Міністерство освіти України

Національний технічний університет "ХПІ"

Кафедра "Системний аналіз та інформаційно-аналітичні технології"

**Звіт**

**Лабораторна робота 6**

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](#_Toc152030923)

[2. Завдання 3](#_Toc152030924)

[3. Теоретичні відомості 3](#_Toc152030925)

[4. Вирішення завдання 4](#_Toc152030926)

[4.1. Розробка алгоритму завдання 4](#_Toc152030927)

[4.2. Змінні програми 5](#_Toc152030928)

[4.3. Розробка програми 5](#_Toc152030929)

[4.4. Налагодження програми 5](#_Toc152030930)

[4.5. Результати роботи програми 5](#_Toc152030931)

[5. Додаткові завдання 6](#_Toc152030932)

[5.1. Умови завдання 6](#_Toc152030933)

[5.2. Алгоритм розв’язання 6](#_Toc152030934)

[5.3. Результати роботи програми 7](#_Toc152030935)

[6. Висновок 7](#_Toc152030936)

**Лабораторна робота №6**

**Робота з матрицями**

Варіант 16

# Мета роботи

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

# Завдання

|  |  |  |
| --- | --- | --- |
| 16 | Відобразити ліву половину матриці на праву дзеркально симетрично щодо вертикальної осі. Значення елементів матриці – випадкові числа. |  |

# 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 заповнити його випадковими значеннями. Розмір масиву і діапазон значень його елементів не задані у виданому варіанті індивідуального завдання. Тому слід просто заповнити масив деякого розміру, і, у разі потреби, змінювати значення вже у коді. В індивідуальних завданнях вказано яку саме 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.

#define N 8;

Далі додаємо масив цілих чисел, який оброблятиметься:

int mas[N][N];

За допомогою циклу зі змінними I та j поступово задаємо випадкові значення для елементів масиву.

## Розробка програми

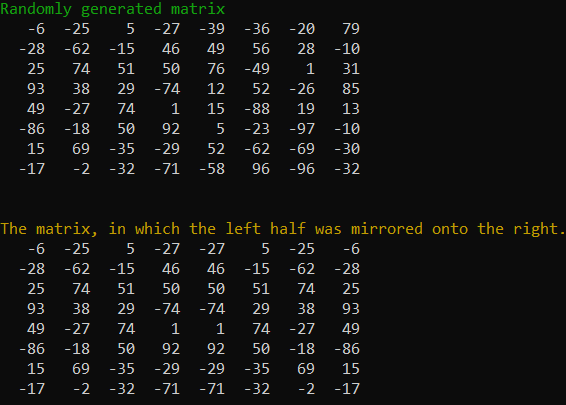
Спочатку програма відображує випадково згенерований масив, далі, за допомогою циклу for(), відображаємо ліву половину матриці на праву дзеркально симетрично щодо вертикальної осі. Останнім кроком в основній частині завдання є вивід нової матриці в консолі.

## 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:



# Додаткові завдання

1. Умови завдання

У двовимірному масиві (кожного разу працювати з **даним** масивом):

а) знайти **максимальний** в кожному рядку і поміняти його місцями з першим елементом цього ж рядка;

б) знайти **мінімальний** в кожному стовпці і поміняти його місцями з останнім елементом цього ж стовпця.

1. Алгоритм розв’язання

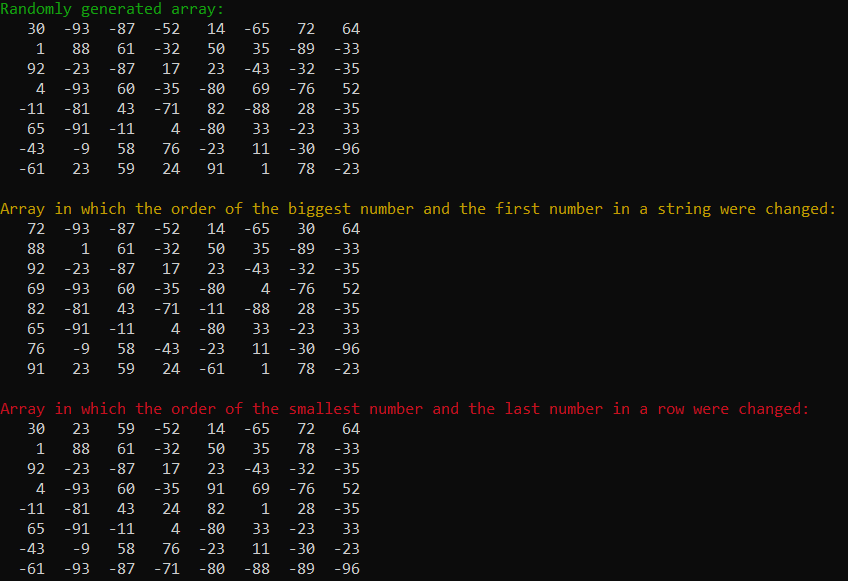
Відповідно до завдання, необхідно знайти максимальне значенняі поміняти його місцями з першим елементом цього ж рядка. Далі необхідно знайти мінімальне значенняв кожному стовпці і поміняти його місцями з останнім елементом цього ж стовпця.

Важливо зробити так, щоб ці завдання не впливали на одне одного. Тому, необхідно заповнити 2 однакові масиви й використати кожен для виконання 1 частини завдання.

Для цього оголошуємо 2 однакові масиви, що заповнюються випадковими значеннями. Далі необхідно, використовуючи тимчасову змінну, змінити значення у цих масивах на ті, що будуть відповідати умові завдання. Наостанок, виводимо початковий і два змінені масиви в консолі.

1. Результати роботи програми

Скріншот результатів роботи програми у консолі надано нижче:



# Висновок

Під час виконання цієї лабораторної роботи необхідно було створити програму, яка буде віддзеркалювати ліву половину матриці на праву симетрично щодо вертикальної осі. Для цього використовуємо подвійний цикл for(), у якому надаємо випадкових значень для елементів масиву.

Однією із причин некоректної роботи програми може бути неправильне обмеження для роботи з вже заповненим масивом. Тобто, потрібно переконатися, що віддзеркалення відбувається горизонтально, а не вертикально. Коректний алгоритм заповнення масиву для цього випадку виглядає наступним чином: необхідно заповнити всі стовпці, а рядки заповнювати лише на половину.

При розв’язанні одного із додаткових завдань цієї лабораторної роботи необхідно було згенерувати масив з випадковими значеннями. У цьому масиві необхідно було знайти найбільше значення у рядку і замінити його з першим значенням цього ж рядка. У другі частині завдання необхідно було знайти найменший елемент стовпця і замінити його з останнім елементом цього ж стовпця.

Для виконання цих додаткових завдань можна використовувати схожий, але протилежний алгоритм. Спочатку встановити значення з яким будемо порівнювати елементи масивів. Для першого завдання це буде значення, що на 1 менше за мінімальне можливе значення масиву, а в другому – навпаки. Потім, ми порівнюємо ці значення з елементами масивів так, як зазначено в умові. Наостанок, за допомогою тимчасової змінної ми міняємо необхідні елементи місцями й виводимо модифіковані масиви на екран.