**Комп‘ютерний практикум №7**

**Тема:** Функції та покажчики на функції.

**Завдання :** Написати програму для обчислення коренів нелінійних рівнянь на заданому проміжку.

***Текст програми:***

#include <stdio.h>

#include <math.h>

char exitchar = 'y', c;

int q, p, j, eo;

int i;

//float ;

double e = 1e-10, d, x, a1, a2, a2c, t, eps;

double fun1(/\*double t,\*/ double x)

{

return (cos(t/x) - 2\*sin(1/x) + 1/x);

}

double fun2(/\*double t, \*/double x)

{

return (sin(log(x)) - cos(log(x)) + t\*log(x));

}

double fun1\_integrate(double x)

{

return ((fun1(x+e)-fun1(x))/e);

}

double fun2\_integrate(double x)

{

return ((fun2(x+e)-fun2(x))/e);

}

main()

{

do

{ system("cls");

chose\_fun();

chose\_method();

if (q==1&&p==1) {;enter\_fun1();half\_division\_method\_fun1();}

if (q==1&&p==2) {;enter\_fun1();newton\_fun1();}

if (q==2&&p==1) {;enter\_fun2();half\_division\_method\_fun2();}

if (q==2&&p==2) {;enter\_fun2();newton\_fun2();}

// printf ("xfin%f=", x);

printf ("Enter y to continue\n");

exitchar = getch();

fflush(stdin);

}

while ( (exitchar == 'y')||(exitchar == 'Y') );

return 0;

}

chose\_fun()

{

printf("This programm will solve cos(t/x)-2sin(1/x)+1/x=0, x=[a1;a2] (press 1)\n");

printf("OR\n");

printf("This programm will solve sin(lnx)-cos(lnx)+t\*lnx=0, x=[a1;a2] (press 2)\n");

do

{

if ( (scanf("%d%c", &q, &c) != 2)||(c!= '\n')||(q != 1)&&(q != 2) )

{

printf("Chose only '1' or '2'\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

}

chose\_method()

{

printf("Chose half division method (press 1)\n");

printf("OR\n");

printf("Newton method (press 2)\n");

do

{

if ( (scanf("%d%c", &p, &c) != 2)||(c!= '\n')||(p != 1)&&(p != 2) )

{

printf("Press only '1' or '2'\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

}

enter\_fun1()

{

do

{

printf("Enter 't'\n");

if ( (scanf("%lf%c", &t, &c) != 2)||(c!= '\n') )

{

printf("Enter only numbers\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

do

{

printf("Enter 'a1'\n");

if ( (scanf("%lf%c", &a1, &c) != 2)||(c!= '\n')||(a1==0) )

{

printf("Enter only numbers except 0 (x!=0)\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

do

{

printf("Enter 'a2'\n");

if ( (scanf("%lf%c", &a2, &c) != 2)||(c!= '\n')||(a2==0))

{

printf("Enter only numbers except 0 (x!=0)\n");

fflush(stdin);

j = 0;

} else if ( (a2==a1) )

{

printf("a2!=a1\n");

fflush(stdin);

j = 0;

}

else if ( a2<a1 )

{

printf("a2 must be higher than a1\n");

fflush(stdin);

j = 0;

}

else j = 1;

} while(j != 1);

do

{

printf("Enter eps (<=6): ");

if ( (scanf("%d%c", &eo, &c) != 2) || (c != '\n') || ((eo != 0) && (eo != 1) && (eo != 2) && (eo != 3) && (eo != 4) && (eo != 5) && (eo != 6)) )

{

printf("Enter only integer numbers from 0 to 6\n");

fflush(stdin);

j = 0;

} else

{

j = 1;

}

} while(j != 1);

{

eps=1/pow(10,eo);

}

}

enter\_fun2()

{

do

{

printf("Enter 't'\n");

if ( (scanf("%lf%c", &t, &c) != 2)||(c!= '\n') )

{

printf("Enter only numbers\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

do

{

printf("Enter 'a1'\n");

if ( (scanf("%lf%c", &a1, &c) != 2)||(c!='\n')||(a1<=0) )

{

printf("Enter only numbers except higher than 0 (x>0)\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

do

{

printf("Enter 'a2'\n");

if ( (scanf("%lf%c", &a2, &c) != 2)||(c!= '\n')||(a2==a1))

