

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

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:

Функція calculateNegativeArithmeticSum (initialValue, step, finalValue)

Реалізація рекурсивного алгоритму

Все функція

Початок

Ініціалізація змінних

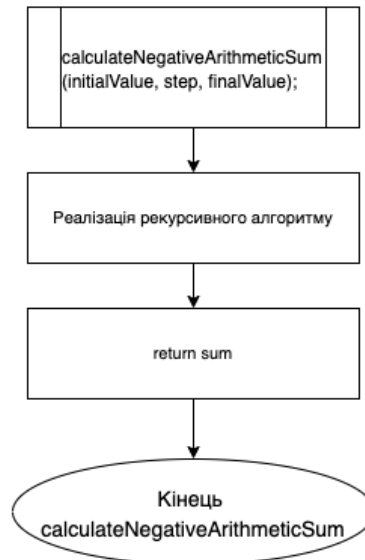
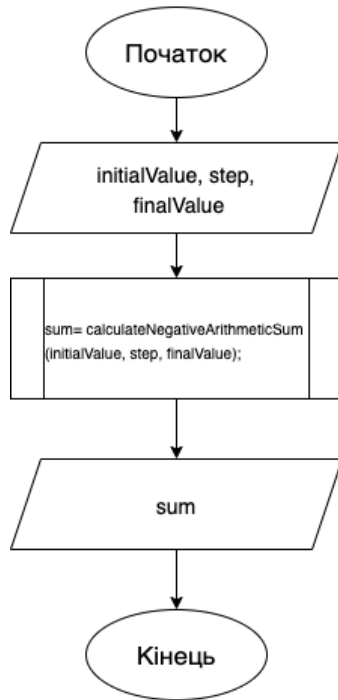
Виклик функції calculateNegativeArithmeticSum(initialValue,step,finalValue)
_____Вивидення sum
Кінець

Крок 2:

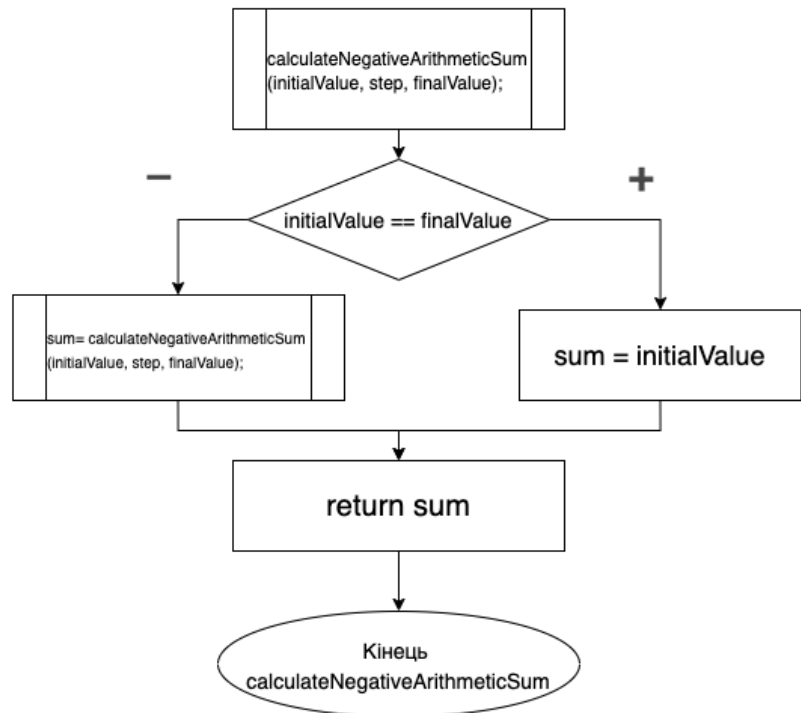
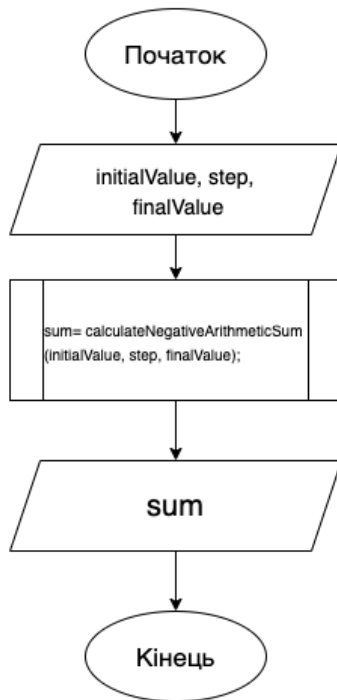
Функція calculateNegativeArithmeticSum (initialValue, step, finalValue)
 якщо initialValue == finalValue
 то
 sum = initialValue;
 інакше
 sum = initialValue + calculateNegativeArithmeticSum
 (initialValue, step, finalValue)
Все функція
Початок
 Ініціалізація змінних
 Виклик функції calculateNegativeArithmeticSum(initialValue,step,finalValue)
 _____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);
6  double sum = 0;
7
8  using namespace std;
9
10 int main() {
11     double initialValue = 3 * M_PI;
12     double step = M_PI_2;
13     double finalValue = -4 * M_PI;
14     sum = calculateNegativeArithmeticSum(initialValue, step, finalValue);
15     cout << sum << endl;
16     return 0;
17 }
18
19 double calculateNegativeArithmeticSum(double initialValue, double step, double finalValue) {
20     if (initialValue == finalValue) {
21         sum = initialValue;
22     } else {
23         sum = initialValue + calculateNegativeArithmeticSum(initialValue - step, step, finalValue);
24     }
25     return sum;
26 }
27

```

6) Випробування алгоритму:

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

9	$(4.7 == -12.5) == \text{false}$
10	$\text{sum} = 4.7 + \text{calculateNegativeArithmeticSum}(3.14, 1.57, -12.5);$
11	$(3.14 == -12.5) == \text{false}$
12	$\text{sum} = 3.14 + \text{calculateNegativeArithmeticSum}(1.5, 1.57, -12.5);$
13	$(1.5 == -12.5) == \text{false}$
14	$\text{sum} = 1.5 + \text{calculateNegativeArithmeticSum}(0, 1.57, -12.5);$
15	$(0 == -12.5) == \text{false}$
16	$\text{sum} = 0 + \text{calculateNegativeArithmeticSum}(-1.5, 1.57, -12.5);$
17	$(-1.5 == -12.5) == \text{false};$
18	$\text{sum} = -1.5 + \text{calculateNegativeArithmeticSum}(-3.14, 1.57, -12.5);$
19	$(-3.14 == 12.5) == \text{false}$
20	$\text{sum} = -3.14 + \text{calculateNegativeArithmeticSum}(-4.7, 1.57, -12.5);$
21	$(-4.7 == -12.5) == \text{false};$
22	$\text{sum} = -4.7 + \text{calculateNegativeArithmeticSum}(-6.28, 1.57, -12.5);$
23	$(-6.28 == -12.5) == \text{false}$
24	$\text{sum} = -6.28 + \text{calculateNegativeArithmeticSum}(-7.8, 1.57, -12.5);$
25	$(-7.8 == -12.5) == \text{false};$
26	$\text{sum} = -7.8 + \text{calculateNegativeArithmeticSum}(-9.4, 1.57, -12.5);$
27	$(-9.4 == -12.5) == \text{false}$
28	$\text{sum} = -9.4 + \text{calculateNegativeArithmeticSum}(-10.9, 1.57, -12.5);$
29	$(-10.9 == -12.5) == \text{false}$
30	$\text{sum} = -10.9 + \text{calculateNegativeArithmeticSum}(-12.5, 1.57, -12.5);$
31	$(-12.5 == -12.5) == \text{true}$
32	$\text{sum} = -12.5$

33	sum = $-10.9 + (-12.5)$; sum = -23.5
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	<code>sum = 7.8 + (-40.8); sum = -32.9</code>
58	<code>return -32.9;</code>
59	<code>sum = 9.4 + (-32.9); sum = -23.5</code>
60	<code>return -23.5</code>
61	<code>виведення-23.5</code>
	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.