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-13 Недельчев Євген Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

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.

**Індивідуальне завдання**

**Варіант 23**

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

**Постановка задачі**

Задане натуральне число А. Розробити рекурсивний алгоритм для обчислення суми цифр та кількості цифр числа А.

**Побудова математичної моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Змінна*** | ***Тип*** | ***Ім’я*** | ***Призначення*** |
| Натуральне число А | Цілий | А | Вхідні дані |
| Кількість цифр у числі А | Цілий | amount | Результат |
| Сума цифр числа А | Цілий | sum | Результат |

Таким чином математичне формулювання задачі зводиться до реалізації двох рекурсивних функцій. Результатом роботи першої є обчислення кількості цифр у числі А. Результат роботи другої – обчислення суми цифр числа А.

Перша рекурсивна функція приймає один параметр:

amountofdigits(int A), де А – число, кількість цифр у якому обчислює функція.

Якщо число А при діленні націло на 10 не дорівнює 0, то функція повертає значення 1 + amountofdigits(A / 10). В противному випадку функція повертає 1.

Умова виходу з рекурсії: A / 10 == 0.

Друга рекурсивна функція приймає один параметр:

sumofdigits(int A), де А – число, сума цифр у якому обчислює функція. Якщо число А при діленні націло на 10 не дорівнює 0, то функція повертає значення A % 10 + sumofdigits(A / 10). В противному випадку функція повертає A % 10.

Умова виходу з рекурсії: A / 10 == 0.

**Розв’язання**

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження кількості цифр у числі.

Крок 3. Деталізуємо дію знаходження суми цифр у числі.

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

**Псевдокод**

Крок 1.

**функція amountofdigits(A)**

реалізація рекурсії

**все функція**

**функція sumofdigits(A)**

реалізація рекурсії

**все функція**

**початок**

введення змінної А

виклик функції amountofdigits(A)

виклик функції sumofdigits(A)

виведення результатів

**кінець**

Крок 2.

**функція amountofdigits(A)**

**якщо** А / 10 != 0

**return** 1 + amountofdigits(A / 10)

**інакше**

**return** 1

**все функція**

**функція sumofdigits(A)**

реалізація рекурсії

**все функція**

**початок**

введення змінної А

amount := amountofdigits(A)

виклик функції sumofdigits(A)

виведення amount, sum

**кінець**

Крок 3.

**функція amountofdigits(A)**

**якщо** А / 10 != 0

**return** 1 + amountofdigits(A / 10)

**інакше**

**return** 1

**все функція**

**функція sumofdigits(A)**

**якщо** А / 10 != 0

**return** A % 10 + sumofdigits(A / 10)

**інакше**

**return** A % 10

**все функція**

**початок**

введення змінної А

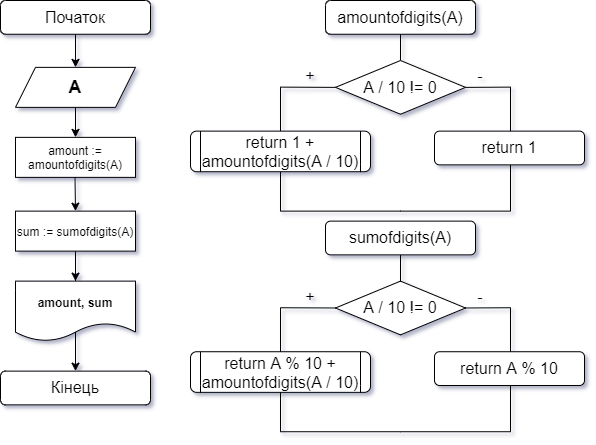
amount := amountofdigits(A)

sum := sumofdigits(A)

