

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 23

Виконав студент ІП-15, Мочалов Дмитро Юрійович

Перевірів Вечерковська Анастасія Сергіївна

Київ 2021

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 23

Задача. Написати рекурсивну функцію для обчислення суми цифр та кількості цифр натурального числа A.

Мат. модель

| Змінна | Тип | Ім'я | Призначення |
|-----------------------------------|-------|--------------|--------------|
| Число з якого рахується результат | цілий | a | Вхідні данні |
| Сума цифр числа | цілий | calcSumOfDig | Результат |
| Кількість цифр числа | цілий | countOfDig | результат |

Таким чином, математичне модулювання зводиться до використання двох рекурсивних функцій які будуть рахувати суму та кількість цифр числа. Рекурсивні функції це ті функції які в своєму тілі викликають себе ж. Суму цифр можна знайти додаючи остачі від ділення числа на 10 ділячи після взяття саме число на 10, поки воно більше або рівне 1. Для взяття остачі від ділення використовуємо %. Для знаходження кількості цифр ми поступово ділимо число на 10 поки воно більше або рівне 1, при цьому додаючи 1 після кожного ділення до результату. Для позначення більше або рівне використовуємо \geq , для цілочисельного ділення /.

Крок1: визначитись з алгоритмом

Крок2: деталізуємо алгоритм знаходження суми цифр за допомогою підпрограми

Крок2: деталізуємо алгоритм знаходження кількості цифр за допомогою підпрограми

Псевдокод

Крок1

Початок

деталізуємо алгоритм знаходження суми цифр за допомогою підпрограми

деталізуємо алгоритм знаходження кількості цифр за допомогою підпрограми

Кінець

Крок2

Початок

calcSumOfDig(a);

деталізуємо алгоритм знаходження кількості цифр за допомогою підпрограми

Кінець

Підпрограми:

calcSumOfDig(a)

якщо $a \geq 1$

то:

повернути $a \% 10 + \text{calcSumOfDig}(a/10)$

інакше

повернути 0;

все якщо

кінець

Крок2

Початок

calcSumOfDig(a);

countOfDig(a);

Кінець

Підпрограми:

calcSumOfDig(a)

якщо $a \geq 1$

то:

повернути $a \% 10 + \text{calcSumOfDig}(a/10)$

інакше

повернути 0;

все якщо

кінець

countOfDig(a)

якщо $a \geq 1$

то:

повернути $1 + \text{calcSumOfDig}(a/10)$

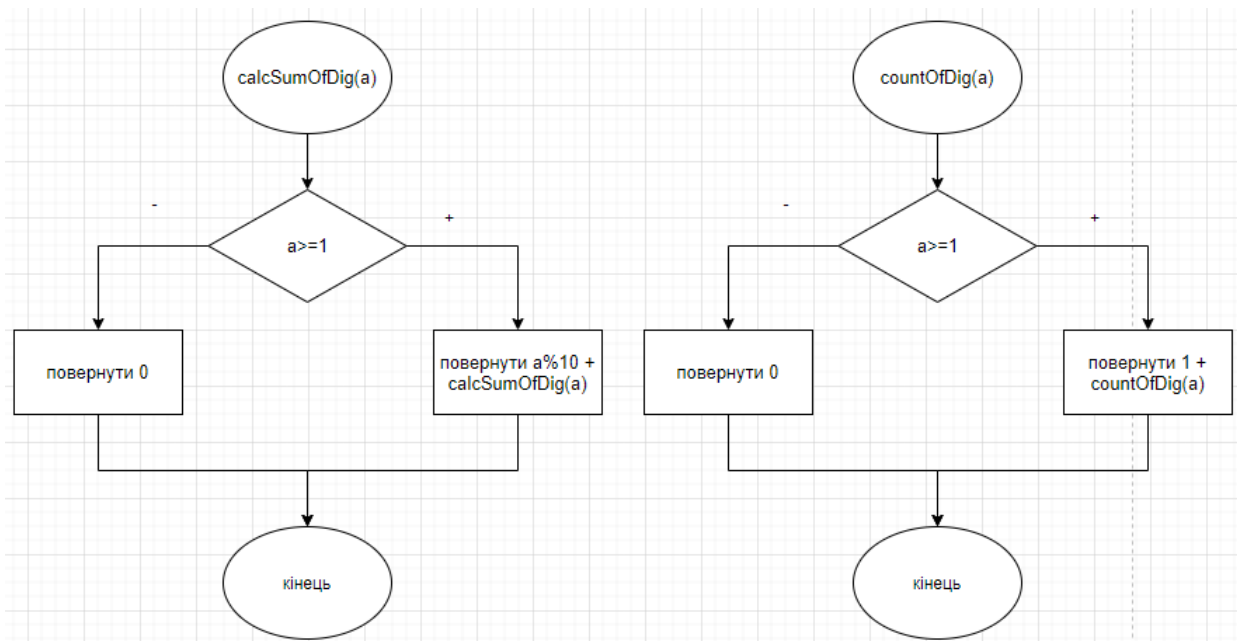
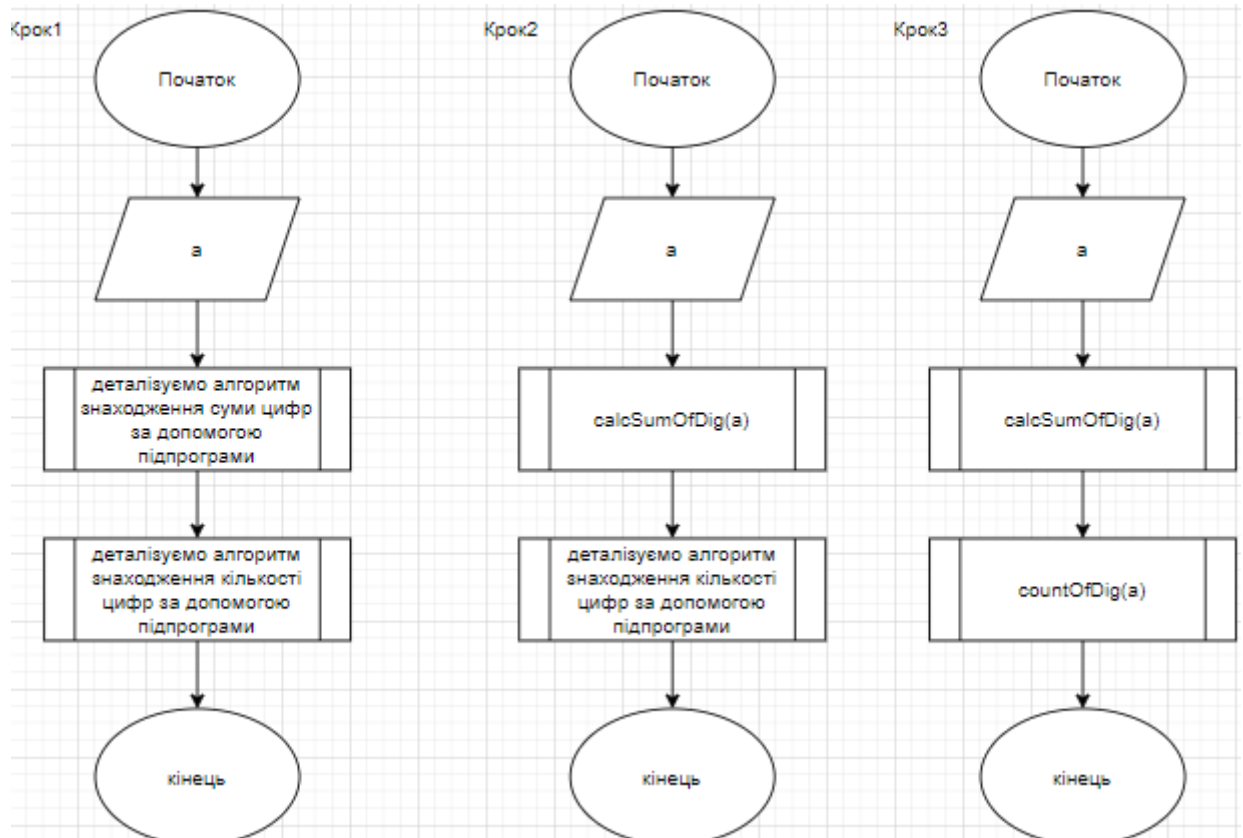
інакше

повернути 0;

все якщо

кінець

Блок-схема



Випробовування алгоритму

| Крок | Дія |
|------|--------------------------------------------------------------|
| | Початок |
| 1 | $a = 12345$ |
| 2 | $\text{calcSumOfDig}(a)$ |
| 3 | $a \geq 1$ |
| 4 | $\text{calcSumOfDig}(12345) = 5 + \text{calcSumOfDig}(1234)$ |
| 5 | $a \geq 1$ |
| 6 | $\text{calcSumOfDig}(1234) = 4 + \text{calcSumOfDig}(123)$ |
| 7 | $a \geq 1$ |
| 8 | $\text{calcSumOfDig}(123) = 3 + \text{calcSumOfDig}(12)$ |
| 9 | $a \geq 1$ |
| 10 | $\text{calcSumOfDig}(12) = 2 + \text{calcSumOfDig}(1)$ |
| 11 | $a \geq 1$ |
| 12 | $\text{calcSumOfDig}(1) = 1 + \text{calcSumOfDig}(0)$ |
| 13 | $a < 1$ |
| 14 | Повернути 0 |
| 15 | $\text{countOfDig}(a)$ |
| 16 | $a \geq 1$ |
| 17 | $\text{countOfDig}(12345) = 1 + \text{countOfDig}(1234)$ |
| 18 | $a \geq 1$ |
| 19 | $\text{countOfDig}(1234) = 1 + \text{countOfDig}(123)$ |

