

Języki formalne i techniki translacji

Laboratorium - Projekt - Testy

program0.imp

```
1  VAR
2      a b
3  BEGIN
4      READ a;
5      WHILE a > 0 DO
6          b := a / 2;
7          b := 2 * b;
8          IF a > b THEN WRITE 1;
9          ELSE WRITE 0;
10         ENDIF
11         a := a / 2;
12     ENDWHILE
13 END
```

program1.imp

```
1  ( sito Eratostenesa )
2  VAR
3      n j sito[100]
4  BEGIN
5      n := 100-1;
6      FOR i FROM n DOWNTO 2 DO
7          sito[i] := 1;
8      ENDFOR
9      FOR i FROM 2 TO n DO
10         IF sito[i] <> 0 THEN
11             j := i + i;
12             WHILE j <= n DO
13                 sito[j] := 0;
14                 j := j + i;
15             ENDWHILE
16             WRITE i;
17         ENDIF
18     ENDFOR
19 END
```

program2.imp

```
1  ( Rozkład liczby na czynniki pierwsze )
2  VAR
3      n m reszta potega dzielnik
4  BEGIN
5      READ n;
6      dzielnik := 2;
7      m := dzielnik * dzielnik;
8      WHILE n >= m DO
9          potega := 0;
10         reszta := n % dzielnik;
11         WHILE reszta = 0 DO
```

```

12         n := n / dzielnik;
13         potega := potega + 1;
14         reszta := n % dzielnik;
15     ENDWHILE
16     IF potega > 0 THEN ( czy znaleziono dzielnik )
17         WRITE dzielnik;
18         WRITE potega;
19     ELSE
20         dzielnik := dzielnik + 1;
21         m := dzielnik * dzielnik;
22     ENDIF
23 ENDWHILE
24 IF n <> 1 THEN ( ostatni dzielnik )
25     WRITE n;
26     WRITE 1;
27 ENDIF
28 END

```

0-div-mod.imp

```

1 ( div-mod.imp
2   1 0
3   1 0 0 0
4 )
5 VAR
6     a b c
7 BEGIN
8     READ a;
9     READ b;
10    c := a / a;
11    WRITE c;
12    c := a / b;
13    WRITE c;
14    c := a % a;
15    WRITE c;
16    c := a % b;
17    WRITE c;
18 END

```

1-numbers.imp

```

1 ( numbers.imp - liczby )
2 VAR
3     a b c t[7] d e f g h i j tab[6]
4 BEGIN
5     WRITE 0;
6     WRITE 1;
7     WRITE 2;
8     WRITE 10;
9     WRITE 100;
10    WRITE 10000;
11    WRITE 1234567890;
12
13    a := 1234566543;

```

```

14      b := 677777177;
15      c := 15;
16      t[2] := 555555555;
17      d := 8888;
18      tab[4] := 11;
19      t[0] := 999;
20      e := 1111111111;
21      tab[0] := 7777;
22      f := 2048;
23      g := 123;
24      t[3] := t[0];
25      tab[5] := a;
26      tab[5] := tab[0] / tab[4];
27      t[5] := tab[0];
28
29      READ h;
30      i := 1;
31      j := h + c;
32
33      WRITE j; ( j = h + 15 )
34      WRITE c; ( c = 15 )
35      WRITE t[3]; ( 999 )
36      WRITE t[2]; ( 555555555 )
37      WRITE t[5]; ( 7777 )
38      WRITE t[0]; ( 999 )
39      WRITE tab[4]; ( 11 )
40      WRITE tab[5]; ( 707 )
41      WRITE tab[0]; ( 7777 )
42  END

```

2-fib.imp

```

1  ( Fibonacci 26
2  ? 1
3  > 121393
4  )
5  VAR
6      tab[1234567890] a b c d e f g h i j k l m n o p q r s t u v w x y z
7  BEGIN
8      READ tab[12345];
9      a:=tab[12345];
10     b:=a;
11     c:=b+a;
12     d:=c+b;
13     e:=d+c;
14     f:=e+d;
15     g:=f+e;
16     h:=g+f;
17     i:=h+g;
18     j:=i+h;
19     k:=j+i;
20     l:=k+j;
21     m:=l+k;
22     n:=m+l;
23     o:=n+m;

```

```

24  p:=o+n;
25  q:=p+o;
26  r:=q+p;
27  s:=r+q;
28  t:=s+r;
29  u:=t+s;
30  v:=u+t;
31  w:=v+u;
32  x:=w+v;
33  y:=x+w;
34  z:=y+x;
35  tab[a]:=z;
36  WRITE tab[a];
37  END

