

```

program lab4;

Uses crt, Math, Graph, wincrt;

var

i,n,n1,gd,gm:integer;

x1,x2,h,x,s1,s,tt,d,q,w,r,nn:real;

gmsX, gmsY: int64;

gx1, gx2: double;

const left=-1.531;


function f(x:real):real;

begin

f:=1*x*x*x+(-1)*x*x+2*x+9;

end;

function f1(x:real):real;

begin

f1:= x*x*x*x/4-x*x*x/3+x*x+9*x;

end;


procedure cg;

begin

    CloseGraph;

    Writeln('Graphics Closed.');
```

End;

```

procedure coord;

begin

writeln(' Enter first coordinate: '); writeln;

readln(x1); writeln;

if x1<left then begin

writeln(' The first coordinate out of range --> its value will be reassigned to -1.531'); writeln;
```

```

x1:=left;
end;

writeln(' Enter second coordinate: '); writeln;
readln(x2); writeln;

if (x1=x2) or (x1>x2) then

    begin
        while (x1=x2) or (x1>x2) do begin
            if x1=x2 then begin
                while x1=x2 do    begin
                    writeln('Coordinates cannot be equal!'); writeln;
                    writeln(' Enter first coordinate: ');    writeln;
                    readln(x1); writeln;
                    if x1<left then begin
                        writeln(' The first coordinate out of range --> its value will be reassigned to -1.531');    writeln;
                        x1:=left; end;
                    writeln(' Enter second coordinate: ');    writeln;
                    readln(x2); writeln;
                    if x1=x2 then writeln('Change coordinates!'); writeln; end; end;
                    if x1>x2 then begin
                        while x1>x2 do    begin
                            writeln('The first coordinate cannot be larger than the second!');    writeln;
                            writeln(' Enter first coordinate: ');    writeln;
                            readln(x1); writeln;
                            if x1<left then begin
                                writeln(' The first coordinate out of range --> its value will be reassigned to -1.531');    writeln;
                                x1:=left; end;
                                writeln(' Enter second coordinate: ');    writeln;
                                readln(x2); writeln;
                                if x1>x2 then writeln('Change coordinates!'); writeln;
                                end; end; end;

```

```

end;

writeln(' Enter number of fragments (if the entered value if fraction, then the whole part will be
taked for the quantity!): '); writeln;

readln(nn);

while trunc(nn)<=0 do begin
if trunc(nn)<=0 then begin

writeln('The number of fragments cannot be negative or equal to zero!');  writeln;

writeln(' Enter number of fragments(if the entered value if fraction, then the whole part will be
taked for the quantity!): '); writeln;

readln(nn);  writeln; end;

        end;

h:=(x2-x1)/trunc(nn);

s:=0;

tt:=0;

for i:=1 to trunc(nn)-1 do

begin

s:=s+f(x1+h*i);

end;

tt:=h*((f(x1)+f(x2))/2+s);

tt:=abs(tt);

end;


procedure met;

begin

if tt>0 then

writeln('l= ', tt:2:5 );

if tt<=0 then

writeln('Enter coordinates!'); writeln;

writeln('Press any key.');
```

```
readln; writeln;
end;
```

```
procedure er;
var l:string;
begin
if tt=0 then begin writeln('Enter coordinates!'); writeln;
writeln('Press ane key');    end;
if tt>0 then begin
d:=abs(f1(x2)-f1(x1));
q:=abs(tt-d);
w:=abs(tt-d)/tt;
r:=abs(trunc(w*100+0.5));
if r>100 then
begin
writeln('Relative error more then 100!'); writeln;
writeln('Press any key');
end
else
begin
if tt>0 then begin
writeln('Approximate value: ', tt:2:2); writeln;
writeln('Exact value: ', d:2:2);    writeln;
writeln('Absolute error: ', q:2:2);    writeln;
writeln('Relative error: ', r:0:0,' %'); writeln;
writeln('Press any key');
end; end; end;
readln; writeln;
end;
```

```

procedure viz; // процедура рисования
var
  gmx, gmy, gx, gdx, gy1, gh, s: real;
  kY, kX, gx0, gy0, gpx, gpy, i, n: integer;
  gs,ss1,ss2,hh1,hh2: string;
begin
  setbkcolor(0);

  cleardevice;

  setcolor(2);

  gx0 := getmaxX div 2;
  gy0 := (getmaxY div 10) * 9;
  gmx := gmsX * 20;
  gmy := gmsY * 10;
  line(gx0 - round(gmx * 50), gy0, getmaxX - 20, gy0);
  line(gx0, getmaxY - 20, gx0, 20);
  setcolor(2);
  outtextXY(getmaxX - 25, gy0 + 10, 'X');
  outtextXY(gx0 + 15, 20, 'Y');
  setcolor(2);
  kX := (getmaxX - 50 - gx0) div round(gmx);
  kY := (getmaxY - 100) div round(gmy);
  for i:= -50 to kX do begin
    if (i=x1) or (i= x2) then
      setcolor(15)
    else
      setcolor(2);
    str(i, gs);
    outtextXY(gx0 + round( i * gmx) - 4, gy0 + 10, gs);
  end
end

```

```

line(gx0 + round(i * gmx), gy0 - 2, gx0 + round(i * gmx), gy0 + 2); end;

setcolor(2);

for i := 1 to kY do

begin

str(i, gs);

outtextXY(gx0 - 20, gy0 - round(i * gmy) - 3, gs);

line(gx0 - 2, gy0 - round(i * gmy), gx0 + 2, gy0 - round(i * gmy)); end;

gx := gx1;

gdx := 0.02;

gpy := 200;

setcolor(green);

while (gx <= gx2) do begin

gpx := gx0 + round(gx * gmx);

gy1 := ( gx * gx * gx - gx * gx + (2) * gx + (9));

gpy := gy0 - round(gy1 * gmy);

if gx = gx1 then

moveto(gpx, gpy);

lineto(gpx, gpy);

gx:= gx + gdx; end;

s:= power((gy0 / gmy), 1 / 3);

setcolor(15);

if x1 <> x2 then begin

if x1 > s then

line(gx0 + round(x1 * gmx), gy0, gx0 + round(x1 * gmx), 0)

else

if x1 > left then

line(gx0 + round(x1 * gmx), gy0, gx0 + round(x1 * gmx), gy0 - round((x1*x1*x1-x1*x1+2*x1+9) * gmy));

if x2 > left then

if x2 > s then

line(gx0 + round(x2 * gmx), gy0, gx0 + round(x2 * gmx), 0)

```

```

else

line(gx0 + round(x2 * gmx), gy0, gx0 + round(x2 * gmx), gy0 - round((x2*x2*x2-x2*x2+2*x2+9) *
gmy)); end;

gh:= 45 / sin(pi / 4) / gmx * 1 / 2;

n:= round((x2 - x1) / gh) + 10;

str(x1:15:2, ss1);

str(x2:15:2, ss2);

hh1:=hh1+ss1;

for i:=n + 50 downto -50 do

begin

gx := x1;

while gx <= x2 do

begin

gx := gx + gdx;

gpx := gx0 + round(gx * gmx);

gy1 := gx + gh * (n -i);

gpy := gy0 - round(gy1 * gmy * (gmx / gmy));

if (gpy < gy0) and (gpy > (gy0 - round((gx*gx*gx-gx*gx+2*gx+9) * gmy))) then

putpixel(gpx, gpy, 15);

end; end;

setcolor(2); // вывод информации в графическом режиме

settextstyle(1,2,1);

outtextXY(20,40, 'Graphics mode control:');

outtextXY(20, 60, 'Left or right for zoom OX. ');

outtextXY(20, 80, 'Up or down for zoom OY. ');

outtextXY(20, 100, 'Page Up or Page Down for zoom in or out. ');

outtextXY(20, 240, 'Press Esc to close vizual mode. ');

outtextXY(20,160,'Function: x^3-x^2+2x+9');

outtextXY(20,180,'Function root: -1.531');

outtextXY(20,200,'Method: trapeze');

outtextXY(20,140,'Info:');

```

```
line(15,25,15,253);  
line(340,25,340,253);  
line(15,25,340,25);  
line(15,253,340,253);  
line(15,120,340,120);  
line(15,220,340,220);  
  
end;
```

```
procedure gr();// процедура приближения графика  
var  
k: char;  
ss1,ss2: string;  
begin  
if tt>0 then begin  
gmsX := 4;  
gmsY := 4;  
gx1 := -4;  
gx2 := 4;  
gd := detect;  
initgraph(gd, gm, '');  
viz;  
repeat  
K:= wincrt.readkey;  
if k = #0 then  
begin  
k:= wincrt.readkey;  
case k of  
#77:
```



```
begin
if gmsX <> 10 then
gmsX := gmsX + 1;
cleardevice;
viz;
end;
#73:
begin
if (gmsX <> 10) and (gmsY <> 10) then
begin
gmsX := gmsX + 1; gmsY := gmsY + 1;
cleardevice;
viz end
else
begin
cleardevice;
viz
end; end;
#81:
begin
if (gmsX <> 1) and (gmsY <> 1) then
begin
gmsX := gmsX - 1; gmsY := gmsY - 1;
cleardevice;
viz; end
else
begin
cleardevice;
viz end end;
#75:
begin
```

```

if gmsX <> 1 then
gmsX := gmsX - 1;
cleardevice;
viz; end;
#72:
begin
if gmsY <> 10 then
gmsY := gmsY + 1;
cleardevice;
viz; end;
#80:
begin
if gmsY <> 2 then
gmsY := gmsY-1;
cleardevice;
viz; end; end; end;
until k= #27;
cg; end
else begin writeln('Enter coordinates!'); writeln(); writeln('Press any key'); readln(); end end;

procedure info;
begin
writeln('Function:  $x^3-x^2+2x+9$ ');
writeln('Function root: -1.531');
writeln('Antiderivative of function:  $x^4/4-x^3/3+x^2+9x$ '); //  $x^4/4-x^3/3+x^2+9x$ ;
writeln('Method: trapezoid'); writeln(); writeln('Press any key'); readln();
end;

```

```
begin
repeat
ClrScr;
textcolor(2);
writeln('1) Coordinates');
writeln('2) Calculate');
writeln('3) Measurement error');
writeln('4) Info');
writeln('5) Graphics mode');
writeln('6) Exit');
writeln('Select programm...'); writeln;
readln(n); writeln;
case n of
1:coord;
2:met;
3:er;
4:info;
5:gr;
end;
until n=6;
end.
```

