

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 складних циклічних  
алгоритмів» Варіант 8

Виконав студент ПІ-12 Волков Вадим Всеволодович  
(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_  
( 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.

Задача 8. Обчислити суму елементів геометричної прогресії, що убуває: початкове значення – 50, кінцеве значення – 1, крок – 2

Побудова математичної моделі.

Змінна	Тип	Ім'я	Призначення
Оброблюване число	Дійсне	n	Параметр функції
Часткова сума прогресії	Дійсне	Return	Результат функції
Сума геометричної прогресії	Дійсне	Res	Результат

Задачею є використання рекурсивного алгоритму для проходження по числам від 50 до 1 із зменшенням у 2 рази на кожному кроці і рахування суми всіх чисел.

### *Розв'язання*

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блоксхеми.

*Крок 1.* Визначимо основні дії.

*Крок 2.* Розрахування Res через GeomProgSum

*Крок 3.* Рекурсивне розрахування значення з перевіркою

*Крок 4.* Повернути рекурсивно розраховане значення

### Псевдокод

крок 1

**початок**

Розрахування Res через GeomProgSum

Виведення Res

**кінець**

**початок** GeomProgSum(n)

Рекурсивне розрахування значення з перевіркою

**кінець**

крок 2

**початок**

Res := GeomProgSum(50)

Виведення Res

**кінець**

**початок** GeomProgSum(n)

Рекурсивне розрахування значення з перевіркою

**кінець**

крок 3

**початок**

Res := GeomProgSum(50)

Виведення Res

**кінець**

**початок** GeomProgSum(n)

**якщо**  $n > 1$

**то**

Порахувати рекурсивно розраховане значення

**інакше**

Return := 1

**все якщо**

повернути Return

**кінець**

крок 4

**початок**

Res := GeomProgSum(50)

Виведення Res

**кінець**

**початок** GeomProgSum(n)

**якщо**  $n > 1$

**то**

Return :=  $n + \text{GeomProgSum}(n / 2)$

**інакше**

Return := 1

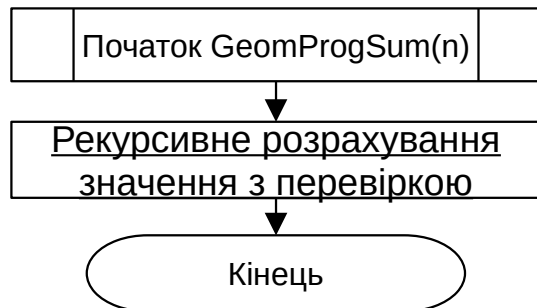
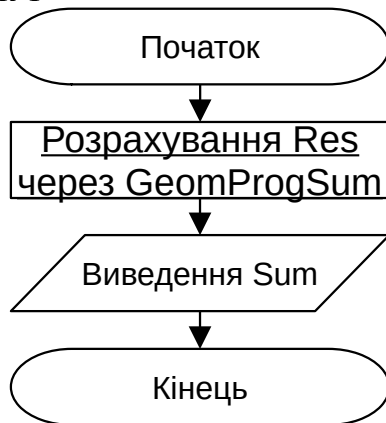
**все якщо**

повернути Return

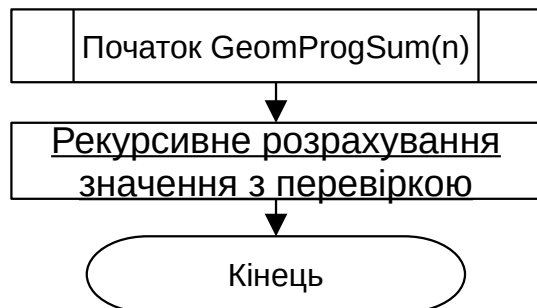
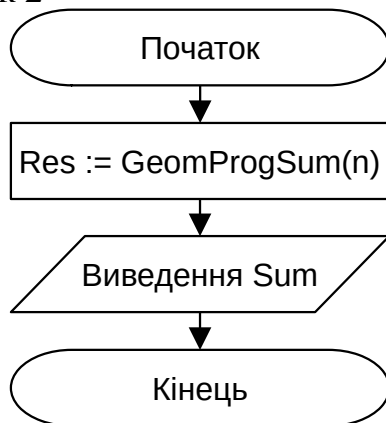
**кінець**

