

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 20

Виконав студент

ІП-15, Ликова Катерина Олександрівна
(шифр, прізвище, ім'я, по батькові)

Перевірив

(прізвище, ім'я, по батькові)

Київ 2021

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

Постановка задачі

Обчислити значення квадрата цілого додатного числа n , якщо відома залежність $n^2 = (n-1)^2 + 2 \cdot (n-1) + 1$

Математична модель

Змінна	Тип	Ім'я	Призначення
Ціле додатне число	Цілий	n	Початкові дані
Функція, що знаходить квадрат заданого числа	Функція	square	Проміжні дані
Параметр функції square	Цілий	k	Проміжні дані
Значення квадрату заданого числа	Цілий	s	Результат

Для того, щоб знайти квадрат цілого додатного числа потрібно створити підпрограму square, що обраховує потрібне значення за допомогою рекурсивної формули. Якщо значення $k = 1$, функція повертає значення 1, в усіх інших випадках виконується така рекурсивна формула: $\text{square}(k - 1) + 2 * (k - 1) + 1$. У функції main вводимо значення n , присвоюємо значення $s = \text{square}(n)$, виводимо отриманий результат.

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Вводимо значення змінної n .

Крок 3. Присвоюємо значення змінній s .

Підпрограма: Деталізація дії знаходження $\text{square}(n)$.

Псевдокод

крок 1

початок

введення значення змінної n

присвоєння значення змінній s

кінець

крок 2

початок

введення n

присвоєння значення змінній s

кінець

крок 3

початок

введення n

s = square(n)

кінець

Підпрограма

square(k)

