**Звіт з лабораторної роботи №4**

**Монастирьова Антона ПФНКТ-1**

**Теоретичні відомості**

**Вказівник** – це змінна, яка може містити адресу деякого іншого об’єкта. Ним може бути інша змінна або масив, структура, функція, клас (в С++) тощо.

**Синтаксис визначення вказівника:**

**<тип> \* <ідентифікатор\_вказівника>;**

В цьому прикладі вказівник поки що не вказує ні на яки об’єкт – він не проініціалізованй.

Якщо p – вказівник на деяку існуючу змінну, то \*p – це сама змінна. Тут \* – є операцією розадресації вказівника.

В мові С для ініціалізація вказівника можлива одним з двох способів:

1. ініціалізація вказівника адресою існуючої змінної;
2. ініціалізація вказівника адресою динамічно виділеної пам’яті (з допомогою функції malloc(), calloc()).

В мові С++ для ініціалізації вказівника може використовуватись також операція new, яка динамічно виділяє необхідний об'єм пам'яті під змінну.

**Синтаксис функцій malloc і calloc:**

**void\* malloc (size\_t size);**

– виділяє size байтів пам'яті в області Heap і повертає вказівник на початок цієї області.

**void\* calloc (size\_t n, size\_t size);**

– повертає вказівник на початок області пам'яті, достатньої для збереження n об'єктів розміру size кожний.

**Масиви**

Синтаксис визначення масиву:

**<тип> <ідентифікатор\_масиву> [кількість\_елементів];**

Ідентифікатор масиву – константний вказівник на його нульовий елемент, а \*(iArray + i)– це елемент масиву з індексом i.

Синтаксис визначення двовимірного масиву:

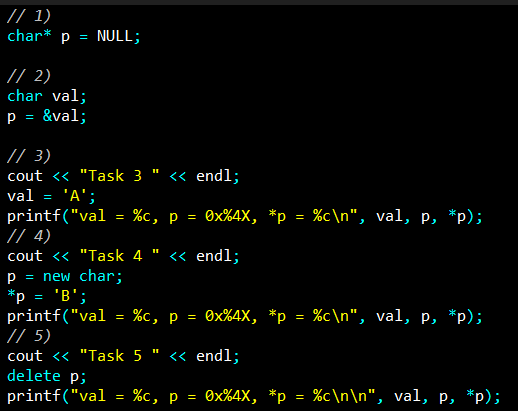
**<тип> <ідентифікатор\_масиву> [кількість\_елементів1] [кількість\_елементів2];**

Створення динамічного масиву: двовимірний масив використовується як масив вказівників на рядки масиву (тобто масив масивів – саме так компілятор інтерпретує багатовимірні масиви).

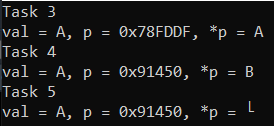
**Завдання аудиторної роботи**

1. Створити вказівник **p** на тип **char**.
2. Присвоїти вказівнику **p** адресу деякої існуючої змінної **val**.
3. Присвоїти змінній **val** деяке значення – вивести на екран: вказівник **p**, змінну **val**, та значення **\*p**.
4. Виділити пам'ять під динамічну змінну, пов’язану з вказівником **p**. Присвоїти цій змінній деяке значення – вивести на екран: вказівник **p**, змінну **val**, та значення **\*p**. Порівняти з попереднім виводом.
5. Звільнити пам'ять та знову вивести на екран: вказівник **p**, змінну **val**, та значення **\*p**. Які висновки можна зробити про вказівник p?

Код:



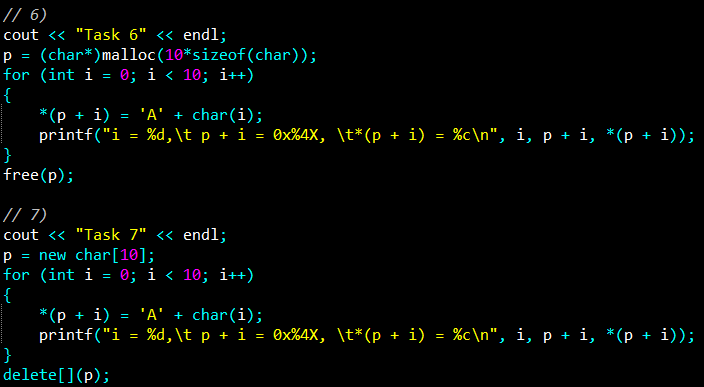
Результат виконання:

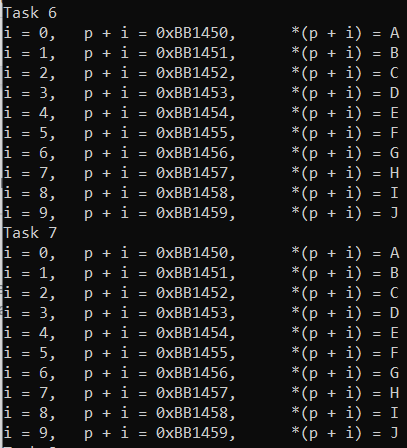


Спочатку був створений вказівник, потім йому присвоєний адрес змінної val. Потім змінній було присвоєно значення, та виведено її значення, значення вказівника, тобто адрес змінної, та значення розадресованого вказівника, тобто значення змінної. У наступному завданні було виділено пам’ять під динамічну змінну, та задано по цьому адресу значення ‘B’. Як бачимо, значення val залишилося тим самим, адже ми змінюємо значення за іншим адресом.

6) Виділити пам'ять під масив типу **char** із 10 елементів, пов’язавши його із вказівником **p** (використати стиль С – **malloc ()** чи **calloc ()**). Присвоїти їм деякі значення. Вивести на екран ці значення елементів масиву та їх адреси. Які висновки можна зробити? Звільнити пам'ять.

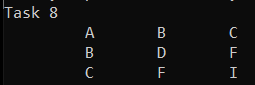
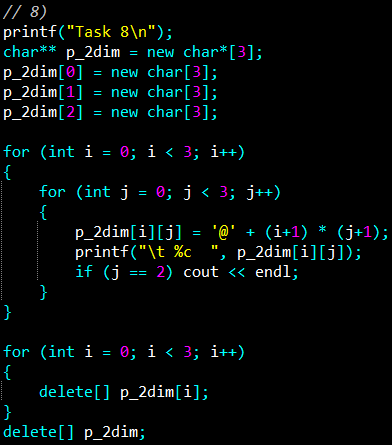
7) Виконати завдання 6 ще раз в стилі С++ (використати операції **new** та **delete**)





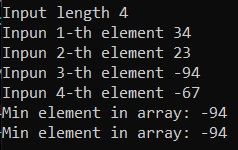
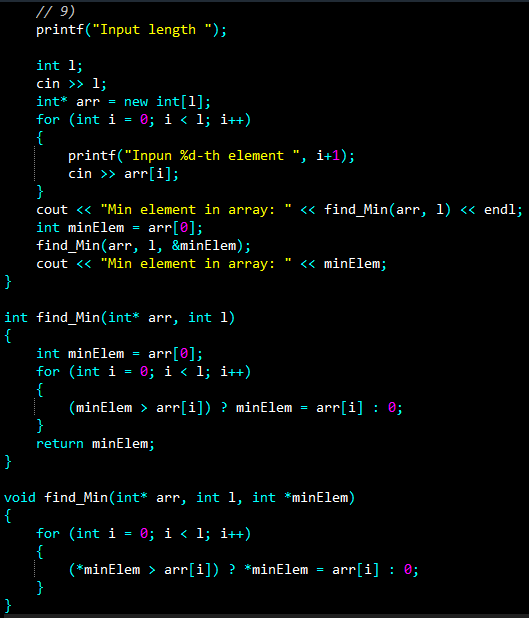
В обох завданнях спочатку було виділено пам’ять під 10 елементів типу char, та за допомогою циклу було занесено на ці місця різні значення та виведено в консоль. Під кінець було звільнено пам’ять, тобто видалено ці масиви.

8) Визначити вказівник **p\_2dim** на двовимірний масив з елементами типу **сhar** із 3\*3 елементів. Виділити пам'ять під цей масив. Як це зробити? Присвоїти довільні значення елементам цього масиву. Як ви звертаєтесь до елементів? Вивести значення елементів масиву на екран у вигляді матриці. Для виділення пам’яті використайте довільний стиль С чи С++. Коректно звільнить пам'ять.