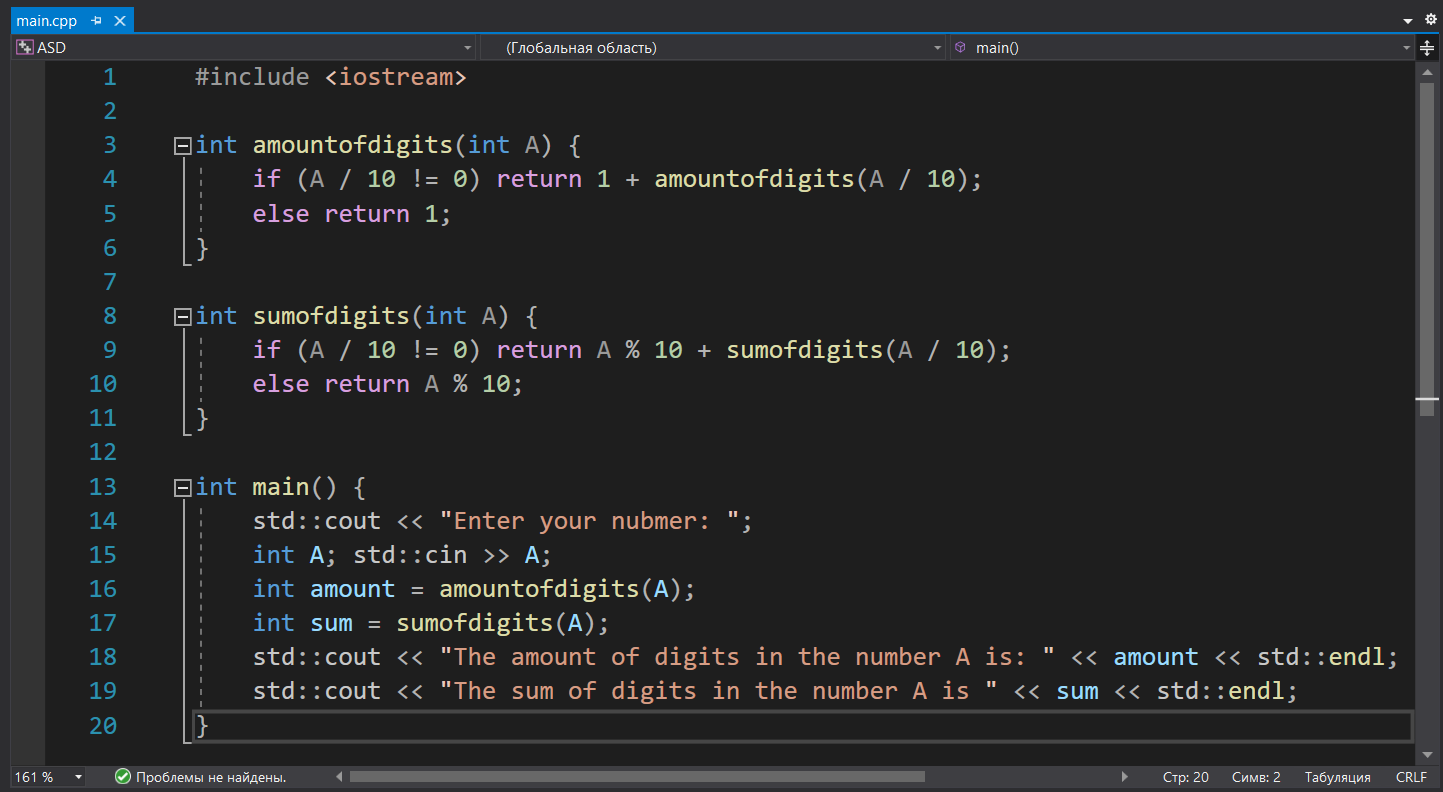
виведення amount, sum

**кінець**

**Блок-схема**

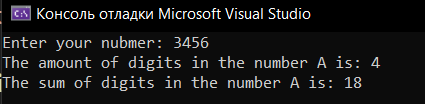


**Код програми**

****

**Тестування алгоритму**

|  |  |
| --- | --- |
| Блок | Дія |
| 1 | Введення: А = 3456 |
| 2 | amount = amountofdigits(A) |
| 3 | A / 10 != 0 == true |
|  | return 1+amountofdigits(A / 10) // до кроку 2 |
| 4 | A / 10 != 0 == true |
|  | return 1+amountofdigits(A / 10) // до кроку 3 |
| 5 | A / 10 != 0 == true |
|  | return 1+amountofdigits(A / 10) // до кроку 4 |
| 6 | A / 10 != 0 == false |
|  | return 1 // до кроку 5 |
| 7 | sum = sumofdigits(A) |
| 8 | A / 10 != 0 == true |
|  | return A % 10 + sumofdigits(A / 10) //до кр 7 |
| 9 | A / 10 != 0 == true |
|  | return A % 10 + sumofdigits(A / 10) //до кр 8 |
| 10 | A / 10 != 0 == true |
|  | return A % 10 + sumofdigits(A / 10) //до кр 9 |
| 11 | A / 10 != 0 == false |
|  | return A % 10 // до кроку 10 |
| 12 | Виведення: amount = 4, sum = 18 |

****

**Висновки**

Під час виконання роботи я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм.