Języki formalne i techniki translacji Laboratorium - Projekt - Testy

```
program0.imp
1 VAR
      a b
2
3 BEGIN
    READ a;
5
      WHILE a > 0 DO
           b := a / 2;
6
7
           b := 2 * b;
8
           IF a > b THEN WRITE 1;
9
           ELSE WRITE 0;
           ENDIF
10
          a := a / 2;
11
      ENDWHILE
12
13 END
   program1.imp
1 ( sito Eratostenesa )
2 VAR
3
       n j sito[100]
4 BEGIN
5
      n := 100-1;
       FOR i FROM n DOWNTO 2 DO
6
7
           sito[i] := 1;
8
      ENDFOR
      FOR i FROM 2 TO n DO
9
           IF sito[i] <> 0 THEN
10
               j := i + i;
11
               WHILE j <= n DO
12
13
                    sito[j] := 0;
14
                    j := j + i;
15
               ENDWHILE
16
               WRITE i;
17
           ENDIF
      ENDFOR
18
19 END
   program2.imp
1 ( Rozklad liczby na czynniki pierwsze )
2 VAR
3
       n m reszta potega dzielnik
4 BEGIN
5
      READ n;
       dzielnik := 2;
7
       m := dzielnik * dzielnik;
      WHILE n >= m DO
8
9
           potega := 0;
           reszta := n % dzielnik;
10
           WHILE reszta = 0 DO
11
```

```
12
               n := n / dzielnik;
13
               potega := potega + 1;
               reszta := n % dzielnik;
14
15
          ENDWHILE
           IF potega > 0 THEN ( czy znaleziono dzielnik )
16
17
               WRITE dzielnik;
               WRITE potega;
18
19
           ELSE
               dzielnik := dzielnik + 1;
20
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
   O-div-mod.imp
1 ( div-mod.imp
 2 1 0
   1 0 0 0
 3
4 )
 5 VAR
6
           a b c
7 BEGIN
8
           READ a;
9
          READ b;
          c := a / a;
10
11
          WRITE c;
12
           c := a / b;
          WRITE c;
13
14
          c := a % a;
          WRITE c;
15
16
          c := a % b;
17
           WRITE c;
18 END
   1-numbers.imp
 1 ( numbers.imp - liczby )
2 VAR
           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;
          WRITE 1234567890;
11
12
13
        a := 1234566543;
```

```
14
            b := 677777177;
            c := 15;
15
            t[2] := 55555555;
16
            d := 8888;
17
            tab[4] := 11;
18
            t[0] := 999;
19
            e := 1111111111;
20
           tab[0] := 7777;
21
            f := 2048;
22
            g := 123;
23
24
            t[3] := t[0];
25
            tab[5] := a;
           tab[5] := tab[0] / tab[4];
26
27
            t[5] := tab[0];
28
29
            READ h;
30
            i := 1;
            j := h + c;
31
32
33
            WRITE j; ( j = h + 15 )
            WRITE c; (c = 15)
34
            WRITE t[3]; ( 999 )
35
            WRITE t[2]; ( 55555555 )
36
            WRITE t[5]; ( 7777 )
37
            WRITE t[0]; ( 999 )
38
            WRITE tab[4]; ( 11 )
39
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;
     j:=i+h;
18
19
     k := j + i;
20
     1 := k + j;
21
    m:=l+k;
22
    n:=m+1;
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;
     v:=u+t;
30
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;
      i[0]:=0;
12
13
      FOR j FROM 1 TO n DO
14
                    k := j-1;
15
            1 := k-1;
16
                    i[j]:=i[k]+1;
                    f[j]:=f[k]+f[1];
17
18
            s[j] := s[k] * i[j];
19
       ENDFOR
20
       WRITE s[n];
       WRITE f[n];
21
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
                             s[i]:=m*s[j];
17
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
            FOR i FROM tc[n] TO tc[0] DO
12
                    tc[i] := ta[i] * tb[i];
13
14
            ENDFOR
            FOR i FROM O TO n DO
15
                     WRITE tc[i];
16
17
            ENDFOR
18 END
   6-mod-mult.imp
1 (a ^ b mod c
2 ? 1234567890
3 ? 1234567890987654321
4 ? 987654321
5 > 674106858
6)
7 VAR
       a b c wynik pot wybor
8
9 BEGIN
10
       READ a;
       READ b;
11
12
       READ c;
13
       wynik:=1;
14
       pot:=a%c;
       WHILE b>0 DO
15
16
                     wybor:=b%2;
                     IF wybor=1 THEN
17
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
           READ a;
11
12
           READ b;
13
           READ c;
14
           FOR i FROM 111091 TO 111110 DO
15
                    FOR j FROM 209 DOWNTO 200 DO
                             FOR k FROM 11 TO 20 DO
16
                                     a := a + k;
17
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
   12 23 34
2
    507 4379 0
3
4 )
5 VAR
6
           a b c
7 BEGIN
8
           READ a;
9
           READ b;
10
           READ c;
           FOR i FROM 9 DOWNTO 0 DO
11
12
                    FOR j FROM O TO i DO
13
                             FOR k FROM O TO j DO
14
                                     a := a + k;
15
                                     c := k * j;
16
                                     c := c + i;
```

```
17
                                      b := b + c;
                              ENDFOR
18
                     ENDFOR
19
20
            ENDFOR
21
            WRITE a;
22
            WRITE b;
            WRITE c;
23
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)
            FOR i FROM O TO m DO \,
11
12
                     w := w * q;
13
                     w := w \% n;
14
                     tab[i] := w;
15
            ENDFOR
16
            (wypisywanie nieposortowanej tablicy)
            FOR i FROM O TO m DO
17
                     WRITE tab[i];
18
19
            ENDFOR
            WRITE 1234567890;
20
21
            (sortowanie)
            FOR i FROM 1 TO m DO
22
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
                                      k := j;
31
32
                                      j := 0;
33
                             ENDIF
34
                     ENDWHILE
35
                     tab[k] := x;
36
            ENDFOR
            (wypisywanie posortowanej tablicy)
37
            FOR i FROM O TO m DO
38
                     WRITE tab[i];
39
            ENDFOR
40
41 END
```