

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

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2

з дисципліни: «Інформатика 1»

	<p>Виконав : Рокунець Андрій Олександрович</p> <p>Група: РЕ-12</p> <p>Викладачі: доцент Катін П.Ю.</p> <p>Оцінка: _____</p> <p>Підпис: _____</p>
--	--

Лабораторна робота №2

1. Мета: написати програму для розрахунку інтегралів 4 способами.
2. Задана функція: $y=dx/\text{row}(x,2)$. Межі: -2,1.

3. Метод	n = 10	n = 100	n = 1000	n = 10000	N
Лівих прямокутників	3.495	3.04545	3.0045045	3.00045005	300
Правих прямокутників	3.495	3.04545	3.0045045	3.00045005	300
Трапецій	4.098	3.092223	3.00902247	3.00090022	426
Сімпсон	4.092	3.092082	3.00902098	3.00090021	426

4. Код:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
```

```
unsigned int method_select( void );
double f( double x );
void print_result(double a, double b, unsigned int n, double l );
double left_rectangle( double a, double b, unsigned int n );
double right_rectangle( double a, double b, unsigned int n );
double trapeze( double a, double b, unsigned int n );
double Simpson( double a, double b, unsigned int n );
```

```
int main()
{
    unsigned int num_of_method; // 1 - left, 2 - right, 3 - trapeze,
    4 - Simpson
    double l, l1, l2, a , b;
    unsigned int N, n;
    double eps = 0.0001;
```

```
printf("a=");
scanf("%lf",& a);
printf("b=");
scanf("%lf",& b);
printf("n=");
scanf("%u",& n);
```

```
num_of_method = method_select();
```

```
if (num_of_method == 1 ){
    l = left_rectangle(a, b, n);
    print_result(a, b, n, l);
```

```
    N = 0;
    do{
        N = N + 2;
        l1 = left_rectangle(a, b, N);
        l2 = left_rectangle(a, b, N+2);
    }while( fabs(l2-l1) > eps );
```

```
    printf("\n\nN=%u, l1(N)=%lf\n", N, l1);
}
```

```
else if( num_of_method == 2 ){
    l=right_rectangle(a, b, n);
    print_result(a, b, n, l);
    N = 0;
    do{
        N = N + 2;
        l1 = right_rectangle(a, b, N);
        l2 = right_rectangle(a, b, N+2);
    }while( fabs(l2-l1) > eps );
```

```

        printf("\n\nN=%u, I1(N)=%lf\n", N, I1);
    }
    else if( num_of_method == 3 ){
        I=trapeze(a, b, n);
        print_result(a, b, n, I);
        N = 0;
        do{
            N = N + 2;
            I1 = trapeze(a, b, N);
            I2 = trapeze(a, b, N+2);
        }while( fabs(I2-I1) > eps );

        printf("\n\nN=%u, I1(N)=%lf\n", N, I1);
    }
    else if( num_of_method == 4 ){
        I=Simpson(a, b, n);
        print_result(a, b, n, I);
        N = 0;
        do{
            N = N + 2;
            I1 = Simpson(a, b, N);
            I2 = Simpson(a, b, N+2);
        }while( fabs(I2-I1) > eps );

        printf("\n\nN=%u, I1(N)=%lf\n", N, I1);
    }
    return 0;
}

```

```

unsigned int method_select(void)
{
    unsigned int rest;

```

```
printf("\n\n Choose your method:\n1 - left \n2 - right \n3 -  
trapeze \n4 - Simpson\n>");  
scanf("%u", &rest);
```

```
while( rest < 1 || rest > 4 ){  
    printf("\nERROR. Choose Method: 1, 2, 3 or 4: ");  
    scanf("%u", &rest);  
}
```

```
switch(rest){  
    case 1:  
        printf("\nYou chose method of LEFT RECTANGLE");  
        break;  
  
    case 2:  
        printf("\nYou chose method of RIGHT RECTANGLE");  
        break;  
  
    case 3:  
        printf("\nYou chose method of TRAPECIUM");  
        break;  
  
    case 4:  
        printf("\nYou chose method of SIMPSON");  
        break;  
}
```

```
// left_rectangle  
return rest;  
}  
double f( double x )  
{  
    double y;  
    y = pow(x,2);  
    return y;  
}
```

```

}
double left_rectangle(double a, double b, unsigned int n)
{
    double h;
    double x;
    double sum = 0;

    h = ( b - a ) / n;
    x = a;

    for (int k = 0; k <= n-1; k++ ){
        sum = sum + f(x);
        x = x + h;
    }

    return sum * h;
}

void print_result(double a, double b, unsigned int n, double l)
{
    system("cls");
    printf("\n*****");
    printf("\n*      Results      *");
    printf("\n*****\n");

    printf("a=%.2lf b=%.2lf n=%u l = %.8lf", a, b, n, l);
}

// right_rectangle
double right_rectangle(double a, double b, unsigned int n)
{
    double h;
    double x;
    double sum = 0;
    h = ( b - a ) / n;
    x = a;
    for (int k = 1; k <= n; k++ ){

```

```

        sum = sum + f(x);
        x = x + h;
    }
    return sum * h;
}

void print_result2(double a, double b, unsigned int n, double l)
{
    system("cls");
    printf("\n*****");
    printf("\n*    Results    *");
    printf("\n*****\n");

    printf("a=%.2lf b=%.2lf n=%u l = %.8lf", a, b, n, l);
}

// trapeze
double trapeze(double a, double b, unsigned int n)
{
    double h;
    double x;
    double sum = f(a) + f(b);
    h = ( b - a ) / n;
    x = a;
    for (int k = 1; k <= n-1; k++) {
        sum = sum + 2*f(x);
        x = x + h;
    }
    return sum * h/2;
}

void print_result3(double a, double b, unsigned int n, double l)
{
    system("cls");
    printf("\n*****");
    printf("\n*    Results    *");
    printf("\n*****\n");
    printf("a=%.2lf b=%.2lf n=%u l = %.8lf", a, b, n, l);
}

```

```

}
// Simpson
double Simpson(double a, double b, unsigned int n)
{
    double h;
    double x;
    double sum = f(a) + f (b);

    h = ( b - a ) / n;
    x = a;
    int i;
    for (int k = 1; k <= n-1; k++ ){
        i = 2 + 2*(k % 2);
        sum = sum + i*f(x);
        x = x + h;
    }
    return sum * h/3;
}

void print_result4(double a, double b, unsigned int n, double l)
{
    system("cls");
    printf("\n*****");
    printf("\n*    Results    *");
    printf("\n*****\n");
    printf("a=%.2lf b=%.2lf n=%u l = %.8lf", a, b, n, l);

}

```


