# 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 № 7 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів розгалуження»

Варіант 7

Виконав студент ІП-11 Головня Олександр Ростиславович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

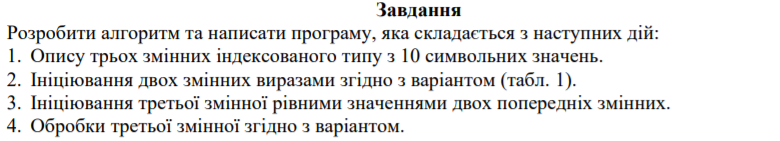
**Лабораторна робота №7**

**Дослідження лінійного пошуку в послідовностях**

**Мета** – дослідити методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

**Індивідуальне завдання:**

Варіант 7.





**Постановка задачі**

Створюємо масиви та допоміжні змінні. Заповнюємо перший масив, заповнюємо другий. Знаходимо рівні значення двох масивів та присвоюємо ці значення у третій. Обробляємо масив(Знаходимо суму значень та кількість елементів) і виводимо те що нас запитують, а саме: середнє арифметичне

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| 1 | Натуральний | Array1 | Початкове дане |
| 2 | Натуральний | Array2 | Початкове дане |
| 3 | Натуральний | Array3 | Проміжні дані |
| 4 | Натуральний | n | Проміжні дані |
| 5 | Дійсний | s | Проміжні дані |
| 6 | Дійсний | average | Вихідні дані |

**Розв’язання**

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

Крок 1. Визначимо основні дії:

Крок 2. Деталізація заповнення першого масива(Лічильником)

Крок 3. Деталізація заповнення другого масива(Лічильником)

Крок 4. Деталізація заповнення третього масива(За допомогою складних циклів і за певних математичних умов)

Крок 5. Знаходження суми та к-сть елементів(Лічильник)

Крок 6. Знаходження та вивід Середн. арифм.

**Псевдокод**

Крок 1

**Початок**

Заповнення першого масива

Заповнення другого масива

Заповнення третього масива

Знаходження суми та к-сть елементів

Знаходження та вивід Середн. арифм.

**Кінець**

Крок 2

**Початок**

Для (i = 0; i <a; i++)

array1[i] = (95 - 3\*i)

Заповнення другого масива

Заповнення третього масива

Знаходження суми та к-сть елементів

Знаходження та вивід Середн. арифм.

**Кінець**

Крок 3

**Початок**

Для (i = 0; i <a; i++)

array1[i] = (95 - 3\*i)

Для (i = 0; i < a; i++)

array2[i] = (74 + 3\*i)

Заповнення третього масива

Знаходження суми та к-сть елементів

Знаходження та вивід Середн. арифм.

**Кінець**

Крок 4

**Початок**

Для (i = 0; i <a; i++)

array1[i] = (95 - 3\*i)

Для (i = 0; i < a; i++)

array2[i] = (74 + 3\*i)

Для (i = 0; i < a; i++)

Для j = 0; j < a; j++)

Якщо (array1[i] == array2[j])

**То** array3[i] = array1[i]

Якщо (array3[i] != array1[i])

**То** array3[i] = 0

Знаходження суми та к-сть елементів

Знаходження та вивід Середн. арифм.

**Кінець**

Крок 5

**Початок**

Для (i = 0; i <a; i++)

array1[i] = (95 - 3\*i)

Для (i = 0; i < a; i++)

array2[i] = (74 + 3\*i)

Для (i = 0; i < a; i++)

Для j = 0; j < a; j++)

Якщо (array1[i] == array2[j])

**То** array3[i] = array1[i]

Якщо (array3[i] != array1[i])

**То** array3[i] = 0

Для (i = 0; i < a; i++)

Якщо ((int)array3[i] < 82)

**То** s = s + (int)array3[i];

n = n+1;

Знаходження та вивід Середн. арифм.

