Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни

«Основи програмування 1.

Базові конструкції»

«Організація циклічних процесів. Складні цикли»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Вітковська Ірина Іванівна

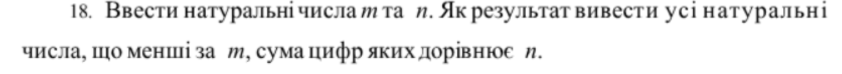
Київ 2021

**Лабораторна робота 5**

**Організація циклічних процесів. Складні цикли**

**Мета** – вивчити особливості організації складних циклів.

**Варіант №18.**



**Постановка задачі.** Результатом розв’язку є усі числа, які відповідають умовам задачі. Для визначення результату повинне бути задані натуральні числа m та n. Інших початкових даних для розв’язку не потрібно.

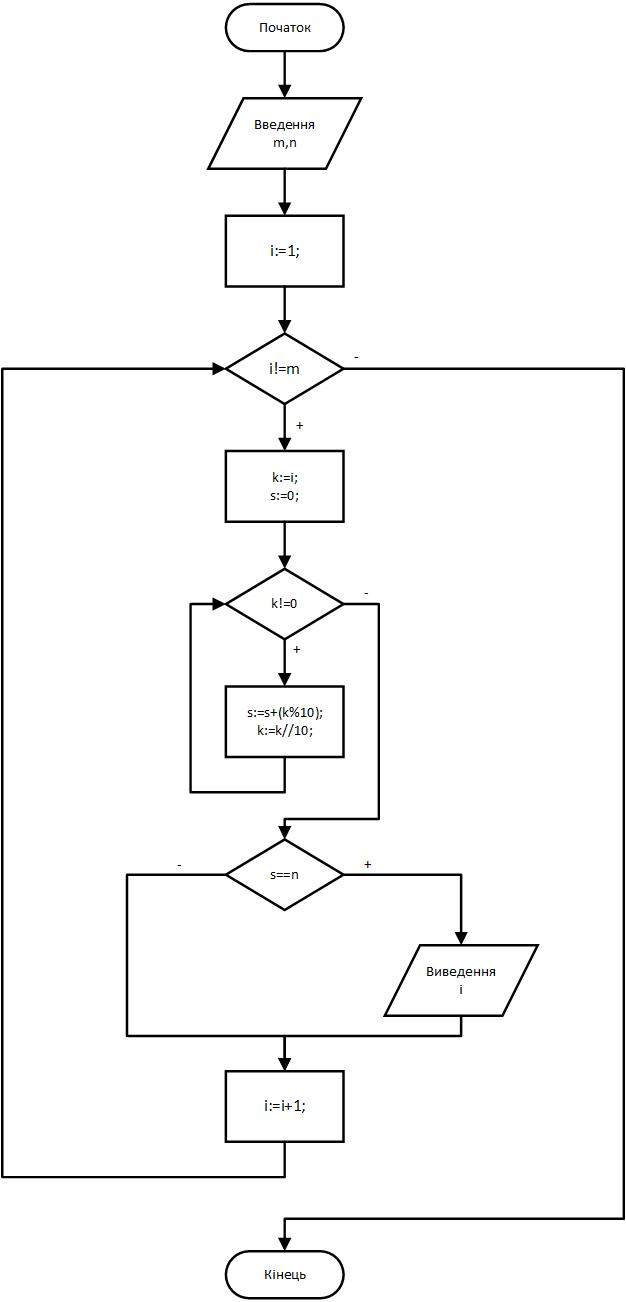
**Побудова математичної моделі.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Натуральне число m | Цілий | m | Початкове дане |
| Натуральне число n | Цілий | n | Початкове дане |
| Змінна для операцій з числом, що перевіряється | Цілий | k | Проміжний результат |
| Сума цифр числа | Цілий | s | Проміжний результат |
| Змінна для послідовного перебору чисел | Цілий | i | Проміжний результат |

Математичне формулювання задачі зводиться до знаходження цифр чисел, що перевіряються, шляхом виділення остачі від ділення на 10, перевірки рівности суми цих цифр і заданого натурального числа n. Якщо рівність виконується, виводимо число, що перевіряється.

