

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 12

Виконав студент Дойчев Костянтин Миколайович
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота №6

Тема: Дослідження рекурсивних алгоритмів

Мета – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Варіант 12

1) Постановка задачі:

Обчислити суму елементів арифметичної прогресії, що убуває:
початкове значення – 3π , кінцеве – -4π , крок – $\pi/2$

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Створимо функцію для обчислення суми елементів арифметичної прогресії, що убуває

2) Побудова математичної моделі:

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Початкове значення прогресії	Дійсний	initialValue	Вхідні дані
Крок зменшення членів прогресії	Дійсний	step	Вхідні дані

Кінцеве значення прогресії	Дійсний	finalValue	Вхідні дані
Сума членів прогресії	Дійсний	sum	Вихідні дані
Функція розрахунку суми арифметичної прогресії	Дійсний	calculateNegativeArithmeticSum(initialValue, step, finalValue)	Проміжні дані

Таким чином, математичне формулювання задачі зводиться до реалізації рекурсивного алгоритму розрахунку спадної арифметичної прогресії.

Створимо рекурсивну ф-цію calculateNegativeArithmeticSum, яка прийматиме 3 аргументи:

- 1) Початкове значення (initialValue)
- 2) Крок прогресії (step)
- 3) Кінцеве значення (finalValue)

Ця функція буде викликати сама себе до тих пір, поки initialValue не буде дорівнювати finalValue, в такому разі в змінну sum запишеться це finalValue, а тоді зворотнім ходом рекурсії ми будемо додавати до початкового значення, значення, яке повернула попередній виклик функції.

3) Псевдокод алгоритму

Крок 1:

Підпрограма

calculateNegativeArithmeticSum (initialValue, step, finalValue, sum)

Реалізація рекурсивного алгоритму

Все підпрограма

Початок

Ініціалізація змінних

Виклик функції calculateNegativeArithmeticSum(initialValue,step,finalValue,
sum)

_____ Вивидення sum

Кінець

Крок 2:

Підпрограма

calculateNegativeArithmeticSum (initialValue, step, finalValue, sum)

якщо initialValue == finalValue

то

sum = initialValue;

інакше

sum = initialValue + calculateNegativeArithmeticSum
(initialValue - step, step, finalValue)

повернути sum

Все підпрограма

Початок

Ініціалізація змінних

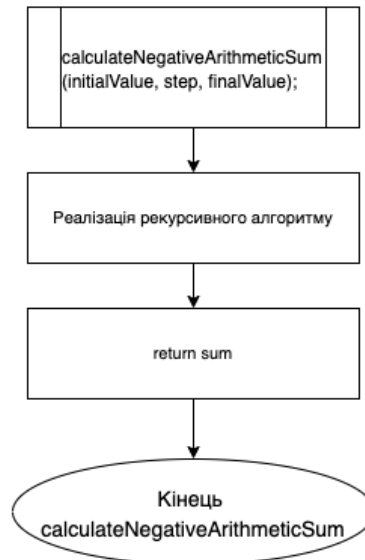
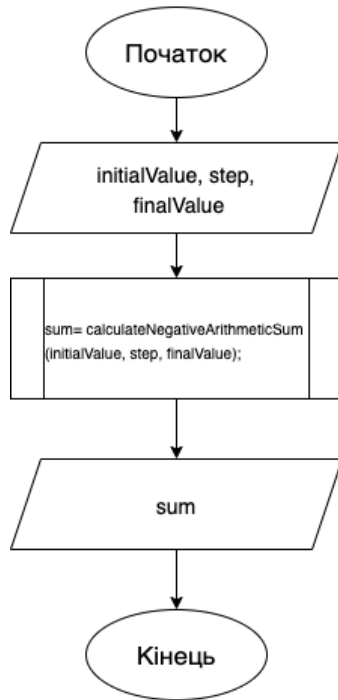
Виклик функції calculateNegativeArithmeticSum(initialValue,step,finalValue)