| | |
|----|------------------------------------------------------|
| 20 | $a \geq 1$ |
| 21 | $\text{countOfDig}(123) = 1 + \text{countOfDig}(12)$ |
| 22 | $a \geq 1$ |
| 23 | $\text{countOfDig}(12) = 1 + \text{countOfDig}(1)$ |
| 24 | $a \geq 1$ |
| 25 | $\text{countOfDig}(1) = 1 + \text{countOfDig}(0)$ |
| 26 | $a < 1$ |
| 27 | Повернути 0 |
| | Кінець |

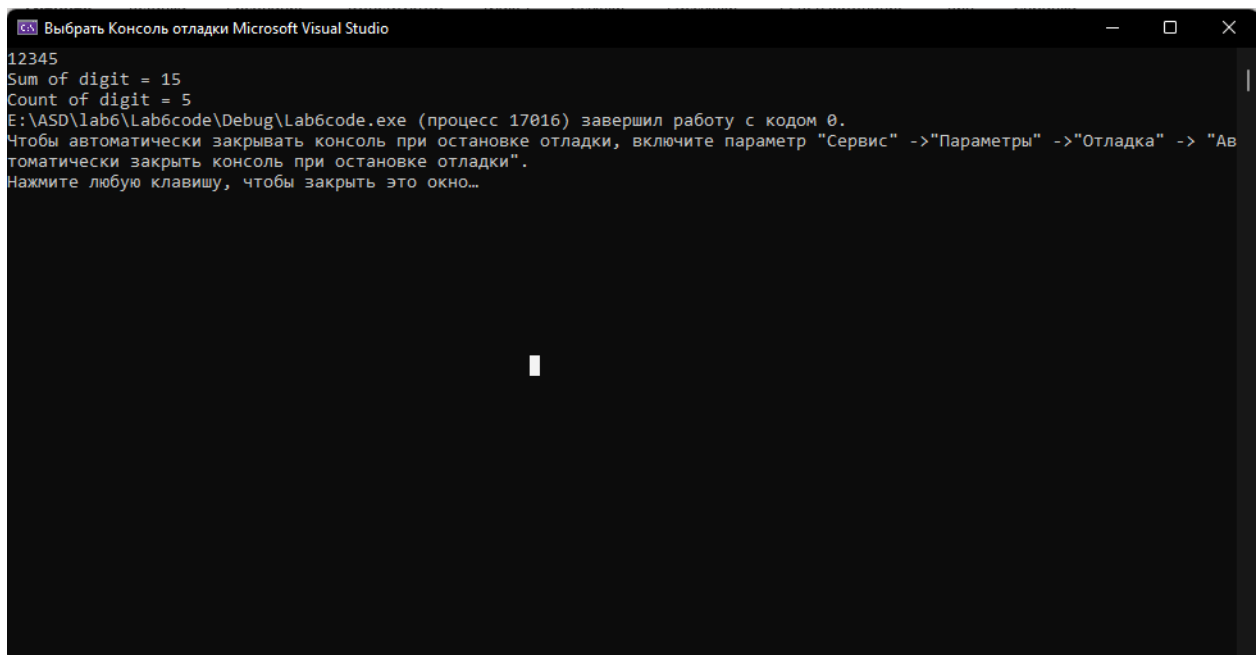
Код

```

1      #include <iostream>
2
3
4      int calcSumOfDig(int a) {
5          if (a >= 1) {
6              return a % 10 + calcSumOfDig(a / 10);
7          }
8          else return 0;
9      }
10
11     int countOfDig(int a) {
12         if (a >= 1) {
13             return 1 + countOfDig(a/10);
14         }
15         else return 0;
16     }
17
18
19     int main()
20     {
21         int a;
22         std::cin >> a;
23         std::cout <<"Sum of digit = " << calcSumOfDig(a) << std::endl;
24         std::cout <<"Count of digit = " << countOfDig(a);
25     }

```

Тестування



```

Выбрать Консоль отладки Microsoft Visual Studio
12345
Sum of digit = 15
Count of digit = 5
E:\ASD\lab6\Lab6code\Debug\Lab6code.exe (процесс 17016) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```


Висновок: Ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм.