.75_{10}-10$	$-1.11111_{10}-1$
2)	274	100000	0
	-38900000	-.0000088	18
	.001	.00000765	0
			-0.01

Zadanie 1,9

Tworzą nazwy zmiennych: 1,3,9,13,15,17,19.

Nie tworzą nazw zmiennych:

- 2 użyto znaku /,
- 4 nazwa char została zastrzeżona dla jednej z procedur standardowych,
- 5 znak \vee nie jest dopuszczalny w nazwach,
- 6 użyto kropki,
- 7 użyto znaku minus,
- 8 użyto kreski,
- 10 pierwszym symbolem jest cyfra a nie litera,

11 użyto znaku podkreślenia,
 12 nazwa zarezerwowana dla funkcji standardowej,
 14 użyto polskiej litery ł,
 16 użyto obniżonego indeksu, co w ALGOLu nie jest dozwolone,
 18 pierwszym symbolem jest cyfra a nie litera,
 20 użyto znaku :

Zadanie 1.13

Tworzą nazwy zmiennych indeksowanych: 1,2,4,5,7,9.

Nie tworzą nazw zmiennych indeksowanych:

- 3 użyto nawiasów okrągłych zamiast kwadratowych,
- 6 brak jednego nawiasu zamykającego,
- 8 Δ jest raz nazwą tablicy jednowymiarowej a raz dwuwymiarowej.

Uwaga: nazwa 4) jest poprawna, gdyż 1-2 stanowi wyrażenie arytmetyczne o wartości -1.

Zadanie 1.14

- | | |
|---------------------------|---------------------------------|
| 1) $a[1,m+1]$ | 6) $f1[0,k]-f1[1,k-1]$ |
| 2) $\text{beta}[1-1,1+1]$ | 7) Twewm |
| 3) $x[1,1]$ | 8) $z[\text{alfa},\text{beta}]$ |
| 4) $v[k1,k2]$ | 9) $U[n[k+1]-1,0,m[1+2]]$ |
| 5) $A[p,q,r,s]$ | |

Zadanie 1.16

- 1) $\text{sqrt}(2)$
- 2) $\text{sqrt}(1+\text{sqrt}(3.325))$
- 3) $\sin(\text{alfa})$
- 4) $\sin(\text{alfa}+\text{beta})$
- 5) $\arctan(z)$
- 6) $\exp(x+2)$

- 7) $\sin(\sqrt{x})$
 8) $\ln(\cos(x))$
 9) $\sqrt{\ln(\cos(\arctan(\sqrt{\cos(f)})))}$

Zadanie 2.6

1)

$$\frac{2a-3b}{(5a+b)^{-5}}$$

2) $(\sqrt{z^2+z+1})^3 - (\sqrt{z^2-z+1})^3 + 2e^{kz}$

3)

$$\frac{a^{3k-2n-1}}{b^{2k+3} + \frac{(c-d)^{k-n}}{(4b-c)^4}}$$

4)

$$\frac{a_{11}a_{22} - a_{12}a_{21}}{a_{11}^2 + a_{22}^2}$$

5)

$$\frac{\sin(2\alpha-3\beta)\cos\sqrt{x}}{\sin^3(\varphi - \frac{\pi}{4})} + 1$$

6)

$$[(2a+3b+4c)^{3a+4b+5c}]^{a-b} + \frac{(a+b)^c}{a+1}$$

7)

$$\left[\frac{((x-2y) - (z+3y)(x-z) - 1) + \frac{1}{y}}{(x+y+z)^2} + 5 \right] \cdot \frac{x}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}}$$

8)

$$\frac{x_a - (x_b + \frac{x_c}{x_a} - \frac{1}{(x_b + x_c)^{-k+1} - 1})}{(\frac{x_c - x_b}{x_a} + 1) - x_c}$$

9)

$$\ln \left| \frac{\cos a_\alpha}{\cos a_\beta} \right| \cdot \sin(\lambda \frac{\pi}{2} - \delta)$$

10)

$$\text{dla } x^2 \leq 1: \quad 2x^3 - x^2 + 4x - 5,$$

$$\text{dla } x^2 > 1: \quad (x^5 - 5x^3 + x)e^x$$

11)

$$\text{dla } x \leq y: \quad \sqrt{y^2 - x^2},$$

$$\text{dla } x > y, x > 0: \quad 0,$$

$$\text{dla } x > y, x \leq 0: \quad x - 2y,$$

12)

$$\text{dla } x_1 \leq x_2: \quad 1,$$

$$\text{dla } x_2 < x_1, x_2 < \frac{1}{2}x_1: \quad -1 + 2(x_1 - x_2),$$

$$\text{dla } x_2 < x_1, x_2 \geq \frac{1}{2}x_1: \quad -1 + 5(x_1 - x_2).$$

Zadanie 2.7

Są wyrażeniami algolowskimi: 1,2,4,5,8,9,12,18,20,23. Nie są wyrażeniami algolowskimi:

- 3 brak jednego nawiasu otwierającego,
- 6 dwa operatory \uparrow - bezpośrednio po sobie,
- 7 tablica A raz jest jednowymiarowa, a raz dwuwymiarowa,
- 10 dwa operatory $/$ - bezpośrednio po sobie,
- 11 użyto nawiasów kwadratowych zamiast okrągłych (sin nie może być nazwą tablicy, gdyż oznacza funkcję standardową sinus),
- 13 zamiast 2b powinno być $2 \times b$,
- 14 zamiast A(0,2,4) powinno być A[0,2,4],
- 15 nieprawidłowo użyto nawiasów kwadratowych, które w ALGOLu można używać tylko dla indeksów,
- 16 w warunku "jeśli" A oznacza zmienną boolowską, a po else zmienną liczbową,
- 17 po then jest wyrażenie arytmetyczne, a po else wyrażenie boolowskie,
- 19 wyraz then nie jest podkreślony,
- 21 po then następuje wyrażenie alternatywne, a powinno być proste,
- 22 po if następuje wyrażenie arytmetyczne zamiast boolowskiego,
- 24 wyrażenie alternatywne nie może być łączone operatorem arytmetycznym, należało zatem drugie wyrażenie alternatywne - tak jak pierwsze - przez wzięcie w nawiasy okrągłe uczynić wyrażeniem prostym.

Zadanie 2.8

- 1) $(2xx\sqrt{2-3xx+5})x(y-x\sqrt{3}/(x\sqrt{2-y\sqrt{2}}+1))x(1+2xxx(\sqrt{x}-y\sqrt{1/3}))\sqrt{2}$
- 2) $(a+b)\sqrt{(1-j)+((\sqrt{a}+1)/\sqrt{a+1}))\sqrt{5}$
- 3) $1/(1+1/(1+1/(1+1/(1+1/x))))$
- 4) $a\sqrt{(b+c)}\sqrt{(d+e\sqrt{(f+g)})}$
- 5) $(x1-x2/x3/(x4-x1\sqrt{2-x2\sqrt{3}}))/(x3\sqrt{(x1/x2+1)-x2\sqrt{(x3x(x1-\sqrt{x4}))})}$
- 6) $\sqrt{(A\sqrt{2+B\sqrt{2}})}x\sin(\alpha+\beta)x\sin(f1)/\cos(f1)x\exp(\alpha f x f1)$
- 7) $(xx(x+1)+2)xx-3x(\ln(\text{abs}(1-x\sqrt{2}))+3)\sqrt{.2}$
- 8) $(a[1,2]xb[2,2]-a[2,1]xb[1,1])/(a[2,2]xb[1,2]-a[1,1]xb[2,1])x\sqrt{(a[3,3]\sqrt{2+a[4,4]\sqrt{2}}\sqrt{2+1})}$
- 9) if X<Y then X/Y else if X=Y then 1 else $X\sqrt{3}-Y\sqrt{3}$
- 10) if $a\sqrt{2}=b\sqrt{2}$ then $(x\sqrt{2+1})\sqrt{(2/3)}$ else $1/x\sqrt{2}$

- 11) if $x \neq y$ then $1/(x-y)$ else if $x \neq 0$ then $1/x$ else 0
- 12) if $X \times Y \leq 0$ then X else if $Z > 0$ then $X + Z$ else if $\text{alfa} < 1.5707963$ then $X \times \sin(\text{alfa})$ else if $\text{delta} < \text{epsylon}$ then $\text{epsylon} \times 2 \times \exp(-\text{alfa} \times X)$ else if $Z \times \text{epsylon} < -8$ then $1 + \text{epsylon} \times X \times Y$ else $X + Y \times Z + \text{alfa} \times \text{delta} \times \text{epsylon}$

Zadanie 2.9

- 1) 12, 2) -2, 3) 8, 4) 2, 5) 3,
6) 7, 7) 3, 8) 2, 9) 1, 10) 0

Zadanie 2.11

Przykład 1:	a) <u>false</u> ,	b) <u>true</u> ,	c) <u>false</u> .
Przykład 2:	a) <u>false</u> ,	b) <u>true</u> ,	c) <u>true</u> .
Przykład 3:	a) <u>true</u> ,	b) <u>false</u> ,	c) <u>true</u> .
Przykład 4:	a) <u>true</u> ,	b) <u>false</u> ,	c) <u>true</u> .
Przykład 5:	a) <u>true</u> ,	b) <u>false</u> ,	c) <u>true</u> .
Przykład 6:	a) <u>false</u> ,	b) <u>true</u> ,	c) <u>true</u> .
Przykład 7:	a) <u>false</u> ,	b) <u>true</u> ,	c) <u>false</u> .
Przykład 8:	a) <u>true</u> ,	b) <u>false</u> ,	c) <u>true</u> .
Przykład 9:	a) <u>true</u> ,	b) <u>true</u> ,	c) <u>true</u> .
Przykład 10:	a) <u>false</u> ,	b) <u>false</u> ,	c) <u>true</u> .
Przykład 11:	a) <u>false</u> ,	b) <u>true</u> ,	c) <u>true</u> .
Przykład 12:	a) <u>true</u> ,	b) <u>true</u> ,	c) <u>true</u> .

Zadanie 2.12

- 1) Tak; jest to wyrażenie równoznaczne z true.
- 2) Nie; po then następuje wyrażenie alternatywne, a musi być proste.
- 3) Nie; równoważność nie jest działaniem na wartościach liczbowych, lecz na boolowskich.
- 4) Nie; relacja łączy tylko dwa proste wyrażenia arytmetyczne, a tu po prawej stronie operatora > podano wyrażenie alternatywne.
- 5) Tak; jest to wyrażenie równoznaczne z false.
- 6) Nie; nawiasów zamykających jest za mało.

- 7) Nie, wyraz else nie jest podkreślony.
 8) Tak; jest to wyrażenie równoznaczne z następującym wyrażeniem booleowskim:

if $x > 0$ then $2=y$ else $3=-y$

Wyrażenie to ma na przykład dla $x=y=1$ wartość false, a dla $x=-4, y=-3$ wartość true.

- 9) Nie, gdyż operatory logiczne mogą łączyć tylko proste wyrażenia booleowskie.
 10) Nie, gdyż X występuje w relacji $X > 0$ jako zmienna liczbowa, a w wyrażeniu $X \vee Z$ jako zmienna booleowska.

Zadanie 2.13

- 1) $x \geq 1 \wedge x < 0 \vee x > 3 \wedge x \leq 7$
- 2) $x \geq 1 \wedge x < 0 \vee y > -3$
- 3) if $x < 0$ then $y \geq 0 \vee y \leq 2$ else if $x = 0$ then $y = 2$ else $y \geq 2$
- 4) $x > 0 \Rightarrow y > z$, co jest równoznaczne z wyrażeniem alternatywnym
if $x > 0$ then $y > z$ else true
- 5) $x = 0 \Rightarrow y = 0$, co jest równoznaczne z wyrażeniem
 $x = 0 \vee y = 0 \vee x \neq 0 \wedge y \neq 0$
- 6) $x + y \leq 1 \Rightarrow x - y < 2$
- 7) $x + y \leq 1 \equiv x - y < 2$
- 8) $(x > 0 \Rightarrow y > 0) \Rightarrow x = y \vee y = z$
- 9) $k = 5 \times (k \div 5)$, jeżeli k jest deklarowane jako całkowite;
 w przeciwnym razie można użyć wyrażenia $k = 5 \times \text{entier}(k/5)$
- 10) $X = s_x(X; s) \equiv Y \neq s_x(Y; s)$
- 11) $(x > 0 \Rightarrow y > 0) \Rightarrow (x > 0 \Rightarrow z > 0)$
- 12) $\neg, (A[1] = A[2] \wedge A[2] = A[3] \wedge A[3] = A[4] \wedge A[4] = A[5] \wedge A[5] = A[6])$
- 13) $a[1] \wedge 2 + a[2] \wedge 2 + a[3] \wedge 2 + a[4] \wedge 2 + a[5] \wedge 2 \neq 0$
- 14) $Z = 2 \times (Z \div 2) \wedge Z > 0$
- 15) $3 \times q \neq \text{entier}(3 \times q)$
- 16) $(X = 1 \equiv Y = 1) \wedge (Y = 1 \equiv Z = 1)$, co jest równoznaczne z wyrażeniem
 $X = 1 \wedge Y = 1 \wedge Z = 1 \vee X \neq 1 \wedge Y \neq 1 \wedge Z \neq 1$

- 17) $c = \text{entier}(c) \Rightarrow c > 3$
 18) $A = \text{entier}(A) \equiv B = \text{entier}(B)$
 19) if $\alpha \geq -1/\alpha \leq 1$ then $\beta \geq -.5/\beta < .5$ else $\beta = .5$
 20) $\text{abs}(x) + \text{abs}(y) \leq 1$

Rozwiązanie w postaci

if $x \geq 0$ then (if $y \geq 0$ then $x+y \leq 1$ else $x-y \leq 1$) else if $y \geq 0$
then $y-x \leq 1$ else $-x-y \leq 1$

nie byłyby błędne, ale znacznie dłuższe.

Zadanie 2.14

Sprawdzenie wyrażeń 1-4 wykonujemy rozpatrując wszystkie cztery możliwe przypadki, tj.

a	b
<u>true</u>	<u>true</u>
<u>true</u>	<u>false</u>
<u>false</u>	<u>true</u>
<u>false</u>	<u>false</u>

Prawdziwość wyrażenia 5 wynika z prawdziwości wyrażeń 4 i 1.

Zadanie 3.5

- 1) $X=0.5$, $Y=0.108$.
 2) $X=3$, $Y=-5$.
 3) $X=1$.

Zadanie 3.10

- 1) if $c > 0$ then $XX := 7xz + 6$ else $XX := 5xz + 2$
 2) if $a^2 + b^2 > 15$ then go to ITER[$2x + j$] else go to K13K
 3) if $a < b/\alpha \geq 0$ then go to RAK[3] else go to RAK[m+n]
 4) if $wrm < -1$ then $BoB := A > B \wedge C$ else $BoB := B \wedge C$

Zadanie 3.12a) begin real x;

X := -2;

L: x := (x $\sqrt[3]{3-2x+5}$)/(3x $\sqrt[3]{2-2}$); X := X-x;if abs(x) \geq 8 then go to Lend;

Zmienna nielokalna X oznacza szukany pierwiastek

b) begin real x, x1, x2, f1, f2, f;

x := x1 := -3; x2 := -2;

f1 := x1 $\sqrt[3]{3-2x1+5}$;f2 := x2 $\sqrt[3]{3-2x2+5}$;

L: X := (x2xf1 - x1xf2)/(f1 - f2);

if abs(X-x) $<$ 8 then go to K;f := X $\sqrt[3]{3-2X+5}$;if f=0 then go to K; x := X;if fxf1 > 0 then begin x1 := X; f1 := f end elsebegin x2 := X; f2 := f end; go to L;K: end;Zadanie 3.13

Popelniono następujące błędy:

- w trzecim wierszu zamiast 3a powinno być 3x a,
- w czwartym wierszu użyto etykiety o tej samej nazwie, co jedna ze zmiennych,
- w czwartym wierszu zmienna b musiałaby być booleowska, a poprzednio występowała jako zmienna liczbowa,
- w czwartym wierszu użyto zmiennej c, która nie była deklarowana,
- w piątym wierszu po lewej stronie instrukcji podstawienia występują zmienne a i k, które są zmiennymi różnych typów,
- w szóstym wierszu ponownie dwuznaczność symbolu a,
- w siódmym wierszu po else jest instrukcja, która nie będąc wyrażeniem nie może być podstawiana na zmienną X.
- w ostatnim wierszu po end brak średnika.

Uwaga: podstawienie w drugim wierszu na zmienną całkowitą k wartości 0.1 nie jest błędem, ponieważ - jak to już było powiedziane - w takim przypadku zostaje podstawiona najbliższa liczba całkowita, w danym przypadku zamiast 0.1 liczba 0.

