**Міністерство Освіти і Науки України**

**Київський Національний Університет імені Тараса Шевченка**

**Факультет Інформаційних Технологій**

Звіт з лабораторної роботи № 7

з дисципліни **Основи програмування**

Виконав студент групи ІР-12

Федоренко Ярослав Юрійович

Викладач: ас. Пороховніченко І.А.

**Київ – 2023**

**Лабораторна робота №7**

**Варіант № 18**

**Тема:** Одновимірні масиви.

**Мета:** Ознайомитися з одновимірними масивами як складними типами даних. Навчитися будувати алгоритми із застосуванням впорядкованих послідовностей даних.

**Завдання № 1**



**Блок – схема**





**­Код**

**main.cpp**

#include <iostream>

#include "functions.h"

using namespace std;

int main()

{

char q = 'y';

do {

int length = 0;

cout << "Введіть кількість елементів масиву: ";

cin >> length;

int \* array = new int[length];

createArray(array, length);

int newLength = removeExtraElements(array, length, length);

cout << endl << "Масив після видалення зайвих елементів: ";

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

cout << array[i] << " ";

}

delete[] array;

cout << endl << endl << "Бажаєте повторити? (y/n): ";

cin >> q;

if (q != 'y')

cout << endl << "До побачення!";

else

cout << endl;

} while (q == 'y');

return 0;

}

**functions.cpp**

#include "functions.h"

void createArray(int \* array, const int length)

{

int method = 0;

cout << "Введіть 1 для вводу елементів масиву вручну або 2 для генерації випадкових чисел: ";

cin >> method;

switch(method) {

case 1:

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

cout << "Введіть " << i << " елемент масиву: ";

cin >> array[i];

}

return;

case 2:

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

array[i] = rand() % 100;

}

return;

default:

cout << endl << "Невірний метод!" << endl;

break;

}

return createArray(array, length);

}

int countElement(int \* array, const int length, const int index)

{

int count = 0;

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

if (array[i] == array[index]) {

count++;

}

}

return count;

}

void removeElement(int \*& array, const int length, const int element, const int count)

{

int \* newArray = new int[length - (count - 2)];

int foundCount = 0;

int j = 0;

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

if (array[i] == element) {

foundCount++;

if (foundCount > 2) {

continue;

}

}

newArray[j++] = array[i];

}

delete[] array;

array = newArray;

}

int removeExtraElements(int \*& array, const int length, int updatedLength)

{

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

int count = countElement(array, length, i);

if (count > 2) {

removeElement(array, length, array[i], count);

return removeExtraElements(

array,

length - (count - 2),

updatedLength - (count - 2)

);

}

}

return updatedLength == 0 ? length : updatedLength;

}

**Тестування**

При введених значеннях 2, 1, 3, 4, 8, 9, 2, 2, 2, 1, 1 отримуємо 2, 1, 3, 4, 8, 9, 2, 1. Бачимо, що всі повторення елементів більше двох разів видалені. Програма працює коректно.

**Завдання № 2**



**Блок – схема**





**Код**

**main.cpp**

#include <iostream>

#include "functions.h"

using namespace std;

int main()

{

char q = 'y';

do {

int lengthA = 0;

cout << "Введіть кількість елементів масиву A: ";

cin >> lengthA;

int \* A = new int[lengthA];

createArray(A, lengthA);

cout << endl;

int lengthB = 0;

cout << "Введіть кількість елементів масиву B: ";

cin >> lengthB;

int \* B = new int[lengthB];

createArray(B, lengthB);

int lengthD = 0;

int \* D = new int[lengthD];

lengthD = addEvens(D, A, lengthD, lengthA);

lengthD = addEvens(D, B, lengthD, lengthB);

cout << endl << "D: ";

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

cout << D[i] << " ";

}

delete[] A;

delete[] B;

delete[] D;

cout << endl << endl << "Бажаєте повторити? (y/n): ";

cin >> q;

if (q != 'y')

cout << endl << "До побачення!";

else

cout << endl;

} while (q == 'y');

return 0;

}

**functions.cpp**

#include "functions.h"

void createArray(int \* array, const int length)

{

int method = 0;

cout << "Введіть 1 для вводу елементів масиву вручну або 2 для генерації випадкових чисел: ";

cin >> method;

switch(method) {

case 1:

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

cout << "Введіть " << i << " елемент масиву: ";

cin >> array[i];

}

return;

case 2:

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

array[i] = rand() % 100;

}

return;

default:

cout << endl << "Невірний метод!" << endl;

break;

}

return createArray(array, length);

}

int getEvensCount(int \*& array, int length)

{

int evensCount = 0;

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

if (array[i] % 2 == 0) {

evensCount++;

}

}

return evensCount;

}

int addEvens(int \*& container, int \*& array, int containerSize, int arraySize)

{

int evensCount = getEvensCount(array, arraySize);

int newLength = containerSize + evensCount;

int \* newContainer = new int[newLength];

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

newContainer[i] = container[i];

}

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

if (array[i] % 2 == 0) {

newContainer[containerSize++] = 2 \* array[i];

}

}

delete[] container;

container = newContainer;

return newLength;

}

**Тестування**

Вводимо 2 вектори: 1, 2, 3, 4 і 5, 6, 7, 8. Отримуємо вектор D 4, 8, 12, 16. Програма працює коректно.

**Висновок**

В цій лабараторній роботі я ознайомився з одновимірними масивами і навчився будувати алгоритми із застосуванням впорядкованих послідовностей даних.