

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Інформатика 1»

	<p>Виконав: Сологуб Артем Миколайович</p> <p>Група: РЕ-11</p> <p>Викладачі: доцент Катін П.Ю.</p> <p>Оцінка: _____</p> <p>Підпис: _____</p>
--	---

Київ – 2021

**Мета роботи:** Створення програми для розв'язання нелінійних рівнянь

**Ключові моменти:**

1) Для того щоб отримати час затрачений на знаходження кореня я використовую бібліотеку `<time.h>` та функцію `clock()` яка повертає значення часу у даний момент.

Пишу на початку обчислень `clock_t t1= clock();`

Функція `clock()` повертає час, пройдений з моменту старту програми, що визвала функцію `clock()`.

2) За допомогою `getch()` записую у змінну `i` значення яке ввів користувач з клавіатури.

`i = getch();`

4) `system("cls");` використовується для очищення консолі

**Код:**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <math.h>
```

```
#include <conio.h>
```

```
#include <time.h>
```

```
double F2(double x)
```

```
{
```

```
    return pow(x,3)+3.5*x;
```

```
}
```

```
void Dump(double xi, double fault,int N, double time)
```

```
{
```

```
    printf("xi = %lf \n", xi);
```

```

printf("fault = %lf \n", fault);

printf("Count of iterations = %d \n", N);

printf("Calculation time = %lf \n", time);
}

int main()
{

double x1, x2, xi, time,fault;

unsigned int N, n=0, var;

char i;

printf("1 method \n 2 method \n");

scanf("%d",&var);

switch(var)
{
case 1:

do

{

printf("Left limit: \n");

scanf("%lf", &x1);

printf("Right limit: \n");

scanf("%lf",&x2);


while(F2(x1)>0 && F2(x2)<0);

printf("Fault: \n");

scanf("%lf", &fault);

printf("Max count of iterations: \n");

clock_t t1= clock();

do

{

```

```

Xi=((F2(x2)*x1)-(F2(x1)*x2))/(F2(x2)-F2(x1));

if(F2(xi)>0)

{

    x2=xi;

}

else

{

    x1=xi;

}

if(n==N)

{

    printf(" number of iterations \n");

    printf("Press the esc key to complete or any other to continue \n");

    i=getch();

    if((int)i==27)

    {

        system("cls");

    }

    else

    {

        system("cls");

    }

}

n++;

}

while(fabs(F2(xi))>fault);

clock_t t2=clock();

F3(xi, fabs(F2(xi)), n, (double)(t2-t1)/CLOCKS_PER_SEC);

break;

```

case 2:

```
do
{
    printf("Left limit: \n");
    scanf("%lf", &x1);
    printf("Right limit: \n");
    scanf("%lf",&x2);
}
while(F2(x1)>0 && F2(x2)<0);
printf("Fault: \n");
scanf("%lf", &fault);
printf("max count iterations\n");
clock_t t3=clock();
scanf("%d", &N);
do
{
    xi=(x1+x2)/2;
    if(F2(xi)>0)
    {
        x2=xi;
    }
    else
    {
        x1=xi;
    }
    if(n==N)
    {
        printf("number of iterations reached \n");
        printf("Press only the esc key to complete or any other to continue \n");
```

```
i=getch();

if((int)i==27)

{

    system("cls");

    break;

}

else

{

    system("cls");

}

}

n++;

}

while(fabs(F2(xi))>fault);

clock_t t4=clock();

F3(xi, fabs(F2(xi)), n, (double)(t4-t3)/CLOCKS_PER_SEC);

break;

}

return 0;

}
```