## Блок схема

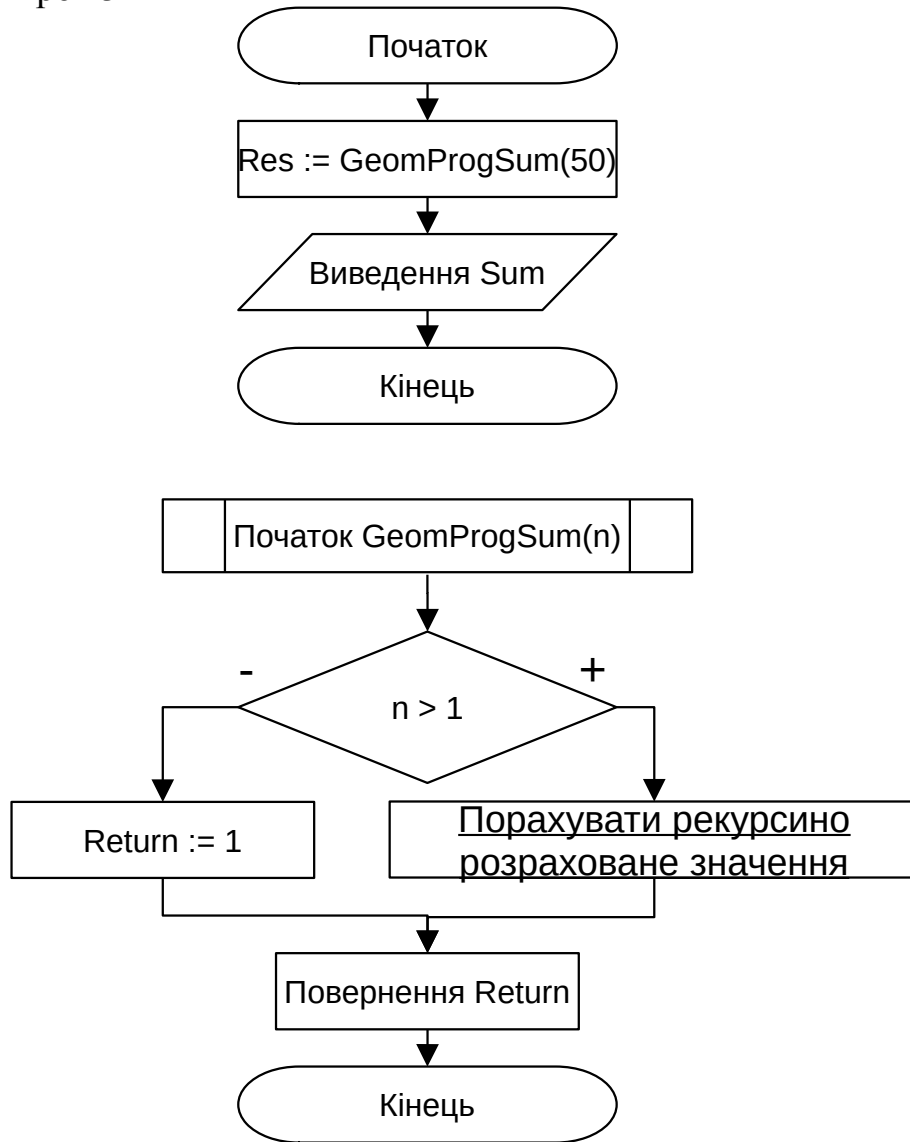
### Крок 1



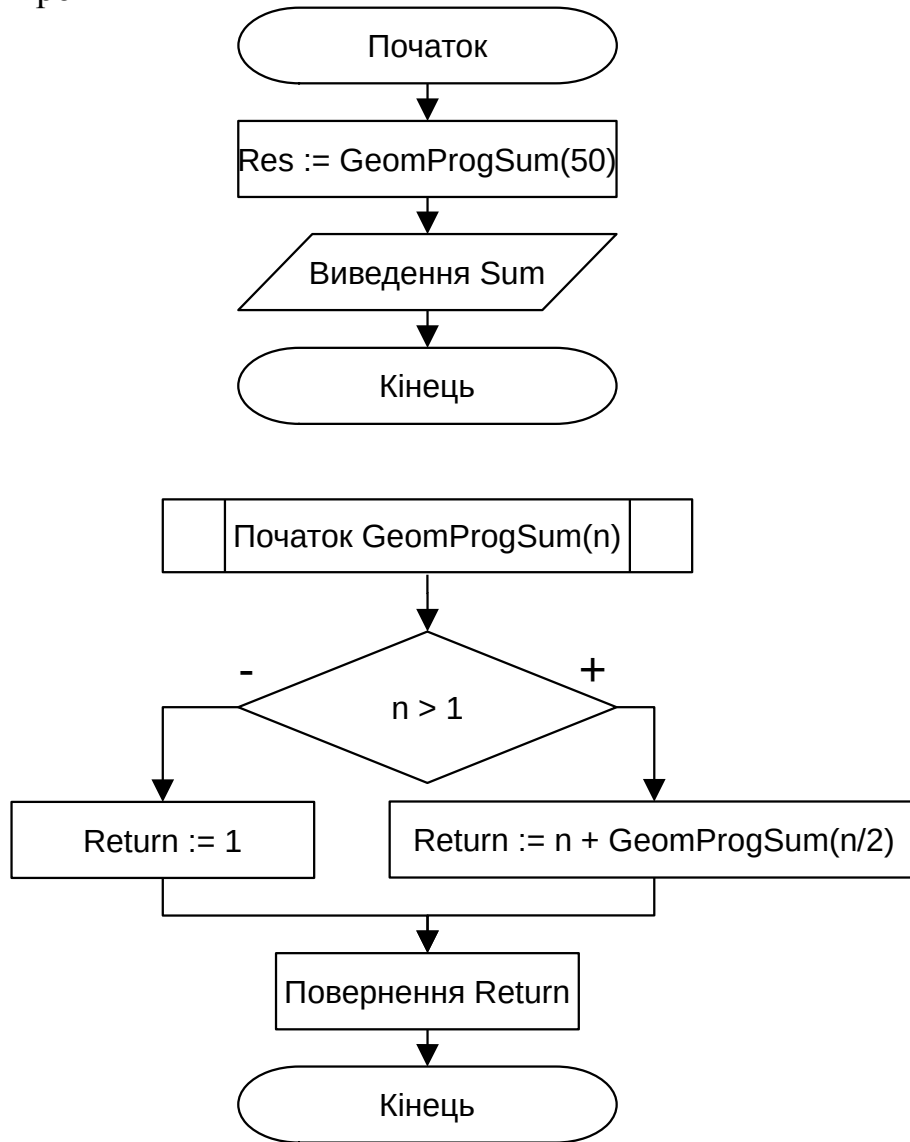
### Крок 2



Крок 3



Крок 4



Код програми:

Python:

```
#!/usr/bin/env python
```

```
def GeomProgSum(n):  
    if n > 1:  
        Return = n + GeomProgSum(n/2)  
    else:  
        Return = 1  
    return Return
```

```
Res = GeomProgSum(50)  
print("Результат: "+str(Res))
```

C++:

```
// Lab6.cpp
```

```
#include <iostream>  
#include <windows.h>
```

```
double geomProgSum(double n) {  
    double returnVal;  
    if (n > 1) {  
        returnVal = n + geomProgSum(n / 2);  
    } else {  
        returnVal = 1.0;  
    }  
    return returnVal;  
}
```

```
int main() {  
    SetConsoleCP(CP_UTF8);  
    SetConsoleOutputCP(CP_UTF8);  
    double result = geomProgSum(50.0);  
    std::cout << "Результат: " << result << "\n";  
}
```

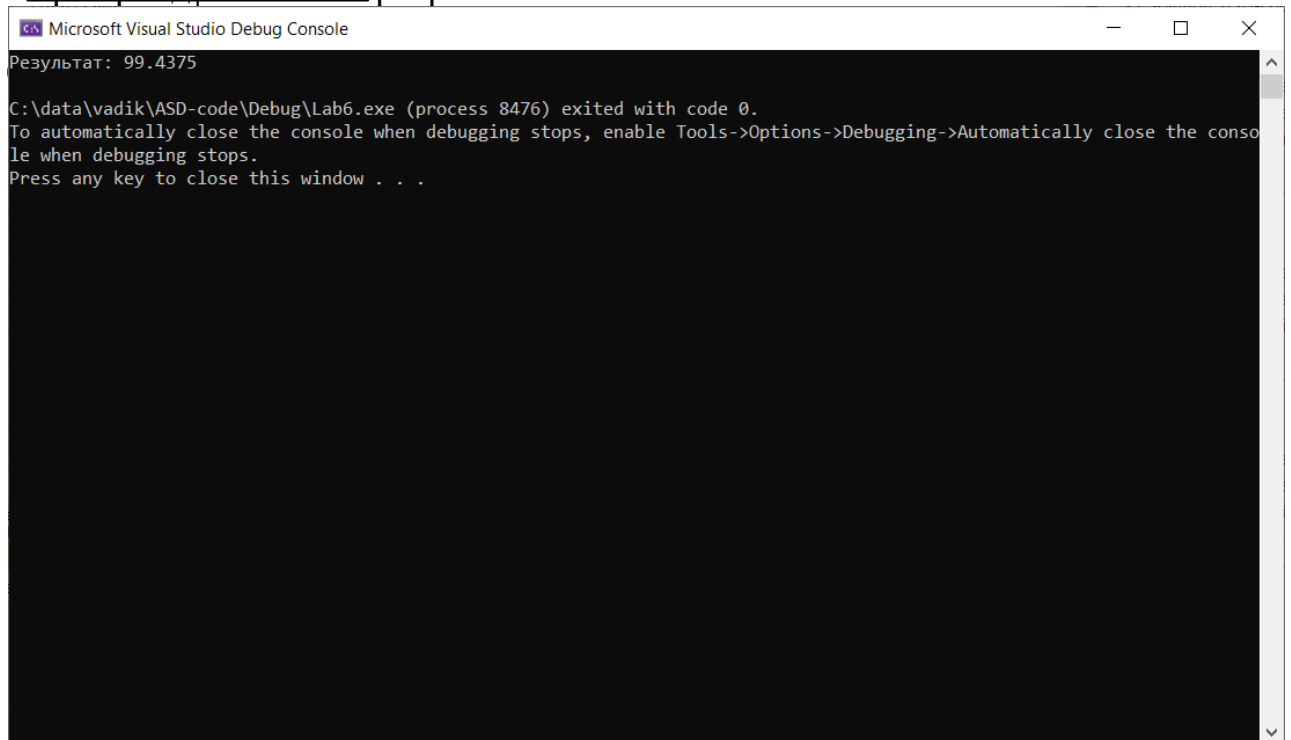


Перевіримо правильність алгоритма на довільних конкретних значеннях початкових даних:

Блок	Дія
	Початок
1	$50 > 1$ - так
2	виклик GeomProgSum(50/2)
3	$25 > 1$ - так
4	виклик GeomProgSum(25/2)
5	$12.5 > 1$ - так
6	виклик GeomProgSum(12.5/2)
7	$6.25 > 1$ - так
8	виклик GeomProgSum(6.25/2)
9	$3.125 > 1$ - так
10	виклик GeomProgSum(3.125/2)
11	$1.5625 > 1$ - так
12	виклик GeomProgSum(1.5625/2)
13	$0.78125 > 1$ - ні
14	Return := 1
15	повернення 1
16	Return := $1.5625 + 1 = 2.5625$
17	повернення 2.5625
18	Return := $3.125 + 2.5625 = 5.6875$
19	повернення 5.6875
20	Return := $6.25 + 5.6875 = 11.9375$
21	повернення 11.9375
22	Return := $12.5 + 11.9375 = 24.4375$
23	повернення 24.4375
24	Return := $25 + 24.4375 = 49.4375$
25	повернення 49.4375
26	Return := $50 + 49.4375 = 99.4375$
27	повернення 99.4375
28	Виведення 99.4375
	кінець

Починаючи з всіх  
чисел від 100 до 200  
обчислювати суму  
чисел поки не  
залишиться 1 цифра і  
виведення її

програми:



```
Microsoft Visual Studio Debug Console
Результат: 99.4375
C:\data\vadik\ASD-code\Debug\Lab6.exe (process 8476) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

### *Висновок:*

На основі цього алгоритму по знаходження значення геометричної прогресії за допомогою рекурсивної функції, що повертає частковий результат, було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм.