# 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

«Основи програмування-1.

Базові конструкції»

«Одновимірні масиви»

Варіант 20

Виконав студент Лошак Віктор Іванович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 7**

**Мета:**

Вивчити особливості обробки одновимірних масивів.

**Тема:**

Для заданих цілочисельних масивів M(z) i K(z) знайти найменше серед тих значень масиву М які не входять до масиву К(вважати що є хоча б одне таке значення).

**Математична модель:**

Для обчислення виразу при заданих масивах M(z) i K(z) використаємо підпрограми. Для спрощення обрахунків використаємо дві підпрограми. Перша підпрограма буде проводити ініціалізацію переданого їй масиву. Друга підпрограма буде розраховувати найменше значення з тих що є унікальними для М. Окрема підпрограма буде виводити значення елементів масивів на екран. Аргументи передаватимемо як вказівники, оскільки функції призначені для обробки масивів. Для ініціалізації масивів використаємо арифметичні цикли та ГПВЧ стандартної бібліотеки. Для форматування полів виведення та отримання випадкових чисел використаємо бібліотеки <iomanip> <cstdlib> та <ctime> в с++.

**1)**Приклад коду на С++:

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <iomanip>>

using namespace std;

// оголошення функцій

void arrayInit(int\* p, int size);

int findTheSmallestNotMatching(int\* m, int\* k, int size);

void printArray(int\* arr, int size, int arrayNum);

void main( )

{

// встановлюємо точку відліку ГПВЧ

srand(time(0));

//ввід розмірності масивів

int inputSize;

cout << "Enter the number of elements you want to enter: ";

cin >> inputSize;

cout << "\n";

//динамічне виділення масивів

int\* m = new int[inputSize];

int\* k = new int[inputSize];

// ініціалізація масивів рандомними значеннями

arrayInit(m, inputSize);

arrayInit(k, inputSize);

//виведення значенб елементів

printArray(m, inputSize, 1);

printArray(k, inputSize, 2);

// виведення результатів обчислень

int result = findTheSmallestNotMatching(m, k, inputSize);

if (result) cout << "\nThe smallest not matching element is: " << result << endl;

else cout << "\nThe smallest not matching element is not found." << endl;

}

void arrayInit(int\* p, int size)

{

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

{

\*(p + i) = (rand( ) % size) + 1;

}

}

int findTheSmallestNotMatching(int\* m, int\* k, int size)

{

bool foundMatch;

int theSmallest = INT\_MAX;

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

{

foundMatch = false;

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

{

if (\*(m + i) == \*(k + j)) foundMatch = true;

}

if (!foundMatch && theSmallest > \*(m + i))

{

theSmallest = \*(m + i);

}

}

return theSmallest != INT\_MAX ? theSmallest : 0;

}

void printArray(int\* arr, int size, int arrayNum)

{

cout << setw(3) << "The lements of the " << arrayNum << " array are:\n";

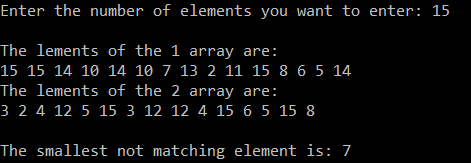
for (int i = 0; i < size; ++i)

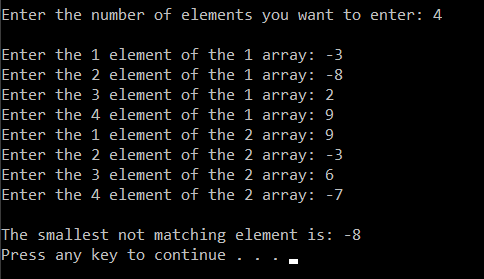
cout << \*(arr + i) << " ";

cout << endl;

}

Виконання коду на С++:





**Висновок:**

Отже за допомогою даного алгоритму ми успішно обрахували і відобразили на екрані значення заданого найменшого унікального для M елемента при довільних ввідних даних, за допомогою використання підпрограм на компільованій мові С++ .Ми також створили алгоритми лінійного пошуку значення у масиві та аналізували умову для знаходження оптимальних шляхів виконання завдання.