_____ Вивидення sum

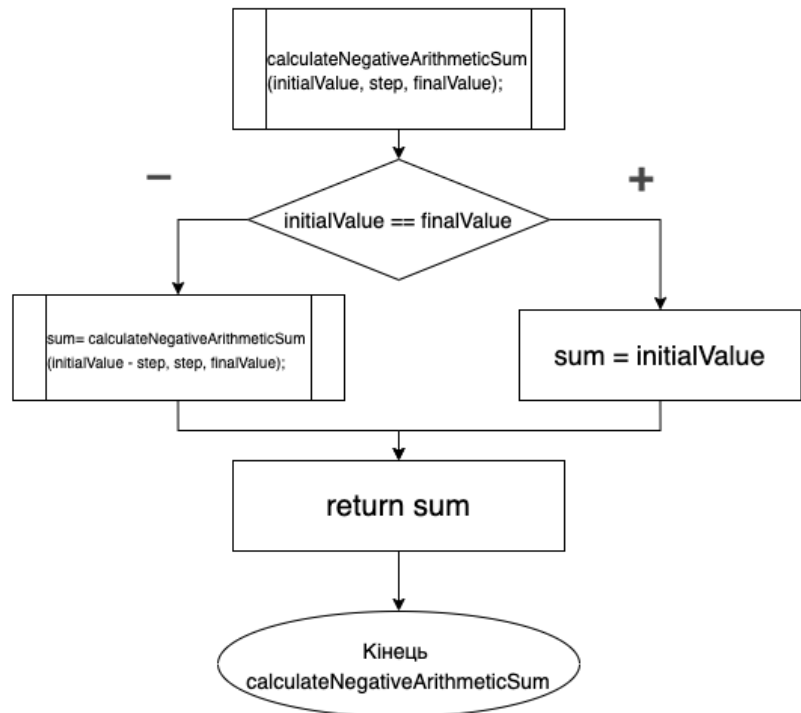
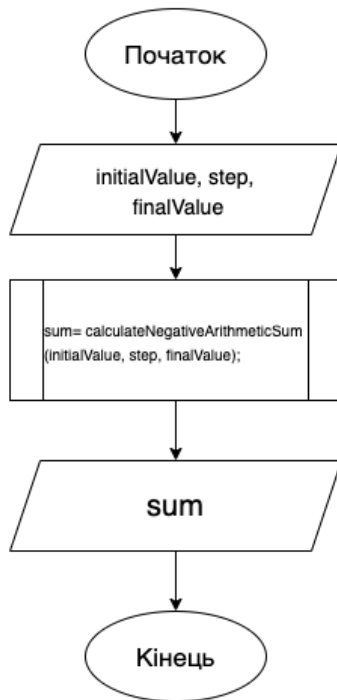
Кінець

4) Блок схема алгоритму

Крок 1



Крок 2



5) Код програми:

```

1  #define _USE_MATH_DEFINES
2  #include <iostream>
3  #include <cmath>
4
5  double calculateNegativeArithmeticSum(double initialValue, double step, double finalValue, double sum);
6
7  using namespace std;
8
9  int main() {
10     double initialValue = 3 * M_PI;
11     double step = M_PI_2;
12     double finalValue = -4 * M_PI;
13     double sum = 0;
14     sum = calculateNegativeArithmeticSum(initialValue, step, finalValue, sum);
15     cout << "Sum: " << sum << endl;
16     return 0;
17 }
18
19 double calculateNegativeArithmeticSum(double initialValue, double step, double finalValue, double sum) {
20     if (initialValue == finalValue) {
21         sum = initialValue;
22     } else {
23         sum = initialValue + calculateNegativeArithmeticSum(initialValue - step, step, finalValue, sum);
24     }
25     return sum;
26 }
27

```

6) Випробування алгоритму:

Блок	Дія
	Початок
1	initialValue:= 3π; finalValue = -4π; step= π/2;
2	calculateNegativeArithmeticSum(3π,π/2, -4π);
3	(3π == -4π) == false
4	sum = 3π + calculateNegativeArithmeticSum(3π - π/2, π/2, -4π);
5	(7.85 == -12.5) == false
6	sum = 7.8 + calculateNegativeArithmeticSum(6.28,1.57, -12.5);
7	(6.2 == -12.5) == false
8	sum = 6.2 + calculateNegativeArithmeticSum(4.7,1.57, -12.5);
9	(4.7 == -12.5) == false

10	<code>sum = 4.7 + calculateNegativeArithmeticSum(3.14,1.57, -12.5);</code>
11	<code>(3.14 == -12.5) == false</code>
12	<code>sum = 3.14 + calculateNegativeArithmeticSum(1.5,1.57, -12.5);</code>
13	<code>(1.5 == -12.5) == false</code>
14	<code>sum = 1.5 + calculateNegativeArithmeticSum(0,1.57, -12.5);</code>
15	<code>(0 == -12.5) == false</code>
16	<code>sum= 0 + calculateNegativeArithmeticSum(-1.5,1.57, -12.5);</code>
17	<code>(-1.5 == -12.5) == false;</code>
18	<code>sum = -1.5 + calculateNegativeArithmeticSum(-3.14,1.57, -12.5);</code>
19	<code>(-3.14 == 12.5) == false</code>
20	<code>sum = -3.14 + calculateNegativeArithmeticSum(-4.7,1.57, -12.5);</code>
21	<code>(-4.7 == -12.5) == false;</code>
22	<code>sum= -4.7 + calculateNegativeArithmeticSum(-6.28,1.57, -12.5);</code>
23	<code>(-6.28 == -12.5) == false</code>
24	<code>sum= -6.28 + calculateNegativeArithmeticSum(-7.8,1.57, -12.5);</code>
25	<code>(-7.8 == -12.5) == false;</code>
26	<code>sum = -7.8 + calculateNegativeArithmeticSum(-9.4,1.57, -12.5);</code>
27	<code>(-9.4 == -12.5) == false</code>
28	<code>sum= -9.4 + calculateNegativeArithmeticSum(-10.9,1.57, -12.5);</code>
29	<code>(-10.9 == -12.5) == false</code>
30	<code>sum= -10.9 + calculateNegativeArithmeticSum(-12.5,1.57, -12.5);</code>
31	<code>(-12.5 == -12.5) == true</code>
32	<code>sum = -12.5</code>
33	<code>sum = -10.9 + (-12.5); sum = -23.5</code>

34	return -23.5
35	sum = $-9.4 + (-23.5)$; sum = -32.9;
36	return -32.9;
37	sum = $-7.8 + (-32.9)$; sum = -40.8;
38	return -40.8
39	sum = $-6.2 + (-40.8)$; sum = -47.1;
40	return -47.1
41	sum = $-4.7 + (-47.1)$; sum = -51.8
42	return -51.8
43	sum = $-3.14 + (-51.8)$; sum = -54.9
44	return -54.9
45	sum = $-1.5 + (-54.9)$; sum = -56.5
46	return -56.5;
47	sum = $0 + (-56.5)$; sum = -56.5
48	return -56.5
49	sum = $1.5 + (-56.5)$; sum = -54.9
50	return -54.9
51	sum = $3.14 + (-54.9)$; sum = -51.8
52	return -51.8
53	sum = $4.7 + (-51.8)$; sum = -47.1
54	return -47.1
55	sum = $6.2 + (-47.1)$; sum = -40.8;
56	return -40.8
57	sum = $7.8 + (-40.8)$; sum = -32.9

58	return -32.9;
59	sum = 9.4 + (-32.9); sum = -23.5
60	return -23.5
61	виведення-23.5
	Кінець

7) Виновки:

Дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. Розв'язав задачу, розробив мат. модель та псевдокод. Побудував блок схему, протестував алгоритм та написав код.