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

*Блок-схема*



**Виконання мовою Python.**

**Код програми:**

m=int(input("Введіть натуральне число m: "))

n=int(input("Введіть натуральне число n: "))

i=1 #Ініціалізуємо змінну для перебору чисел від 1 до m

while i!=m: #Починаємо цикл, аби перевірити всі числа, менші за m

k=i #Призначаємо змінну k, яка дорівнює поточному і, аби проводити операції над числом, що перевіряється

s=0 #Присвоюємо змінній суми цифр числа значення 0

while k!=0: #Починаємо цикл для обчислення суми цифр числа

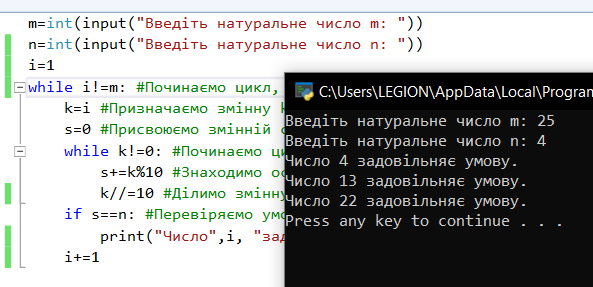
s+=k%10 #Знаходимо остачу від ділення на 10 - останню цифру змінної k - і додаємо її до змінної суми

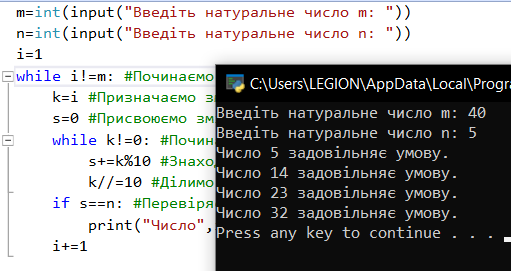
k//=10 #Ділимо змінну k націло на 10, тоді останньою цифрою стане передостання, і ми зможемо її відділити в наступній ітерації

if s==n: #Перевіряємо умову, чи сума цифр дорівнює заданому n

print("Число",i, "задовільняє умову.") #Якщо так, виводимо число, що перевірялося

i+=1 #Збільшуємо і на 1, аби в наступній ітерації перевіряти наступне число **Випробування алгоритму.**

****

****

**Виконання мовою C++.**

**Код програми:**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ukr");

int m, n, k, s; //Оголошуємо змінні m,n - задані в умові; k - змінна для значення числа, що перевіряється; s - змінна для суми цифр числа

cout << "Введiть натуральнi числа m та n:" << endl;

cin >> m >> n; //Вводимо з клавіатури числа m i n

int i = 1; //Інціалізуємо змінну для перебору чисел від 1 до m

while (i!=m) { //Починаємо цикл, аби перевірити всі числа, менші за m

k = i; //Призначаємо змінну k, яка дорівнює поточному і, аби проводити операції над числом, що перевіряється

s = 0; //Присвоюємо змінній суми цифр числа значення 0

while (k != 0) { //Починаємо цикл для обчислення суми цифр числа

s += k % 10; //Знаходимо остачу від ділення на 10 - останню цифру змінної k - і додаємо її до змінної суми

k /= 10; //Ділимо змінну k націло на 10, тоді останньою цифрою стане передостання, і ми зможемо її відділити в наступній ітерації

}

if (s == n) { //Перевіряємо умову, чи сума цифр дорівнює заданому n

cout << "Число " << i << " задовiльняє умову." << endl; //Якщо так, виводимо число, що перевірялося

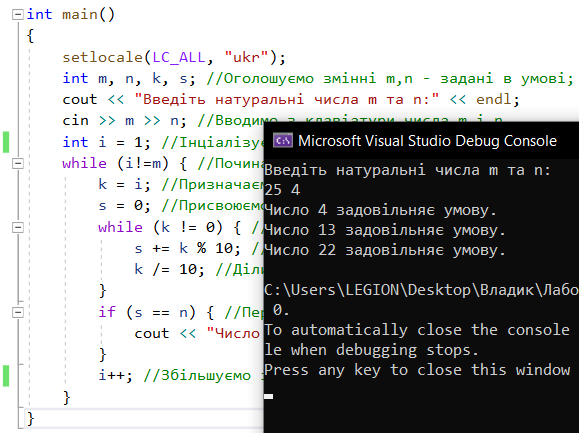
}

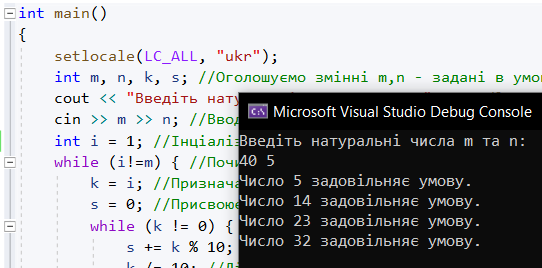
i++; //Збільшуємо i на одиницю, аби перевірити наступне число в наступній ітерації

}

}

**Випробування алгоритму.**

****

****

**Висновок.** Отже, у цій роботі я вивчив особливості організації складних циклів. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі, та блок-схему, яка пояснює логіку алгоритму. Використовуючи два ітераційних цикли з передумовою while, один з яких був вкладений у інший, перевіряючи задані умови, отримуємо коректний результат.