Для створення двовимірного масиву було створено одновимірний масив вказівників, де кожен елемент масиву вказує на одновимірні масиви. За допомогою вкладеного циклу та звертання до члену масиву як p\_2dim[i][j] було надано кожній комірці значення та виведено їх в консоль. Після цього видалено кожний динамічний одновимірний масив усередині масиву p\_2dim, а потім і сам масив вказівників.

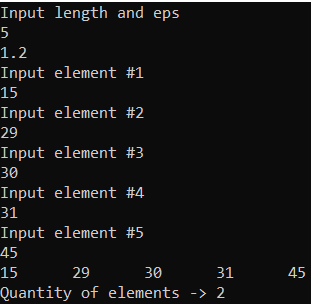
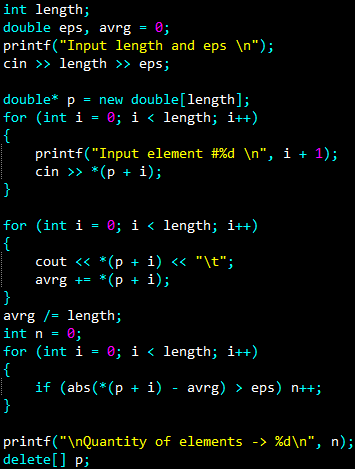
9) Створити метод для обчислення мінімального елементу в одновимірному масиві. У списку параметрів методу – вказівник на масив та кількість елементів в ньому. Зробити 2 варіанти методу: в одному мінімум повертається як результат методу, в іншому результат методу **void**, а мінімум повертається через список параметрів.



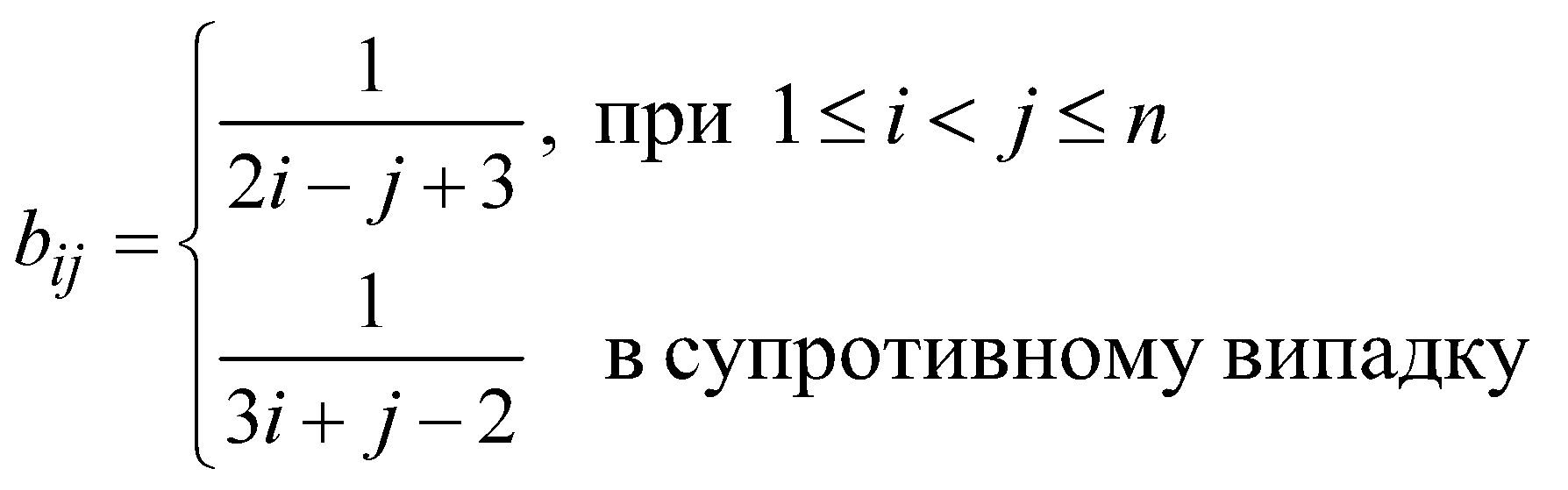
Спочатку було створено масив, у якому буде знайдено мінімальний елемент. Метод find\_Min() з двома параметрами приймає вказівник на перший елемент масиву та кількість його елементів. У циклі за найменше значення прийнято перший елемент масиву, а потім йде перебір кожного члена масиву, якщо він менше за попередній, то змінній minElem присвоюється його значення. Метод з трьома параметрами відрізняється тим, що йому третім параметром передано вказівник на змінну minElem, за цим адресом змінюється сама змінна в методі main, а не її копія.

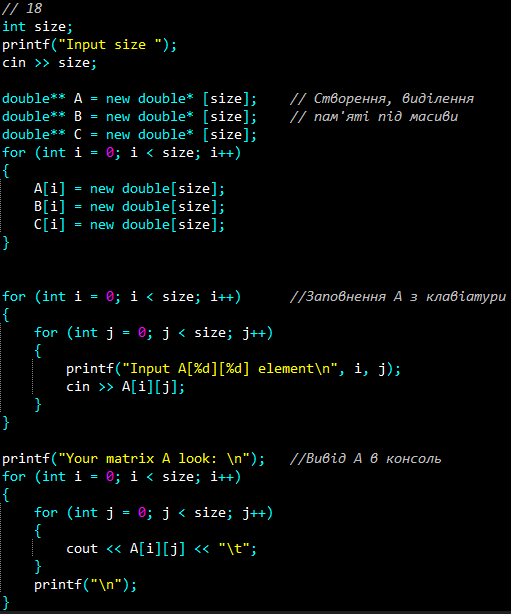
**Завдання домашньої роботи**

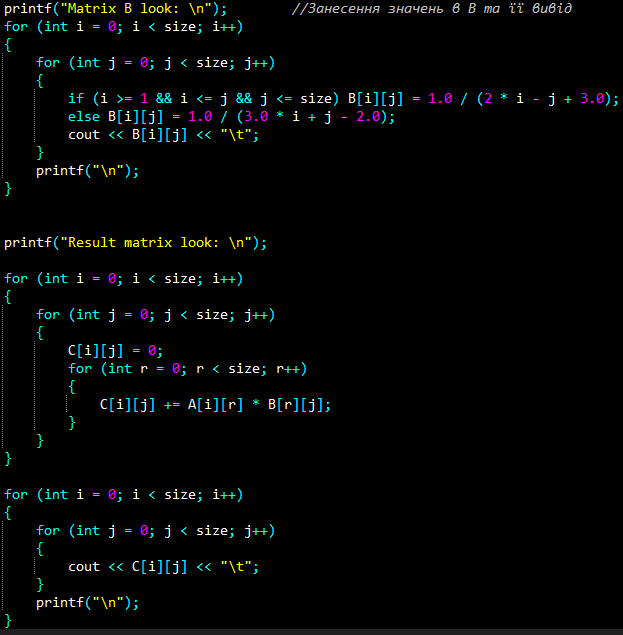
4. Обчислити кількість елементів масиву, які відрізняються від середнього арифметичного більше ніж на деяку задану величину eps.

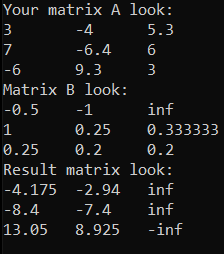
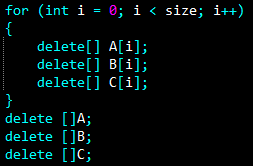


Зчитано значення довжини масиву, величини eps, створено динамічний масив та занесено в його значення. У наступного циклі було виведено масив в консоль та обчислено суму елементів, яку використано, щоб знайти середнє арифметичне. У наступному циклі виконується перебір елементів, те перевірка, чи є різниця кожного елемента більша за величину eps.

Для заданої квадратної матриці А заданого порядку n визначити добуток матриць A\*B, де елементи матриці B визначаються за формулою:





(Після введення А)

Спочатку було створено 3 двовимірних масиви(матриці), у які було занесено значення(в А з консолі, у В за формулою). Також було виведено масиви в консоль. Потім було обчислено добуток А\*В за правилами множення матриць за допомогою вкладених циклів. Перед тим, як заносити в комірку матриці C значення, йому присвоєно нуль. Після цього цю матрицю виведено та очищено місце з-під всіх масивів.

**Висновок**

Я навчився використовувати вказівники, передавати їх в функцію, щоб вона могла змінювати значення змінної в методі main. Також навчився створювати динамічні масиви, одновимірні та двовимірні, виконувати дії над їх елементами. Було написано програми: яка обчислює добуток двох матриць, яка шукає мінімальний елемент одновимірного масиву.