**Кінець**

Крок 6

**Початок**

Для (i = 0; i <a; i++)

array1[i] = (95 - 3\*i)

Для (i = 0; i < a; i++)

array2[i] = (74 + 3\*i)

Для (i = 0; i < a; i++)

Для j = 0; j < a; j++)

Якщо (array1[i] == array2[j])

**То** array3[i] = array1[i]

Якщо (array3[i] != array1[i])

**То** array3[i] = 0

Для (i = 0; i < a; i++)

Якщо ((int)array3[i] < 82)

**То** s = s + (int)array3[i];

n = n+1;

average = s / n

Вивід average

**Кінець**

**Блок-схема**

**Крок 1**



**Крок 2**



**Крок 3**



**Крок 4**

**Крок 5** 

**Крок 5** 



**Код програми:**

#include<iostream>

using namespace std;

int main() {

const int a = 10;

double average = 0;

char array1[a], array2[a], array3[a];

int n = 0;

double s = 0;

cout << "first array: " << endl;

for (int i = 0; i <a; i++)

{

array1[i] = (95 - 3\*i);

cout << (int)array1[i] <<"\t"<<array1[i] << endl;

}

cout<<"second array: "<<endl;

for (int i = 0; i < a; i++)

{

array2[i] = (74 + 3\*i);

cout << (int)array2[i] << "\t" << array2[i] << endl;

}

cout << "third array: " << endl;

for (int i = 0; i < a; i++)

{

for (int j = 0; j < a; j++)

{

if (array1[i] == array2[j])

array3[i] = array1[i];

}

if (array3[i] != array1[i])

array3[i] = 0;

cout << (int)array3[i] << "\t" << array3[i] << endl;

}

for (int i = 0; i < a; i++)

{

if (((int)array3[i] < 82) && ((int)array3[i]>0))

{

s = s + (int)array3[i];

n = n+1;

}

}

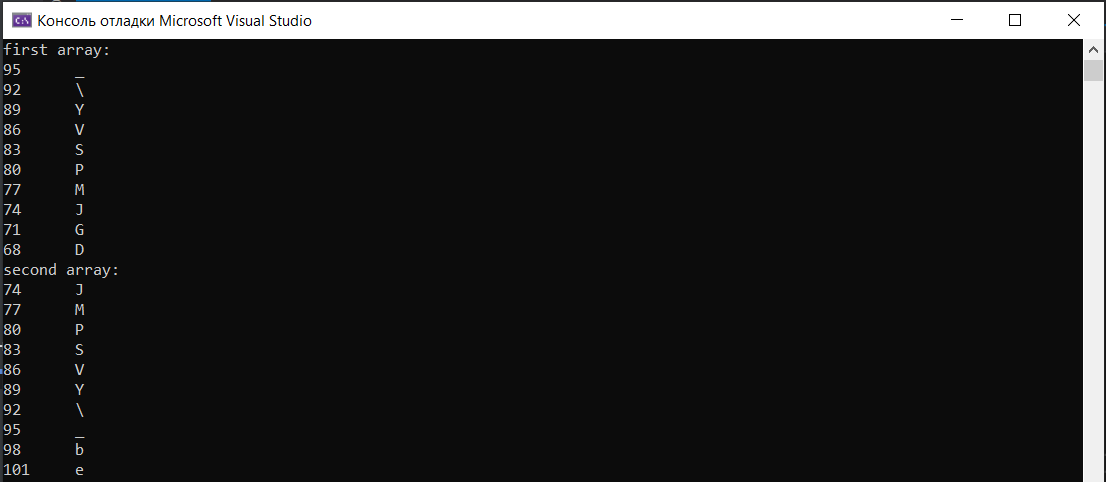
average = s / n;

cout <<"suma: " << s << endl;

cout <<"amount: " << n << endl;

cout <<"average: "<< average << endl;

}



**Тестування**

|  |  |
| --- | --- |
| Масив | Елементи |
| 1 | 95 \_  92 \  89 Y  86 V  83 S  80 P  77 M  74 J  71 G  68 D |
| 2 | 74 J  77 M  80 P  83 S  86 V  89 Y  92 \  95 \_  98 b  101 e |
| 3(Спільні елементи) | 95 \_  92 \  89 Y  86 V  83 S  80 P  77 M  74 J  0  0 |
| Сума елементів коди < 82 | 80 + 77 + 74 + 0 + 0 = 231 |
| Кількість цих елементів | 3 |
| Середнє арифметичне | 231 / 5 = 77 |

**Висновок**

Отже, ми дослідили методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій, а саме: для дослідження лінійного пошуку використали декілька операторів повторення та один складний цикл з повторенням, за допомогою яких звертались до кожного елемента послідовності.