НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп’ютерних систем

Лабораторна робота №4

з дисципліни “Алгоритми та методи обчислень”

Тема: “ **ЧИСЛЕННІ ІНТЕГРУВАННЯ** ”

Варіант 4

Виконала: студентка III курсу

КВ-72 Дорош Карина

Перевірив: Зорін Ю. М.

Київ 2019

Завдання для лабораторної роботи

1. Відповідно до варіанту (табл. 4.2) за допомогою залишкового члена АНАЛІТИЧНО визначити крок інтегрування h, що забезпечує необхідну точність ε (визначається у тексті програми).

Обчислити інтеграл I_h з кроком h і визначити абсолютну похибку Δ, прийнявши за точне значення, обчислене за формулою Ньютона-Лейбніца. Результати п.1 подати у вигляді:

Задана похибка, ε	Крок інтегрування	Точне значення інтеграла	Отримана похибка, Δ

2. За допомогою методу подвійного перерахунку досягти тієї ж похибки Δ, що й у п.1. Вивести значення отриманого кроку й абсолютної похибки. Результати п.2 подати у вигляді:

Задана похибка, Δ	Крок інтегрування	Отримана похибка

Варіант	Інтеграл	Первісна
4	5	6
4	$\int_0^{15} \frac{\sqrt{2x+1}}{2}\,dx$	$\frac{2x+1}{6}\sqrt{2x+1}$

Код програми

main.cpp

```cpp
#include "header.h"

int main() {
	//calculating integral by Newton-Leibniz formula
	double NewtonLeibniz = funcAntiderivative(b) - funcAntiderivative(a);
	cout << "\nIntegral sqrt(2x+1)/2 from 0 to 15:" << endl;
	cout << "\nI = " << setprecision(10) << NewtonLeibniz << endl << endl;
	double delta = firstTable(NewtonLeibniz);
	secondTable(delta, NewtonLeibniz);
	return 0;
}
```

header.h

```cpp
#pragma once
#include <iostream>
#include <cmath>
#include <iomanip>
#define b 15
#define a 0
#define h 0.04
using namespace std;
double func(double x);
double funcAntiderivative(double x);
double methodTrapezoid(int n);
double firstTable(double NewtonLeibniz);
void secondTable(double delta, double NewtonLeibniz);
```

calculating.cpp

```cpp
#include "header.h"
double func(double x) {
	return sqrt(2 * x + 1) / 2;

}

double funcAntiderivative(double x) {
	return (2 * x + 1) / 6 * sqrt(2 * x + 1);
}

double methodTrapezoid(int n) {
	if (n < 1)
		return 0;
	double result = func(a) / 2;
	double h1 = (b - a) / (double) n;
	for (int i = 1; i < n; i++) {
		result += func(a + h1 * i);
```

```
        }
        result += func(b) / 2;
        return result * h1;
}

double firstTable(double NewtonLeibniz) {
        double eps = 0.01;
        //double h = sqrt(12 * eps / (b - a));
        int n = (b - a) / h;
        double integralUsingTrapezoid = methodTrapezoid(n);
        double delta = fabs(NewtonLeibniz - integralUsingTrapezoid);
        cout << " ---------------------------------------------------" << endl;
        cout << "| eps    |       h       |     I       |      delta     |" << endl;
        cout << " ---------------------------------------------------" << endl;
        cout << "|" << eps << setw(3) << "  |    " << h << setw(7) <<" |" <<
integralUsingTrapezoid << "|" << delta << setw(2) << "|" << endl;
        cout << " ---------------------------------------------------" << endl;

        return delta;
}

void secondTable(double delta, double NewtonLeibniz) {
        //using method of multiply recalculation
        int n = 1 / sqrt(delta);
        double integralUsingTrapezoid1 = methodTrapezoid(n);
        n *= 2;
        double integralUsingTrapezoid2 = methodTrapezoid(n);
        cout << " ------------------------------------------" << endl;
        cout << "|     delta     |        h        |     Rh     |" << endl;
        //since in the generalized trapezoidal formula the order p is 2, then here we
divide by 3
        do{
               integralUsingTrapezoid1 = integralUsingTrapezoid2;
               n *= 2;
               integralUsingTrapezoid2 = methodTrapezoid(n);
               cout << " ------------------------------------------" << endl;
               cout << "|" << setprecision(9) << delta << "|" << setprecision(12) << (b -
a) / (double)n << "|" << setprecision(5) << fabs(NewtonLeibniz - integralUsingTrapezoid2)
<< "|" << endl;
        }while (fabs(integralUsingTrapezoid2 - integralUsingTrapezoid1) / 3 > delta);


        cout << " ------------------------------------------" << endl;

}
```

WolframAlpha computational intelligence.

integrate from 0 to 15 sqrt(2*x +1)/2

Extended Keyboard Upload Examples Random

Assuming "from 0 to 15" is referring to variable ranges | Use "from" as a word instead

Definite integral: More digits Step-by-step solution

$$\int_0^{15} \frac{1}{2}\sqrt{2x+1}\,dx = \frac{1}{6}\left(31\sqrt{31}-1\right) \approx 28.600$$