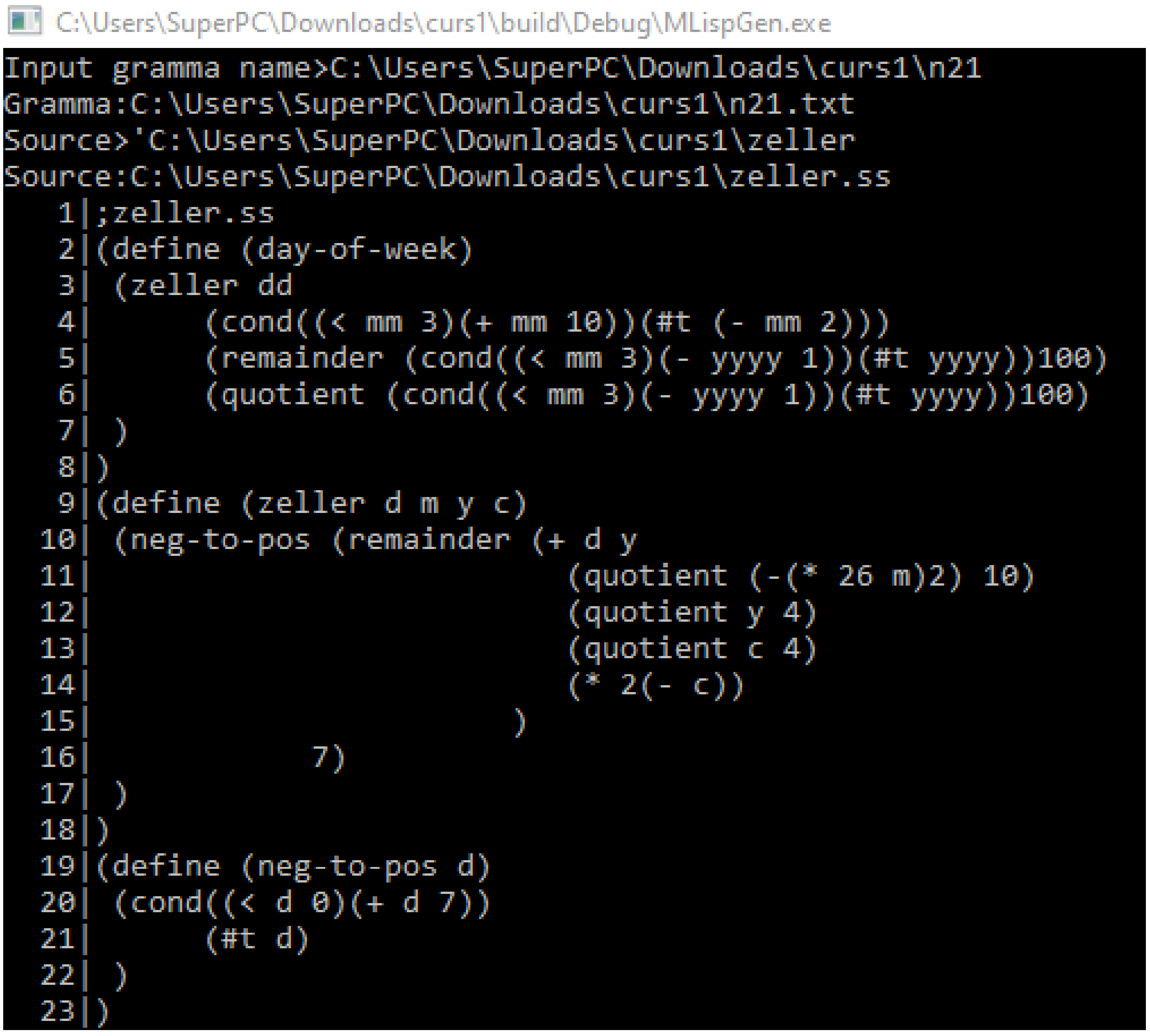
Студент: Пищик Е.С.
Группа: М8О-206Б-19
Номер по списку: 21

«СИСТЕМЫ ПРОГРАММИРОВАНИЯ»
Курсовая работа 2021.
Часть 1.

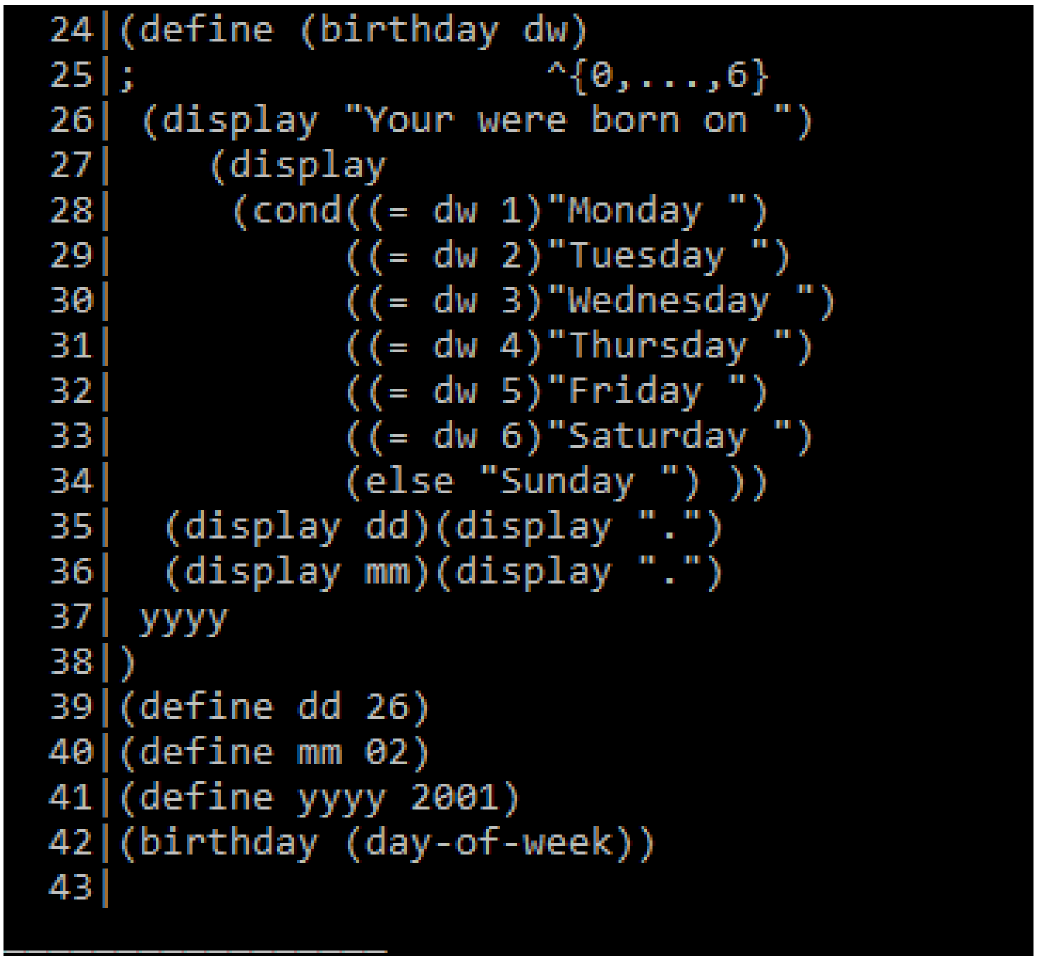
Перечень документов в отчете.
Вариант грамматики:21

Контрольная задача №1 – zeller.

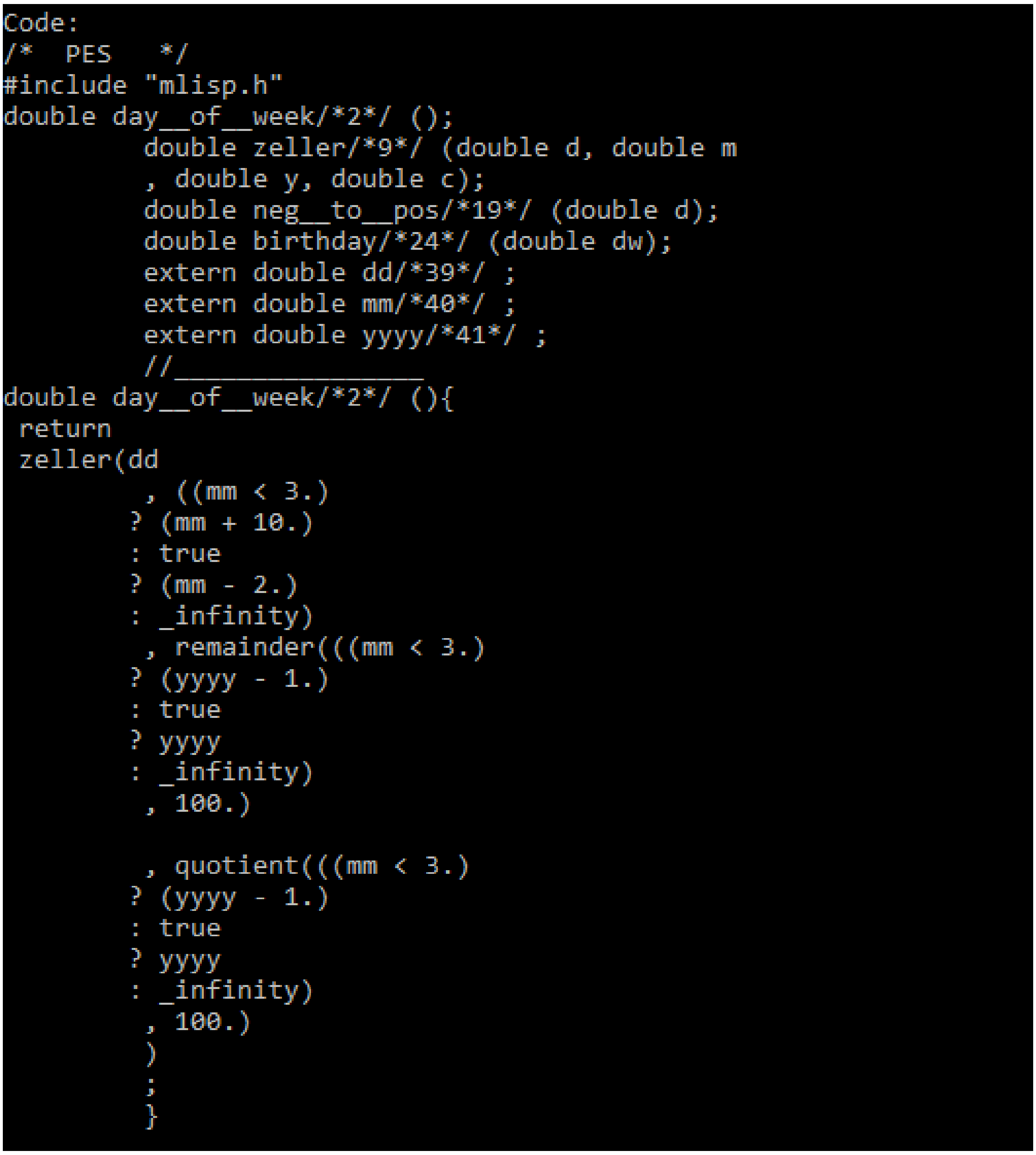
Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).
>



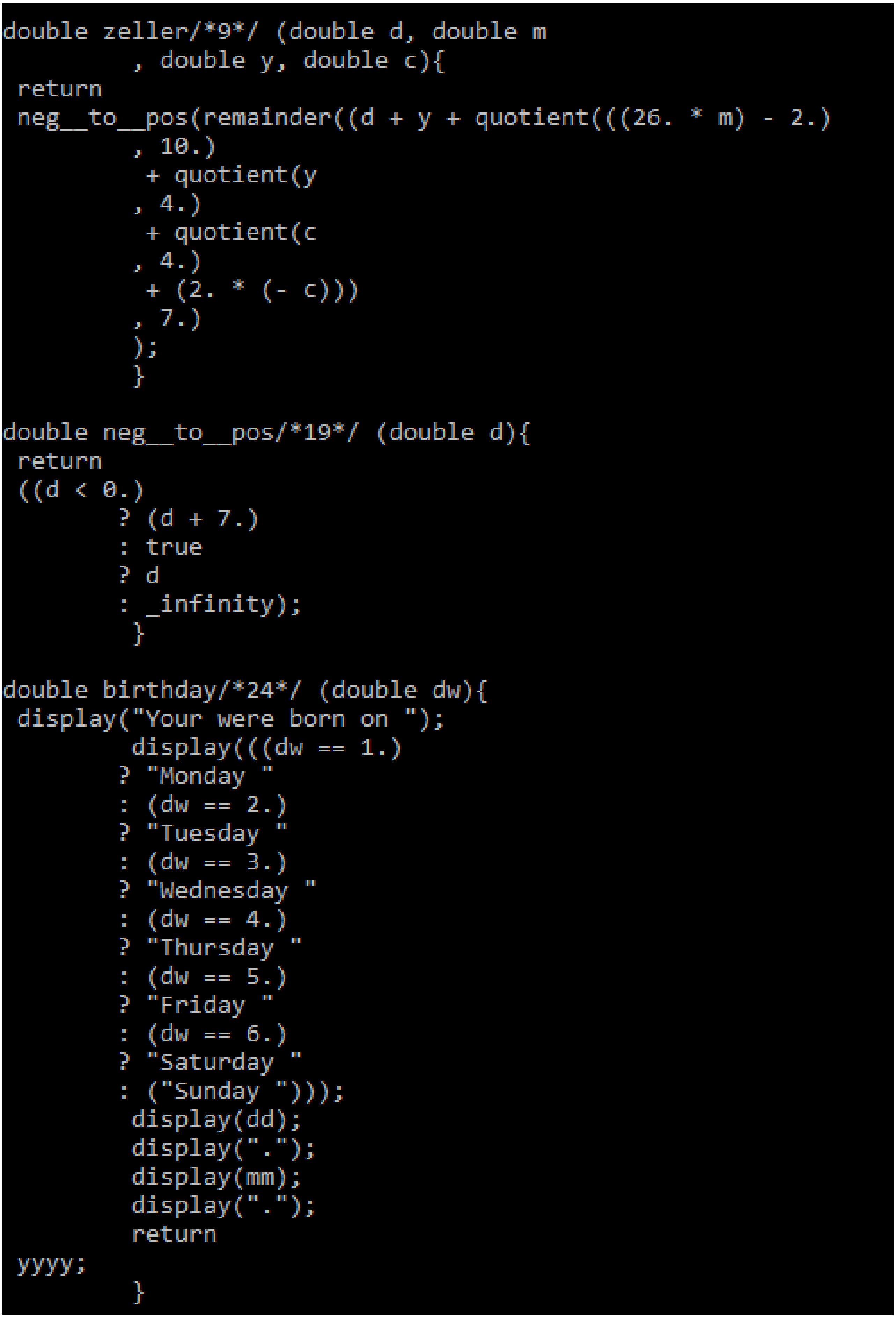
```
C:\Users\SuperPC\Downloads\curs1\build\Debug\MLispGen.exe
Input gramma name>C:\Users\SuperPC\Downloads\curs1\n21
Gramma:C:\Users\SuperPC\Downloads\curs1\n21.txt
Source>'C:\Users\SuperPC\Downloads\curs1\zeller
Source:C:\Users\SuperPC\Downloads\curs1\zeller.ss
   1|;zeller.ss
   2|(define (day-of-week)
   3| (zeller dd
   4|      (cond((< mm 3)(+ mm 10))(#t (- mm 2)))
   5|      (remainder (cond((< mm 3)(- yyyy 1))(#t yyyy))100)
   6|      (quotient (cond((< mm 3)(- yyyy 1))(#t yyyy))100)
   7| )
   8|)
   9|(define (zeller d m y c)
  10| (neg-to-pos (remainder (+ d y
  11|                          (quotient (-(* 26 m)2) 10)
  12|                          (quotient y 4)
  13|                          (quotient c 4)
  14|                          (* 2(- c))
  15|                       )
  16|            7)
  17| )
  18|)
  19|(define (neg-to-pos d)
  20| (cond((< d 0)(+ d 7))
  21|      (#t d)
  22| )
  23|)
```



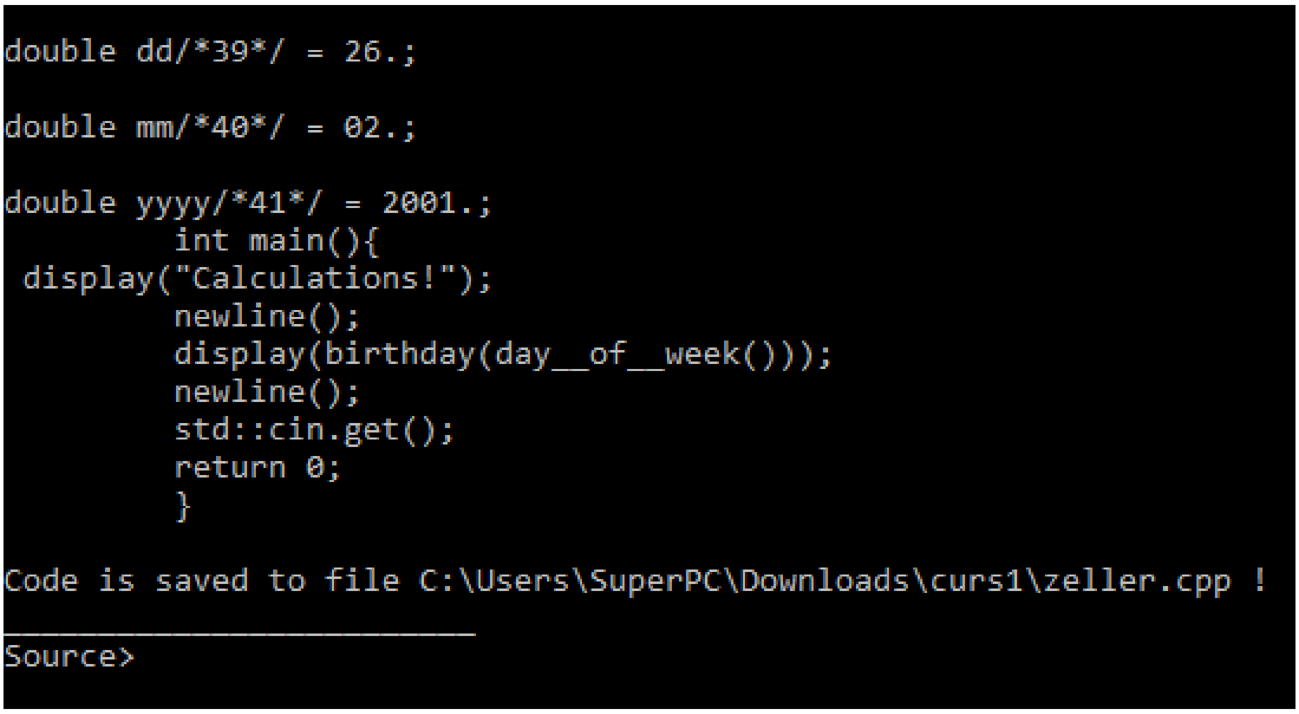
```scheme
(define (birthday dw)
;                  ^{0,...,6}
 (display "Your were born on ")
    (display
     (cond((= dw 1)"Monday ")
          ((= dw 2)"Tuesday ")
          ((= dw 3)"Wednesday ")
          ((= dw 4)"Thursday ")
          ((= dw 5)"Friday ")
          ((= dw 6)"Saturday ")
          (else "Sunday ") ))
  (display dd)(display ".")
  (display mm)(display ".")
 yyyy
)
(define dd 26)
(define mm 02)
(define yyyy 2001)
(birthday (day-of-week))
```



```
Code:
/*  PES   */
#include "mlisp.h"
double day__of__week/*2*/ ();
         double zeller/*9*/ (double d, double m
         , double y, double c);
         double neg__to__pos/*19*/ (double d);
         double birthday/*24*/ (double dw);
         extern double dd/*39*/ ;
         extern double mm/*40*/ ;
         extern double yyyy/*41*/ ;
         //________________
double day__of__week/*2*/ (){
 return
 zeller(dd
         , ((mm < 3.)
        ? (mm + 10.)
        : true
        ? (mm - 2.)
        : _infinity)
         , remainder(((mm < 3.)
        ? (yyyy - 1.)
        : true
        ? yyyy
        : _infinity)
         , 100.)

         , quotient(((mm < 3.)
        ? (yyyy - 1.)
        : true
        ? yyyy
        : _infinity)
         , 100.)
         )
         ;
         }
```



```
double zeller/*9*/ (double d, double m
         , double y, double c){
 return
 neg__to__pos(remainder((d + y + quotient(((26. * m) - 2.)
         , 10.)
          + quotient(y
         , 4.)
          + quotient(c
         , 4.)
          + (2. * (- c)))
         , 7.)
         );
         }

double neg__to__pos/*19*/ (double d){
 return
 ((d < 0.)
        ? (d + 7.)
        : true
        ? d
        : _infinity);
         }

double birthday/*24*/ (double dw){
 display("Your were born on ");
         display(((dw == 1.)
        ? "Monday "
        : (dw == 2.)
        ? "Tuesday "
        : (dw == 3.)
        ? "Wednesday "
        : (dw == 4.)
        ? "Thursday "
        : (dw == 5.)
        ? "Friday "
        : (dw == 6.)
        ? "Saturday "
        : ("Sunday ")));
         display(dd);
         display(".");
         display(mm);
         display(".");
         return
 yyyy;
         }
```



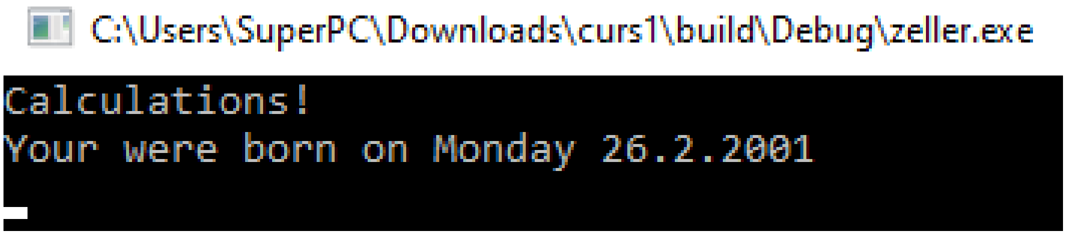
```
double dd/*39*/ = 26.;

double mm/*40*/ = 02.;

double yyyy/*41*/ = 2001.;
         int main(){
 display("Calculations!");
         newline();
         display(birthday(day__of__week()));
         newline();
         std::cin.get();
         return 0;
         }

Code is saved to file C:\Users\SuperPC\Downloads\curs1\zeller.cpp !
_________________________
Source>
```

Распечатка файла zeller.cpp .

```
>
/*  PES   */
#include "mlisp.h"
double day__of__week/*2*/ ();
        double zeller/*9*/ (double d, double m
        , double y, double c);
        double neg__to__pos/*19*/ (double d);
        double birthday/*24*/ (double dw);
        extern double dd/*39*/ ;
        extern double mm/*40*/ ;
        extern double yyyy/*41*/ ;
        //___________________
double day__of__week/*2*/ (){
 return
 zeller(dd
        , ((mm < 3.)
       ? (mm + 10.)
       : true
       ? (mm - 2.)
       : _infinity)
        , remainder(((mm < 3.)
       ? (yyyy - 1.)
       : true
       ? yyyy
       : _infinity)
        , 100.)
        , quotient(((mm < 3.)
       ? (yyyy - 1.)
       : true
       ? yyyy
       : _infinity)
        , 100.)
        )
        ;
        }

double zeller/*9*/ (double d, double m
        , double y, double c){
 return
 neg__to__pos(remainder((d + y + quotient(((26. * m) - 2.)
        , 10.)
         + quotient(y
        , 4.)
         + quotient(c
        , 4.)
         + (2. * (- c)))
        , 7.)
        );
        }

double neg__to__pos/*19*/ (double d){
 return
 ((d < 0.)
       ? (d + 7.)
       : true
       ? d
       : _infinity);
        }

double birthday/*24*/ (double dw){
 display("Your were born on ");
        display(((dw == 1.)
       ? "Monday "
       : (dw == 2.)
       ? "Tuesday "
       : (dw == 3.)
       ? "Wednesday "
```

```
      : (dw == 4.)
      ? "Thursday "
      : (dw == 5.)
      ? "Friday "
      : (dw == 6.)
      ? "Saturday "
      : ("Sunday ")));
       display(dd);
       display(".");
       display(mm);
       display(".");
       return
 yyyy;
       }

double dd/*39*/ = 26.;

double mm/*40*/ = 02.;

double yyyy/*41*/ = 2001.;
       int main(){
 display("Calculations!");
       newline();
       display(birthday(day__of__week()));
       newline();
       std::cin.get();
       return 0;
       }
```

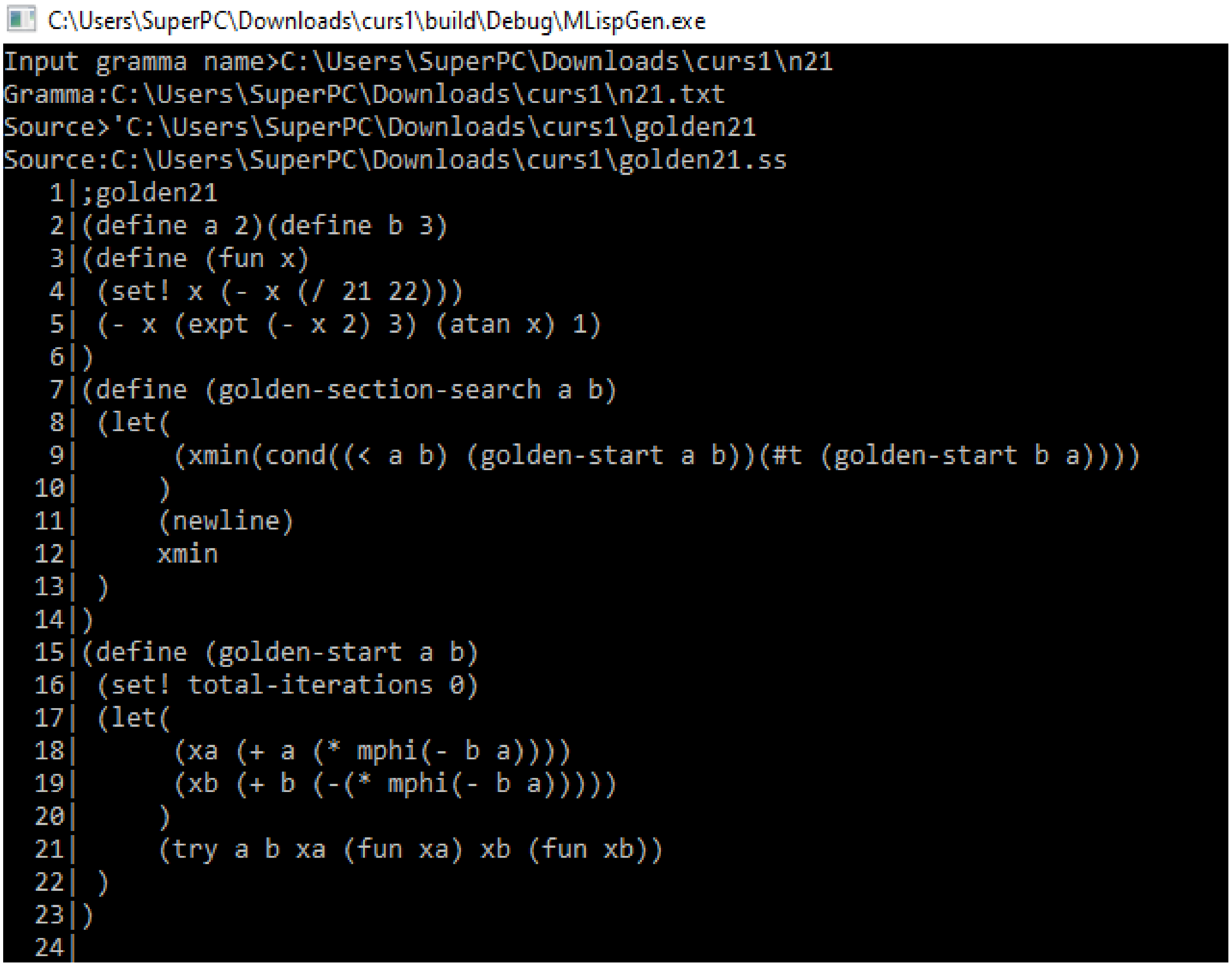
Скриншот запуска задачи на C++.
>



```
C:\Users\SuperPC\Downloads\curs1\build\Debug\zeller.exe
Calculations!
Your were born on Monday 26.2.2001
_
```

