

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 лінійних алгоритмів»

Варіант 19

Виконав студент ІІ-15 Левченко Владислав В'ячеславович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 2021

19. Знайти натуральне число від 1 до n з максимальною сумою дільників.

Побудова математичної моделі

Постановка задачі. Для вирішення цієї задачі потрібно використовувати рекурсію, яка буде створена окремою функцією. Рекурсивна функція (sumOfDiv), яку ми задамо буде рахувати сумму дільників заданого числа, а в основній програмі будемо перевіряти послідовно усі числа до заданого.

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Число, яке задає користувач	Натуральне	number	Вхідне дане
Показчик циклу	Натуральне	i	Проміжне дане
Перший аргумент циклу	Натуральне	numb	Проміжне дане
Другий аргумент циклу	Натуральне	count	Проміжне дане
Змінна, яка містить результат рекурсії	Натуральне	storage	Проміжне дане
Змінна, що містить найбільшу суму дільників	Натуральне	maxDivN	Проміжне дане
Змінна результату (числа з найбільшою сумою дільників)	Натуральне	maxCounter	Проміжне, вихідне дане

Крок 1. Визначимо основні дії

Крок 2. Створення основної програми і задання змінних

Крок 3. Створення циклу, що перевіряє поточне число

Псевдокод

Крок 1

початок

Створення основної програми і задання змінних

Створення циклу, що перевіряє поточне число

кінець

Крок 2

початок

Основна програма:

Введення number

maxDivN:=0

maxCounter:=0

Створення циклу, що перевіряє поточне число

Виведення maxCounter

кінець

Крок 3

початок

Підпрограми:

sumOfDiv(numb, count = 1)

якщо count > numb

то повернути 0

все якщо

якщо numb % count == 0

то повернути count + sumOfDiv(numb, count+1)

інакше повернути sumOfDiv(numb, count+1)

все якщо

Основна програма:

Введення number

maxDivN:= 0

maxCounter:= 0

для i **від** 1 **до** number

storage:=sumOfDiv(i)

якщо maxDivN < storage

то maxDivN = storage

maxCounter = i

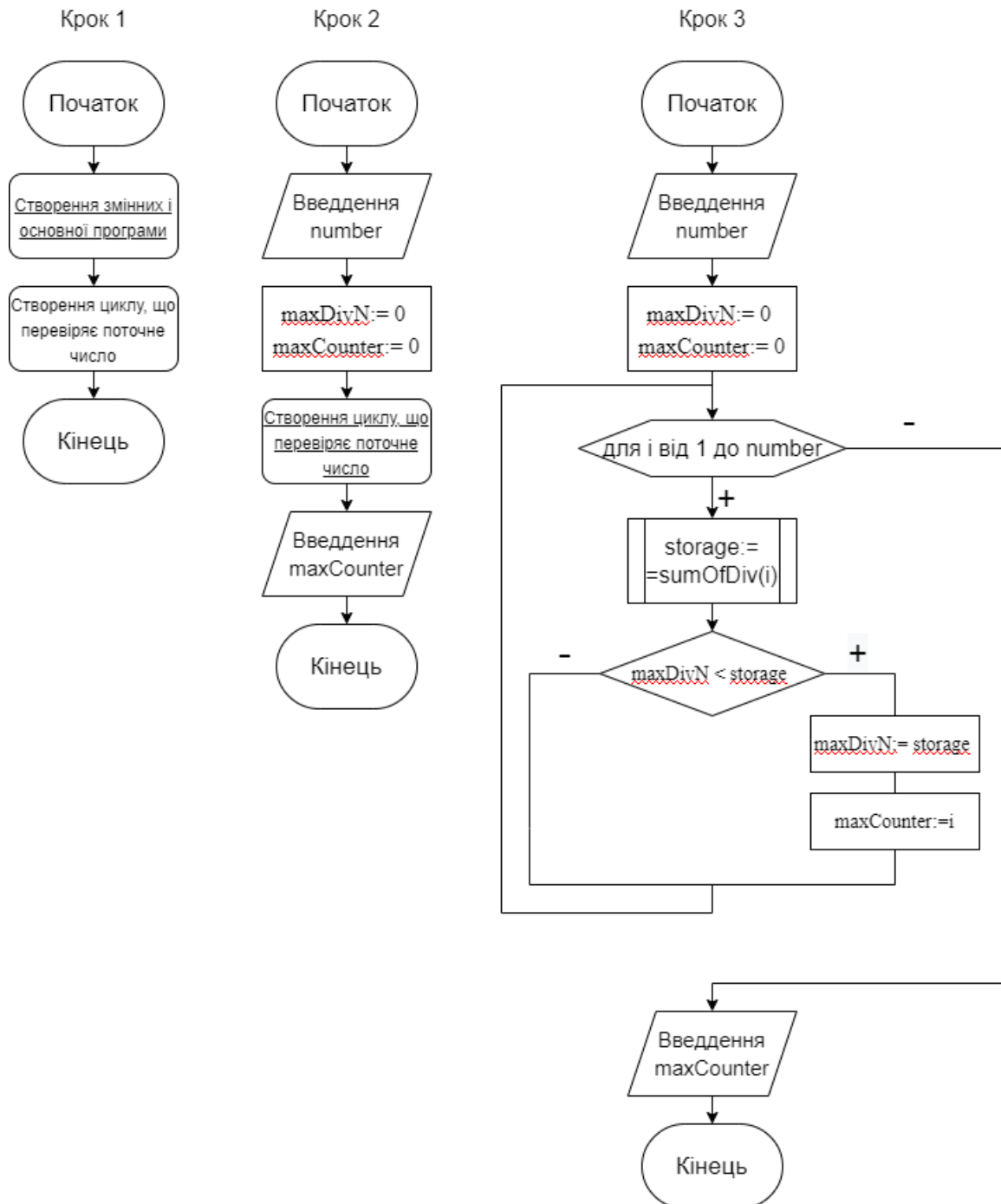
все якщо

все повторити

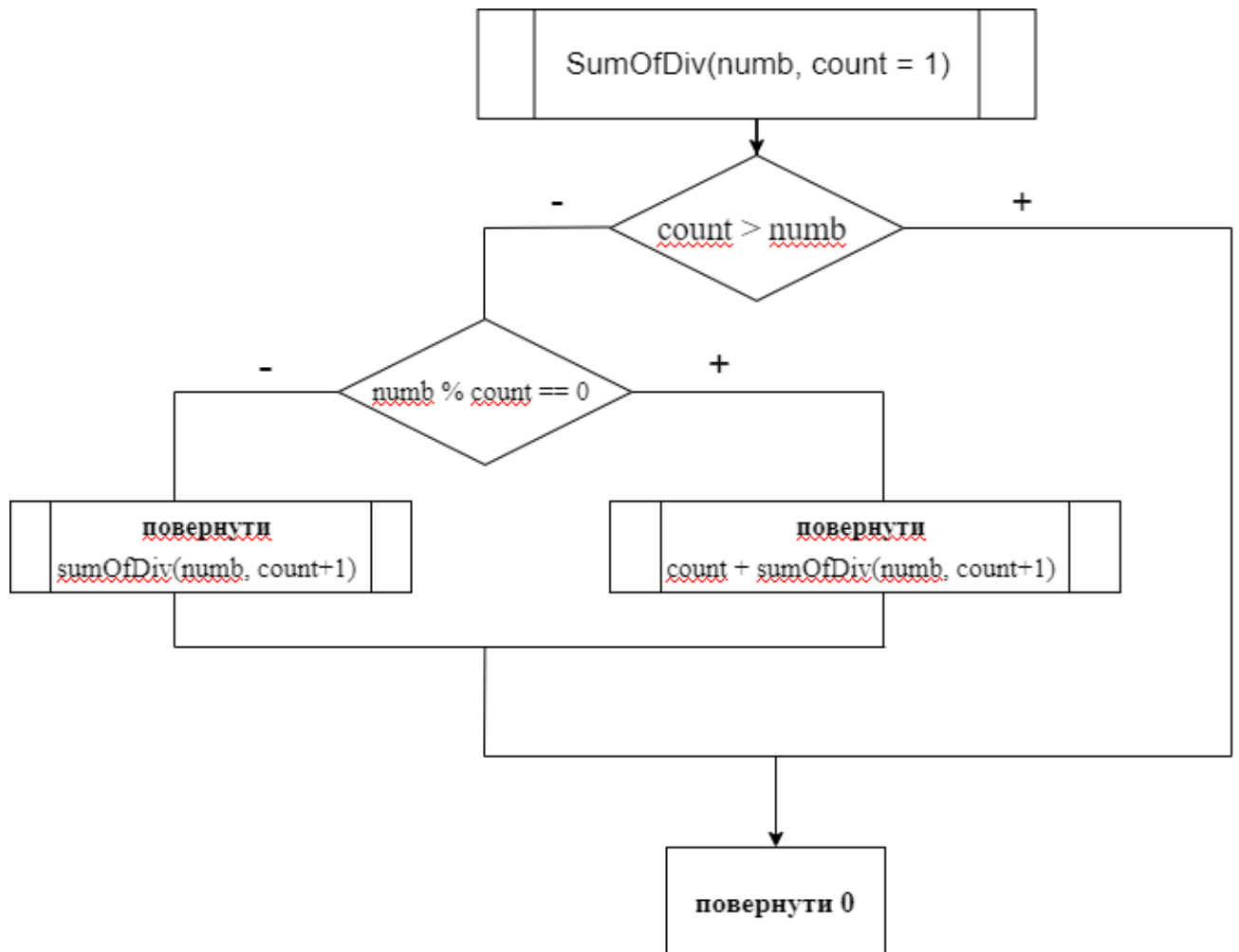
Виведення maxCounter

кінець

Блок схема основної програми



Блок схема підпрограми



Код програми і тестування

```
1  #include <iostream>
2  #include <math.h>
3  #include <cstdio>
4  #include <cstdlib>
5
6  using namespace std;
7
8  int sumOfDiv(int numb, int count = 1) {
9
10     if (count > numb) {return 0;}
11     if (numb % count == 0)
12     {return count + sumOfDiv(numb, count+1);}
13     else {return sumOfDiv(numb, count+1);}
14 }
15
16 int main() {
17     int storage;
18
19     int maxDivN = 0;
20     int maxCounter = 0;
21
22     int number;
23     std::cout << "Enter the number: ";
24     std::cin >> number;
25
26     for (int i = 1; i <= number; i++) {
27
28         storage = sumOfDiv(i);
29
30
31         if (maxDivN < storage) {
32             maxDivN = storage;
33             maxCounter = i;
34         }
35     }
36     std::cout << "The number with the biggest sum of div = " << maxCounter;
37 }
```

input

```
Enter the number: 5
The number with the biggest sum of div = 4
```

Випробування

Блок	Дія
	початок
1	Введення number = 5
2.1	i = 1; sumOfDiv(int numb, int count = 1); numb = 1;
2.2	1 > 1 == false; (1 % 1 == 0) == true; виклик count + sumOfDiv(int numb, int count = 2); 2 > 1; return 0; maxCounter = 1
3.1	i = 2; sumOfDiv(int numb, int count = 1); numb = 2;
3.2	1 > 2 == false; (2 % 1 == 0) == true; виклик count + sumOfDiv(int numb, int count = 2); 2 > 2 == false; (2 % 1 == 0) == true; виклик count + sumOfDiv(int numb, int count = 3); 3 > 2 == true; return 0; maxCounter = 3
