Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2**

з дисципліни: «Інформатика 1

|  |  |
| --- | --- |
|  | Виконав:  Демчук В’ячеслав Андрійович  Група: РЕ-11  Викладач: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

Мета роботи: навчитися працювати з функціями користувача та за їх допомогою скласти програму для обчислення визначеного інтегралу кількома методами, дослідити отримані результати.

Функція: x^2

Код програми:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

double function(double x){

return x = x\*x;

}

double right\_columns(double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit;

double integral = 0;

for (int i = 1; i <= interval; i++)

{

integral = integral + function(pointer) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double left\_columns (double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit + width\_columns;

double integral = 0;

for (int i = 1; i <= interval; i++)

{

integral = integral + function(pointer) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double trapeze (double left\_limit, double right\_limit, int interval) {

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit;

double integral = 0;

for (int i = 1; i < interval; i++)

{

integral = integral + ((function(pointer) + function(pointer + width\_columns)) / 2) \* width\_columns;

pointer = left\_limit + i \* width\_columns;

}

return integral;

}

double simpson(double left\_limit, double right\_limit, double interval) {

double integral=0;

double width\_columns = (right\_limit - left\_limit) / interval;

double pointer = left\_limit + width\_columns;

while (pointer < right\_limit)

{

integral += 4\*function(pointer);

pointer += width\_columns;

if (pointer >= right\_limit) break;

integral += 2\*function(pointer);

pointer += width\_columns;

}

integral = (width\_columns / 3) \* (integral + function(left\_limit) + function(right\_limit));

return integral;

}

int main()

{

setlocale(LC\_ALL, "Ukr");

double first\_integral, second\_integral, left\_limit, right\_limit, error;

int spacing\_option, metod\_integration;

unsigned long int counter, interval = 2;

{

printf("\n\n\t++===================================================++\n");

printf("\t|| Лабораторна робота №2 ||\n");

printf("\t++---------------------------------------------------++\n");

printf("\t|| Обчислити визначений інтеграл ||\n");

printf("\t++===================================================++\n\n");

}

{//введення початкових данних

printf("\t Введіть ліву межу\t");

scanf("%lf", &left\_limit);

printf("\t Введіть праву межу\t");

scanf("%lf", &right\_limit);

}

{//вибір метода

printf("\n\t\t 1 - Правих прямокутників");

printf("\n\t\t 2 - Лівих прямокутників");

printf("\n\t\t 3 - Трапецій");

printf("\n\t\t 4 - Сімпсона");

printf("\n\t Оберіть метод обчислення інтеграла:\t");

scanf("%d", &metod\_integration);

}

{//вибір способа

printf("\n\t\t 1 - Через розраховану кількість проміжків");

printf("\n\t\t 2 - Через свою кількість проміжків");

printf("\n\t Оберіть спосіб обчислення визначеного інтеграла:\t");

scanf("%d", &spacing\_option);

}

if (spacing\_option == 1)

{

printf("\n\t\t Від 3 до 5 порядку");

printf("\n\t Введіть погрішність для обчислення визначеного інтеграла:\t");

scanf("%lf", &error);

error = 1/pow(10, error);

switch (metod\_integration) {

case 1: {

do {

first\_integral = right\_columns (left\_limit, right\_limit, interval);

second\_integral = right\_columns (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 2: {

do {

first\_integral = left\_columns (left\_limit, right\_limit, interval);

second\_integral = left\_columns (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 3: {

do {

first\_integral = trapeze (left\_limit, right\_limit, interval);

second\_integral = trapeze (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

case 4: {

do {

first\_integral = simpson (left\_limit, right\_limit, interval);

second\_integral = simpson (left\_limit, right\_limit, interval+=2);

counter = interval;

} while (fabs(second\_integral - first\_integral) >= error);} break;

}}

if (spacing\_option == 2)

{

printf("\n\t Введіть кількість проміжків для обчислення визначеного інтеграла:\t");

scanf("%lu", &counter);

switch (metod\_integration) {

case 1: {

first\_integral = right\_columns (left\_limit, right\_limit, counter);

} break;

case 2: {

first\_integral = left\_columns (left\_limit, right\_limit, counter);

} break;

case 3: {

first\_integral = trapeze (left\_limit, right\_limit, counter);

} break;

case 4: {

first\_integral = simpson (left\_limit, right\_limit, counter);

} break;

}}

{//результати

printf("\n\t Ліва межа:\t");

printf("%lf", left\_limit);

printf("\n\t Права межа:\t");

printf("%lf", right\_limit);

printf("\n\t Кількість проміжків:\t");

printf("%lu", counter);

printf("\n\t Погрішність обчислень:\t");

printf("%lf", error);

printf("\n\n\t Початкова функція:\t");

printf("x^2");

printf("\n\t Аналітичне інтегрування:\t");

printf("(x^3)/3");

printf("\n\t Результат обчислень:\t");

printf("%lf\n\n", first\_integral);

}

printf("\n\t\t\tПісля натиснення будь-якої клавіші програма повториться\n\n");

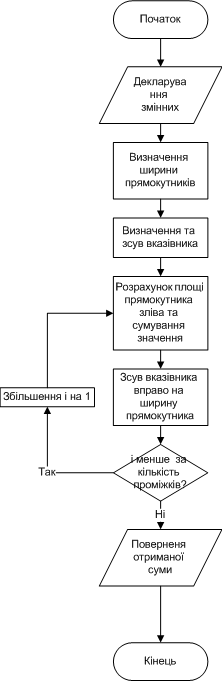
getch(); system("cls"); main();

return 0;

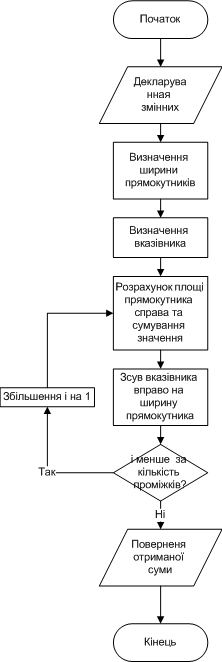
}

Блок-схеми програми:

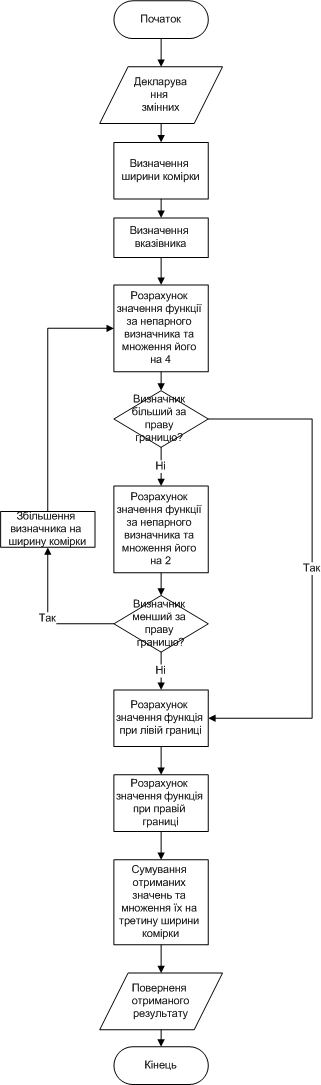
Метод лівих прямокутників:



Метод правих прямокутників:



Метод Сімпсона:



Метод трапецій:

