Івано-Франківський національний технічний університет нафти і газу

Кафедра Комп’ютерних технологій ДНУ

Звіт до лабораторної роботи №1

«Алгоритми обробки масивів»

з курсу «Алгоритми та структури даних»

Виконала:

ст.гр. ІП-22-4

Томин Є.В.

Перевірив:

Григорчук Л. І.

Івано-Франківськ

2023

**Тема.** Алгоритми обробки масивів

**Мета:** сформувати та закріпити практичні навички розробки алгоритмів обробки масивів з урахуванням їх ефективності та вмінням аналізувати алгоритми, оцінювати їх обчислювальну складність та ефективність**.**

**Постановка завдання:**

Мовою С/С++ розробити програму, яка реалізує такі дії:

-створює статичний масив з N=100 елементів;

-заповнює масив n випадковими цілими числами з діапазону, границі якого вводяться з клавіатури;

-виводить заповнену частину (з n елементів) на екран;

- виконує обробку масиву з n елементів згідно з завданням за варіантами. Передбачити обробку помилкових ситуацій. Зокрема:

- якщо в результаті обробки масиву після видалення елементів масив виявиться порожнім, видати відповідне повідомлення;

- якщо додавання елементів призведе до виходу за межі масиву (кількість елементів перевищить N), видати відповідне повідомлення.

Програму слід оформити таким чином, щоб усі функції та завдання можна було б продемонструвати за один сеанс роботи програми.

**Варіант 12**

**Завдання:**

**1.**Сформувати новий масив з елементів даного масиву, які є простими числами. Підрахувати їх кількість. Результуючий масив вивести на екран.

**2.** Вставити елемент на позицію з номером К. Номер К та значення елемента ввести з клавіатури. Попередньо К-тий елемент та всі елементи , розташовані після цієї позиції, посунути на одну позицію вправо. Якщо додавання елементів у масив неможливе (кількість елементів після додавання перевищить N-максимальну кількість елементів у масиві), то видати відповідне повідомлення. У випадку успішної операції вивести на екран масив до та після перетворення.

**Теоретичні відомості**

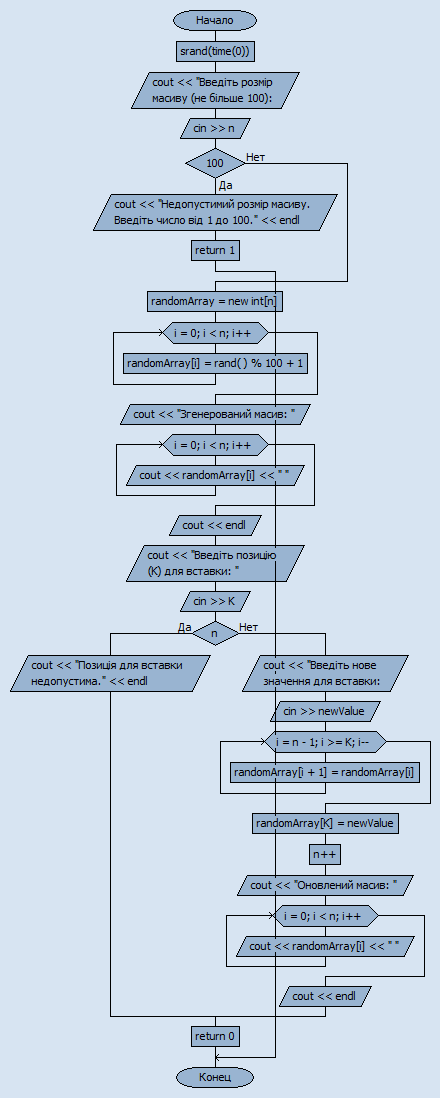
**Масив** – це впорядкована індексована сукупність однотипних елементів, що мають спільне ім’я. Елементами масиву можуть бути дані різних типів. Кожен елемент масиву однозначно визначається ім’ям та індексами (для одновимірного масиву – індексом).

**Алгоритм** — це послідовність дій, яка буде виконана для досягнення цілі, маніпулюючи вхідними даними. Алгоритм може відрізнятися в залежності від того, яку структуру даних ми обираємо. Майже завжди є можливість вирішити проблему декількома способами, застосовуючи різні алгоритми. Якісь будуть ефективними, а інші — неефективними.

**Структура даних** — це спосіб організації даних. Залежно від того, якими даними ми хочемо маніпулювати і що з ними робити, — повинні обрати ту чи іншу структуру даних, яка ідеально підходить для нашого випадку.

**Виконання**

**Блок-схема**

****

**Завдання 1**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

bool isPrime(int number) {

if (number <= 1) return false;

if (number <= 3) return true;

if (number % 2 == 0 || number % 3 == 0) return false;

for (int i = 5; i \* i <= number; i += 6) {

if (number % i == 0 || number % (i + 2) == 0) return false;

}

return true;

}

int main() {

srand(time(0));

int n;

cout << " Введіть розмір масиву (не більше 100): ";

cin >> n;

if (n <= 0 || n > 100) {

cout << "Недопустимий розмір масиву. " << endl;

return 1;

}

int lowerBound;

cout << "Нижня межа: ";

cin >> lowerBound;

int upperBound;

cout << "Верхня межа: ";

cin >> upperBound;

int \*randomArray = new int[n];

for (int i = 0; i < n; i++) {

randomArray[i] = rand() % (upperBound - lowerBound + 1) + lowerBound;

}

cout << "Згенерований масив: ";

for (int i = 0; i < n; i++) {

cout << randomArray[i] << " ";

}

cout << endl;

int \*primeArray = new int[n];

int primeCount = 0;

for (int i = 0; i < n; i++) {

if (isPrime(randomArray[i])) {

primeArray[primeCount] = randomArray[i];

primeCount++;

}

}

cout << "Масив простих чисел: ";

for (int i = 0; i < primeCount; i++) {

cout << primeArray[i] << " ";

}

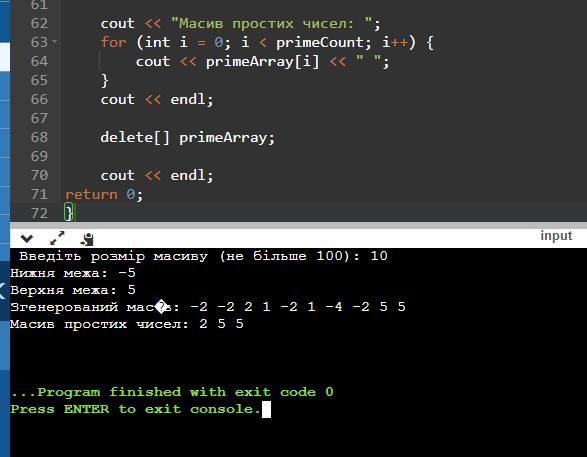
cout << endl;

delete[] primeArray;

cout << endl;

return 0;

}



**Мал.1** Виконано завдання 1, програма генерує масив з простими числами.

**Завдання 2**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

bool isPrime(int number) {

if (number <= 1) return false;

if (number <= 3) return true;

if (number % 2 == 0 || number % 3 == 0) return false;

for (int i = 5; i \* i <= number; i += 6) {

if (number % i == 0 || number % (i + 2) == 0) return false;

}

return true;

}

int main() {

srand(time(0));

int n;

cout << " Введіть розмір масиву (не більше 100): ";

cin >> n;

if (n <= 0 || n > 100) {

cout << "Недопустимий розмір масиву. " << endl;

return 1;

}

int lowerBound;

cout << "Нижня межа: ";

cin >> lowerBound;

int upperBound;

cout << "Верхня межа: ";

cin >> upperBound;

int \*randomArray = new int[n];

for (int i = 0; i < n; i++) {

randomArray[i] = rand() % (upperBound - lowerBound + 1) + lowerBound;

}

cout << "Згенерований масив: ";

for (int i = 0; i < n; i++) {

cout << randomArray[i] << " ";

}

cout << endl;

int \*primeArray = new int[n];

int primeCount = 0;

for (int i = 0; i < n; i++) {

if (isPrime(randomArray[i])) {

primeArray[primeCount] = randomArray[i];

primeCount++;

}

}

cout << "Масив простих чисел: ";

for (int i = 0; i < primeCount; i++) {

cout << primeArray[i] << " ";

}

cout << endl;

delete[] primeArray;

cout << endl;

int k;

cout << "Вибнріть номер позиції К для вставки: ";

cin >> k;

if (k < 0 || k >= n) {

cout << "Недопустимий номер позиції" << endl;

delete[] randomArray;

return 1;

}

int newValue;

cout << "Введіть нове значення для вставки:";

cin >> newValue;

if (n == 100) {

cout << "Масив уже має максимальний розмір; вставка неможлива." << endl;

delete[] randomArray;

return 1;

}

n++;

for (int i = n - 1; i > k; i--) {

randomArray[i] = randomArray[i - 1];

}

randomArray[k] = newValue;

cout << "Масив після вставки: ";

for (int i = 0; i < n; i++) {

cout << randomArray[i] << " ";

