**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

**Індивідуальне завдання №1**

з дисципліни «Алгоритмізація та програмування»

Тема роботи: «Рекурентні співвідношення. Обробка файлів даних.»

**Варіант №1**

**Виконав студент**

**групи КН-12**

**Сімак О. С.**

**Перевірив**

**Круглов О. І.**

**Київ – 2020**

**Завдання 1**

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: a, b – проміжок, змінні дійсного типу.

*Вихідні дані*: res – результати розрахунку дробового типу.

*Математична модель задачі*:cres = h \*

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| a, b – проміжок, змінні дійсного типу. | 1) res =h \*  2) Вивести res | res – результати розрахунку дробового типу. |

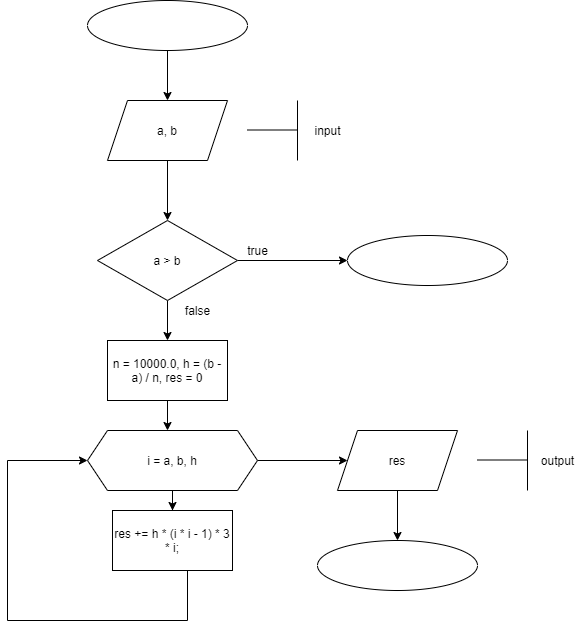


Рисунок 1 - Схема алгоритму задачі 1.

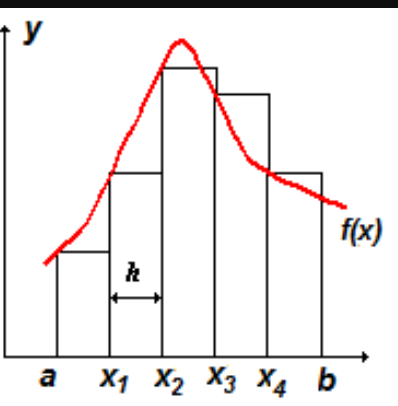
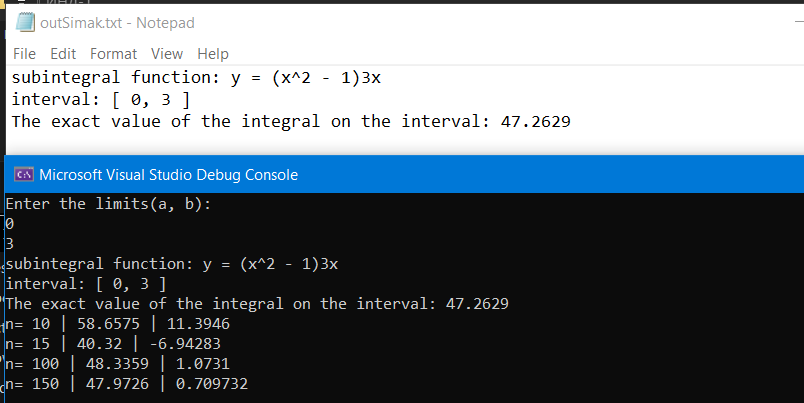


Рисунок 2 – Графічне подання методу лівих прямокутників.

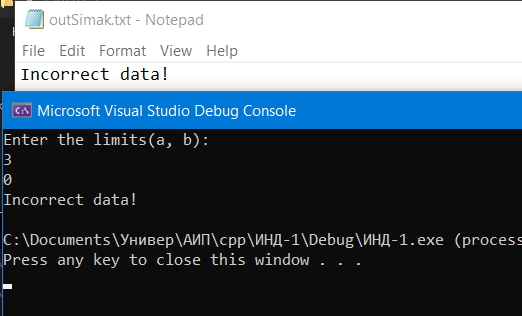
Метод лівих прямокутників отримав свою назву через те, що висота прямокутників на проміжних відрізках дорівнюють значенням функції в лівих кінцях даних відрізків.

1. **Тестові приклади.**

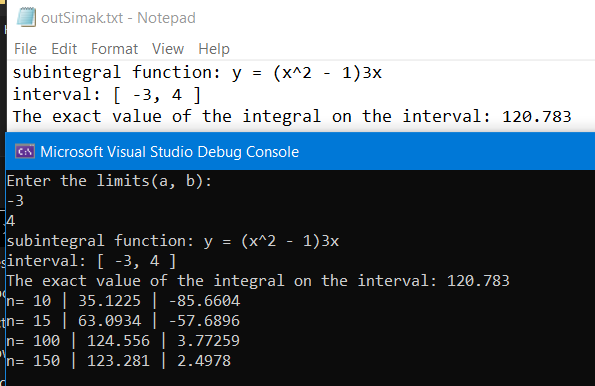
* a = 0; b = 3;



* a = 3; b = 0;



* a = -3; b = 4;



1. **Текст програми:**

#include <iostream>

#include <fstream>

using namespace std;

double function(double x) {

return (x \* x - 1) \* 3 \* x;

}

int tableOutput(int a, int b, float res, float n) {

float newRes = 0, h = (b - a) / n;

for (float i = a; i < b; i += h) {

newRes += h \* function(i);

}

cout << "n= " << n << " | " << newRes << " | " << newRes - res << endl;

return 0;

}

// Метод левых треугольников

int main()

{

ofstream fout("outSimak.txt");

if (!fout) {

cout << "File for writing isn't open! \n";

system("pause");

return 0;

}

int a, b;

cout << "Enter the limits(a, b): " << endl;

cin >> a >> b;

if (a > b) {

cout << "Incorrect data!" << endl;

fout << "Incorrect data!" << endl;

return 0;

}

double n = 10000.0, h = (b - a) / n, res = 0;// n - кол-во интервалов, h - шаг разбиения

for (float i = a; i < b; i += h) {

res += h \* function(i);

}

cout << "subintegral function: y = (x^2 - 1)3x" << endl << "interval: [ " << a << ", " << b << " ]" << endl << "The exact value of the integral on the interval: " << res << endl;

fout << "subintegral function: y = (x^2 - 1)3x" << endl << "interval: [ " << a << ", " << b << " ]" << endl << "The exact value of the integral on the interval: " << res << endl;

fout.close();

//для таблицы вывода

tableOutput(a, b, res, 10.0);

tableOutput(a, b, res, 15.0);

tableOutput(a, b, res, 100.0);

tableOutput(a, b, res, 150.0);

return 0;

}