 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №11**

з дисципліни «Алгоритмізація та програмування»

на тему: «Двовимірні масиви»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**Завдання:**

1. Ознайомитися з теоретичними відомостями створення, обробки та виводу на екран елементів двовимірних масивів.
2. Побудувати блок-схему алгоритму для вирішення задачі відповідно до свого варіанту.
3. Відповідно до свого варіанту розробити програмний застосунок, що і включає реалізацію функції користувача для обробки двовимірного статичного масиву з використанням звертань до елементів через операцію індексації «[ ]» (глобальні змінні не використовувати, функція main має бути призначена тільки для виклику функцій користувача, обов’язковими є реалізація функцій користувача для введення, обробки та виведення на екран елементів масиву):

***В матриці з дійсних чисел знайти добуток елементів в кожному стовпці. Результат записати в одновимірний масив. Вивести початкову матрицю та отриманий одновимірний масив на екран.***

1. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема:**





**Код:**

#include <iostream>

#include <conio.h>

using namespace std;

#define width 3

#define hight 3

/\*

функция для корректного ввода елементов массива

параметр - массив

ничего не возвращает, изменяет массив

\*/

void input(int(\* a)[hight])

{

for (int i = 0; i < hight; i++)

for (int j = 0; j < width; j++)

{

cout << "Enter A[" << i << "][" << j << "] : ";

while (!(cin >> a[i][j]))

{

cout << "Try correct number!\n";

cin.clear();

cin.ignore();

fflush(stdin);

}

}

}

/\*

функция для вывода массива

параметр - массив

ничего не возвращает, выводит массив

\*/

void output(int(\*a)[hight])

{

cout << "\nYour array is :\n\n";

for (int i = 0; i < hight; i++)

{

for (int j = 0; j < width; j++)

cout << a[i][j] << " ";

cout << "\n\n";

}

}

/\*

функция для суммирования столбцов двухмерного массива

параметр - двухмерный массив, массив для записи значений сумм

ничего не возвращает, изменяет нулевой массив записывая суммы столбцов

\*/

void mult\_of\_columns(int(\*a)[hight], int\* b)

{

for (int i = 0; i < hight; i++)

for (int j = 0; j < width; j++)

b[j] += a[i][j];

}

/\*

функция для инициализации массива, ввода, вывода и вывода сумм столбцов двухмерного массива

нет параметров

ничего не возвращает, выводит полученый массив с суммами

\*/

void my\_array()

{

int arr[hight][width];

input(arr);

int final\_arr[width] = {};

mult\_of\_columns(arr, final\_arr);

output(arr);

cout << "Sums of columns are :\n\n";

for (int i = 0; i < width; i++)

cout << final\_arr[i] << " ";

}

void main()

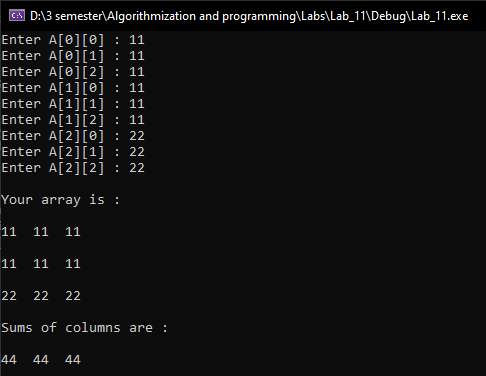
{

my\_array();

\_getch();

}

**Результати:**

****

**Контрольні питання:**

1. ***Яким чином задаються двовимірні масиви?***

Багатовимірний масив в мові C++ організований за принципом «масиву масивів». Загальний формат його оголошення наступний:

type array\_name [N1][N2]...[NM];

тут M - число індексів (або розмірність) масиву. Індекси змінюються в межах від 0 до N1-1, від 0 до N2-1, від 0 до N3-1, ... , від 0 до NM-1 відповідно.

1. ***В якій послідовності виконується заповнення елементів багатовимірних масивів в мові програмування С++?***

Елементи заповнюються по рядках (від нульового до кінцевого). У кожного рядка заповнюється елемент теж від нульового до кінцевого.

Приклад:

1 2 3

4 5 6

7 8 9

1. ***Скільки необхідно операторів циклів для виведення на екран значень елементів двовимірного масиву?***

Необхідно 2 оператора циклу.

1. ***Навести приклади можливих способів ініціалізації двовимірного масиву.***

int a[2][2] = {{1,2}{3,4}}

int b[2][2]{{1,2}{3,4}}

int c[][] = {{1,2}{3,4}}

int d[][]{{1,2}{3,4}}

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

cin >> e[i][j];

1. ***Яка умова визначення елементів на головній(другорядній) діагоналі двовимірного масиву? Навести приклади.***

Головна діагональ – номери індексів співпадають. e.g. arr[i][i]

Другорядна діагональ - індекс стовпця – це максимальний індексу стовпця мінус індекс рядка. e.g. arr[i][n-i-1]

1. ***Яка умова визначення всіх елементів k-го стовпчика (рядка) двовимірного масиву? Навести приклад.***

for (int i = 0; i < N; i++)

cout << arr[i][k]; - індекс стовпчика – константа.

1. ***Чи можна, використовуючи покажчик, а не покажчик на покажчик, обробити всі елементи двовимірного масиву?***

Можна, якщо пам’ять виділяється під кожний рядок окремо. Тоді для кожного рядка потрібно записувати в покажчик адресу його першого елементу. У випадку коли пам’ять виділена одним блоком - покажчику достатньо передати адресу першого елементу.

1. ***Як можна під час виконання програми змінити розмір двовимірного масиву (кількість елементів в ньому)? Навести відповідний приклад.***

Необхідно створити новий масив потрібного розміру, переписати елементи старого масиву в новий і видалити старий масив.

int size = 10;

int\* arr = new int[size];

void resize()

{

size\_t newSize = size \* 2;

int\* newArr = new int[newSize];

memcpy( newArr, arr, size \* sizeof(int) );

size = newSize;

delete [] arr;

arr = newArr;

}

1. ***Перелічити відмінності в роботі із статичними та динамічними масивами.***

*Статичний масив:*

* розмір масиву задається під час його оголошення.
* не можна змінити розмір
* оперативна пам’ять під масив виділяється до початку виконання програми в області, яка називається стек.
* ім’я масиву - покажчик-константа.

*Динамічний масив:*

* розмір масива оголошується в процесі виконання програми.
* можемо змінити розмір
* оперативна пам’ять під масив запрошується і виділяється динамічно з області, яка називається – куча.
* ім’я масиву - покажчик-змінна.

1. ***В чому полягають переваги та недоліки використання масивів?***

Перевагами:

* доступ за константний час до будь-якого елементу масиву
* пам'ять витрачається тільки на дані (не враховуючи константні затрати пам’яті ОС, що виникають при реалізації структури).

Недоліки:

* статичність незмінної структури
* додавання та видалення елементів в масивах – затратна на пам’ять операція.