```

3-fib-factorial.imp

```

1  ( Silnia + Fibonacci
2  ? 20
3  > 2432902008176640000
4  > 17711
5  )
6  VAR
7      f[101] s[101] i[101] n k l
8  BEGIN
9      READ n;
10     f[0]:=1;
11     s[0]:=1;
12     i[0]:=0;
13     FOR j FROM 1 TO n DO
14         k:=j-1;
15         l:=k-1;
16         i[j]:=i[k]+1;
17         f[j]:=f[k]+f[l];
18         s[j]:=s[k]*i[j];
19     ENDFOR
20     WRITE s[n];
21     WRITE f[n];
22  END

```

4-factorial.imp

```

1  ( Silnia
2  ? 20
3  > 2432902008176640000
4  )
5  VAR
6      s[101] n m a j
7  BEGIN
8      READ n;
9      s[0]:=1;
10     m:=n;
11     FOR i FROM 1 TO m DO
12         a:=i%2;

```

```

13             j:=i-1;
14             IF a=1 THEN
15                 s[i]:=s[j]*m;
16             ELSE
17                 s[i]:=m*s[j];
18             ENDIF
19             m:=m-1;
20         ENDFOR
21         WRITE s[n];
22     END

```

5-tab.imp

```

1  ( tab.imp )
2  VAR
3      n j ta[25] tb[25] tc[25]
4  BEGIN
5      n := 25 - 1;
6      tc[0] := n;
7      tc[n] := n - n;
8      FOR i FROM tc[0] DOWNTO tc[n] DO
9          ta[i] := i;
10         tb[i] := n - i;
11     ENDFOR
12     FOR i FROM tc[n] TO tc[0] DO
13         tc[i] := ta[i] * tb[i];
14     ENDFOR
15     FOR i FROM 0 TO n DO
16         WRITE tc[i];
17     ENDFOR
18 END

```

6-mod-mult.imp

```

1  ( a ^ b mod c
2  ? 1234567890
3  ? 1234567890987654321
4  ? 987654321
5  > 674106858
6  )
7  VAR
8      a b c wynik pot wybor
9  BEGIN
10     READ a;
11     READ b;
12     READ c;
13     wynik:=1;
14     pot:=a%c;
15     WHILE b>0 DO
16         wybor:=b%2;
17         IF wybor=1 THEN
18             wynik:=wynik*pot;
19             wynik:=wynik%c;
20         ENDIF

```

```

21         b:=b/2;
22         pot:=pot*pot;
23         pot:=pot%c;
24     ENDWHILE
25     WRITE wynik;
26 END

```

7-loopiii.imp

```

1  ( loopiii.imp - zagniezdzone petle
2      0 0 0
3      31000 40900 2222010
4
5      1 0 2
6      31001 40900 2222012
7  )
8  VAR
9      a b c
10 BEGIN
11     READ a;
12     READ b;
13     READ c;
14     FOR i FROM 111091 TO 111110 DO
15         FOR j FROM 209 DOWNTO 200 DO
16             FOR k FROM 11 TO 20 DO
17                 a := a + k;
18             ENDFOR
19             b := b + j;
20         ENDFOR
21         c := c + i;
22     ENDFOR
23     WRITE a;
24     WRITE b;
25     WRITE c;
26 END

```

8-for.imp

```

1  ( for.imp
2      12 23 34
3      507 4379 0
4  )
5  VAR
6      a b c
7  BEGIN
8      READ a;
9      READ b;
10     READ c;
11     FOR i FROM 9 DOWNTO 0 DO
12         FOR j FROM 0 TO i DO
13             FOR k FROM 0 TO j DO
14                 a := a + k;
15                 c := k * j;
16                 c := c + i;

```

```

17                                     b := b + c;
18                                 ENDFOR
19          ENDFOR
20      ENDFOR
21      WRITE a;
22      WRITE b;
23      WRITE c;
24  END

```

9-sort.imp

```

1  ( sort.imp
2  )
3  VAR
4      tab[22] x q w j k n m
5  BEGIN
6      n := 23;
7      m := n - 2;
8      q := 5;
9      w := 1;
10     (generowanie nieposortowanej tablicy)
11     FOR i FROM 0 TO m DO
12         w := w * q;
13         w := w % n;
14         tab[i] := w;
15     ENDFOR
16     (wypisywanie nieposortowanej tablicy)
17     FOR i FROM 0 TO m DO
18         WRITE tab[i];
19     ENDFOR
20     WRITE 1234567890;
21     (sortowanie)
22     FOR i FROM 1 TO m DO
23         x := tab[i];
24         j := i;
25         WHILE j > 0 DO
26             k := j - 1;
27             IF tab[k] > x THEN
28                 tab[j] := tab[k];
29                 j := j - 1;
30             ELSE
31                 k := j;
32                 j := 0;
33             ENDIF
34         ENDWHILE
35         tab[k] := x;
36     ENDFOR
37     (wypisywanie posortowanej tablicy)
38     FOR i FROM 0 TO m DO
39         WRITE tab[i];
40     ENDFOR
41  END

```