Міністерство освіти України

Національний технічний університет "ХПІ"

Кафедра "Системний аналіз та інформаційно-аналітичні технології"

**Звіт**

**Лабораторна робота 5**

з дисципліни "Об’єктно орієнтоване програмування"

Виконав: студент групи КН-1223Б

Фільчак М.В.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

[1. Мета роботи 3](#_Toc150259069)

[2. Завдання 3](#_Toc150259070)

[3. Теоретичні відомості 3](#_Toc150259071)

[4. Вирішення завдання 4](#_Toc150259072)

[4.1. Розробка алгоритму завдання 4](#_Toc150259073)

[4.2. Змінні програми 5](#_Toc150259074)

[4.3. Розробка програми 5](#_Toc150259075)

[4.4. Налагодження програми 6](#_Toc150259076)

[4.5. Результати роботи програми 6](#_Toc150259077)

[5. Висновок 6](#_Toc150259078)

**Лабораторна робота №5**

**Робота з масивами**

Варіант 16

# Мета роботи

Метою лабораторної роботи є отримання практичних навичок в роботі з масивами мови C++.

# Завдання

|  |  |
| --- | --- |
| 16 | Порахувати кількість елементів з однаковим значенням у введеному з клавіатури масиві. |

# Теоретичні відомості

Масив у мові програмування C++ є структурою даних, яка дозволяє зберігати набір змінних однакового типу. Масиви використовуються для зберігання даних, таких як числа, рядки або об'єкти, і дозволяють легко отримувати доступ до окремих елементів за їхніми індексами без необхідності введення окремого імені змінної для кожного з них.

Масиви можуть бути одновимірними та багатовимірними. В одновимірних масивах для доступу до елементу масиву використовується один індекс. В багатовимірних масивах для доступу до елементу масиву використовується декілька індексів. Масив, як і будь-який інший об’єкт в мові *С*, має бути оголошений перед тим, як він буде використаний:

Загальна форма опису одновимірного масиву:

тип ім’я\_масиву[розмір];

*тип* – це тип елементів масиву. Він ще називається базовим типом.

*розмір* – кількість елементів в масиві;

*ім’я\_масиву* – безпосередньо ім’я масиву, за яким здійснюється доступ до елементів масиву.

Індекс масиву визначає елемент масиву, до якого треба забезпечити доступ, і вказується в квадратних дужках після імені масиву. Індекс масиву – це цілочисельний вираз, значення якого може бути в діапазоні від 0 до значення, рівного розміру масиву, зменшеного на 1. Наприклад:

int A[10];

У результаті, в пам’яті комп’ютера виділяється 10 комірок типу int. Одна комірка займає 4 байти, всього буде виділено 40 байт пам’яті. Номер першої комірки починається з нуля. Ці комірки об’єднані спільним іменем A.

int B[3][3];

Таким чином оголошується двовимірний цілочисельний масив. Його можна розглядати як набір з трьох 3-елементних масивів цілих чисел.

Елементи масиву зберігаються в пам’яті послідовно. Якщо, наприклад, перший елемент символьного масиву *id* зберігається за адресою 5000, то другий зберігатиметься за адресою 5001, третій – за адресою 5002.

Двовимірний масив розташовується в пам’яті за рядками: спочатку – нульовий рядок, потім – перший і тд. Так, елементи *B*[3][3] зберігаються в наступному порядку: [0][0], [0][1], [0][2], [1][0], [1][1], [1][2], [2][0], [2][1], [2][2].

Якщо в масиві не дуже багато елементів і їх значення відомі заздалегідь, масив можна ініціалізувати разом з його оголошенням, розмістивши перелік значень у фігурних дужках:

int N[10] = {0,1,2,3,4,5,6,7,8,9};

Якщо ж в масиві багато елементів, заповнити його їх значеннями доведеться програмно в циклі.

# Вирішення завдання

## Розробка алгоритму завдання

Відповідно до завдання, нам необхідно оголосити масив цілих чисел і надати можливість оператору заповнити його відповідними значеннями. Розмір масиву і діапазон значень його елементів задані у виданому варіанті індивідуального завдання. В індивідуальних завданнях вказано також, яку обробку масиву слід провести.

Для всіх варіантів завдання слід мати на увазі наступне:

нуль вважається додатним числом, якщо в завданні не обумовлений якийсь інший його статус;

коли мова йде про якісь послідовності чисел, мається на увазі послідовність з довжиною, більшою одиниці;

у тих випадках, коли завдання вимагає виконання якихось обчислень, дозволяється виконувати їх з тією точністю, яку забезпечують операції цілочисельної арифметики.