Zadanie 3.14

Wykorzystane etykiety: AA, A, B, BB /instrukcja go to
 Kad [8] była równoznaczna z instrukcją pustą.
 Końcowe wartości zmiennych: a=8, b=64.

Zadanie 3.16

Z=3. Wykorzystane etykiety: W1, STOP.

Zadanie 3.19

```

begin integer m,n,p; real s;
  for m:= 1 step 1 until i do
    for n:= 1 step 1 until k do
      begin s:= 0;
        for p:= 1 step 1 until j do
          s:= s+A[m,p]×B[p,n];
          C[m,n]:= s
        end
      end;

```

Uwaga: dla ekonomii użyto zmiennej s zamiast C[m,n] w czasie sumowania, ponieważ wyszukiwanie zmiennej indeksowanej trwa w maszynie znacznie dłużej niż wyszukiwanie zmiennej bez indeksu.

Zadanie 3.20

```

.....
i:= 0;
for c:= 1.668, 4.608, 6.951, 8.002, 10.168,
  12.376, 14.589, 18.207, 20.113, 22.724 do

```

begin

i:= i+1; A[i]:= c

end;

.....

Nie można tu było zamiast c użyć A[i] jako zmiennej kontrolowanej, ponieważ wtedy indeks i zostałby na wstępie ustalony dla całej instrukcji "dla".

Zadanie 3.21

5,

3, 4, 7,

-14, -17, -23, -32, -44, -59, -77, -98,

-94, -84, -68, -46, -18, 16, 56,

54, 50, 44, 36, 26, 14, 0, 16, 15, 15, 16, 18, 21, 25, 30,

30, 29, 27, 24,

24, 23, 21,

21, 20,

20,

1, 19, 18; -1.

Zadanie 3.24

begin comment Sumowanie szeregu; integer n; real a;

n:= -1; S:= 0; a:= 1;

for n:= n+2 while a>= -8 do

begin a:= 1/n/4/n;

$S := S + (-1)^{((n-1):2)} \times a$ end

end;

Zadanie 1.5

procedure dzielenie(a,b) przez:(c,d) daje:(x,y);

value a,b,c,d; real a,b,c,d,x,y;

begin real u;

u:= $c^2 + d^2$;

x:= $(a \times c + b \times d) / u$;

y:= $(b \times c - a \times d) / u$

end;

Zadanie 4.6

procedure MnozMac(A) przez:(B) daje:(C) Wierszy w A:(i)

Kolumn w A:(j) Kolumn w B:(k);

value i,j,k; integer i,j,k; array A,B,C;

begin integer m,n,p; real s;

for m:= 1 step 1 until i do

for n:= 1 step 1 until k do

begin s:= 0;

for p:= 1 step 1 until j do s:= s+A[m,p]×B[p,n];

C[m,n]:= s;

end end MnozMac;

Zadanie 4.7

Rozwiązanie zadania nie jest jednoznaczne. Oto przykład rozwiązania:

```

procedure ZAMIANA(x,y) na biegunowe:(r,fi);
value x,y; real x,y,r,fi;
begin r:= sqrt(x2+y2);
      fi:= if y=0 then (if x>0 then 0 else 3.14159265) else
           arctan(-x/y)+(if y<0 then 4.71238898 else 1.57079633)
end;

```

Zadanie 4.8

Po wyeliminowaniu pierwszej niewiadomej wykorzystamy pierwszą kolumnę macierzy współczynników A dla przechowywania indeksów, wskazujących miejsce odpowiedniej niewiadomej. Jeżeli na przykład będzie $A[j,1]=k$, znaczyć to będzie, że x_k po zakończeniu obliczeń występuje jako $A[j,n+1]$. W celu uporządkowania niewiadomych według wyjściowej kolejności posłużymy się drugą kolumną macierzy A. W tej kolumnie będą ostatecznie wartości niewiadomych po wykonaniu procedury. W ten sposób oszczędzamy miejsce w pamięci maszyny, umożliwiając rozwiązywanie większych układów równań. A oto proponowany program:

```

procedure UkładRownLin(A) Liczba rown:(n) Etykieta:(Zdegenerowany);
comment Macierz współczynników A ma n wierszy i n+1 kolumn, z których ostatnia jest złożona z wyrazów wolnych. Wyniki otrzymuje się w drugiej kolumnie tablicy A tzn.  $X_k=A[k,2]$ ;
value n; integer n; array A; label Zdegenerowany;
begin integer i,j,k,p,q; real max,r;
for i:= 1 step 1 until n do
  begin max:= A[i,1]; p:= q:= i;
    for j:= i step 1 until n do
      for k:= i step 1 until n do
        if abs(A[j,k])>abs(max) then

```

```

    begin p:= j; q:= k; max:= A[p,q] end;
    if max=0 then go to Zdegenerowany;
    for j:= 1 step 1 until n do
        begin r:= A[j,q]; A[j,q]:= A[j,1]; A[j,1]:= r end;
    for k:= 1 step 1 until n+1 do
        begin r:= A[p,k]; A[p,k]:= A[i,k]; A[i,k]:= r end;
    for k:= i+1 step 1 until n+1 do A[i,k]:= A[i,k]/max;
    for j:= i+1 step 1 until n do
        for k:= i+1 step 1 until n+1 do
            A[j,k]:= A[j,k]-A[j,i]×A[i,k];
        if i=1 then for j:= 1 step 1 until n do A[j,1]:= j;
        k:= A[q,1]; A[q,1]:= A[i,1]; A[i,1]:= k ?
    end;
    for i:= n step -1 until 1 do
        for k:= i+1 step 1 until n do
            A[i,n+1]:= A[i,n+1]-A[i,k]×A[k,n+1];
    for i:= 1 step 1 until n do
        A[A[i,1],2]:= A[i,n+1]
    end procedure;

```

Zadanie 4.9

```

begin array M[1:3,1:4];
M[1,1]:= 2; M[1,2]:= -1; M[1,3]:= -3; M[1,4]:= 4;
M[2,1]:= 1; M[2,2]:= 1; M[2,3]:= 1; M[2,4]:= -4;
M[3,1]:= -1; M[3,2]:= 1; M[3,3]:= 0; M[3,4]:= 3;
UkladRownLin(M,3,Koniec);

```

```
x:= M[1,2]; y:= M[2,2]; z:= M[3,2];
```

```
Koniec: end;
```

Zadanie 4.11

Jeżeli w momencie wywołania procedury Ilskal zmienna P nie miała określonej wartości, wartość x również nie będzie określona.

Jeżeli natomiast w momencie wywołania procedury Ilskal zmienna P miała określoną wartość, otrzymamy

$$x = 20 \times A[t, P, u] \times B[P] / \sqrt{t^2 + u^2}$$

gdzie t, P, u miałyby wartości, jak w momencie wywołania procedury Ilskal, ponieważ podany fragment programu byłby równoważny następującemu:

```
.....
begin real k,a,b; k:= 10;
a:= A[t,P,u]; b:= B[P];
begin real s; s:= 0;
  for P:= 1 step 1 until k do s:= s+a*x;
  x:= 2*x/sqrt(t^2+u^2)
end end;
.....
```

Zadanie 4.12

Deklaracja procedury:

```
procedure Ilskal(a,b) Rzad:(k,p) Wynik:(y);
```

```
value k; integer k,p; real y,a,b;
```

```
begin real s; s:= 0;
```

```
  for p:= 1 step 1 until k do s:= s+a*x;
```

```
  y:= s
```

```
end;
```

Fragment programu z wywołaniem tej procedury wyglądałby następująco:

```
.....
Ilskal(A[t,P,u],B[P],l0,P,Y);
x:= 2*Y/sqrt(t/2-u/2);
.....
```

Zadanie 4.13

```
real procedure tg(x); value x; real x; tg:= sin(x)/cos(x);
```

Zadanie 4.14

```
real procedure F(x) Tablica wartosci:(T) Poczatek:(x0) Krok:(h);
value x,x0,h; real x,x0,h; array T;
comment T[i]=F(x0+i*xh);
begin integer m;
x:= (x-x0)/h;
m:= entier(x);
x:= x-m;
F:= T[m]*(x-1)*(x-2)/2+
    T[m+1]*x*(x-2)+
    T[m+2]*x*(x-1)/2 end procedure;
```

Zadanie 4.15

1) Funkcja całkowana jako procedura funkcyjna:


```

real procedure CALKA(Funkcja) Przedzial:(a,b) Liczba czesci:(n);
value a,b,n; integer n; real a,b; real procedure Funkcja;
begin real h,s,t; integer i;

  h:= (b-a)/n; s:= Funkcja(a)+Funkcja(b);

  t:= 0;

  for i:= 1 step 2 until n-1 do t:= t+Funkcja(a+i×h);

  s:= s+4×t; t:= 0;

  for i:= 2 step 2 until n-2 do t:= t+Funkcja(a+i×h);

  s:= s+2×t;

  CALKA:= h/3×s end CALKI;

```

Dla obliczenia podanej całki stosujemy program:

```

begin real procedure f(x); value x; real x; f:= 1/ln(x);

  I:= CALKA(f,2,3,20) end;

```

zakładając, że deklaracja zmiennej I musiała być uczynion wcześniej.

2) Funkcja całkowana traktowana jako wyrażenie:

```

real procedure CALKA(funkcja) Zmienna:(t) Przedzial:(a,b)
  Liczba czesci:(n);

  value a,b,n; integer n; real funkcja,t,a,b;

  begin real h,s,u;

    s:= 0; h:= (b-a)/n; u:= 0;

    for t:= a,b do s:= s+funkcja;

    for t:= a+h step 2×h until b-0.9×h do u:= u+funkcja;

    s:= s+4×u; u:= 0;

    for t:= a+2×h step 2×h until b-1.9×h do u:= -u+funkcja;

    s:= s+2×u;

    CALKA:= h/3×s end CALKI;

```

Zamiast $b-h$ i $b-2 \times h$ użyto $b-0.9 \times h$ i $b-1.9 \times h$ ze względu na możliwe błędy zaokrągleń przy obliczaniu coraz to nowych wartości t .

Dla obliczenia danej całki wystarczy teraz jedna instrukcja:

$I := \text{CALKA}(1/\ln(x), x, 2, 3, 20);$

Zadanie 4.17

```

real procedure I(a,b,n);
value a,b,n; real a,b; integer n;
I:= if n=0 then b-a else
      if n=1 then ln(abs(cos(a)/cos(b))) else
      ((sin(b)/cos(b))n-(sin(a)/cos(a))n)/(n-1) -
      I(a,b,n-2);

```

Zadanie 4.18

```

integer procedure C(m,n); value m,n; integer m,n;
C:= if n=0 or m=n then 1 else C(m-1,n-1)+C(m-1,n);

```

Zadanie 4.19

```

real procedure T(n,x); value n,x; integer n; real x;
T:= if n=0 then 2 else
      if n=1 then x else
      x*T(n-1,x)-T(n-2,x)/4;

```

Zadanie 4.21

Wobec tego, że krok całkowania ma być około 0.1, przyjmujemy dla całki zewnętrznej podział przedziału całkowania na 40 części. Natomiast przedział całkowania całki wewnętrznej jest zmienny. Aby uzyskać odpowiedni podział na parzystą liczbę części, jak tego wymaga metoda Simpsona, dzielimy go na

$$2 \times \text{entier} \left(\frac{\sqrt{x} - (-\sqrt{4+x^2})}{0.2} \right)$$

czyli na

$$2 \times \text{entier}((\text{sqrt}(x) + \text{sqrt}(4+x^2))/0.2)$$

części. Wobec tego dla obliczenia danej całki podwójnej wystarczy instrukcja:

```
I:= CALKA(CALKA(sin(sqrt(x^2+y^2))),y,-sqrt(4+x^2),sqrt(x),
          2*entier((sqrt(x)+sqrt(4+x^2))/0.2)),x,1,5,40);
```

Wykorzystaliśmy tu drugi wariant procedury dla obliczania całki.

Zadanie 4.22

```
begin real procedure P(u); value u; real u;
P:= sqrt(3*x+2*xu);
*x:= P(P(P(t^2+1))) end;
```

Należy tu pamiętać, że - zgodnie z regułami budowy ciała procedury - tylko instrukcja

```
P:= -sqrt(3*x+2*xu)
```

stanowi w powyższym programie ciało procedury i na niej kończą się deklaracje podanego bloku.

Zadanie powyższe można rozwiązać również bez rekurencyjnego wywoływania procedury, a mianowicie za pomocą programu:

```
begin integer i;
x:= t^2+1;
for i:= 1,2,3,4 do x:= sqrt(3*x+2*x) end;
```

Zadanie 4.23

$$y = \frac{a}{1 + \frac{b}{1 + \frac{c}{1 + \frac{d}{1 + \frac{e}{1+x}}}}}$$

Program tego obliczenia można ułożyć następująco:

```

begin real u;
y:= x;
for u:= e,d,c,b,a do y:= u/(1+y) end;

```

Zadanie 4.25

Dla uwypuklenia deklaracji procedury program przepiszemy w postaci

```

begin integer i;
procedure p;  x:= x+i;
i:= 4;  x:= 1;
p;
begin integer i;
i:= 2;
p
end end;

```

Prześledzimy wykonanie programu. Mamy do czynienia z kolizją dwu zmiennych o tej samej nazwie i. Wobec tego zmienimy nazwę zmiennej w bloku wewnętrznym, aby usunąć kolizję. Otrzymujemy wtedy program:

```

begin integer i;

procedure p;  x:= x+1;

i:= 4;  x:= 1;

p;

begin integer k;

k:= 2;

p

end end;

```

Na początku jest $i=4$ i $x=1$. Wykonanie procedury p daje z kolei $x=5$. Wchodzimy do bloku wewnętrznego, gdzie $k=2$. Następuje po raz drugi wykonanie procedury p . Istota przykładu polega na wyjaśnieniu tego drugiego wykonania: mamy wykonać przez tę procedurę podstawienie $x:=x+1$. Jaką wartość na i należy tu podstawić: wartość zewnętrznego i , czy wartość wewnętrznego i , tzn. obecnie k ? Odpowiedź jest następująca: W ciele procedury parametr i jest parametrem nielokalnym. Ponieważ deklaracja procedury znajduje się na zewnątrz bloku wewnętrznego, w jej ciele obowiązuje ostatnia wartość zewnętrznego i . Wobec tego wykonanie po raz drugi procedury p , tzn. jej wykonanie w bloku wewnętrznym, polega na wykonaniu podstawienia $x:=x+i$, a nie $x:=x+k$, co daje ostatecznie $x=9$.

Zadanie 5.3

- 1) $+.dd_1ddd$
- 2) $nddd.d000$
- 3) $+ddd.00_n+dd$
- 4) $-nd_1dd_100.00_n+d$
- 5) $ddd_1000.000_n-dd$

Zadanie 5.4

$\langle +a \rangle$	$\langle -a.d \rangle$	$\langle a.dd \rangle$	$\langle \pm ddaaaaa \rangle$
$; +6_{10}2;$	$; 5.7_{10}2;$	$; 5.67_{10}2;$	$; + 567;$
$; -6_{10}2;$	$; -5.7_{10}2;$	$; -5.67_{10}2;$	$; - 567;$
$; +0;$	$; .0;$	$; .00;$	$; + 0;$
$; +0;$	$; .0;$	$; .00;$	$; + 0;$
$; +1_{10}3;$	$; 1.4_{10}3;$	$; 1.39_{10}3;$	$; + 1388;$
$; -7_{10}5;$	$; -7.0_{10}5;$	$; -6.97_{10}5;$	$; -697000;$

$\langle -nd.ddd \rangle$	$\langle +d.d_{10}ddd \rangle$	$\langle -d.d_{10}+ddd \rangle$
$; 56.700_{10}1;$	$; +5.7_{10}2;$	$; 5.7_{10}+2;$
$; -56.700_{10}1;$	$; -5.7_{10}2;$	$; -5.7_{10}+2;$
$; -0.004;$	$; -3.7_{10}-3;$	$; -3.7_{10}-3;$
$; 0.000;$	$; +.0$	$; .0$
$; 13.884_{10}2;$	$; +1.4_{10}3;$	$; 1.4_{10}+3;$
$; -69.700_{10}4;$	$; -7.0_{10}5;$	$; -7.0_{10}+5;$

$\langle +d.d_{10}+ddd \rangle$	$\langle -daaa.d0000 \rangle$
$; +5.7_{10}+ 2;$	$; 567.0$
$; -5.7_{10}+ 2;$	$; -567.0$
$; -3.7_{10}- 3;$	$; -.00368;$
$; +.0$	$; .00000;$
$; +1.4_{10}+ 3;$	$; 1388.4$
$; -7.0_{10}+ 5;$	$; -6970.0_{10}2;$

$\downarrow +d.dd00_{10}+d\downarrow$	$\downarrow +d.dd.d_{10}0_{10}+d\downarrow$
; +.567 ₁₀ +3;	;+5 67.0 ;
; -.567 ₁₀ +3;	; -5 67.0 ;
; -3.68 ₁₀ -3;	; - 36.7 9 ₁₀ -4;
; +.0000 ;	; + .0 0 ;
; +1.39 ₁₀ +3;	; + 13.8 8 ₁₀ +2;
; -.697 ₁₀ +6;	; - 69.7 0 ₁₀ +4;

Użyto tutaj średników dla lepszej orientacji co do położenia drukowanych znaków. W oryginalnym druku średników tych oczywiście nie było.

Zadanie 5.6

```

if z<0 then <<
.....Funkcja> else if z=0 then <<0,0,0> else <<
.....XYZ.....YZX.....ZXY
.....5>

```

Zadanie 5.11

```

begin real x,X;  X:= -2;
for x:= (X3-2X+5)/(3X2-2) while abs(x)>=8 do X:= X-x;
output(<+n.ddddd>,X) end;

```

Zadanie 5.13

```

begin real x,X,x1,x2,f1,f2,f;

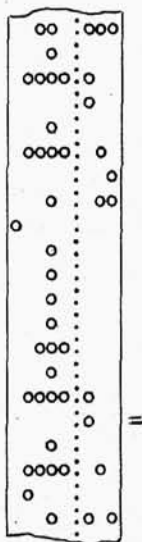
```

```

x:= x1:= -3;   X:= x2:= -2;
f1:= x1^3-2*x1+5;   f:= f2:= x2^3-2*x2+5;
for X:= (x2*f1-x1*f2)/(f1-f2) while f#0^abs(X-x)>=10-8 do
  begin f:= X^3-2*X+5;
    if f*f1>0 then begin x1:= X;   f1:= f end else
      if f*f2>0 then begin x2:= X;   f2:= f end end;
write(└-d.ddd┘,d┘,X) end;

```

Zadanie 5.15



Zakładając, że w momencie rozpoczęcia przedruku karetki flexowritera znajduje się w ustawieniu początkowym, otrzymamy z przedruku powyższego odcinka taśmy papierowej następujący tekst:

$x = 13$

$y = -5$

Zadanie 5.17

-5.3900₁₀+1 7.4788₁₀+4 wartosci kontrolne

Suma = 990

KONIEC DRUKU

Zadanie 5.19

```
output(⟨-d.ddddd⟩, outtext(⟨X,Y,Z1=...⟩), X,
      outsp(6), Y, outsp(6), Z)
```

Zadanie 5.21

```
output(⟨dddd⟩, outcr, outtext(⟨U1⟩), outsp(12), outtext(⟨U2⟩),
      outsp(12), outtext(⟨U3⟩), outsp(12), outtext(⟨U4⟩), outsp(12),
      outtext(⟨U5⟩), outcr, U1, outsp(10), U2, outsp(10), U3, outsp(10), U4,
      outsp(10), U5)
```

albo bez użycia instrukcji outcr:

```
output(⟨dddd⟩, outtext(⟨<
U1.....U2.....U3.....U4.....U5
⟩), U1, outsp(10), U2, outsp(10), U3, outsp(10), U4, outsp(10), U5)
```

Zadanie 5.23

```
write(⟨+n.ddd10-d⟩, writecr, p, writecr, q, writecr, r, writecr, s)
```

Zadanie 5,26

```
output(↑dddd↑,outer,gettext(↑<_U1↑),outchar(30),gettext(↑<_U2↑),
outchar(30),gettext(↑<_U3↑),outchar(30),gettext(↑<_U4↑),outchar(30),
gettext(↑<U5↑),outer,U1,outchar(30),U2,outchar(30),U3,outchar(30),
U4,outchar(30),U5)
```

Uwaga: Zamiast outer można również używać outchar(64).

Zadanie 5,27

```
outchar(60); outchar(23); outchar(0); outchar(4); outchar(0);
outchar(58);
```

Zadanie 5,29

- 1) writechar(if X=5X(X:5) then 64 else 30)
- 2) writechar(29)

Zadanie 5,31

```
begin comment Tablice funkcji Psi;
real w,a,b; integer u,v;
outer; outer; outer;
gettext(↑<_.....Tablice_..funkcji_..Psi
_u.....0.....2.....4.....6.....8
↑);
```

```

for u:= 0 step 5 until 95 do begin
for v:= 0,1,2,3,4 do begin output({ddd},u+v); outsp(1);
for w:= 0, 0.2, 0.4, 0.6, 0.8 do begin
outsp(5); a:= u+v+w; b:= sqrt(ava+1);
output({n.ddd},(b-a)/(b+a)) end; outcr end; outcr end;
outcr; outcr; outcr end;

```

To samo zadanie można rozwiązać również inaczej, na przykład:

```

begin comment Tablice funkcji Psi;
real a; integer u;
outcr; outcr; outcr;
outtext(outsp(22),{<Tablice,,funkcji,,Psi
u,,,,,,,,0,,,,,,,,2,,,,,,,,4,,,,,,,,6,,,,,,,,8};
for u:= 0 step 2 until 998 do begin
if u=10x(u:10) then begin outcr; output({ddd},u:10); outsp(1) end;
outsp(5); a:= sqrt(uxu+100); output({n.ddd},(a-u)/(a+u));
if u+1=50x((u+1):50) then outcr end; outcr; outcr; outcr end;

```

Zadanie 5.34

```

begin comment Rozwiązanie układu m równan liniowych;
integer m; procedure URL(A) Liczba rownan:(n) Etykieta:
(Zdegenerowany);
- tu następuje ciąg dalszy procedury z zadania 4.8 -
end procedury;
input(m);
begin array C[1:m,1:m+1]; integer s;
input(C);
URL(C,m,Q);

```

```

outtext(outcr,⟨<Rozwiązanie układu równan liniowych:⟩,outcr);
for s:= 1 step 1 until m do output(⟨+n.ddddd⟩,C[s,2],outchar(64));
go to Koniec;
Q: writetext(⟨<Układ zdegenerowany⟩);
Koniec: end end;

```

W przypadku układu zdegenerowanego został tu przewidziany druk sygnalizujący na maszynie do pisania. Przy perforacji wyników outchar(64) daje kod powrotu karetki, który jest zarazem terminatorem kończącym zapis liczby. Perforację wyników poprzedzono perforacją nagłówka z tekstem:

Rozwiązanie układu równan liniowych:

Zadanie 5.35

Wzory przekształćmy do wygodniejszej dla programu postaci:

$$S_x = \sqrt{\frac{\sum_{k=1}^m w_k x_k^2}{\sum_{k=1}^m w_k} - \bar{X}^2}$$

$$S_y = \sqrt{\frac{\sum_{k=1}^m w_k y_k^2}{\sum_{k=1}^m w_k} - \bar{Y}^2}$$

$$S_{xy}^2 = \frac{\sum_{k=1}^m w_k x_k y_k}{\sum_{k=1}^m w_k} - \bar{X}\bar{Y}$$

i w programie będziemy sukcesywnie liczyli sumy:

$$\sum_{k=1}^m w_k x_k, \quad \sum_{k=1}^m w_k y_k, \quad \sum_{k=1}^m w_k, \quad \sum_{k=1}^m w_k x_k^2, \quad \sum_{k=1}^m w_k x_k y_k, \quad \sum_{k=1}^m w_k y_k^2$$

używając do nich jako zmiennych roboczych $X, Y, r, Sx, Sy, Sxy2$, które później otrzymują inne znaczenia.

A oto program:

```

begin comment Obliczenie współczynników regresji i korelacji;
integer m,i; real x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;
X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;
input(m);
for i:= 1 step 1 until m do
    begin input(x,y,w);
        X:= X+w*x; Y:= Y+w*y; r:= r+w;
        Sx:= Sx+w*x/2; Sy:= Sy+w*y/2; Sxy2:= Sxy2+w*x*y
    end;
X:= X/r; Y:= Y/r;
Sx:= Sx/r-X/2; Sy:= Sy/r-Y/2; Sxy2:= Sxy2/r-X*Y;
Ryx:= Sxy2/Sx; Rxy:= Sxy2/Sy;
Sx:= sqrt(Sx); Sy:= sqrt(Sy);
r:= Sxy2/Sx/Sy;
begin procedure Druk(a,b); value b; string a; real b;
    writetext(a,<_=>,write(<-d.dddd_>,b),<_>);
    writecr; Druk(<_X>,X); Druk(<_Y>,Y);
    writecr; Druk(<_Sx>,Sx); Druk(<_Sy>,Sy);. Druk(<_Sxy2>,Sxy2);
    writecr; Druk(<_Ryx>,Ryx); Druk(<_Rxy>,Rxy); Druk(<_r>,r)
end end;
```

Zadanie 5.37

Należałoby na początku taśmy papierowej wydziurkować nie tylko liczbę równań m , ale również po niej wartość $m+1$, a w programie po opuszczeniu deklaracji integer m i instrukcji input(m) następny blok rozpocząć od

```
begin array C[1:inone,1:inone];
```

Uwaga: Nie można na taśmie wydziurkować tylko liczby m bez $m+1$ i deklarować

```
begin array C[1:inone,1:inone+1];
```

ponieważ za drugim inone będzie czytana następna liczba z taśmy. Ale można by użyć tej ostatniej deklaracji, gdyby zamiast m i $m+1$ taśma papierowa rozpoczynała się liczbą m powtórzoną dwukrotnie.

Zadanie 5.41

Wystarczy na przykład bezpośrednio po etykiecie S6 wstawić instrukcję:

```
if kbon then go to S4;
```

Bardzo szybko po wciśnięciu klucza KB maszyna wydrukuje aktualne wartości x , y i $R(x,y)$, potem znaki

S=

i będzie oczekiwała na napisanie na maszynie do pisania indeksu dla przełącznika S. W tym momencie możemy się zdecydować, co robić dalej.

Zadanie 5.43

```
begin comment Obliczenie współczynników regresji i korelacji;
```

```
real x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;
```

```
X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;
```

```
Q: if inchar 17 then begin input(x,y,w);
```

```
    X:= X+wxx; Y:= Y+wxy; r:= r+w;
```

```
    Sx:= Sx+wxx1/2; Sy:= Sy+wxy1/2; Sxy2:= Sxy2+wxxxy;
```

```
    go to Q end;
```

```
X:= X/r; Y:= Y/r;
```

Dalszy ciąg programu jak w zadaniu 5.35.

Zadanie 5,44

begin comment Obliczenie współczynników regresji i korelacji;

real c,x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;

X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;

c:= inchar;

Q: if inchar \neq c then begin input(x,y,w);

Dalszy ciąg programu jak w zadaniu 5.43.

Zadanie 5,46

go to if typechar=49 then A else B

Gdy na maszynie do pisania przyciśniemy po zapaleniu się zielonego światła klawisz "a", nastąpi skok do A. Po przyciśnięciu innego klawisza (z symbolem właściwym), nastąpi skok do B.

Zadanie 5,47

begin comment Rozwiązanie układu równań;

real x,y,h,r1,r2,Rx,Ry,p,s;

real procedure R(u,v); value u,v; real u,v;

R:= $(u\sqrt{3}-2xv\sqrt{3}+u\sqrt{2}+v-1)\sqrt{2}+(2xu\sqrt{3}+3xv\sqrt{3}+u-2xv\sqrt{2}+6)\sqrt{2}$;

S: writetext(\langle ...Skok do... \rangle); s:= typechar; go to

if s=49 then a else if s=50 then b else if s=51 then c else

if s=52 then d else e;

a: writecr; writetext(\langle x,y= \rangle);

x:= typein; y:= typein; go to S;

```

b: writecr; writetext( $\langle \langle h = \_ \rangle \rangle$ ); h:= typein; go to S;
c: writecr; write( $\langle -d.dddd, \_ \rangle$ ), writetext( $\langle \langle x = \_ \rangle \rangle$ ), x,
  writetext( $\langle \langle \_, \_ \rangle \rangle$ ), y, writetext( $\langle \langle \_, \_, R = \_ \rangle \rangle$ ), R(x,y));
go to S;
d: r1:= R(x,y);
e: Rx:= R(x+h,y)-r1; Ry:= R(x,y+h)-r1;
  p:= h/sqrt(Rx2+Ry2);
  r2:= R(x-pRx,y-pRy);
  if r1>r2 then begin r1:= r2; x:= x-pRx; y:= y-pRy;
    go to e end;
go to c end;

```

Zadanie 5.48

```

begin comment Obliczenie współczynników regresji i korelacji;
real c,x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;
X:= Y:= r:= Sx:= Sy:= Sxy2:= 0; c:= typechar;
P: if inchar c then go to P;
Q: if inchar c then begin input(x,y,w);
  Dalszy ciąg programu jak w zadaniu 5.43.

```

Założenia dla taśmy papierowej z danymi:

- między kolejnymi trójkami liczb w każdej serii prócz terminatora jest jeszcze co najmniej jeden kod właściwy,
- po terminatorze kończącym zapis ostatniej liczby w serii następuje bezpośrednio kod charakterystyczny dla tej serii, będący zawsze kodem właściwym,
- po kodzie charakterystycznym sygnalizującym koniec jednej serii a przed pierwszą liczbą następnej serii musi być kod charakterystyczny tej następnej serii (może on występować między innymi terminatorami),
- przed pierwszą liczbą pierwszej serii musi być (niekoniecznie bezpośrednio) kod charakterystyczny pierwszej serii,

- każdy kod charakterystyczny występuje na danej taśmie papierowej dokładnie dwa razy: raz przed i raz po serii, którą charakteryzuje.

Zadanie 5.49

- 1) outcr;
- 2) outtext(~~4~~<
- ~~1~~);
- 3) outchar(64);
- 4) outchar(typechar);

Zadanie 5.51

- 1) begin comment Obliczenie współczynników regresji i korelacji;

real x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;

X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;

P: if inchar~~4~~25 then go to P;

Q: if inchar~~4~~25 then begin input(x,y,w);

Dalszy ciąg programu jak w zadaniu 5.43.

- 2) begin comment Obliczenie współczynników regresji i korelacji;

real x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;

X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;

P: if inchar~~4~~25 then go to P;

T: if lyn~~4~~31 then go to T;

Q: if lyn~~4~~44 then begin input(x,y,w);

Dalszy ciąg programu jak w zadaniu 5.43.

3) begin comment Obliczenie współczynników regresji i korelacji;

real x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;

X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;

P: if inchar+25 then go to P;

Q: if lyn+25 then begin input(x,y,w);

Dalszy ciąg programu jak w zadaniu 5.43.

Uwaga: W wariancie 2 kod PUNCH ON, a w wariancie 3 zarówno kod PUNCH OFF, jak i PUNCH ON muszą następować bezpośrednio po terminatorze kończącym zapis poprzedniej liczby.

Zadanie 5.53

begin comment Obliczenie współczynników regresji i korelacji;

real c,x,y,w,X,Y,Sx,Sy,Sxy2,Rxy,Ryx,r;

X:= Y:= r:= Sx:= Sy:= Sxy2:= 0;

c:= inchar; x:= inchar;

Q: if char=c then begin input(x,y,w);

Dalszy ciąg programu jak w zadaniu 5.43.

Uwaga: Po wprowadzeniu wartości charakterystycznej na c wprowadzamy jakąkolwiek inną wartość przejściowo na x, aby później - przez pobranie tej różnej od c wartości za pomocą procedury char - uruchomić procedurę input (x,y,w).

Założenia dla taśmy papierowej:

- terminator kończący ostatnią liczbę musi być kodem charakterystycznym, występującym poza tym tylko jeszcze jeden raz, a mianowicie jako pierwszy kod właściwy na taśmie dla podstawienia odpowiadającej mu liczby na zmienną roboczą c;

- po kodzie charakterystycznym rozpoczynającym taśmę, którego odpowiadająca liczba zostaje podstawiona na zmienną c, a jeszcze przed pierwszą wyperforowaną liczbą, musi być co najmniej jeden kod właściwy, oczywiście różny od charakterystycznego, dla podstawienia jakiegokolwiek różnej od c wartości na zmienną c dla uruchomienia procedury input.

- liczby mogą być od siebie oddzielone nawet tylko jednym terminatorem.

Zadanie 5.55

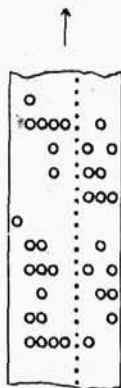
W obu przypadkach kod znaków 9 i).

Zadanie 5.56

```

begin integer c,s; s:= 0;
P: c:= inchar; if c#131 then begin
    if c#4 then go to P else
        begin setchar(c);
            a:= s+inone;
            go to P end end;
write(⟨ndddd⟩,s) end;

```

Zadanie 5.59Zadanie 5.61

Maszyna do pisania wydrukuje:

Numer problemu:

Zadanie 5.66

Zadanie można rozwiązać następująco:

```
c:= drumplace;
to drum(A);
drumplace:= c;
from drum(B);
```

Prostsze jednak jest rozwiązanie następujące:

```
drumplace:= - to drum(A) + drumplace;
from drum(B);
```

albo

```
drumplace:= drumplace + to drum(A)×0;
from drum(B);
```

Należy tu pamiętać o dynamiczności wyrażenia

- to drum(A) + drumplace

które nie jest wcale równoznaczne z wyrażeniem

drumplace - to drum(A)

(patrz paragraf 4.24).

Zadanie 5.67

1) drumplace:= c-(m-1+l)×n+j;

from drum(P);

2) drumplace:= c-(m-1)×n;

from drum(C);

```

3)  for p:= 1 step 1 until m do
      begin drumplace:= c-(m-p+1)*xn+j;
          from drum(P);
          D[p]:= P[1] end;

```

Zadanie 5.68

```

1)  drumplace:= c-1*xn+j;
    from drum(P);

2)  drumplace:= c-(1-1)*xn;
    from drum(C);

3)  for p:= 1 step 1 until m do
      begin drumplace:= c-p*xn+j;
          from drum(P);
          D[p]:= P[1] end;

```

Zadanie 5.69

```

begin comment Rozwiazanie ukladu n algebraicznych rownan liniowych;
integer n;  n:= inone;
begin array C,D[1:n+1],X[1:n],P[1:1];
integer c,i,j,k,m,p,q;  real max,r;
c:= drumplace;
for i:= 1 step 1 until n do

```

```

begin input(C); to drum(C) end wprowadzania macierzy wspolczyn
                                nikow A na beben;

for j:= 1 step 1 until n do
    X[j]:= j; comment Tablica X bedzie przejsciowo przechowywala
                                wskazniki niewiadomych dla kolejnych kolumn macierzy
                                wspolczynnika A a potem same niewiadome;

for i:= 1 step 1 until n do
    begin drumplace:= c-i×n;

        from drum(P); comment pobranie z bebn A[i,i];

        max:= P[1]; p:= q:= 1;

        for j:= 1 step 1 until n do
            begin drumplace:= c-(j-1)×(n+1);

                from drum(C);

                for k:= 1 step 1 until n do
                    if abs(C[k])>abs(max) then

                        begin p:= j; q:= k; max:= C[k] end end;

                comment zostal wybrany wspolczynnik najwiekszy co do wartosci
                                bezwzglednej;

                if max=0 then

                    begin writetext({<Uklad,zdegenerowany>}); go to KONIEC end;

                comment teraz nastapi zamiana wierszy i kolumn, tak aby
                                najwiekszy wspolczynnik stal sie wyrazem A[i,i];

                k:= c-(p-1)×(n+1); m:= c-(i-1)×(n+1);

                drumplace:= k; from drum(C);

                drumplace:= m; from drum(D);

                drumplace:= k; to drum(D);

                drumplace:= m; to drum(C); comment koniec zamiany wierszy;

                for j:= 1 step 1 until n do
                    begin p:= c-j×(n+1); k:= p+q; p:= p+i;

```

```

    drumplace:= k; from drum(P); r:= P[1];
    drumplace:= p; from drum(P);
    drumplace:= k; to drum(P); P[1]:= r;
    drumplace:= p; to drum(P) end zamiany kolumn;
k:= X[q]; X[q]:= X[i]; X[i]:= k; comment zamiana
                                wskaznikow niewiadomych;

drumplace:= m; from drum(C);
for k:= i+1 step 1 until n+1 do C[k]:= C[k]/max;
drumplace:= m; to drum(C); comment przekształcenie
                                i-go wiersza;

for j:= i+1 step 1 until n do
    begin k:= c-(j-1)×(n+1);
        drumplace:= k; from drum(D);
        for p:= i+1 step 1 until n+1 do
            D[p]:= D[p]-D[i]×C[p];
        drumplace:= k; to drum(D) end przekształcenia
                                j-go wiersza;
    end eliminacji i-go wiersza i i-ej kolumny;
comment zakonczono eliminacje, nastapi obliczenie niewiadomych
początkowo na pierwszych n miejscach tablicy D a po przerzuceniu
tablicy X na pierwsze n miejsc tablicy C niewiadome - już w pier
wotnej kolejnosci - beda zawarte w tablicy X;
for i:= n step -1 until 1 do
    begin drumplace:= c-(i-1)×(n+1); from drum(C);
        D[i]:= C[n+1];
        for k:= i+1 step 1 until n do
            D[i]:= D[i]-C[k]×D[k] end liczenia niewiadomych;
    for i:= 1 step 1 until n do
        C[i]:= X[i];
    for i:= 1 step 1 until n do

```

```

X[C[i]]:= D[i]; comment uporządkowanie niewiadomych
                    wg pierwotnej kolejności;

outer; outer; outer;

outtext({Rozwiązanie układu równan:
});

for i:= 1 step 1 until n do
    output({+n.ddddd10+d},X[i],outchar(if i=5*(i:5) then 64 else 30));

outer; outer; outer;

KONIEC: end end programu;

```

Zadanie 5.71

Wystarczy na początku zamiast `c:= drumplace` dać `c:= 4095`, a potem zmienić wszędzie `to drum` na `to buf` i `from drum` na `from buf`.

Można jeszcze uprościć nieco program, deklarując zamiast tablicy `P` zmienną `P` typu real i wprowadzając potem w związku z tym następujące zmiany:

zamiast:	wprowadzamy:
<code>max:= P[1];</code>	<code>max:= P;</code>
<code>from drum(P); r:= P[1];</code>	<code>from buf(r);</code>
<code>P[1]:= r; drumplace:= p; to drum(P) <u>end</u></code>	<code>drumplace:= p;</code>
	<code>to buf(r) <u>end</u></code>

Takim programem można liczyć układy aż do 63 równań i tyluż niewiadomych ($63 * 64 = 4032 < 4096$).


```

    else begin drumplace:= d-(j-b-1)×(n+1);
              from drum(C) end;
    for k:= 1 step 1 until n do
      if abs(C[k])>abs(max) then
        begin p:= j; q:= k; max:=C[k] end end;
    comment zostal wybrany wspolczynnik najwiekszy co do wartosci
    bezwzglednej;
    if max=0 then begin writetext(⟨<Uklad,zdegenerowany⟩);
    go to KONIEC end;
    comment Teraz nastapi zamiana wierszy i kolumn tak, aby naj-
    wiekszy wspolczynnik stal sie A[i,1];
    k:= if p<b then c-(p-1)×(n+1) else d-(p-b-1)×(n+1);
    m:= if i<b then c-(i-1)×(n+1) else d-(i-b-1)×(n+1);
    drumplace:= k; if p<b then from buf(C) else from drum(C);
    drumplace:= m; if i<b then from buf(D) else from drum(D);
    drumplace:= k; if p<b then to buf(D) else to drum(D);
    drumplace:= m; if i<b then to buf(C) else to drum(C);
    comment koniec zamiany wierszy p-go z i-tym;
    for j:= 1 step 1 until b do
      begin p:= c-j×(n+1); k:= p+q; p:= p+i;
        drumplace:= k; from buf(Q); r:= Q;
        drumplace:= p; from buf(Q);
        drumplace:= k; to buf(Q);
        drumplace:= p; to buf(r) end;
    for j:= b+1 step 1 until n do
      begin p:= d-(j-b)×(n+1); k:= p+q; p:= p+i;
        drumplace:= k; from drum(P); r:= P[1];
        drumplace:= p; from drum(P);