4.1	i = 3; sumOfDiv(int numb, int count = 1); numb = 3;
4.2	1 > 3 == false; (3 % 1 == 0) == true; виклик count + sumOfDiv(int numb, int count = 2); 2 > 3 == false; (3 % 2 == 0) == false; виклик sumOfDiv(int numb, int count = 3); 3 > 3 == false; (3 % 3 == 0) == true; виклик count + sumOfDiv(int numb, int count = 4); 4 > 3 == true; return 0; maxCounter = 4
5.1	i = 4; sumOfDiv(int numb, int count = 1); numb = 4;
5.2	1 > 4 == false; (4 % 1 == 0) == true; виклик count + sumOfDiv(int numb, int count = 2); 2 > 4 == false; (4 % 2 == 0) == true; виклик count + sumOfDiv(int numb, int count = 3); 3 > 4 == false; (4 % 3 == 0) == false; виклик sumOfDiv(int numb, int count = 4); 4 > 4 == false; (4 % 4 == 0) == true; виклик count + sumOfDiv(int numb, int count = 5); 5 > 4 == true; return 0; maxCounter = 7
6.1	i = 5; sumOfDiv(int numb, int count = 1); numb = 5;
6.2	1 > 5 == false; (5 % 1 == 0) == true;

	виклик count + sumOfDiv(int numb, int count = 2); 2>5 == false; (5%2 == 0) == false; виклик sumOfDiv(int numb, int count = 3); 3>5 == false; (5%3 == 0) == false; виклик sumOfDiv(int numb, int count = 4); 4>5 == false; (5%4 == 0) == false; виклик sumOfDiv(int numb, int count = 5); 5>5 == false; (5%5 == 0); == true; виклик count + sumOfDiv(int numb, int count = 6); 6>5 == true; return 0; maxCounter = 7
7	Виведення maxCounter
	кінець

Висновки: під час виконання 6 лабораторної роботи ми навчилися використовувати рекурсивну функцію і задавши значення number:=5 визначили, що число з найбільшою сумою дільників до 5 - це число 4.