## Змінні програми

Для реалізації алгоритму необхідні наступні змінні.

Спочатку задаємо розмір масиву за допомогою постійного значення.

const int N = 100;

Далі додаємо масив цілих чисел, який оброблятиметься:

int mas[N];

За допомогою циклу зі змінними I та j, а також цілочисельної змінної поступово задаємо значення для елементів масиву.

int num;

За допомогою останніх двох змінних ми рахуємо кількість разів, що ті чи інші значення масиву з’явилися.

int count\_negative\_num;

int count;

## Розробка програми

Спочатку оператор повідомляє розмір масиву, що він буде заповнювати. Це виконується за допомогою обмеження роботи циклів функцій до значення, що його ввів користувач. Далі оператор повинен заповнити масив деякими значеннями. Для цього необхідно використати цикл з лічильником в кожній ітерації якого оператор задає значення для наступного елементу масиву. Після закінчення циклу заповнення вміст масиву виводимо на екран.

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

cin >> num;

mas[i] = num;

}

for (int i = 0; i < n; i++) cout << setw(6) << mas[i];

Далі необхідно порахувати кількість елементів з однаковим значенням у введеному з клавіатури масиві. Для цього використовуємо декілька циклів for() та змінні count та count\_negative\_num. Друга змінна змінює значення елементів масиву, що вже повторювалися на інше число. Це необхідно для того, щоб числа, які вже були пораховані не зчитувались по декілька разів, бо вони не будуть проходити оператор if().

if (mas[i] == mas[j]) {

count++;

count\_negative\_num--;

mas[j] = -158;

}

Наступний крок – це виведення елементів масиву поряд з кількістю разів, що вони з’являлися.

cout << mas[i] << " appears " << count << " times." << endl;

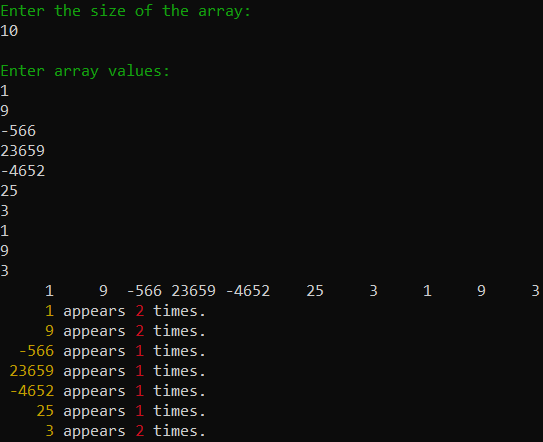
Наприкінці необхідно повторити виведення для того числа, на яке ми змінювали числа, що повторюються.

## Налагодження програми

Налагодження програми починається з перевірки результатів отриманих під час роботи програми. Далі необхідно провести перевірну на тих значеннях, що можуть спричинити збій у роботі програми, або призвести до некоректного чи неправильного перебігу роботи програми. До таких значень можуть належати від’ємні числа, нуль та число, що використовується як заміна для значень, що повторюються. Якщо зазначено у завданні, також слід зробити перевірку на тих значеннях, що повинні викликати виведення повідомлення про помилку.

## Результати роботи програми

Скріншот результатів роботи програми у консолі надано нижче:



# Висновок

Під час виконання цієї лабораторної роботи необхідно було створити програму, яка буде рахувати кількість разів, що числа з’являються в масиві заданому з клавіатури. Для цього використовуємо подвійний цикл for() в якому послідовно порівнюються значення елементів від першого до останнього.

Однією із причин некоректної роботи програми може бути повторне зарахування одного і того ж значення. Ця проблема починає виникати при наявності 3 і більше однакових елементів. Для її вирішення можна використовувати заміну значень для елементів, що вже були пораховані. Тобто, потрібно в разі збігання значень у двох елементів масиву замінювати значення другого на деяке число, що потім буде виключено із циклу перевірки за допомогою оператора if().

При розв’язанні одного із додаткових завдань цієї лабораторної роботи необхідно було видалити, а потім вставити до масиву випадкових значень деяку кількість елементів, що задається оператором.

Для того, щоб вставити елементи в масив можна використати різні підходи. Одним з них є створення нового масиву, що буде вміщувати старий масив та нові значення. Або можна заздалегідь ініціалізовувати більший масив, ніж потрібно для записування початкових даних, а потім заповнити його новими значеннями.

Для видалення елементів з масиву можна змінити значення окремих елементів цього масиву на деяке число, що потім не буде відповідати умові оператора if(), який виводитиме значення масиву на екран консолі.