```

```

drumplace:= k; to drum(P); P[1]:= r;
drumplace:= p; to drum(P) end;
comment koniec zamiany kolumn q-tej z i-ta;
k:= X[q]; X[q]:= X[i]; X[i]:= k;
comment dokonano zamiany wskaznikow niewiadomych, koniecznej
ze wzgledu na zamiane kolumn, a teraz rozpoczyna sie elimi-
nacja i-go wiersza i i-ej kolumny;
drumplace:= m; if i<b then from buf(C) else from drum(C);
for k:= i+1 step 1 until n+1 do C[k]:= C[k]/max;
drumplace:= m; if i<b then to buf(C) else to drum(C);
for j:= i+1 step 1 until n do
begin k:= if j<b then c-(j-1)×(n+1) else d-(j-b-1)×(n+1);
drumplace:= k; if j<b then from buf(D)
else from drum(D);
for p:= i+1 step 1 until n+1 do
D[p]:= D[p]-D[i]×C[p];
drumplace:= k; if j<b then to buf(D)
else to drum(D) end
end eliminacji i instrukcji for i:= 1 step 1 until n;
comment Teraz nastapi obliczenie niewiadomych poczatkowo na
pierwszych n miejscach tablicy D, a po przerzuceniu tablicy X
na pierwsze n miejsc tablicy C niewiadome - juz w pierwotnej
kolejnosci - beda zawarte w tablicy X;
for i:= n step -1 until 1 do
begin drumplace:= if i<b then c-(i-1)×(n+1) else d-(i-b-1)×(n+1);
if i<b then from buf(C) else from drum(C);
D[i]:= C[n+1];
for k:= i+1 step 1 until n do

```

```

D[i]:= D[i]-C[k]×D[k] end liczenia niewiadomych;
for i:= 1 step 1 until n do C[i]:= X[i];
for i:= 1 step 1 until n do X[C[i]]:= D[i];
comment Teraz nastąpi druk obliczonych niewiadomych;
outer; outer; outer;
outtext(⟨<Rozwiązanie układu równan:
⟩); for i:= 1 step 1 until n do
    output(⟨+n.dddddn+d⟩,X[i],outchar(if i=5×(i:5) then 64 else 30));
outer; outer; outer;
KONIEC: end end programu;

```

Zadanie 5.74

```

begin integer i; integer array A[1:512];
for i:= 1 step 1 until 512 do A[i]:= i;
drumplace:= 3000; to buf(A);
to car(39,10,1,3000) end;

```

Zadanie 5.75

```

begin integer i,c; array A[1:512];
from car(28,8,4,4095);
c:= drumplace;
for i:= 0,1,2,3 do begin
    drumplace:= 2559+i×512; from buf(A);
    drumplace:= c-i×512; to drum(A) end end;

```

Przepisywanie do pamięci operacyjnej należało rozbić na części, ponieważ pojemność pamięci operacyjnej nie zezwalała na wprowadzenie do niej tablic o więcej niż 500-600 wyrazach.

Zadanie 5.78

```

begin integer i,c,k; integer array A[1000:1499];
for i:= 1000 step 1 until 1499 do A[i]:= i;
c:= drumplace;
gierdrum(k<<corbuf,k);
begin array corbuf[1:k];
drumplace:= c;
from drum(corbuf);
gierproc(corbuf[2],A,4000,true) end end;

```

W przypadku stosowania translatora GIER ALGOL III BUF powyższy program można zastąpić przez:

```

begin integer i; integer array A[1000:1499];
for i:= 1000 step 1 until 1499 do A[i]:= i;
drumplace:= 4000; to buf(A) end;

```

albo

```

begin integer i; drumplace:= 4000;
for i:= 1000 step 1 until 1499 do to buf(i) end;

```

Drugi z tych wariantów nie wymaga rezerwacji tylu miejsc w pamięci operacyjnej maszyny co pierwszy i dlatego należy go uznać za lepszy.

Zadanie 5.80

```

begin integer c,i,k,b; array A[1:512];

```

```

c:= drumplace;
gierdrum(⟨⟨corbuf⟩,k); gierdrum(⟨⟨bufcar⟩,b);
begin array corbuf[1:k],bufcar[1:b];
drumplace:= c;
from drum(corbuf); from drum(bufcar); c:= drumplace;
gierproc(bufcar[2],28,8,4,4095,false);
for i:= 2559,3071,3583,4095 do begin
    gierproc(corbuf[2],A,i,false); to drum(A) end
end end;

```

Przepisywanie do pamięci operacyjnej należało rozbić na części, ponieważ pojemność pamięci operacyjnej nie zezwalała na wprowadzenie do niej tablic o więcej niż 500-600 wyrazach.

Zadanie 6.18

Wypełniając żądane zadanie na maszynie GIER, powinniśmy otrzymać na maszynie do pisania następujący zapis:

```

S= 2,
x,y= 0,-2, S= 3,
h= 0.1, S= 4,
x= .0000 y= -2.0000 R= 8.450010+2 S= 5,
x= 3.268410-2 y= -1.1007 R= 4.748910-1 S= 3,
h= 0.01, S= 5,
x= 2.893010-2 y= -1.0619 R= 1.443510-1 S= 3,
h= 0.001, S= 5,
x= -5.398610-2 y= -1.0599 R= 1.212210-1 S= 3,
h= 10-4, S= 5,
x= -6.965810-1 y= -9.704910-1 R= 2.595310-5 S= 3,
h= 10-5, S= 5,

```

$$x = -6.9979_{10^{-1}} \quad y = -9.6958_{10^{-1}} \quad R = 2.0531_{10^{-7}} \quad S = 3,$$

$$h = 10^{-6}, \quad S = 5,$$

$$x = -7.0006_{10^{-1}} \quad y = -9.6951_{10^{-1}} \quad R = 3.2895_{10^{-9}} \quad S = 3,$$

$$h = 10^{-7}, \quad S = 5,$$

$$x = -7.0010_{10^{-1}} \quad y = -9.6950_{10^{-1}} \quad R = .0360_{10^{-9}} \quad S = 3,$$

$$h = 10^{-8}, \quad S = 5,$$

$$x = -7.0010_{10^{-1}} \quad y = -9.6950_{10^{-1}} \quad R = .0165_{10^{-9}} \quad S =$$

Ponieważ wartość funkcji R jest wystarczająco bliska zera, a wartości x i y ustaliły się, rozwiązaniem danego układu jest

$$x = -0.70010, \quad y = -0.96950.$$

Wychodząc z punktów: $x=4, y=4$ lub $x=-4, y=4$ lub $x=y=0$ otrzymujemy

$$x = -1.2918, \quad y = -0.41332, \quad R = 1.5471$$

Nie jest to rozwiązaniem, ponieważ w punkcie tym funkcja R - aczkolwiek posiada minimum - nie osiąga wartości zero (lub w granicach błędów zaokrąglenia wartości bliskiej zero). Minimum w tym punkcie powstało na skutek tego, że krzywe, dane przez poszczególne równania, w pobliżu punktu $x = -1.2918, y = -0.41332$ zbliżyły się znacznie do siebie, ale się nie przecięły, ani nie doszły do styczności.

Zadanie 6.20

Po przedrukowaniu informacji wyperforowanych na taśmie papierowej na flexowriterze otrzymuje się następujący tekst:

stack

156880800

block

points

314 8

316 26

317 25

317 29

variables

-

18

0

.600₁₀ 0-.594₁₀ 2

7

4

-1

-1

array

4 × 7

0 0 0 0 0

0 0 1 2 3

4 5 6 7 2

4 6 8 10 12

14 3 6 9 12

15 18 21

array

7

7776 248832 1889568 7962624 24300000

60466176 130691232

stack end

Liczba wydrukowana bezpośrednio po stack jest wartością zmiennej b zadeklarowanej jako own. Po niej następuje otwarcie bloku zewnętrznego i wydrukowanie czterech punktów charakterystycznych dla tego bloku: wejście do ciała procedury MN, oraz etykiety L1, L2, Q. Cały blok wewnętrzny został usunięty z pamięci operacyjnej po jego wykonaniu i dojściu do jego end. Po punktach charakterystycznych zostały wydrukowa-

ne wartości zmiennych, które najłatwiej rozpoznać od końca: ostatnią jest wartość zmiennej booleowskiej b2 (false, co jest wydrukowane jako -1), przed nią jest wartość zmiennej b1 (również false jako -1), a jeszcze przed tym kolejno j=4, i=7 (wartości nieokreślone, ponieważ obie zmienne kontrolowane po zakończeniu instrukcji "dla" są nieokreślone), f (wartość przypadkowa, ponieważ f nie zostało obliczone na skutek pojawienia się sygnału "spill" w czasie liczenia 1/0), d=.600₁₀, a=0, a przed nimi wartość dwu zmiennych roboczych. Po zmiennych następują wartości wyrazów tablic A i C i na tym wyprowadzenie się kończy. Wywołania procedur MN i PR nie ma, gdyż po ich zakończeniu zostały z pamięci operacyjnej usunięte.

Zadanie 6,22

Otrzymuje się:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40