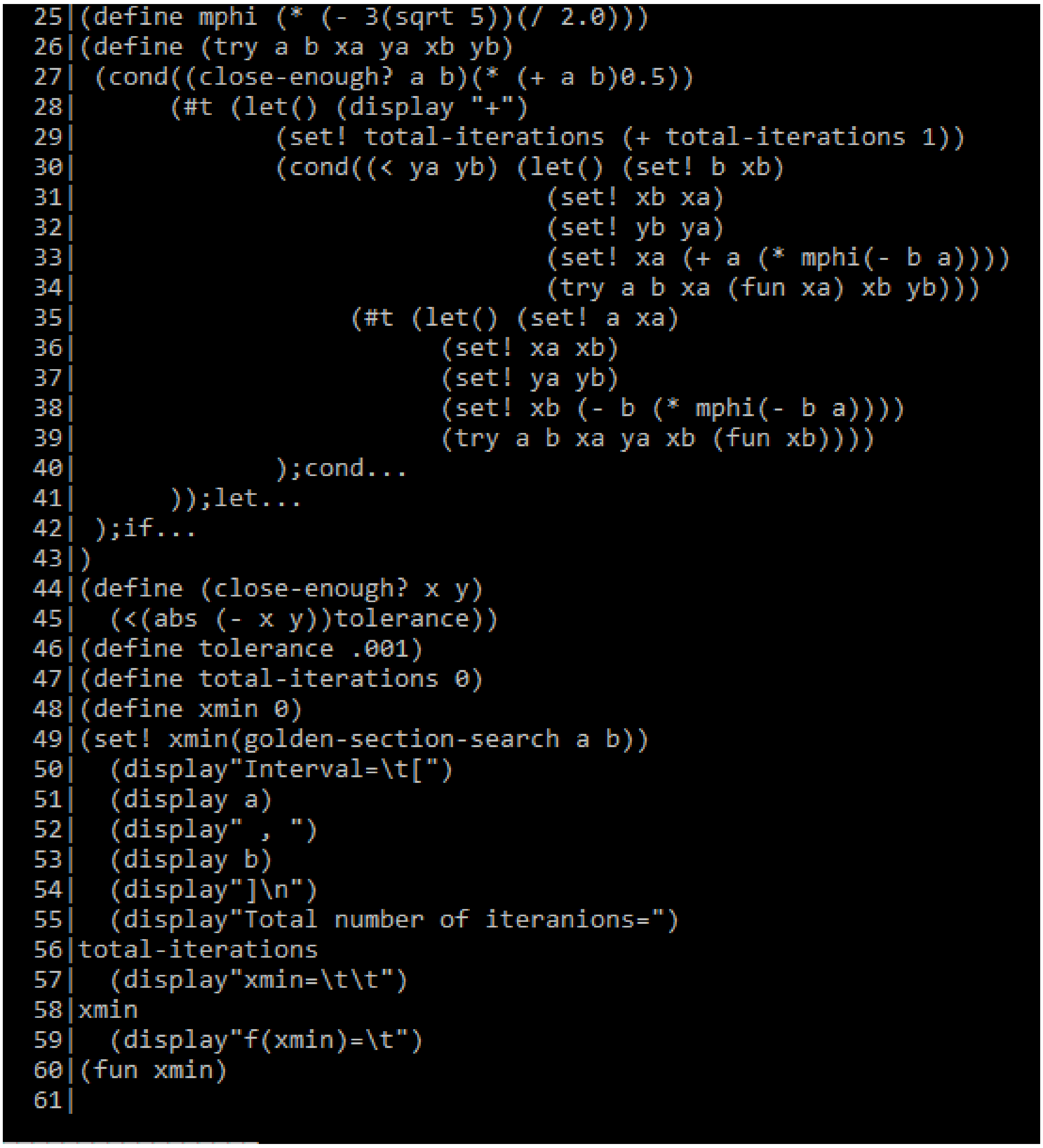
Контрольная задача №2 – golden21.

Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).
>

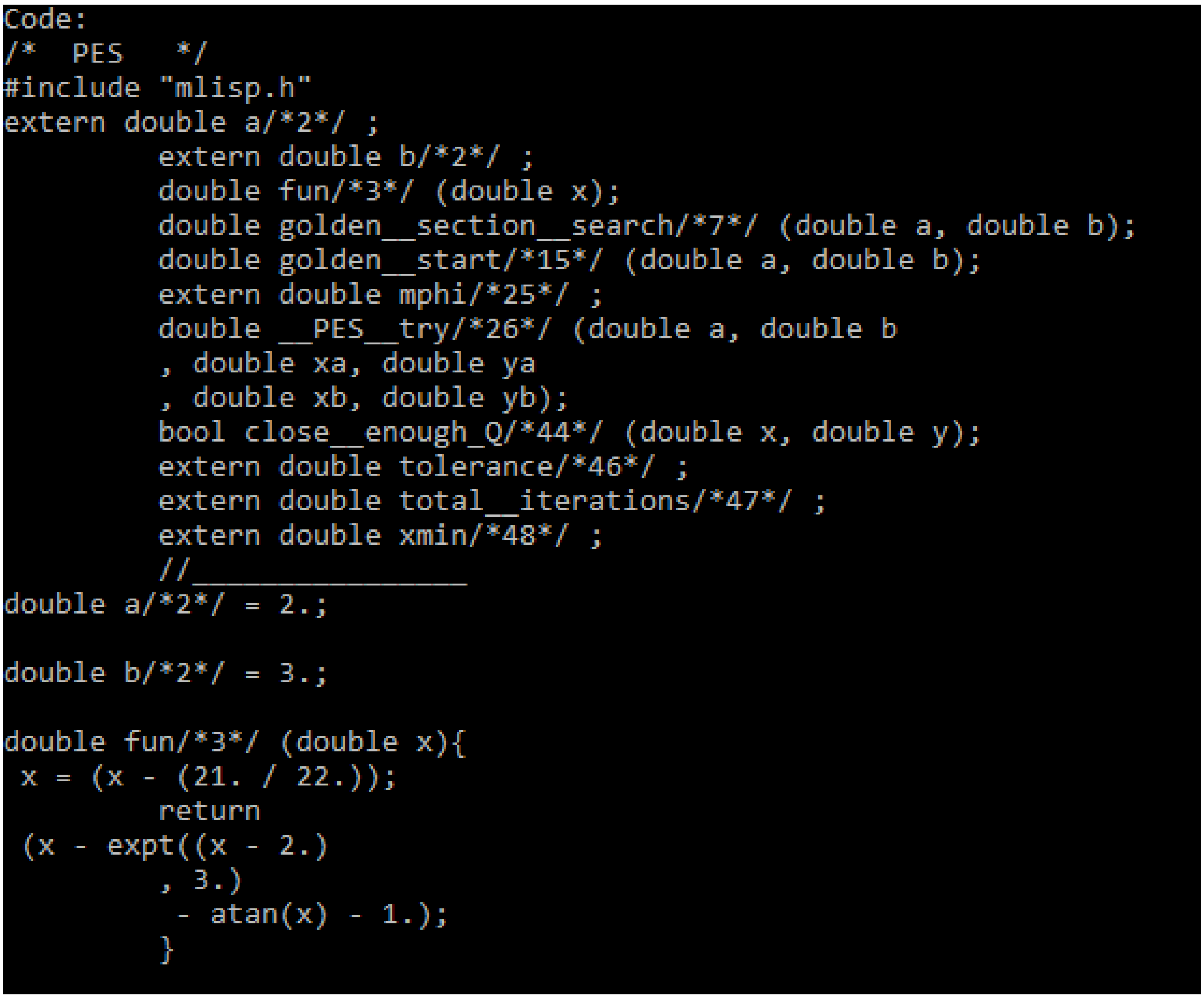


C:\Users\SuperPC\Downloads\curs1\build\Debug\MLispGen.exe

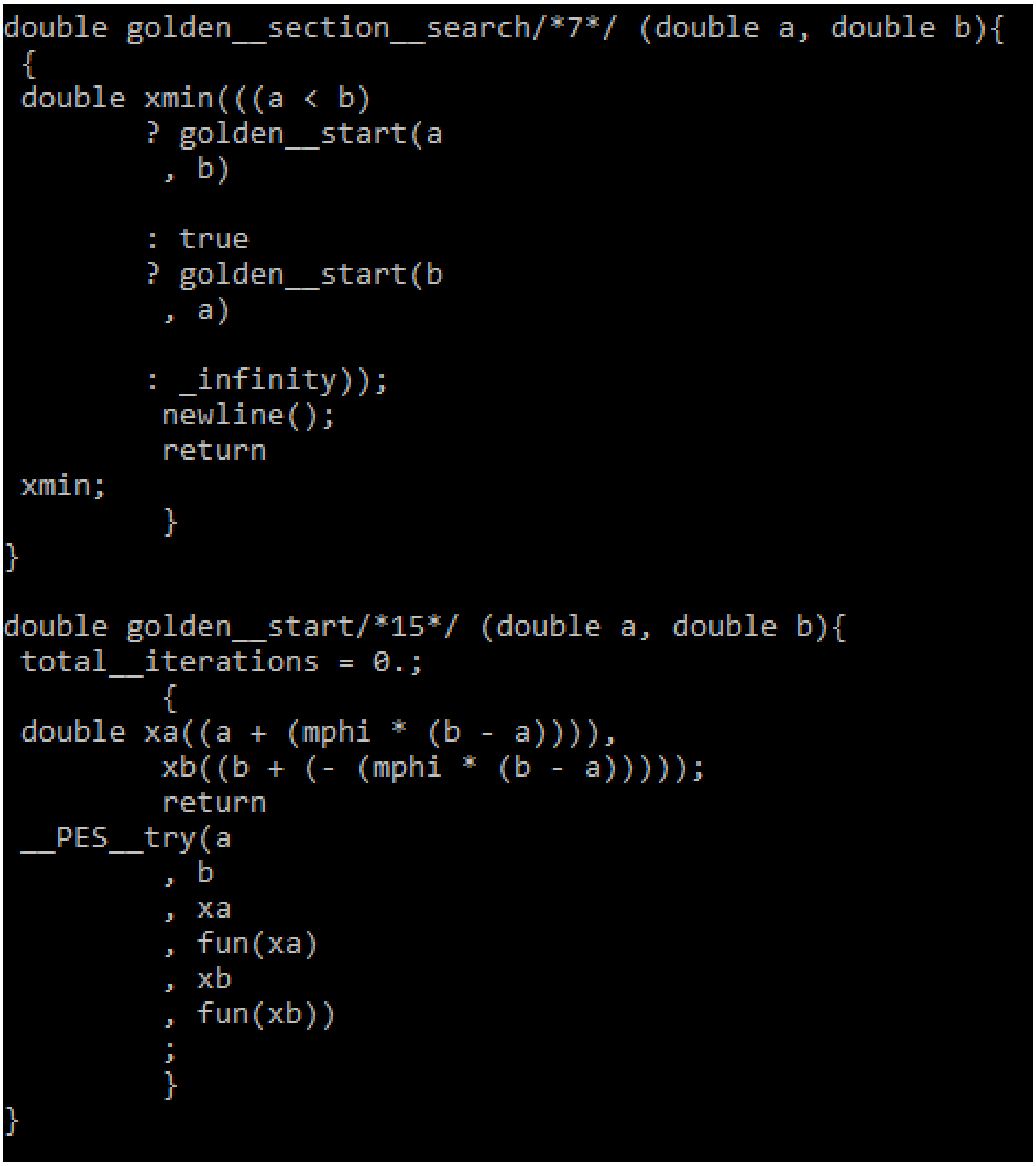
```
Input gramma name>C:\Users\SuperPC\Downloads\curs1\n21
Gramma:C:\Users\SuperPC\Downloads\curs1\n21.txt
Source>'C:\Users\SuperPC\Downloads\curs1\golden21
Source:C:\Users\SuperPC\Downloads\curs1\golden21.ss
;golden21
(define a 2)(define b 3)
(define (fun x)
 (set! x (- x (/ 21 22)))
 (- x (expt (- x 2) 3) (atan x) 1)
)
(define (golden-section-search a b)
 (let(
      (xmin(cond((< a b) (golden-start a b))(#t (golden-start b a))))
     )
     (newline)
     xmin
 )
)
(define (golden-start a b)
 (set! total-iterations 0)
 (let(
      (xa (+ a (* mphi(- b a))))
      (xb (+ b (-(* mphi(- b a)))))
     )
     (try a b xa (fun xa) xb (fun xb))
 )
)
```



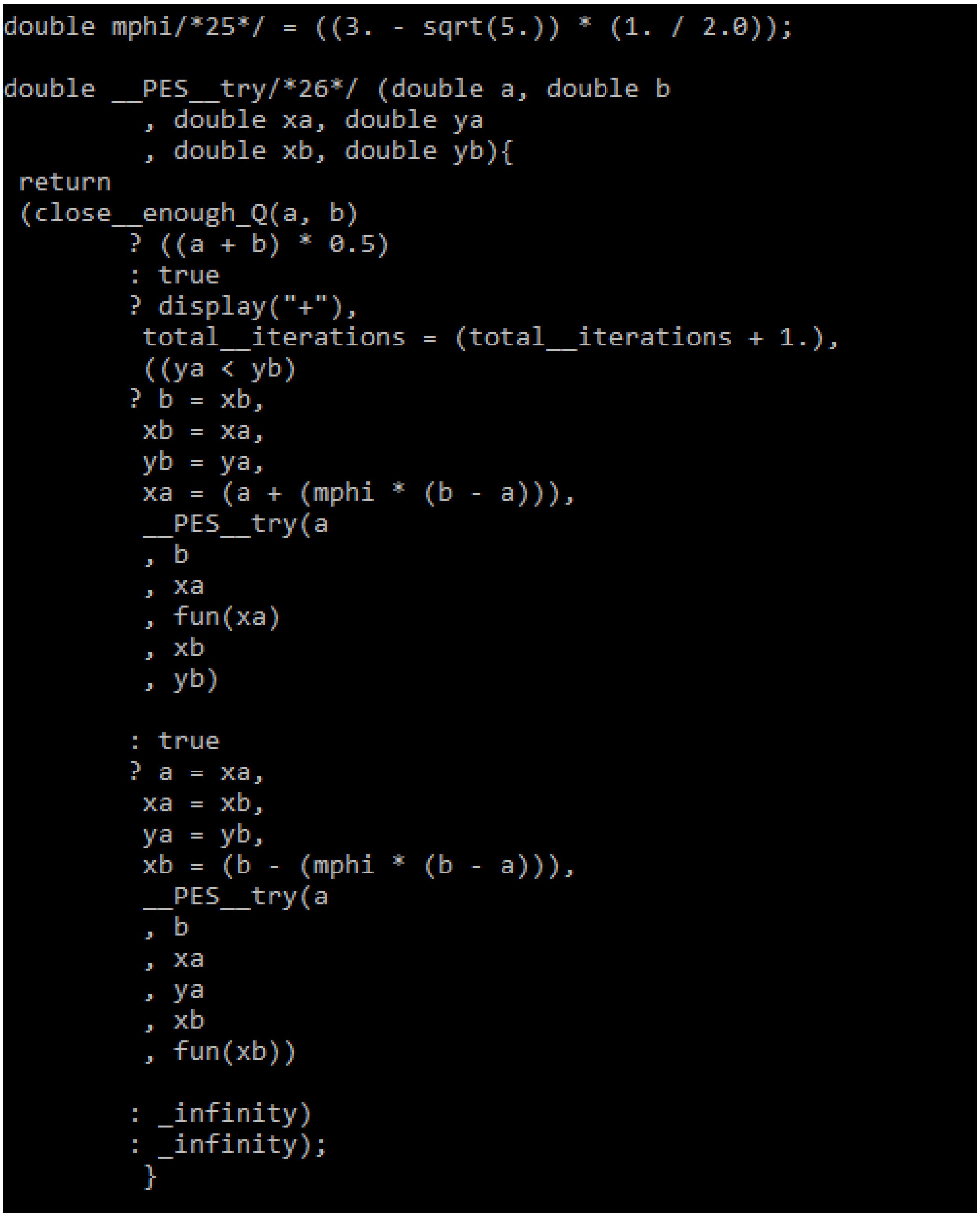
```scheme
(define mphi (* (- 3(sqrt 5))(/ 2.0)))
(define (try a b xa ya xb yb)
 (cond((close-enough? a b)(* (+ a b)0.5))
      (#t (let() (display "+")
             (set! total-iterations (+ total-iterations 1))
             (cond((< ya yb) (let() (set! b xb)
                               (set! xb xa)
                               (set! yb ya)
                               (set! xa (+ a (* mphi(- b a))))
                               (try a b xa (fun xa) xb yb)))
                  (#t (let() (set! a xa)
                        (set! xa xb)
                        (set! ya yb)
                        (set! xb (- b (* mphi(- b a))))
                        (try a b xa ya xb (fun xb))))
             );cond...
      ));let...
 );if...
)
(define (close-enough? x y)
  (<(abs (- x y))tolerance))
(define tolerance .001)
(define total-iterations 0)
(define xmin 0)
(set! xmin(golden-section-search a b))
  (display"Interval=\t[")
  (display a)
  (display" , ")
  (display b)
  (display"]\n")
  (display"Total number of iteranions=")
total-iterations
  (display"xmin=\t\t")
xmin
  (display"f(xmin)=\t")
(fun xmin)
```



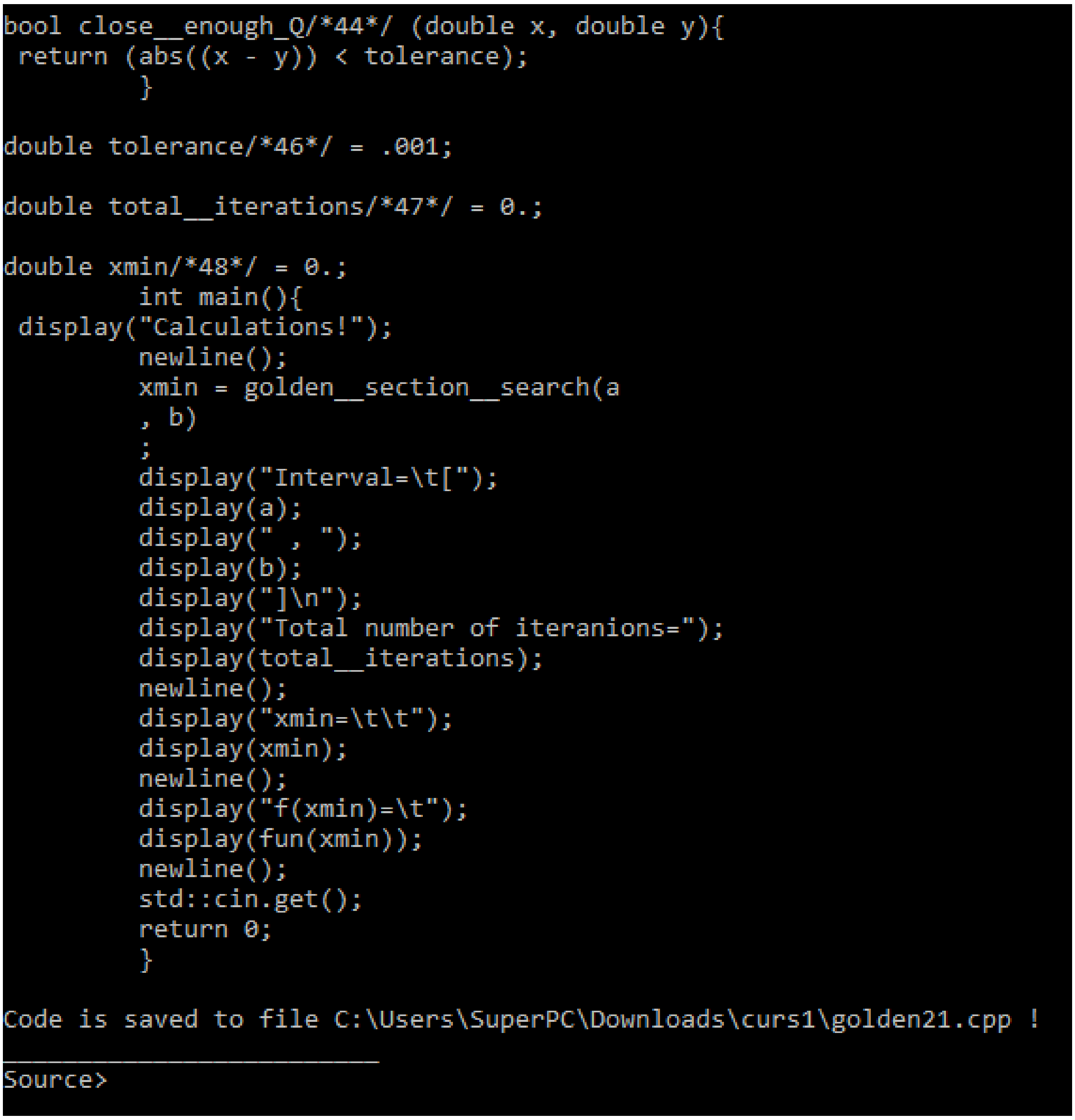
```
Code:
/*  PES   */
#include "mlisp.h"
extern double a/*2*/ ;
         extern double b/*2*/ ;
         double fun/*3*/ (double x);
         double golden__section__search/*7*/ (double a, double b);
         double golden__start/*15*/ (double a, double b);
         extern double mphi/*25*/ ;
         double __PES__try/*26*/ (double a, double b
         , double xa, double ya
         , double xb, double yb);
         bool close__enough_Q/*44*/ (double x, double y);
         extern double tolerance/*46*/ ;
         extern double total__iterations/*47*/ ;
         extern double xmin/*48*/ ;
         //________________
double a/*2*/ = 2.;

double b/*2*/ = 3.;

double fun/*3*/ (double x){
 x = (x - (21. / 22.));
         return
 (x - expt((x - 2.)
         , 3.)
          - atan(x) - 1.);
         }
```



```
double golden__section__search/*7*/ (double a, double b){
 {
 double xmin(((a < b)
        ? golden__start(a
         , b)

        : true
        ? golden__start(b
         , a)

        : _infinity));
         newline();
         return
 xmin;
         }
}

double golden__start/*15*/ (double a, double b){
 total__iterations = 0.;
         {
 double xa((a + (mphi * (b - a)))),
         xb((b + (- (mphi * (b - a)))));
         return
 __PES__try(a
         , b
         , xa
         , fun(xa)
         , xb
         , fun(xb))
         ;
         }
}
```



```
double mphi/*25*/ = ((3. - sqrt(5.)) * (1. / 2.0));

double __PES__try/*26*/ (double a, double b
         , double xa, double ya
         , double xb, double yb){
 return
 (close__enough_Q(a, b)
        ? ((a + b) * 0.5)
        : true
        ? display("+"),
         total__iterations = (total__iterations + 1.),
         ((ya < yb)
        ? b = xb,
         xb = xa,
         yb = ya,
         xa = (a + (mphi * (b - a))),
         __PES__try(a
         , b
         , xa
         , fun(xa)
         , xb
         , yb)

        : true
        ? a = xa,
         xa = xb,
         ya = yb,
         xb = (b - (mphi * (b - a))),
         __PES__try(a
         , b
         , xa
         , ya
         , xb
         , fun(xb))

        : _infinity)
        : _infinity);
         }
```



```
bool close__enough_Q/*44*/ (double x, double y){
 return (abs((x - y)) < tolerance);
         }

double tolerance/*46*/ = .001;

double total__iterations/*47*/ = 0.;

double xmin/*48*/ = 0.;
         int main(){
 display("Calculations!");
         newline();
         xmin = golden__section__search(a
         , b)
         ;
         display("Interval=\t[");
         display(a);
         display(" , ");
         display(b);
         display("]\n");
         display("Total number of iteranions=");
         display(total__iterations);
         newline();
         display("xmin=\t\t");
         display(xmin);
         newline();
         display("f(xmin)=\t");
         display(fun(xmin));
         newline();
         std::cin.get();
         return 0;
         }

Code is saved to file C:\Users\SuperPC\Downloads\curs1\golden21.cpp !
_________________________
Source>
```

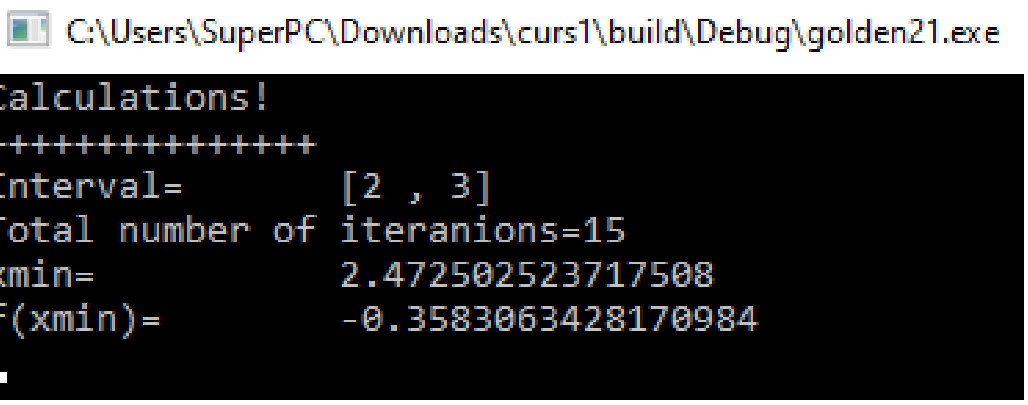
Распечатка файла golden21.cpp .

```
>
/*  PES   */
#include "mlisp.h"
extern double a/*2*/ ;
      extern double b/*2*/ ;
      double fun/*3*/ (double x);
      double golden__section__search/*7*/ (double a,
double b);
      double golden__start/*15*/ (double a, double b);
      extern double mphi/*25*/ ;
      double __PES__try/*26*/ (double a, double b
      , double xa, double ya
```

```
        , double xb, double yb);
        bool close__enough_Q/*44*/ (double x, double y);
        extern double tolerance/*46*/ ;
        extern double total__iterations/*47*/ ;
        extern double xmin/*48*/ ;
        //__________________
double a/*2*/ = 2.;

double b/*2*/ = 3.;

double fun/*3*/ (double x){
 x = (x - (21. / 22.));
        return
 (x - expt((x - 2.)
        , 3.)
         - atan(x) - 1.);
        }

double golden__section__search/*7*/ (double a, double
b){
 {
 double xmin(((a < b)
       ? golden__start(a
        , b)

       : true
       ? golden__start(b
        , a)

       : _infinity));
        newline();
        return
 xmin;
        }
}

double golden__start/*15*/ (double a, double b){
 total__iterations = 0.;
        {
 double xa((a + (mphi * (b - a)))),
        xb((b + (- (mphi * (b - a)))));
        return
 __PES__try(a
        , b
        , xa
        , fun(xa)
        , xb
        , fun(xb))
        ;
        }
}

double mphi/*25*/ = ((3. - sqrt(5.)) * (1. / 2.0));

double __PES__try/*26*/ (double a, double b
        , double xa, double ya
        , double xb, double yb){
 return
 (close__enough_Q(a, b)
       ? ((a + b) * 0.5)
       : true
       ? display("+"),
        total__iterations = (total__iterations + 1.),
        ((ya < yb)
       ? b = xb,
        xb = xa,
        yb = ya,
        xa = (a + (mphi * (b - a))),
        __PES__try(a
        , b
        , xa
        , fun(xa)
        , xb
        , yb)

       : true
       ? a = xa,
        xa = xb,
        ya = yb,
        xb = (b - (mphi * (b - a))),
        __PES__try(a
        , b
        , xa
        , ya
        , xb
        , fun(xb))

       : _infinity)
       : _infinity);
        }

bool close__enough_Q/*44*/ (double x, double y){
 return (abs((x - y)) < tolerance);
        }

double tolerance/*46*/ = .001;

double total__iterations/*47*/ = 0.;

double xmin/*48*/ = 0.;
        int main(){
 display("Calculations!");
        newline();
        xmin = golden__section__search(a
        , b)
        ;
        display("Interval=\t[");
        display(a);
        display(" , ");
        display(b);
        display("]\n");
        display("Total number of iteranions=");
        display(total__iterations);
        newline();
        display("xmin=\t\t");
        display(xmin);
        newline();
        display("f(xmin)=\t");
        display(fun(xmin));
        newline();
        std::cin.get();
        return 0;
        }
```

Скриншот запуска задачи на C++.

>

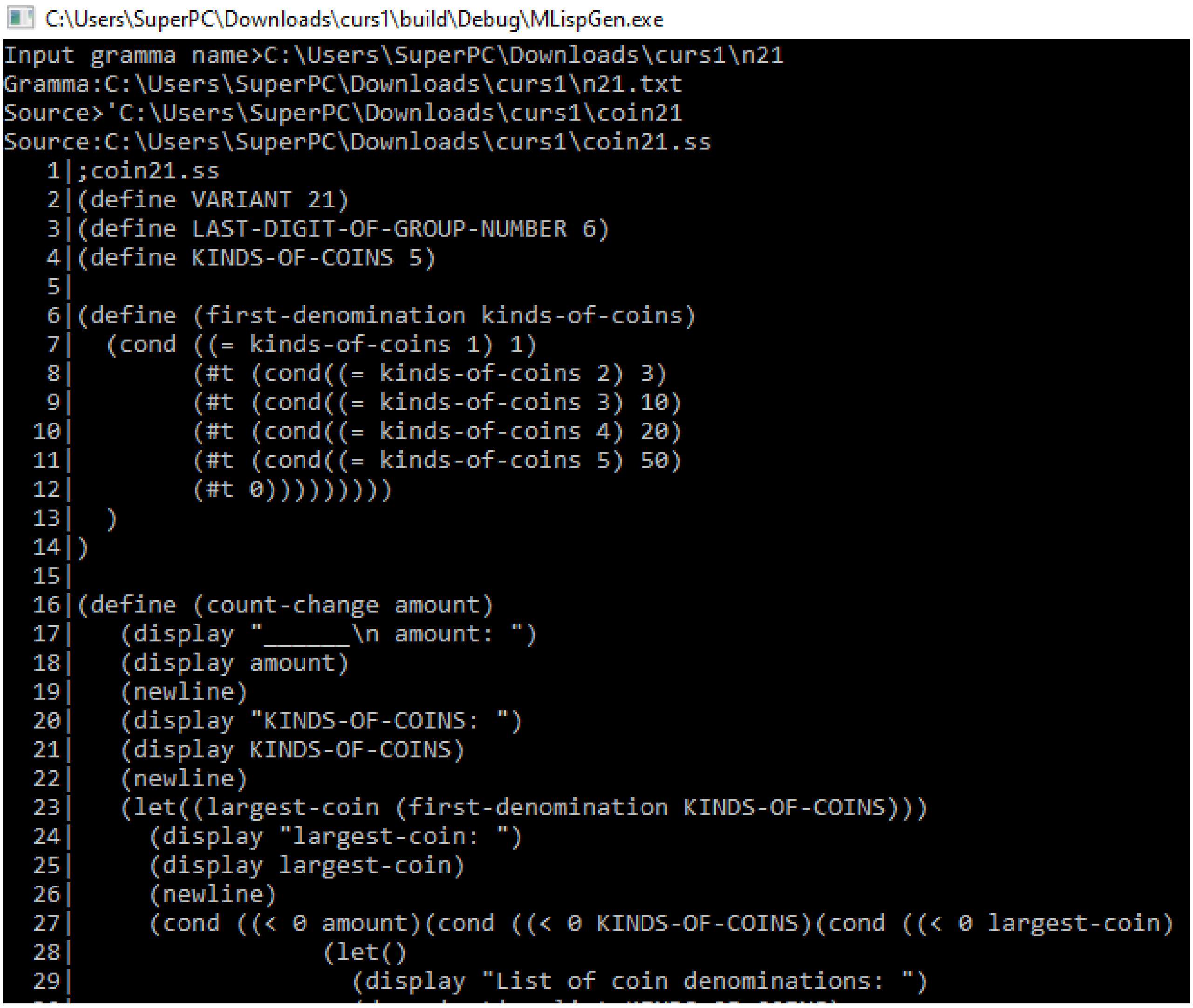


```
C:\Users\SuperPC\Downloads\curs1\build\Debug\golden21.exe
```

```
Calculations!
+++++++++++++++
Interval=        [2 , 3]
Total number of iteranions=15
xmin=            2.472502523717508
f(xmin)=         -0.3583063428170984
```

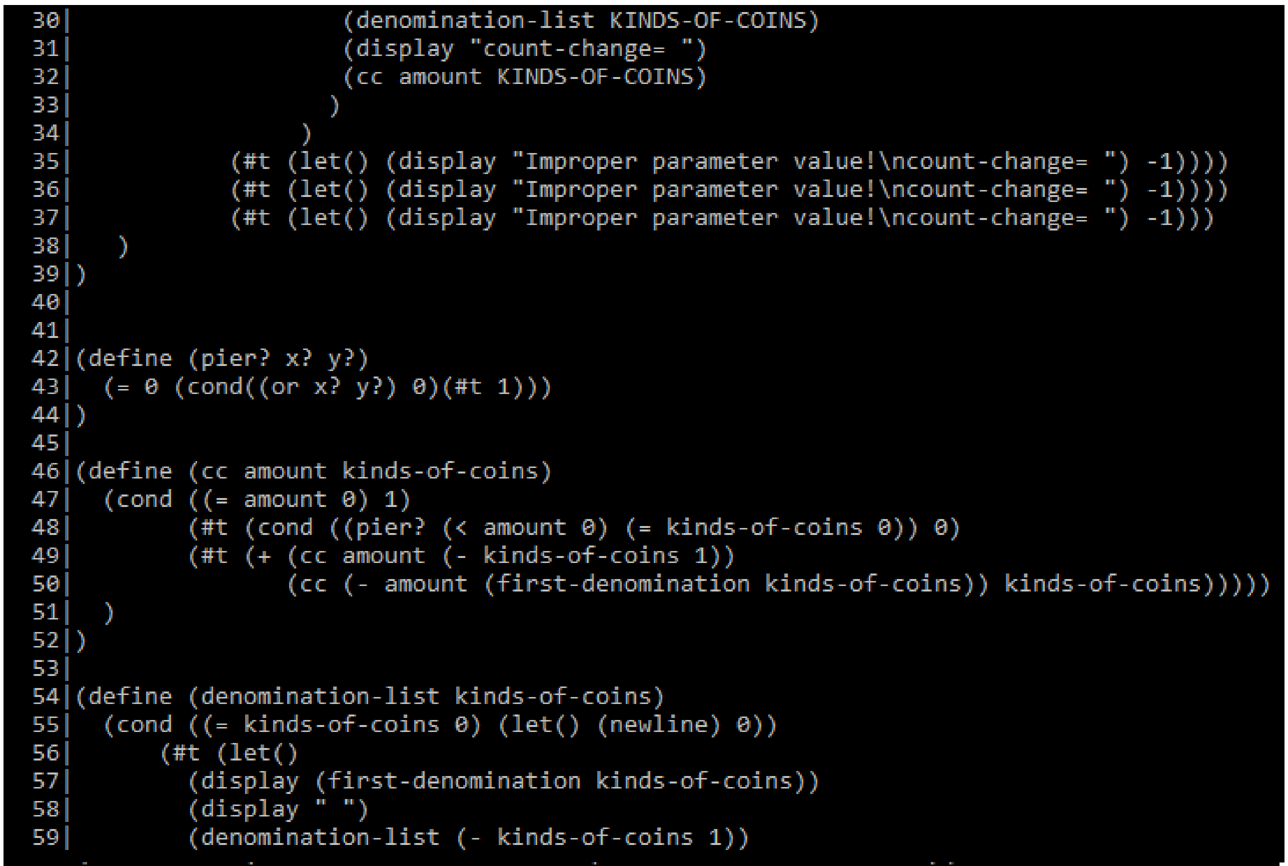
Контрольная задача №3 – coin21.

Полный скриншот трансляции без трассировки (крупный белый шрифт на ярком черном фоне).
>

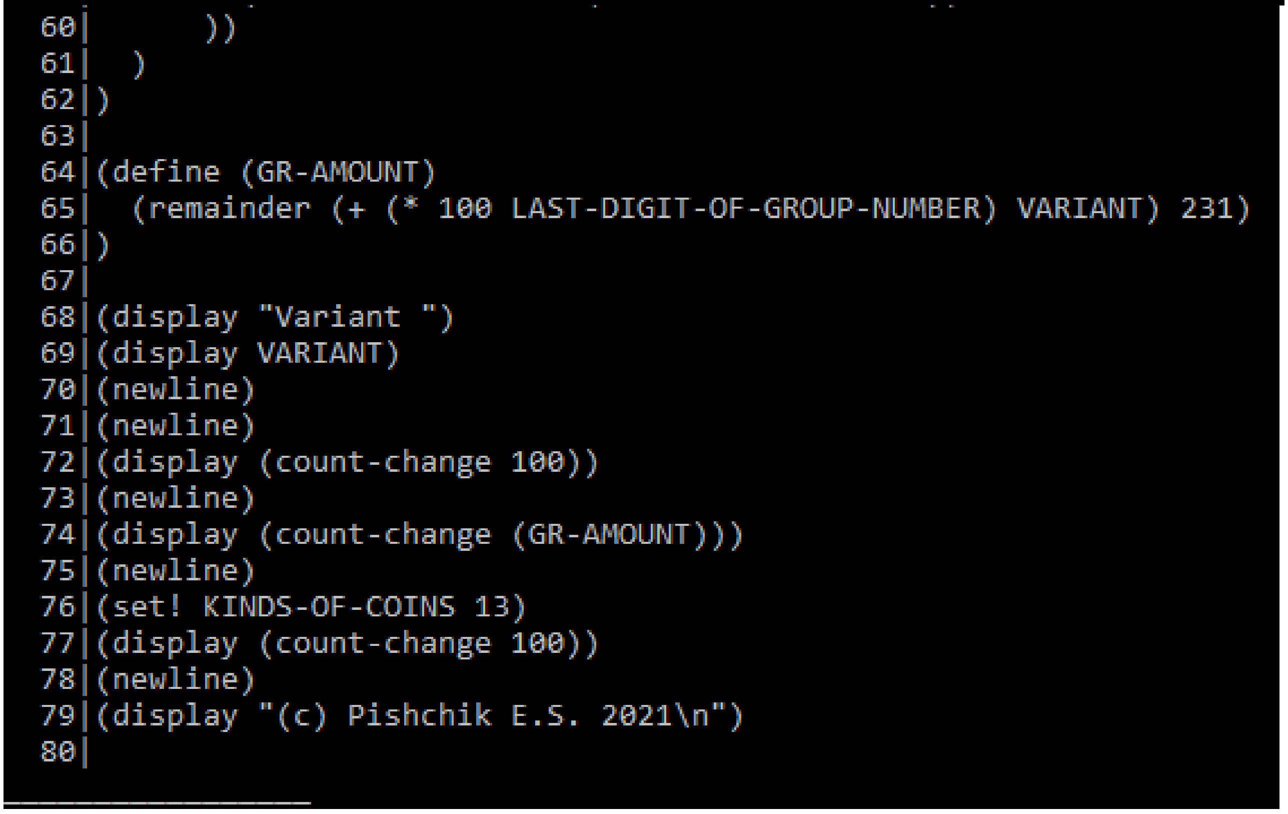


```
C:\Users\SuperPC\Downloads\curs1\build\Debug\MLispGen.exe
```