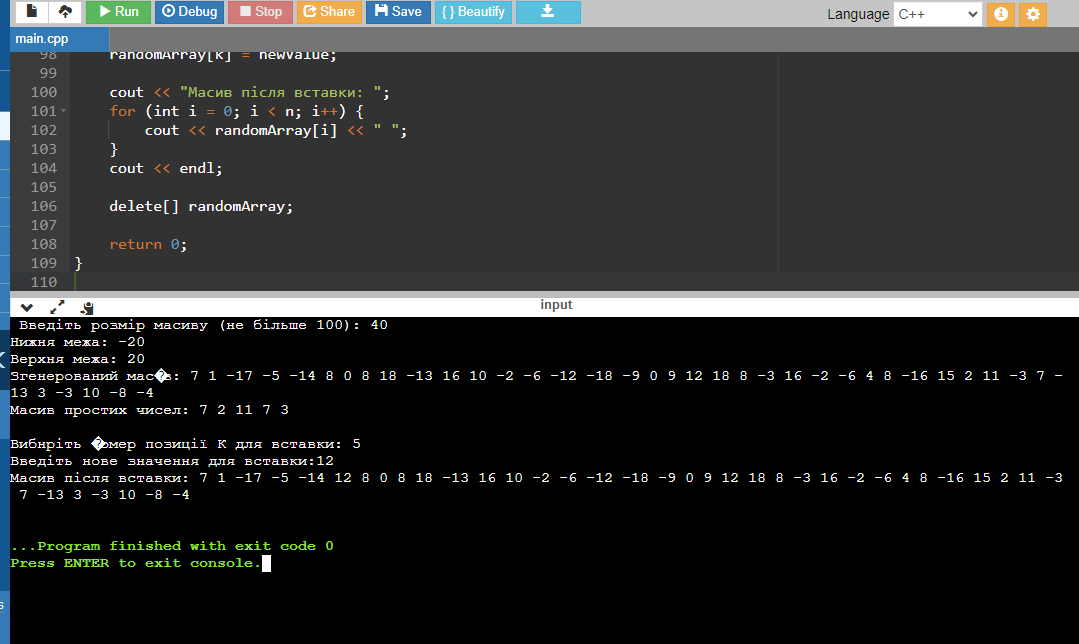
}

cout << endl;

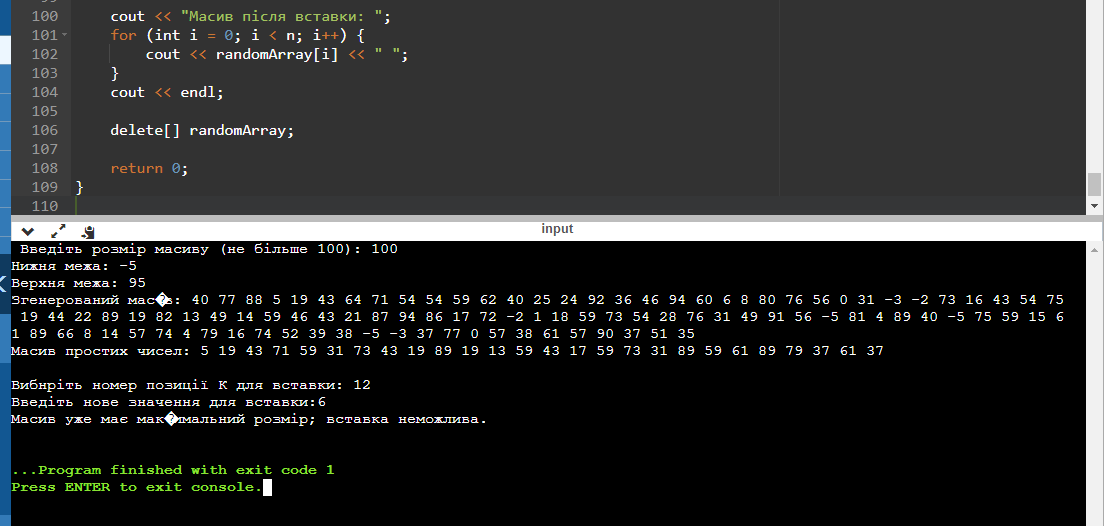
delete[] randomArray;

return 0;

}



**Мал.2** Вставка довільного елемента на позицію К і генерування нового масиву.



**Мал.3** Генерування неможливе, масив – переповнений.

**Висновок:** виконуючи лабораторну роботу я навчилася розробляти алгоритми обробки масивів з урахуванням їх ефективності, навчилася аналізувати алгоритми, оцінювати їх обчислювальну складність та ефективність.