{

printf("a2!=a1\n");

fflush(stdin);

j = 0;

}

else if ( a2<a1 )

{

printf("a2 must be higher than a1\n");

fflush(stdin);

j = 0;

}

else j = 1;

} while(j != 1);

do

{

printf("Enter eps (<=6): ");

if ( (scanf("%d%c", &eo, &c) != 2) || (c != '\n') || ((eo != 0) && (eo != 1) && (eo != 2) && (eo != 3) && (eo != 4) && (eo != 5) && (eo != 6)) )

{

printf("Enter only integer numbers from 0 to 6\n");

fflush(stdin);

j = 0;

} else

j = 1;

} while(j != 1);

{

eps=1.0/pow(10,eo);

}

}

half\_division\_method\_fun1()

{

// float x;

a2c=a2;

// printf ("fun1=%g\n", fun1(a1, t));

// printf ("a1=%g\n", a1);

// printf ("a2=%g\n", a2);

// printf ("x=%g\n\n\n", x);

while (fabs(a2-a1)>eps)

{

// printf ("Zashlo1\n");

x=(a1+a2)/2;

(fun1(a1)\*fun1(x) > 0) ? (a1=x) : (a2=x);

// printf ("fun1=%g\n", fun1(a1, t));

// printf ("a1=%g\n", a1);

// printf ("a2=%g\n", a2);

// printf ("\n\n\neps=%g\n", eps);

// printf ("a2c=%g\n", a2c);

// printf ("x=%g\n", x);

if ( x+eps>=a2c)

{

printf ("There is no answers in this part\n");

return 0;

}

}

printf ("x=%g\n", x);

return x;

}

half\_division\_method\_fun2()

{

// float x;

a2c=a2;

// printf ("fun1=%g\n", fun1(a1, t));

// printf ("a1=%g\n", a1);

// printf ("a2=%g\n", a2);

// printf ("x=%g\n\n\n", x);

while (fabs(a2-a1)>eps)

{

// printf ("Zashlo1\n");

x=(a1+a2)/2;

(fun2(a1)\*fun2(x) > 0) ? (a1=x) : (a2=x);

// printf ("fun1=%g\n", fun1(a1, t));

// printf ("a1=%g\n", a1);

// printf ("a2=%g\n", a2);

// printf ("\n\n\neps=%g\n", eps);

// printf ("a2c=%g\n", a2c);

// printf ("x=%g\n", x);

if ( x+eps>=a2c)

{

printf ("There is no answers in this part\n");

return 0;

}

}

printf ("x=%g\n", x);

return x;

}

newton\_fun1()

{

//printf ("Zashlo1\n");

// double x;

// double d;

i=0;

x=fun1(a2);

// printf ("x=%g\n", x);

do

{

// printf ("Zashlo\n");

d=fun1(x)/fun1\_integrate(x);

// printf ("fun1=%g\n", fun1(x));

// printf ("fun1\_integrate=%g\n", fun1\_integrate(x));

// printf ("d=%g\n", d);

x = x - d;

// printf ("xx=%g\n", x);

// i++;

/\* if (i>100)

{

printf ("There is no answers in this part (i>100)\n");

return 0;

} \*/

if ( (x>a2)||(x<a1) )

{

printf ("Programm can not find answers by this method in those limits\n");

return 0;

}

}

while(fabs(d) >= eps);

{

printf ("x=%g\n", x);

return x;

}

}

newton\_fun2()

{

// float x;

// float d;

x=fun2(a2);

if ( (x>a2)||(x<a1) )

{

printf ("Programm can not find answers by this method in those limits1\n");

return 0;

}

// printf ("x=%f\n", x);

// i=0;

do

{

// printf ("Zashlo\n");

d=fun2(x)/fun2\_integrate(x);

// printf ("fun2\_integrate=%f\n", fun2\_integrate(x));

// printf ("fun2=%f\n", fun2(x));

// printf ("d=%f\n", d);

x = x - d;

// printf ("x1=%f\n", x);

/\*if (i>100)

{

printf ("There is no answers in this part (i>100)\n");

return 0;

} \*/

if ( (x>a2)||(x<a1) )

{

printf ("Programm can not find answers by this method in those limits\n");

return 0;

}

}

while(fabs(d) > eps);

{

printf ("x=%g\n", x);

return x;

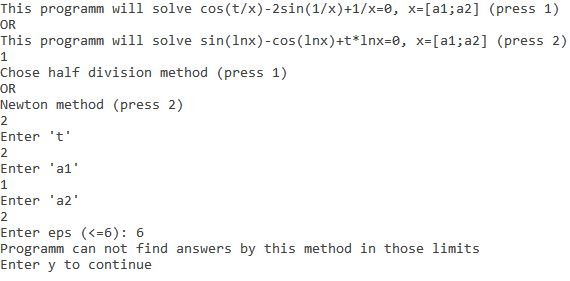
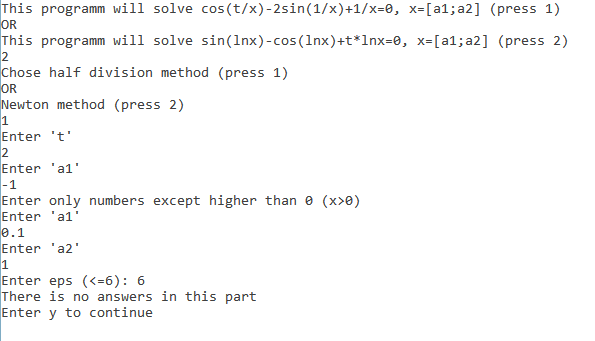
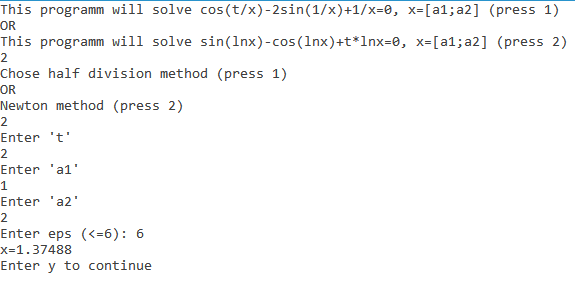
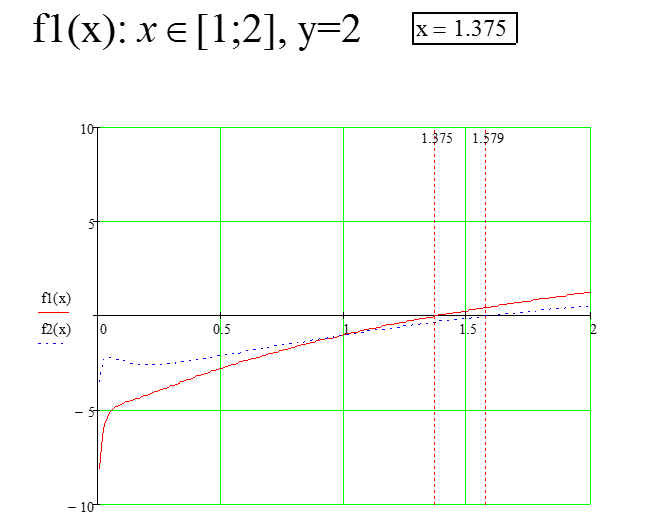
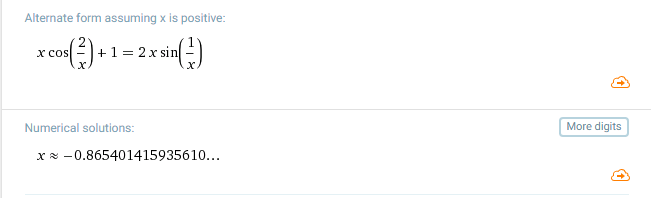
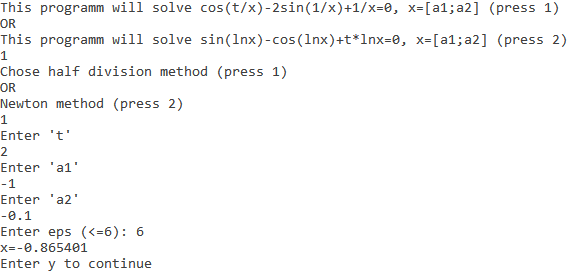
}

}

***Схема до програми:***



***Введені та одержані результати:***



***Теоретичні розрахунки:***

***Висновки***: Програма вирішує поставлене завдання. Теоретичні розрахунки відповідають отриманим. Програма працює корректно.