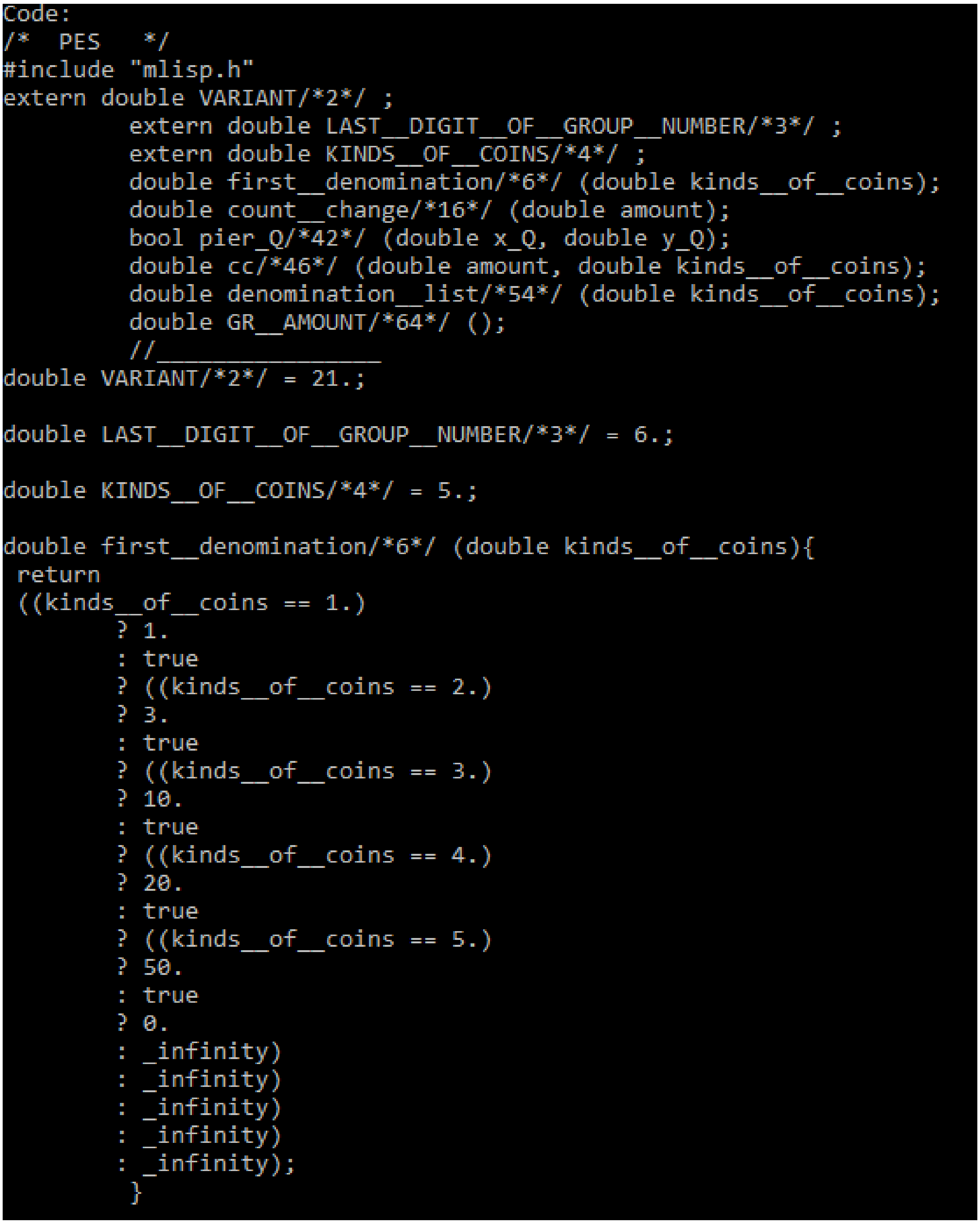
```
Input gramma name>C:\Users\SuperPC\Downloads\curs1\n21
Gramma:C:\Users\SuperPC\Downloads\curs1\n21.txt
Source>'C:\Users\SuperPC\Downloads\curs1\coin21
Source:C:\Users\SuperPC\Downloads\curs1\coin21.ss
;coin21.ss
(define VARIANT 21)
(define LAST-DIGIT-OF-GROUP-NUMBER 6)
(define KINDS-OF-COINS 5)

(define (first-denomination kinds-of-coins)
  (cond ((= kinds-of-coins 1) 1)
        (#t (cond((= kinds-of-coins 2) 3)
        (#t (cond((= kinds-of-coins 3) 10)
        (#t (cond((= kinds-of-coins 4) 20)
        (#t (cond((= kinds-of-coins 5) 50)
        (#t 0)))))))))
  )
)

(define (count-change amount)
   (display "______\n amount: ")
   (display amount)
   (newline)
   (display "KINDS-OF-COINS: ")
   (display KINDS-OF-COINS)
   (newline)
   (let((largest-coin (first-denomination KINDS-OF-COINS)))
     (display "largest-coin: ")
     (display largest-coin)
     (newline)
     (cond ((< 0 amount)(cond ((< 0 KINDS-OF-COINS)(cond ((< 0 largest-coin)
                 (let()
                   (display "List of coin denominations: ")
```



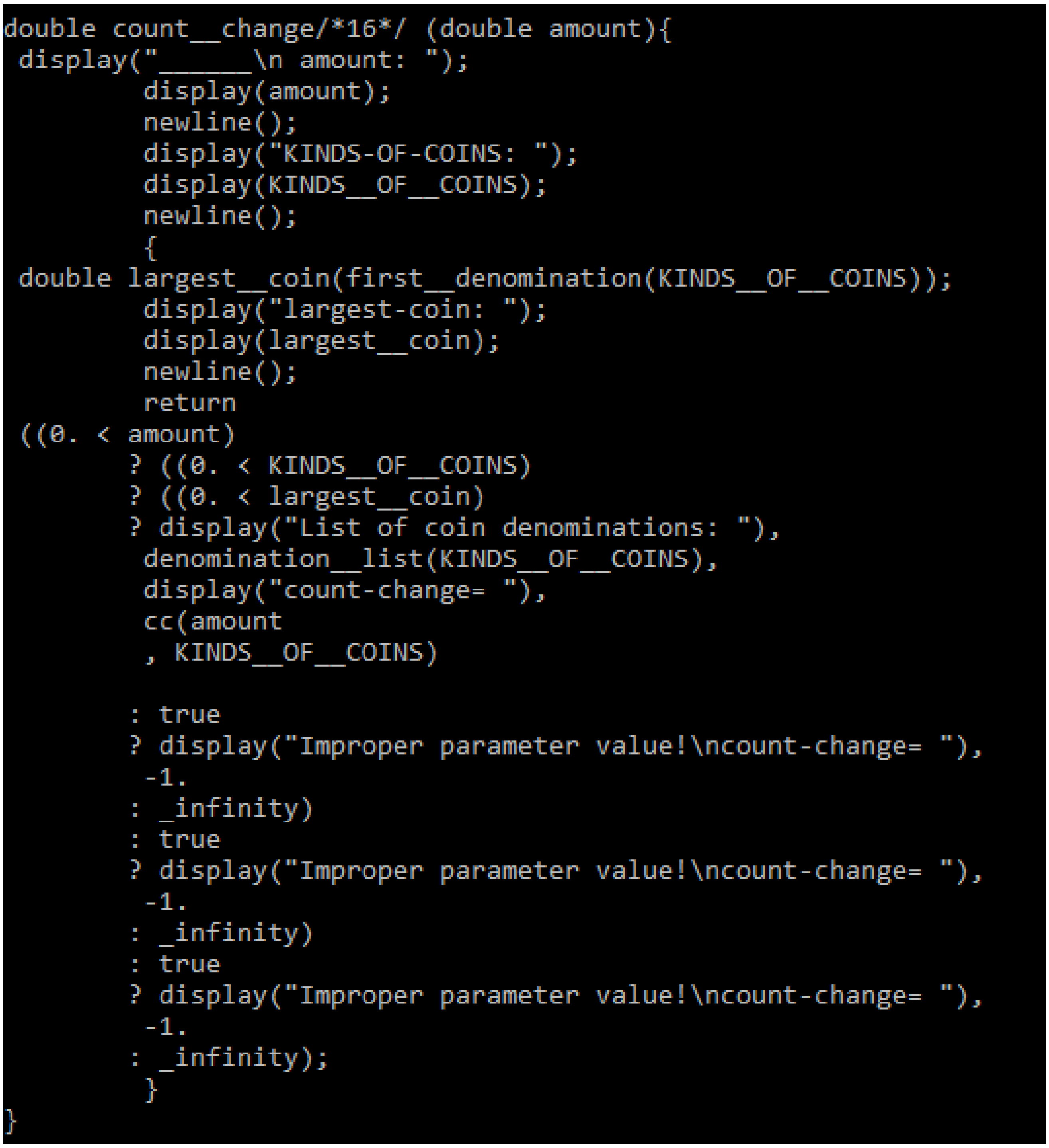
```scheme
                   (denomination-list KINDS-OF-COINS)
                   (display "count-change= ")
                   (cc amount KINDS-OF-COINS)
                  )
                )
           (#t (let() (display "Improper parameter value!\ncount-change= ") -1))))
           (#t (let() (display "Improper parameter value!\ncount-change= ") -1))))
           (#t (let() (display "Improper parameter value!\ncount-change= ") -1)))
   )
)


(define (pier? x? y?)
  (= 0 (cond((or x? y?) 0)(#t 1)))
)

(define (cc amount kinds-of-coins)
  (cond ((= amount 0) 1)
        (#t (cond ((pier? (< amount 0) (= kinds-of-coins 0)) 0)
        (#t (+ (cc amount (- kinds-of-coins 1))
               (cc (- amount (first-denomination kinds-of-coins)) kinds-of-coins)))))
  )
)

(define (denomination-list kinds-of-coins)
  (cond ((= kinds-of-coins 0) (let() (newline) 0))
      (#t (let()
        (display (first-denomination kinds-of-coins))
        (display " ")
        (denomination-list (- kinds-of-coins 1))
```



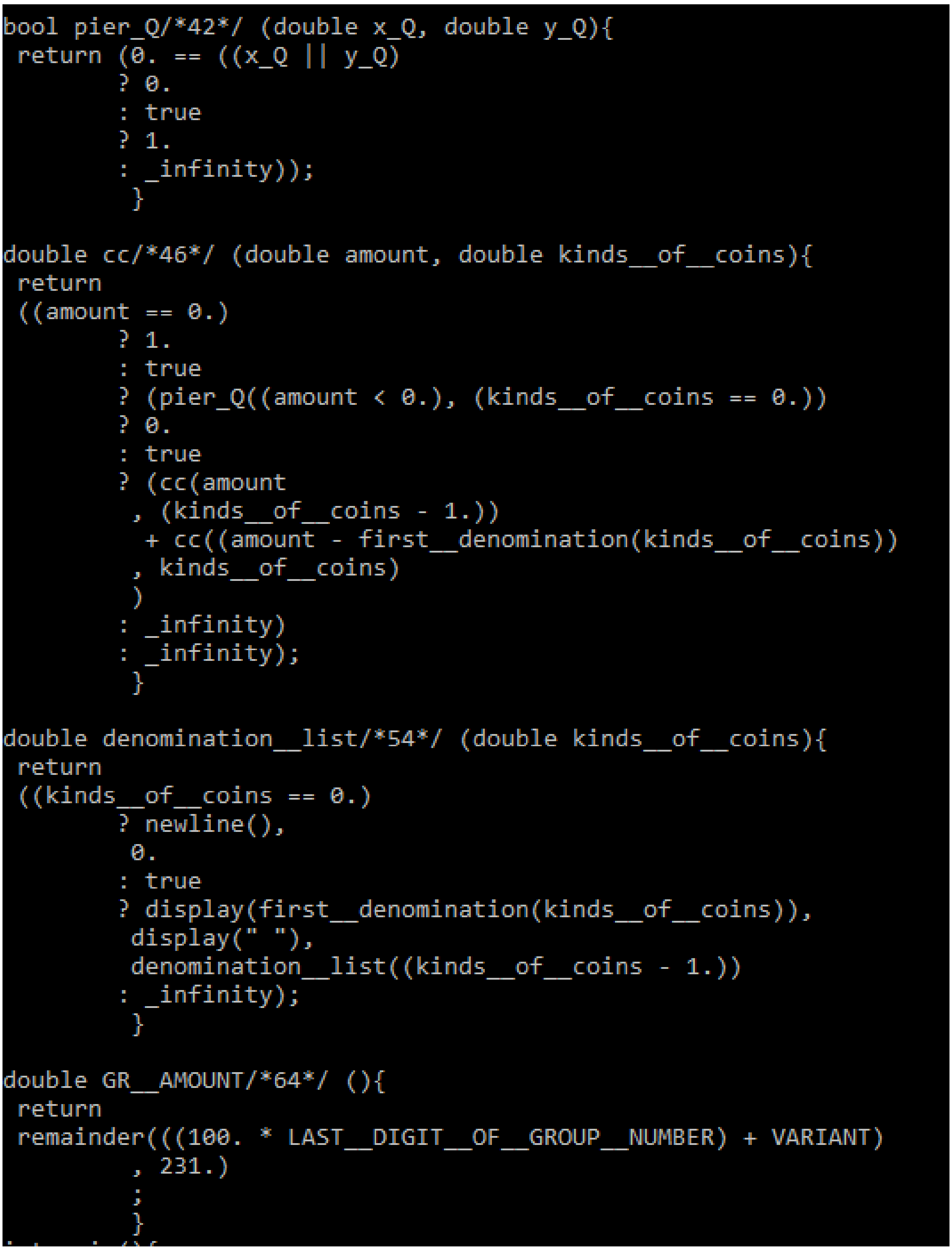
```scheme
      ))
  )
)

(define (GR-AMOUNT)
  (remainder (+ (* 100 LAST-DIGIT-OF-GROUP-NUMBER) VARIANT) 231)
)

(display "Variant ")
(display VARIANT)
(newline)
(newline)
(display (count-change 100))
(newline)
(display (count-change (GR-AMOUNT)))
(newline)
(set! KINDS-OF-COINS 13)
(display (count-change 100))
(newline)
(display "(c) Pishchik E.S. 2021\n")
```



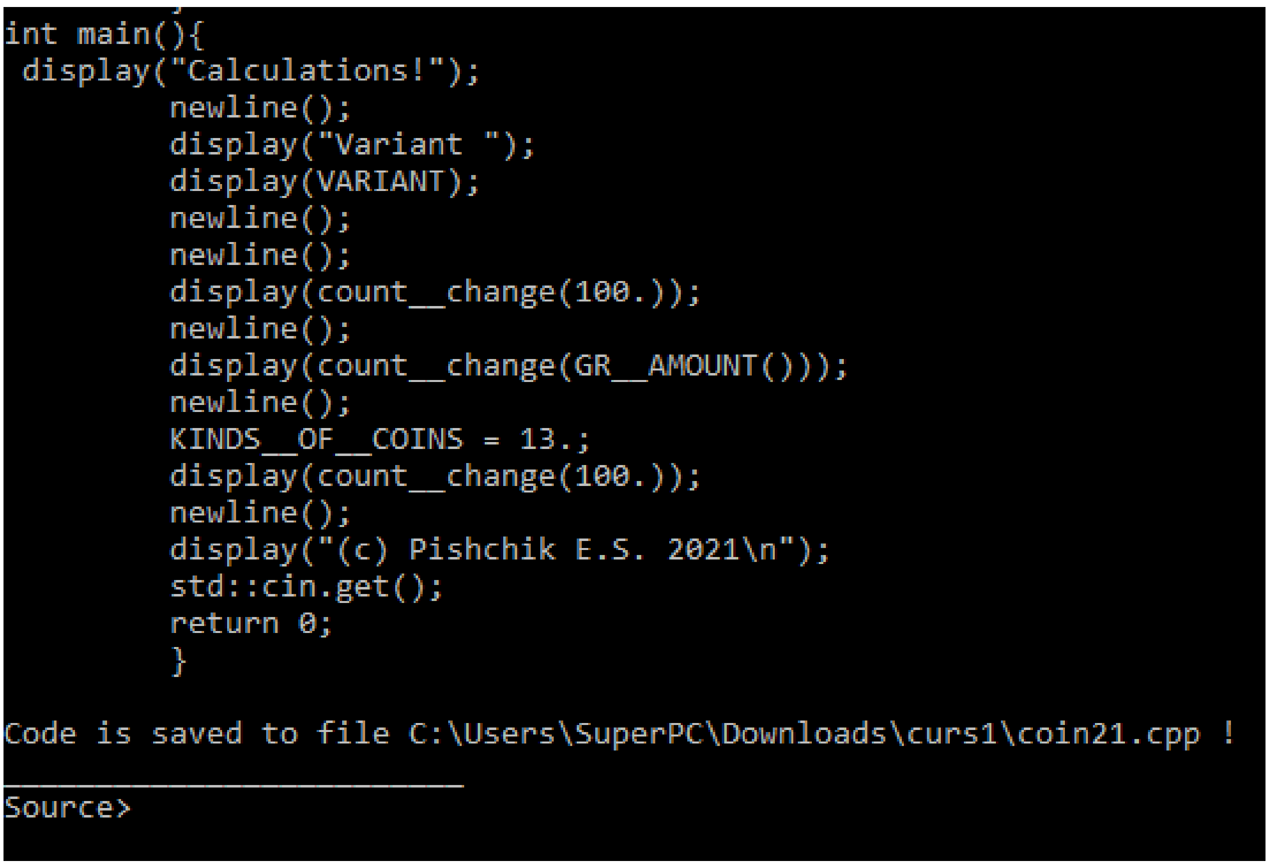
```
Code:
/*  PES   */
#include "mlisp.h"
extern double VARIANT/*2*/ ;
         extern double LAST__DIGIT__OF__GROUP__NUMBER/*3*/ ;
         extern double KINDS__OF__COINS/*4*/ ;
         double first__denomination/*6*/ (double kinds__of__coins);
         double count__change/*16*/ (double amount);
         bool pier_Q/*42*/ (double x_Q, double y_Q);
         double cc/*46*/ (double amount, double kinds__of__coins);
         double denomination__list/*54*/ (double kinds__of__coins);
         double GR__AMOUNT/*64*/ ();
         //________________
double VARIANT/*2*/ = 21.;

double LAST__DIGIT__OF__GROUP__NUMBER/*3*/ = 6.;

double KINDS__OF__COINS/*4*/ = 5.;

double first__denomination/*6*/ (double kinds__of__coins){
 return
 ((kinds__of__coins == 1.)
        ? 1.
        : true
        ? ((kinds__of__coins == 2.)
        ? 3.
        : true
        ? ((kinds__of__coins == 3.)
        ? 10.
        : true
        ? ((kinds__of__coins == 4.)
        ? 20.
        : true
        ? ((kinds__of__coins == 5.)
        ? 50.
        : true
        ? 0.
        : _infinity)
        : _infinity)
        : _infinity)
        : _infinity)
        : _infinity);
         }
```



```
double count__change/*16*/ (double amount){
 display("______\n amount: ");
         display(amount);
         newline();
         display("KINDS-OF-COINS: ");
         display(KINDS__OF__COINS);
         newline();
         {
 double largest__coin(first__denomination(KINDS__OF__COINS));
         display("largest-coin: ");
         display(largest__coin);
         newline();
         return
 ((0. < amount)
        ? ((0. < KINDS__OF__COINS)
        ? ((0. < largest__coin)
        ? display("List of coin denominations: "),
         denomination__list(KINDS__OF__COINS),
         display("count-change= "),
         cc(amount
         , KINDS__OF__COINS)

        : true
        ? display("Improper parameter value!\ncount-change= "),
         -1.
        : _infinity)
        : true
        ? display("Improper parameter value!\ncount-change= "),
         -1.
        : _infinity)
        : true
        ? display("Improper parameter value!\ncount-change= "),
         -1.
        : _infinity);
         }
}
```



```
bool pier_Q/*42*/ (double x_Q, double y_Q){
 return (0. == ((x_Q || y_Q)
        ? 0.
        : true
        ? 1.
        : _infinity));
         }

double cc/*46*/ (double amount, double kinds__of__coins){
 return
 ((amount == 0.)
        ? 1.
        : true
        ? (pier_Q((amount < 0.), (kinds__of__coins == 0.))
        ? 0.
        : true
        ? (cc(amount
         , (kinds__of__coins - 1.))
          + cc((amount - first__denomination(kinds__of__coins))
         , kinds__of__coins)
         )
        : _infinity)
        : _infinity);
         }

double denomination__list/*54*/ (double kinds__of__coins){
 return
 ((kinds__of__coins == 0.)
        ? newline(),
         0.
        : true
        ? display(first__denomination(kinds__of__coins)),
         display(" "),
         denomination__list((kinds__of__coins - 1.))
        : _infinity);
         }

double GR__AMOUNT/*64*/ (){
 return
 remainder(((100. * LAST__DIGIT__OF__GROUP__NUMBER) + VARIANT)
         , 231.)
         ;
         }
```



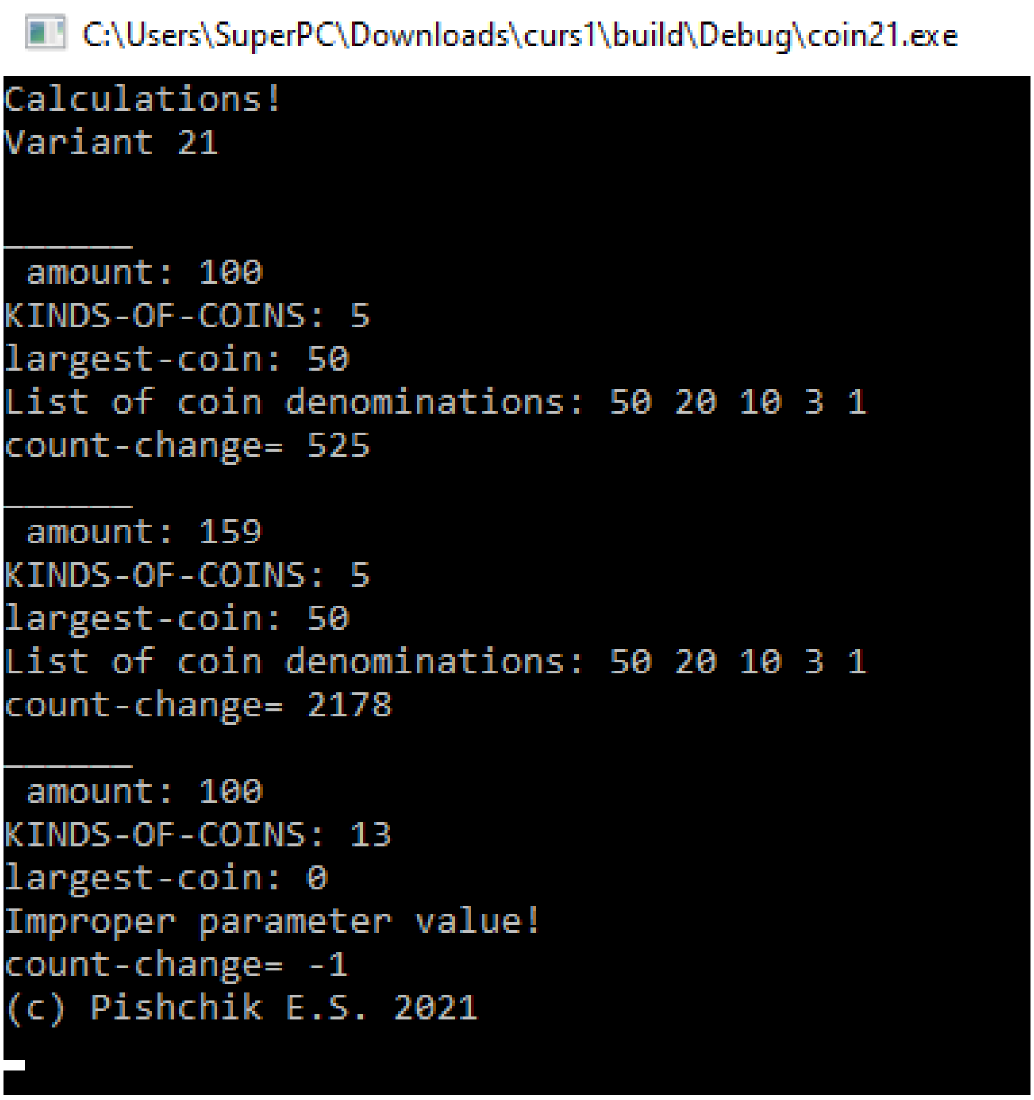
```
int main(){
 display("Calculations!");
        newline();
        display("Variant ");
        display(VARIANT);
        newline();
        newline();
        display(count__change(100.));
        newline();
        display(count__change(GR__AMOUNT()));
        newline();
        KINDS__OF__COINS = 13.;
        display(count__change(100.));
        newline();
        display("(c) Pishchik E.S. 2021\n");
        std::cin.get();
        return 0;
        }

Code is saved to file C:\Users\SuperPC\Downloads\curs1\coin21.cpp !
_________________________
Source>
```

Распечатка файла coin21.cpp .

```
>
/*  PES   */
#include "mlisp.h"
extern double VARIANT/*2*/ ;
       extern double
LAST__DIGIT__OF__GROUP__NUMBER/*3*/ ;
       extern double KINDS__OF__COINS/*4*/ ;
       double first__denomination/*6*/ (double
kinds__of__coins);
       double count__change/*16*/ (double amount);
       bool pier_Q/*42*/ (double x_Q, double y_Q);
       double cc/*46*/ (double amount, double
kinds__of__coins);
       double denomination__list/*54*/ (double
kinds__of__coins);
       double GR__AMOUNT/*64*/ ();
       //____________________
double VARIANT/*2*/ = 21.;

double LAST__DIGIT__OF__GROUP__NUMBER/*3*/ = 6.;

double KINDS__OF__COINS/*4*/ = 5.;
double first__denomination/*6*/ (double
kinds__of__coins){
 return
 ((kinds__of__coins == 1.)
       ? 1.
       : true
       ? ((kinds__of__coins == 2.)
       ? 3.
       : true
       ? ((kinds__of__coins == 3.)
       ? 10.
       : true
       ? ((kinds__of__coins == 4.)
       ? 20.
       : true
       ? ((kinds__of__coins == 5.)
       ? 50.
       : true
       ? 0.
       : _infinity)
       : _infinity)
       : _infinity)
       : _infinity)
       : _infinity);
        }

double count__change/*16*/ (double amount){
 display("_______\n amount: ");
        display(amount);
        newline();
        display("KINDS-OF-COINS: ");
        display(KINDS__OF__COINS);
        newline();
        {
 double
largest__coin(first__denomination(KINDS__OF__COINS));
        display("largest-coin: ");
        display(largest__coin);
        newline();
        return
 ((0. < amount)
       ? ((0. < KINDS__OF__COINS)
       ? ((0. < largest__coin)
       ? display("List of coin denominations: "),
        denomination__list(KINDS__OF__COINS),
        display("count-change= "),
        cc(amount
        , KINDS__OF__COINS)

       : true
       ? display("Improper parameter value!\ncount-change=
"),
        -1.
       : _infinity)
       : true
       ? display("Improper parameter value!\ncount-change=
"),
        -1.
       : _infinity)
       : true
       ? display("Improper parameter value!\ncount-change=
"),
        -1.
       : _infinity);
        }
}

bool pier_Q/*42*/ (double x_Q, double y_Q){
 return (0. == ((x_Q || y_Q)
       ? 0.
       : true
       ? 1.
       : _infinity));
        }

double cc/*46*/ (double amount, double
kinds__of__coins){
 return
 ((amount == 0.)
       ? 1.
       : true
       ? (pier_Q((amount < 0.), (kinds__of__coins == 0.))
       ? 0.
       : true
       ? (cc(amount
       , (kinds__of__coins - 1.))
        + cc((amount -
first__denomination(kinds__of__coins))
       , kinds__of__coins)
       )
      : _infinity)
      : _infinity);
       }

double denomination__list/*54*/ (double
kinds__of__coins){
 return
 ((kinds__of__coins == 0.)
      ? newline(),
       0.
      : true
      ? display(first__denomination(kinds__of__coins)),
       display(" "),
       denomination__list((kinds__of__coins - 1.))
      : _infinity);
       }

double GR__AMOUNT/*64*/ (){
 return
 remainder(((100. *
LAST__DIGIT__OF__GROUP__NUMBER) + VARIANT)
       , 231.)
       ;
       }
int main(){
 display("Calculations!");
       newline();
       display("Variant ");
       display(VARIANT);
       newline();
       newline();
       display(count__change(100.));
       newline();
       display(count__change(GR__AMOUNT()));
       newline();
       KINDS__OF__COINS = 13.;
```

```
        display(count__change(100.));
        newline();
        display("(c) Pishchik E.S. 2021\n");
        std::cin.get();
        return 0;
        }
```

Скриншот запуска задачи на C++.
>



C:\Users\SuperPC\Downloads\curs1\build\Debug\coin21.exe

```
Calculations!
Variant 21


______
 amount: 100
KINDS-OF-COINS: 5
largest-coin: 50
List of coin denominations: 50 20 10 3 1
count-change= 525

______
 amount: 159
KINDS-OF-COINS: 5
largest-coin: 50
List of coin denominations: 50 20 10 3 1
count-change= 2178

______
 amount: 100
KINDS-OF-COINS: 13
largest-coin: 0
Improper parameter value!
count-change= -1
(c) Pishchik E.S. 2021
```

Распечатка файла code-gen.cpp.
>

```
/* $n21 */
#include "code-gen.h"
using namespace std;
void tCG::init(){declarations.clear();
 Authentication = "PES";
//               ^
// replace with your initials!!!
}
int tCG::p01(){ // S -> PROG
      string header = "/*  " + Authentication + "   */\n";
      header += "#include \"mlisp.h\"\n";
      header += declarations;
      header += "//________________________ \n";
      S1->obj = header + S1->obj;
      return 0;}
int tCG::p02(){ //   PROG -> CALCS
      S1->obj = "int main(){\n " + S1->obj +
"std::cin.get();\n\t return 0;\n\t }\n";
      return 0;}

int tCG::p03(){ //   PROG -> DEFS
      S1->obj += "int main(){\n "
                 "display(\"No calculations!\");\n\t
newline();\n\t "
                 "std::cin.get();\n\t return 0;\n\t }\n";
      return 0;}

int tCG::p04(){ //   PROG -> DEFS CALCS
      S1->obj += "int main(){\n "
                 "display(\"Calculations!\");\n\t
newline();\n\t ";
      S1->obj += S2->obj;
      S1->obj += "std::cin.get();\n\t return 0;\n\t }\n";
      return 0;}
int tCG::p05(){ //             E -> $id
      S1->obj = decor(S1->name);
      return 0;}
int tCG::p06(){ //             E -> $int
      S1->obj = S1->name + ".";
      return 0;}
int tCG::p07(){ //             E -> $dec
   S1->obj = S1->name;
      return 0;}
int tCG::p08(){ //       E -> AREX
      return 0;}
int tCG::p09(){ //       E -> COND
      return 0;}
int tCG::p10(){ //       E -> EASYLET
```

```
        return 0;}
int tCG::p11(){ //        E -> CPROC
        return 0;}
int tCG::p12(){ //     AREX -> HAREX E )
        if (S1->count == 0 && S1->name == "/")
                S1->obj = "(1. " + S1->obj + " " + S2->obj + ")";
        else
                S1->obj = "(" + S1->obj + " " + S2->obj + ")";
        return 0;}
int tCG::p13(){ //    HAREX -> ( AROP
        S1->obj = S2->obj;
        S1->name = S2->name;
        return 0;}
int tCG::p14(){ //    HAREX -> HAREX E
        if (S1->count == 0)
                S1->obj = S2->obj + " " + S1->name;
        else
                S1->obj = S1->obj + " " + S2->obj + " " + S1-
>name;
        ++(S1->count);
        return 0;}
int tCG::p15(){ //     AROP -> +
        S1->obj = S1->name;
        return 0;}
int tCG::p16(){ //     AROP -> -
        S1->obj = S1->name;
        return 0;}
int tCG::p17(){ //     AROP -> *
        S1->obj = S1->name;
        return 0;}
int tCG::p18(){ //     AROP -> /
        S1->obj = S1->name;
        return 0;}
int tCG::p19(){ //  EASYLET -> HEASYL E )
        if(S1->count != 0)
                S1->obj += S2->obj + ";\n\t";
        S1->obj += S2->obj;
        ++(S1->count);
        return 0;}
int tCG::p20(){ //   HEASYL -> ( let ( )
        return 0;}
int tCG::p21(){ //   HEASYL -> HEASYL INTER
```

```
      S1->obj += S2->obj + ",\n\t ";
      return 0;}
int tCG::p22(){ //     COND -> ( cond BRANCHES )
      S1->obj = "(" + S3->obj + "_infinity)";
      return 0;}
int tCG::p23(){ // BRANCHES -> CLAUS CLAUS
      S1->obj += S2->obj;
      return 0;}
int tCG::p24(){ //    CLAUS -> ( BOOL E )
      S1->obj += S2->obj + "\n\t? " + S3->obj + "\n\t: ";
      return 0;}
int tCG::p25(){ //      STR -> $str
      S1->obj = S1->name;
      return 0;}
int tCG::p26(){ //      STR -> SCOND
      return 0;}
int tCG::p27(){ //    SCOND -> ( cond SBRANCHES )
      S1->obj = "(" + S3->obj + ")";
      return 0;}
int tCG::p28(){ //        SBRANCHES -> SELSE
      return 0;}
int tCG::p29(){ //    SBRANCHES -> SCLAUS SBRANCHES
      S1->obj = S1->obj + S2->obj;
      return 0;}
int tCG::p30(){ //   SCLAUS -> ( BOOL STR )
      S1->obj = S2->obj + "\n\t? " + S3->obj + "\n\t: ";
      return 0;}
int tCG::p31(){ //    SELSE -> ( else STR )
      S1->obj = "(" + S3->obj + ")";
      return 0;}
int tCG::p32(){ //    CPROC -> HCPROC )
      if (S1->count <= 1)
            S1->obj = S1->obj + ")";
      else
            S1->obj = S1->obj + ")\n\t ";
      return 0;}
int tCG::p33(){ //   HCPROC -> ( $id
      S1->obj = decor(S2->name) + "(";
      return 0;}
int tCG::p34(){ //   HCPROC -> HCPROC E
      if (S1->count)
            S1->obj += "\n\t , ";
```

```
        S1->obj += S2->obj;
        ++(S1->count);
        return 0;}
int tCG::p35(){ //     BOOL -> $bool
        S1->obj += (S1->name == "#t" ? "true" : "false");
        return 0;}
int tCG::p36(){ //     BOOL -> $idq
        S1->obj = decor(S1->name);
        return 0;}
int tCG::p37(){ //     BOOL -> REL
        return 0;}
int tCG::p38(){ //     BOOL -> OR
        return 0;}
int tCG::p39(){ //     BOOL -> CPRED
        return 0;}
int tCG::p40(){ //    CPRED -> HCPRED )
        S1->obj += ")";
        return 0;}
int tCG::p41(){ //   HCPRED -> ( $idq
        S1->obj = decor(S2->name) + "(";
        return 0;}
int tCG::p42(){ //   HCPRED -> HCPRED ARG
        if (S1->count)
                S1->obj += S1->count % 2 ? ", " : "\n\t , ";
    S1->obj += S2->obj;
    ++(S1->count);
        return 0;}
int tCG::p43(){ //      ARG -> E
        return 0;}
int tCG::p44(){ //      ARG -> BOOL
        return 0;}
int tCG::p45(){ //      REL -> ( = E E )
        S1->obj = "(" + S3->obj + " == " + S4->obj + ")";
        return 0;}
int tCG::p46(){ //      REL -> ( < E E )
        S1->obj = "(" + S3->obj + " < " + S4->obj + ")";
        return 0;}
int tCG::p47(){ //       OR -> HOR BOOL )
        S1->obj = "(" + S1->obj + S2->obj + ")";
        return 0;}
int tCG::p48(){ //      HOR -> ( or
    return 0;}
```

```
int tCG::p49(){ //      HOR -> HOR BOOL
      S1->obj += S2->obj + " || ";
      return 0;}
int tCG::p50(){ //    SET -> HSET E )
      S1->obj += S2->obj;
      return 0;}
int tCG::p51(){ //   HSET -> ( set! $id
      S1->obj = decor(S3->name) + " = ";
      return 0;}
int tCG::p52(){ //DISPSET -> ( display E )
      S1->obj = "display(" + S3->obj + ")";
      return 0;}
int tCG::p53(){ //DISPSET -> ( display BOOL )
      S1->obj = "display(" + S3->obj + ")";
      return 0;}
int tCG::p54(){ //DISPSET -> ( display STR )
      S1->obj = "display(" + S3->obj + ")";
      return 0;}
int tCG::p55(){ //DISPSET -> ( newline )
      S1->obj = "newline()";
      return 0;}
int tCG::p56(){ //DISPSET -> SET
      return 0;}
int tCG::p57(){ //  INTER -> DISPSET
      return 0;}
int tCG::p58(){ //  INTER -> E
      return 0;}
int tCG::p59(){ //  CALCS -> CALC
      return 0;}
int tCG::p60(){ //  CALCS -> CALCS CALC
      S1->obj += S2->obj;
      return 0;}
int tCG::p61(){ //   CALC -> E
      S1->obj = "display(" + S1->obj + ");\n\t
newline();\n\t ";
      return 0;}
int tCG::p62(){ //   CALC -> BOOL
      S1->obj = "display(" + S1->obj + ");\n\t
newline();\n\t ";
      return 0;}
int tCG::p63(){ //   CALC -> STR
```

```
      S1->obj = "display(" + S1->obj + ");\n\t
newline();\n\t ";
      return 0;}
int tCG::p64(){ //   CALC -> DISPSET
      S1->obj = S1->obj + ";\n\t ";
      return 0;}
int tCG::p65(){ //   DEFS -> DEF
      return 0;}
int tCG::p66(){ //   DEFS -> DEFS DEF
      S1->obj = S1->obj + "\n" + S2->obj;
      return 0;}
int tCG::p67(){ //    DEF -> PRED
      return 0;}
int tCG::p68(){ //    DEF -> VAR
      return 0;}
int tCG::p69(){ //    DEF -> PROC
      return 0;}
int tCG::p70(){ //   PRED -> HPRED BOOL )
      S1->obj += S2->obj + ";\n\t }\n";
      return 0;}
int tCG::p71(){ //  HPRED -> PDPAR )
      S1->obj += ")";
   declarations += S1->obj + ";\n\t ";
   S1->obj += "{\n return ";
   S1->count = 0;
      return 0;}
int tCG::p72(){ //  PDPAR -> ( define ( $idq
      S1->obj = "bool " + decor(S4->name) + "/*" + S4-
>line + "*/ (";
   S1->count = 0;
      return 0;}
int tCG::p73(){ //  PDPAR -> PDPAR $idq
      if (S1->count)
            S1->obj += S1->count % 2 ? ", " : "\n\t , ";
   S1->obj += "double " + decor(S2->name);
   ++(S1->count);
      return 0;}
int tCG::p74(){ //  PDPAR -> PDPAR $id
      if (S1->count)
            S1->obj += S1->count % 2 ? ", " : "\n\t , ";
   S1->obj += "double " + decor(S2->name);
   ++(S1->count);
```

```
	return 0;}
int tCG::p75(){ //    VAR -> VARDCL E )
	declarations += "extern double " + S1->obj + "/*" +
S1->line + "*/ ;\n\t ";
	S1->obj = "double " + S1->obj + "/*" + S1->line + "*/
= " + S2->obj + ";\n\t ";
	return 0;}
int tCG::p76(){ // VARDCL -> ( define $id
	S1->obj = decor(S3->name);
	return 0;}
int tCG::p77(){ //   PROC -> HPROC BLOCK )
	S1->obj += S2->obj + "}\n";
	return 0;}
int tCG::p78(){ //   PROC -> HPROC E )
	S1->obj += "return\n " + S2->obj + ";\n\t }\n";
	return 0;}
int tCG::p79(){ //  HPROC -> PCPAR )
	S1->obj += ")";
	declarations += S1->obj + ";\n\t ";
	S1->obj += "{\n ";
	return 0;}
int tCG::p80(){ //  HPROC -> HPROC INTER
	S1->obj += S2->obj + ";\n\t ";
	return 0;}
int tCG::p81(){ //  PCPAR -> ( define ( $id
	S1->obj = "double " + decor(S4->name) + "/*" + S4-
>line + "*/ (";
	S1->count = 0;
	S1->name = S4->name;
	return 0;}
int tCG::p82(){ //  PCPAR -> PCPAR $id
	if (S1->count)
		S1->obj += S1->count % 2 ? ", " : "\n\t , ";
	S1->obj += "double " + decor(S2->name);
	++(S1->count);
	return 0;}
int tCG::p83(){ //  BLOCK -> HBLOCK E )
	S1->obj = S1->obj + "return\n " + S2->obj + ";\n\t
}\n";
	return 0;}
int tCG::p84(){ // HBLOCK -> BLVAR )
	S1->obj = S1->obj + ";\n\t ";
```

```
        return 0;}
int tCG::p85(){ // HBLOCK -> HBLOCK INTER
        S1->obj += S2->obj + ";\n\t ";
        return 0;}
int tCG::p86(){ //  BLVAR -> ( let ( LOCDEF
        S1->obj = "{\n double " + S4->obj;
        return 0;}
int tCG::p87(){ //  BLVAR -> BLVAR LOCDEF
        S1->obj += ",\n\t " + S2->obj;
        return 0;}
int tCG::p88(){ // LOCDEF -> ( $id E )
        S1->obj += decor(S2->name) + "(" + S3->obj + ")";
        return 0;}
//________________________
int tCG::p89(){return 0;} int tCG::p90(){return 0;}
int tCG::p91(){return 0;} int tCG::p92(){return 0;}
int tCG::p93(){return 0;} int tCG::p94(){return 0;}
int tCG::p95(){return 0;} int tCG::p96(){return 0;}
int tCG::p97(){return 0;} int tCG::p98(){return 0;}
int tCG::p99(){return 0;} int tCG::p100(){return 0;}
int tCG::p101(){return 0;} int tCG::p102(){return 0;}
int tCG::p103(){return 0;} int tCG::p104(){return 0;}
int tCG::p105(){return 0;} int tCG::p106(){return 0;}
int tCG::p107(){return 0;} int tCG::p108(){return 0;}
int tCG::p109(){return 0;} int tCG::p110(){return 0;}
```