якщо k==1

то

return 1

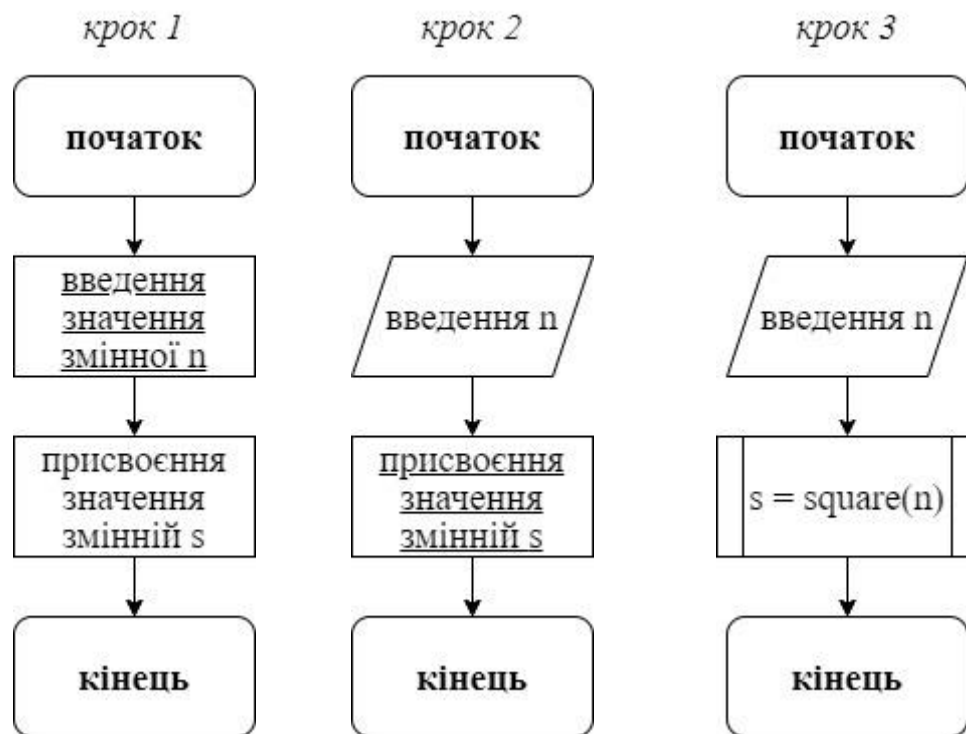
інакше

return square(k - 1) + 2 * (k - 1) + 1

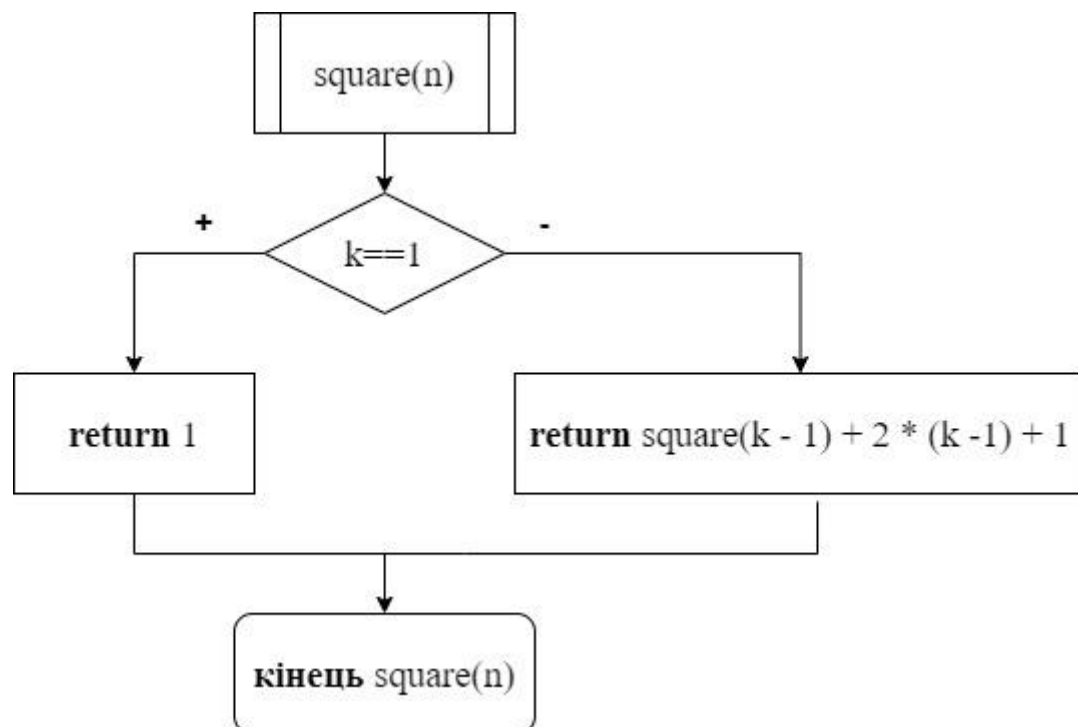
все якщо

кінець square(k)

Блок-схема



Підпрограма:



Код

```

1  #include <iostream>
2  // n - ціле додатне число
3  int square (int k)
4  {
5      if (k == 1)
6          return 1;
7      return square(k - 1) + 2 * (k - 1) + 1;
8  }
9
10 int main()
11 {
12     int n;
13     int s;
14     std::cout << "Enter n: ";
15     std::cin >> n;
16     s = square(n);
17     std::cout << "s: " << s << std::endl;
18
19     system ("pause");
20 }

```

```

Enter n: 5
s: 25
Для продовження натисніть будь-яку клавішу . . .

```

Випробування алгоритму

Блок	Дія
	Початок
1	n=5
2	<p>5>1</p> <p>return square(5 - 1) + 2 * (5 - 1) + 1 = square(4) + 9</p> <p>4>1</p>

	return square(4 - 1) + 2 * (4 - 1) + 1 = square(3) + 7 3>1 return square(3 - 1) + 2 * (3 - 1) + 1 = square(2) + 5 2>1 return square(2 - 1) + 2 * (2 - 1) + 1 = square(1) + 3 1==1 return 1 return square(1) + 3 = 1 + 3 = 4 return square(2) + 5 = 4 + 5 = 9 return square(3) + 7 = 9 + 7 = 16 return square(4) + 9 = 16 + 9 = 25 s = 25
	Кінець

Висновки: Для побудови алгоритму розв’язання заданої задачі я застосувала рекурсивні алгоритми для створення програмної специфікації. Завдяки цьому я вивчила властивості рекурсивних формул та навчилась створювати підпрограми та використовувати їх на практиці.