

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

«Дослідження рекурсивних алгоритмів»

Варіант 6

Виконав студент ІП-13 Вдовиченко Станіслав Юрійович
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

- **Мета** – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.
- **Варіант 6**
Обчислити добуток елементів арифметичної прогресії, що убуває: початкове значення – 35, кінцеве – 0, крок – 10.
- **Постановка задачі**
Треба покроково знайти кожен елемент арифметичної прогресії, перемножити їх, вивести результат.
- **Математична модель**

Змінна	Тип	Ім'я	Призначення
Аргумент підпрограми, що відповідає першому елементу прогресії	Дійсний	first_element	Аргумент функції
Аргумент підпрограми, що відповідає кроку прогресії	Дійсний	step	Аргумент функції
Аргумент підпрограми, що відповідає останньому елементу прогресії	Дійсний	last_element	Аргумент функції
Добуток	Дійсний	product	Проміжне дане, повертає підпрограма
Перший елемент	Дійсний	a1	Вхідні дані
Крок	Дійсний	d	Вхідні дані
Кінцевий елемент	Дійсний	a0	Вхідні дані
Результат	Дійсний	result	Вихідні дані

Формулювання задачі зводиться до реалізації рекурсивного алгоритму, описаного в постановці задачі.

Рекурсивна функція приймає 3 параметри:

multiplication(int first_element, int step, int last_element), де first_element – перший елемент арифметичної прогресії, step – крок прогресії, last_element – кінцевий елемент прогресії.

Якщо поточний елемент прогресії(за поточний елемент приймаємо змінну first_element, яка змінюється при кожному виклику функції) більше за останній, то викликаємо функцію multiplication, але в першому параметрі задаємо (first_element + step), тобто перехід до наступного елемента прогресії. Знаходимо product як добуток поточного елемента(first_element) та значення, яке поверне наступна функція.

Інакше(якщо поточний елемент менше за кінцевий), повертаємо добуток product.

Розв'язання.

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження добутку.

- **Псевдокод**

Крок 1.

Функція multiplication (first_number, step, last_number)

Реалізація рекурсії

Все функція

Початок

Введення даних

Виклик функції

Виведення даних

Кінець

Крок 2.

Функція **multiplication (first_number, step, last_number)**

```
product := 1;  
якщо (first_number >= last_number)  
    product := first_number *  
    multiplication(first_number+step,step,last_number);  
    return product;  
все якщо  
  
інакше  
    return product;  
все інакше
```

Все функція

Початок

Введення даних

Виклик функції

Виведення даних

Кінець

Крок 3.

Функція **multiplication (first_number, step, last_number)**

```
product := 1;  
якщо (first_number >= last_number)  
    product := first_number *  
    multiplication(first_number+step,step,last_number);  
    return product;  
все якщо  
  
інакше  
    return product;  
все інакше
```

Все функція

Початок

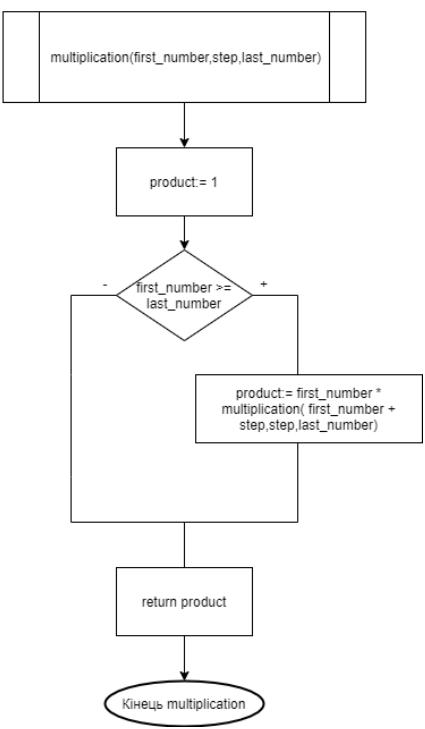
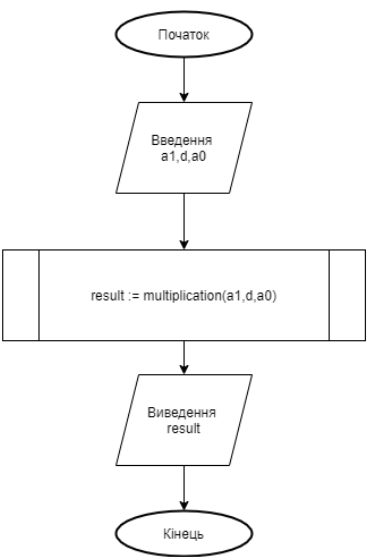
Введення a1,d,a0

result = multiplication(a1,d,a0)

Виведення result

Кінець

• Блок-схема



- Код програми

```
#include <iostream>
using namespace std;

double multiplication(double first_number, double step, double last_number){
    double product = 1;
    if (first_number >= last_number){
        product = first_number * multiplication( first_number first_number+step,step,last_number);
        return product;
    }
    else return product;
}

int main() {
    double a1;
    double d;
    double a0;
    double result;
    cout << "Enter the first number, step, last number: " << endl;
    cin >> a1;
    cin >> d;
    cin >> a0;
    result = multiplication(a1,d,a0);
    cout << "The multiplication is: " << result;
    return 0;
}
```

Enter the first number, step, last number:

35 -5 5

The multiplication is: 3.9375e+08

Process finished with exit code 0

- Випробування алгоритму

Блок	Дія
	Початок
1	Введення first_number = 35, step = -5, last_number = 5
2	Виклик функції multiplication (35,-5,0)
3	product = 1
4	(35 > 5); product = 35 * multiplication (30,-5,0)
5	(30 > 5); product = 30 * multiplication(25,-5,0)
6	(25 > 5); product = 25 * multiplication(20, -5, 0)
7	(20 > 5); product = 20 * multiplication (15, -5,0)
8	(15 > 5); product = 15 * multiplication (10, -5,0)
9	(10 > 5); product = 10 * multiplication (5, -5, 0)
10	(5 > 5); product = 5 * multiplication (0, -5, 0)
11	(0 < 5); product = 1; // повертаємося до кроку 10
12	(5 > 0); product = 5 * 1 = 5; // повертаємося до кроку 9
13	(10 > 0); product = 10 * 5 = 50; // повертаємося до кроку 8
14	(15 > 0); product = 15 * 50 = 750; // повертаємося до кроку 7
15	(20 > 0); product = 20 * 750 = 15 000; // повертаємося до кроку 6
16	(25 > 0); product = 15 000 * 25 = 375 000; // повертаємося до кроку 5
17	(30 > 0); product = 30 * 375 000 = 11 250 000; // повертаємося до кроку 4
18	(35 > 0); product = 35 * 11 250 000 = 393 750 000;
19	Виведення 393 750 000
	Кінець

- **Висновок**

Під час виконання лабораторної роботи я дослідив особливості роботи рекурсивних алгоритмів, набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. Розробив алгоритм для розв'язання поставленої задачі, побудував математичну модель, псевдокод, блок-схему. Написав код програми. Протестував алгоритм та довів його правильність.