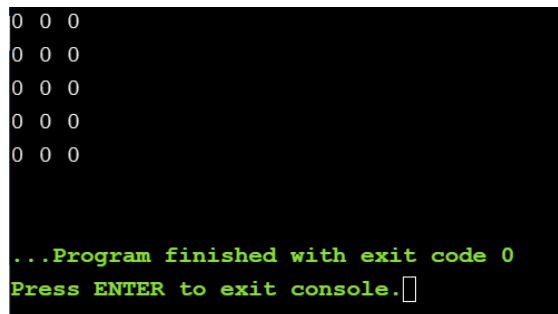


Матриці в STL. Робота з діагоналями матриці.

Матриця – це двовимірний вектор, який можна реалізувати наступним чином:

```
//створення матриці розміром 5x3, заповненої 0
int row = 5; //кількість рядків
int col = 3; //кількість стовпців
vector <vector<int> > matrix (row, vector<int>(col, 0));
for (int i = 0; i < row; i++) {
    for (int j = 0; j < col; j++){
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}
```

Результат роботи програми:



```
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0

...Program finished with exit code 0
Press ENTER to exit console.
```

Існує інший спосіб – створити порожній вектор, а потім у циклі розширити його на відповідну кількість рядків та стовпців:

```
vector <vector<int> > matrix; // порожня матриця нульового розміру
int row;
cout<<"Rows: ";
cin >> row;
matrix.resize (row);          // змінюємо розмір відповідно до кількості рядків

for (int i = 0; i < row; i++) {
    cout<<"Columns: ";
    cin >> col;
    // якщо кількість стовпців задавати в циклі, то рядки можуть бути різного розміру
    matrix[i].resize(col); // змінюємо розмір i-го рядка відповідно до кількості стовпців
    cout<<"Elements: ";
    for (int j = 0; j < col; j++){
        cin >> matrix[i][j];
    }
}
```

```

}
for (int i = 0; i < matrix.size(); i++) {
    for (int j = 0; j < matrix[i].size(); j++){
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}

```

Результат роботи програми:

```

Rows: 2
Columns: 3
Elements: 1 1 1
Columns: 4
Elements: 5 5 5 5
1 1 1
5 5 5 5

...Program finished with exit code 0
Press ENTER to exit console.

```

Діагоналі квадратної матриці

Квадратна матриця — це матриця з однаковою кількістю рядків і стовпців.

Головна діагональ матриці — це уявна лінія, на якій розташовані елементи матриці, яка проходить від лівого верхнього кута до правого нижнього кута матриці (рисунок 1).

Діагональ квадратної матриці, яка проходить від правого верхнього кута до нижнього лівого кута, називається *побічною* (рисунок 1).

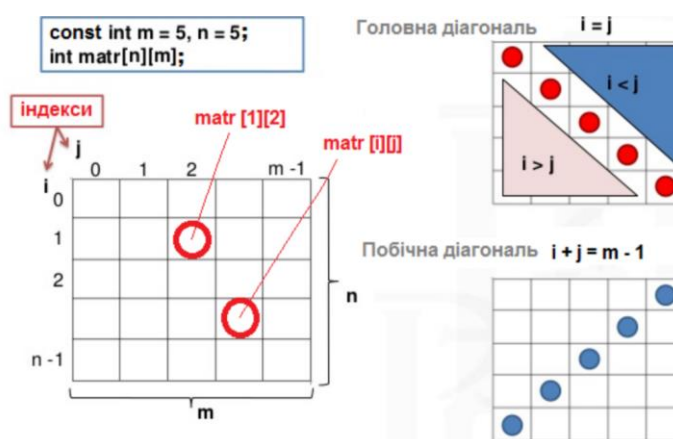


Рисунок 1. Головна та побічні діагоналі матриці

Якщо необхідно обробляти елементи матриці, які розташовані між головною та побічною діагоналлю, можна застосовувати логічні співвідношення для індексів елементів, що зазначені на рисунку 2.

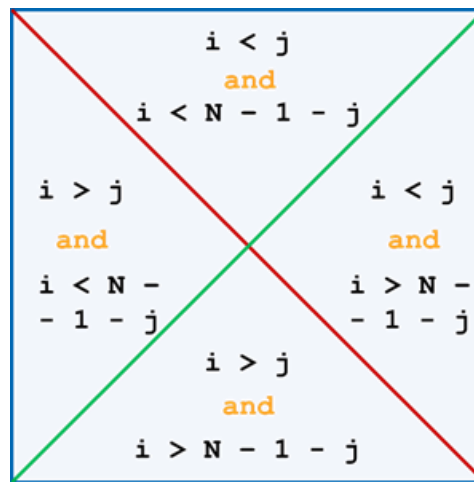


Рисунок 2. Логічні співвідношення для індексів елементів, що розташовані між головною та побічною діагоналлю матриці.

Операції над рядками матрицями:

- додавання рядка в кінець матриці методом `push_back()`:

```
matr.push_back (vector<int>(col)); // de col – кількість стовпчиків в рядку
```

- видалення останнього рядка матриці методом `pop_back()`:

```
matr.pop_back();
```

- додавання рядка перед рядком із заданим номером `K` методом `insert()`:

```
matr.insert (matr.begin() + K , vector<int>(col));
```

- видалення рядка матриці з номером `K` методом `erase()`:

```
matr.erase(matr.begin() + K);
```

- обмін рядків/стовпців методом `vector<> swap()`:

```
matr[k].swap(matr[t]); //обмін місцями рядків k та t.
```

Операції над стовпцями матрицями:

- додавання останнього стовпця матриці методом `push_back()`:

```
for (int i=0; i < matr.size(); i++)
```

```
    matr[i].push_back(0); // додати останній нульовий стовпчик
```

- видалення останнього стовпця матриці методом `pop_back()`:

```
for (int i=0; i < matr.size(); i++)
```

```
    matr[i].pop_back();
```

- вставка стовпця в матрицю перед стовпцем з номером `K` методом `insert`:

```
for (int i=0; i < matr.size(); i++)
```

```
    matr[i].insert ( matr[i].begin() + K , 0 );
```

- видалення стовпця матриці з номером `K` методом `erase()`:

```
for (int i=0; i < matr.size(); i++)
```

```
    matr[i].erase ( matr[i].begin() + K );
```

- обмін стовпців з індексами `K` та `T` функцією `swap()`:

```
for (int i=0; i < matr.size(); i++)  
    swap( matr[i][K], matr[i][T] );
```

Приклади задач роботи з матрицями в STL

Задача 1

Задано цілочисельну квадратну матрицю A (кількість рядків/стовпців вводить користувач, елементи задаються випадковим чином). Підрахувати суми елементів, що стоять на головній і побічній діагоналях. Вивести результат на екран.

Математична постановка задачі

Вхідні дані

n – кількість рядків та стовпців матриці A, ціле ($n > 0$).

A – цілочисельна квадратна матриця розміром $n \times n$;

Вихідні дані

sumMainDiagonal – сума елементів головної діагоналі, ціле.

sumSideDiagonal – сума елементів побічної діагоналі, ціле.

Математична модель задачі

Визначимо:

sumMainDiagonal = 0;

sumSideDiagonal = 0.

Для $i=0, n$:

sumMainDiagonal = sumMainDiagonal + A[i][i];

sumSideDiagonal = sumSideDiagonal + A[i][n-1-i];

Перевірка працездатності програми

1) Для перевірки працездатності задамо $n = 3$ (непарне n):

4	7	8
6	4	6
7	3	10

Сума елементів головної діагоналі: $4+4+10=18$.

Сума елементів побічної діагоналі: $8+4+7=19$.

```

n: 3
4 7 8
6 4 6
7 3 10
Main Diagonal: 18
Side Diagonal: 19

...Program finished with exit code 0
Press ENTER to exit console.

```

2) Для перевірки працездатності задамо $n = 2$ (парне n):

4	7
8	6

Сума елементів головної діагоналі: $4+6=10$.

Сума елементів побічної діагоналі: $8+7=15$.

```

n: 2
4 7
8 6
Main Diagonal: 10
Side Diagonal: 15

...Program finished with exit code 0
Press ENTER to exit console.

```

Висновок: працездатність програми підтверджено тестовими розрахунками.

Текст програми

```

#include <iostream>
#include <vector>
#include <random>
using namespace std;

int main() {
    vector <vector<int> > A;
    int n;
    cout<<"n: "; cin >> n;
    if (n > 0){
        A.resize(n);

        for (int i = 0; i < n; i++) {
            A[i].resize(n);
            for(int j = 0; j < n; j++){
                A[i][j] = rand() % 10 + 1;
            }
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++){
                cout << A[i][j] << " ";
            }
        }
    }
}

```

```

    }
    cout << endl;
}

int sumMainDiagonal = 0;
int sumSideDiagonal = 0;
for (int i = 0; i < n; i++){
    sumMainDiagonal += A[i][i];
    sumSideDiagonal += A[i][n-1-i];
}
cout << "Main Diagonal: " << sumMainDiagonal <<
        "\nSide Diagonal: " << sumSideDiagonal;
} else cout << "n is incorrect!";

return 0;
}

```

Задача 2

Задано цілочисельну квадратну матрицю A (кількість рядків/стовпців та елементи задаються випадковим чином). Видалити з середини матриці рядок та стовпчик. Поміняти місцями перший та останній рядки та стовпці матриці. Вивести результат на екран.

Математична постановка задачі

Вхідні дані

n – кількість рядків та стовпців матриці A , ціле ($n > 0$).

A – цілочисельна квадратна матриця розміром $n \times n$;

Вихідні дані

A – цілочисельна квадратна матриця після модифікації;

Математична модель задачі

Позначимо:

$s = \text{size} / 2$ – індекс середнього рядка та стовпця матриці, де size – поточний розмір матриці.

Видалити рядок $A[s]$.

Для $i = 0, \text{size}-1$: видалити $A[i][s]$.

Обміняти рядки $A[0]$ та $A[\text{size}-1]$.

Для $i = 0, \text{size}-1$: обміняти елементи $A[i][0]$ та $A[i][\text{size}-1]$.

Перевірка працездатності програми

1) Для значення $n=5$ (непарне n), $s = 5 / 2 = 2$.

7	11	7	3	2
5	1	7	4	2
9	8	6	4	8

5	10	11	3	1
11	9	6	1	5

Маємо отримати:

7	11	3	2
5	1	4	2
5	10	3	1
11	9	1	5

Після перестановки першого та останнього рядків/стовпців маємо отримати:

5	9	1	11
2	1	4	5
1	10	3	5
2	11	3	7

```

7 11 7 3 2
5 1 7 4 2
9 8 6 4 8
5 10 11 3 1
11 9 6 1 5

7 11 3 2
5 1 4 2
5 10 3 1
11 9 1 5

5 9 1 11
2 1 4 5
1 10 3 5
2 11 3 7

...Program finished with exit code 0
Press ENTER to exit console.

```

УВАГА! Перевірку для парної кількості рядків/стовпців матриці та висновки щодо працездатності роботи програми зробити самостійно.

Текст програми

```

#include <iostream>
#include <vector>
#include <random>
using namespace std;

void print_matrix(vector <vector<int> > &M){
    for (int i = 0; i < M.size(); i++) {

```

```

        for (int j = 0; j < M[i].size(); j++){
            cout << M[i][j] << " ";
        }
        cout << endl;
    }
    cout << "_____ " << endl;
}

int main() {
    vector <vector<int> > A;
    int n = rand() % 10 + 1;

    A.resize(n);

    for (int i = 0; i < n; i++) {
        A[i].resize(n);
        for(int j = 0; j < n; j++){
            A[i][j] = rand() % 11 + 1;
        }
    }

    print_matrix(A);

    int s = A.size() / 2;

    A.erase(A.begin() + s);

    for (int i = 0; i < A.size(); i++)
        A[i].erase(A[i].begin() + s);

    print_matrix(A);

    A[0].swap(A[A.size()-1]);

    for (int i=0; i < A.size(); i++)
        swap( A[i][0], A[i][A.size()-1] );

    print_matrix(A);

    return 0;
}

```

Лабораторна робота №10(5)

Тема роботи: «Багатовимірні масиви даних бібліотеки STL»

Завдання: 1) розробити для кожної задачі МПЗ і СА відповідно до варіанту завдання; 2) скласти програми мовою C++; 3) розробити тестові приклади та провести аналіз працездатності програм; 4) провести трасування однієї задачі; 5) оформити звіт (зразок оформлення див. у прикладах задач 1-2, що наведені вище).

При виконанні завдання працюємо з класом vector<> бібліотеки STL.

Увага! Якщо студент здав працездатну програму викладачу під час проведення

першого заняття, в звіті до цієї задачі наводиться тільки варіант завдання, текст програми та тестові приклади.

Варіанти завдання:

Студент обирає свій варіант V завдання за формулою :

$$\text{int } V = ((C + \text{SecretCode}) \% 5) ? (C + \text{SecretCode}) \% 5 : 5;$$

де C – номер в списку групи, SecretCode – кодове число, яке оголошується викладачем на початку лабораторного заняття.

Варіант 1

1. Задана матриця розміром KxP (K та P вводить користувач). Видалити з матриці: 1) рядок, що містить максимальний елемент матриці; 2) стовбець, який містить значення, що було найближчим до максимального.
2. Задана матриця розміром KxP (K та P вводить користувач). 1) Після рядка, що містить всі додатні значення вставити рядок з нулями. 2) Перед стовпцем, елементи якого утворюють спадну послідовність вставити стовпець з одиницями.
3. Задана матриця розміром KxP (K та P вводить користувач). 1) Поміняти місцями рядок, що містить всі парні елементи з рядком, що містить всі непарні елементи (вважати, що такі рядки існують по одному). 2) Поміняти місцями стовпці, що містять мінімальний та максимальний елементи.
4. Задана квадрата матриця (розмір матриці може бути довільний). Вивести на екран елементи головної та побічної діагоналей. Обнулити елементи вище головної та побічної діагоналі (позначено на схемі).

5. Задана прямокутна матриця розміром KxP (P >= K). 1) Знайти суми елементів головної діагоналі та для кожної діагоналей, що їй паралельні. 2) Знайти добутки елементів побічної діагоналі та для кожної діагоналей, що їй паралельні.

Варіант 2

1. Задана матриця розміром KxP (K та P вводить користувач). Видалити з матриці: 1) рядок, що має найменшу суму елементів; 2) стовбець, який містить мінімальний елемент.
2. Задана матриця розміром KxP (K та P вводить користувач). 1) Перед рядком, елементи якого утворюють послідовність, що зростає, вставити рядок з нулями. 2) Після стовпця, що містить перший від'ємний елемент вставити стовпець з одиницями.
3. Задана матриця розміром KxP (K та P вводить користувач). 1) Поміняти

місцями рядок, що містить всі від'ємні елементи з рядком, що містить всі додатні елементи (вважати, що такі рядки існують по одному). 2) Поміняти місцями стовпці, що містять однакову кількість парних та непарних елементів (вважати, що такі стовпці існують по одному).

4. Задана квадранта матриця (розмір матриці може бути довільний). Вивести на екран елементи головної та побічної діагоналей. Обнулити елементи нижче побічної та вище головної діагоналі (позначено на схемі).

5. Задана прямокутна матриця розміром $K \times P$ ($P \geq K$). 1) Знайти добутки елементів головної діагоналі та для всіх діагоналей, що їй паралельні. 2) Знайти кількості парних елементів побічної діагоналі та для кожної діагоналі, що їй паралельні.

Варіант 3

1. Задана матриця розміром $K \times P$ (K та P вводить користувач). Видалити з матриці: 1) рядок, в якому знаходиться мінімальний елемент; 2) стовбець, який містить елемент найближчий до мінімального.

2. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Після рядка, елементи якого всі непарні вставити рядок з одиницями. 2) Перед стовпцем, що містить найбільшу кількість парних значень вставити стовпець з нулями.

3. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Поміняти місцями рядок, що містить максимальний елемент матриці, з рядком, що містить мінімальний елемент матриці. 2) Поміняти місцями перший та останній стовпці, в яких всі елементи однакові.

4. Задана квадранта матриця (розмір матриці може бути довільний). Вивести на екран елементи головної та побічної діагоналей. Обнулити елементи нижче головної та побічної діагоналі (позначено на схемі).

5. Задана прямокутна матриця розміром $K \times P$ ($P \geq K$). 1) Знайти максимальні елементи для головної діагоналі та для всіх діагоналей, що їй паралельні. 2) Знайти суми парних елементів побічної діагоналі та для кожної діагоналі, що їй паралельні.

Варіант 4

1. Задана матриця розміром $K \times P$ (K та P вводить користувач). Видалити з матриці: 1) рядок, в якому знаходиться мінімальна кількість парних елементів; 2) стовбець, який містить максимальний елемент матриці.
2. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Перед рядком, елементи якого утворюють спадну послідовність вставити рядок з нулями. 2) Після стовпця, що містить всі від'ємні значень вставити стовпець з нулями.
3. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Поміняти місцями рядок, що містить перше від'ємне число в матриці, з рядком, що містить останнє від'ємне число в матриці. 2) Поміняти місцями перший та останній стовпці, в яких суми елементів однакові.
4. Задана квадранта матриця (розмір матриці може бути довільний). Вивести на екран елементи головної та побічної діагоналей. Обнулити елементи нижче головної та вище побічної діагоналі (позначено на схемі).

5. Задана прямокутна матриця розміром $K \times P$ ($P \geq K$). 1) Знайти добутки додатних елементів для головної діагоналі та для всіх діагоналей, що їй паралельні. 2) Знайти кількості значень, що кратні X , для елементів побічної діагоналі та для кожної діагоналі, що їй паралельні.

Варіант 5

1. Задана матриця розміром $K \times P$ (K та P вводить користувач). Видалити з матриці: 1) рядок, в якому сума елементів максимальна; 2) останній стовбець, в якому всі елементи від'ємні.
2. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Перед рядком, елементи якого парні вставити рядок з двійками. 2) Після стовпця, що містить мінімальний елемент матриці вставити стовпець з нулями.
3. Задана матриця розміром $K \times P$ (K та P вводить користувач). 1) Поміняти місцями перший та останній рядки, елементи яких утворюють арифметичну прогресію. 2) Поміняти місцями перший стовпець, що містить більше парних значень, ніж непарних, з першим стовпцем, що містить менше парних значень, ніж непарних.
4. Задана квадранта матриця (розмір матриці може бути довільний). Вивести на екран елементи головної та побічної діагоналей. Обнулити елементи між головною та побічною діагоналлю (позначено на схемі).

5. Задана прямокутна матриця розміром $K \times P$ ($P \geq K$). 1) Знайти кількості додатних елементів для головної діагоналі та для всіх діагоналей, що їй паралельні. 2) Знайти суми непарних значень для елементів побічної діагоналі та для кожної діагоналі, що їй паралельні.

Контрольні запитання

1. Як реалізувати матрицю з використанням контейнеру вектор STL? Наведіть приклад опису. Як встановити розмір матриці для заданої кількості рядків та стовпців?
2. Як встановити розмір матриці для змінної кількості стовпців в кожному рядку?
3. Дайте визначення головної та побічної діагоналям матриці?
4. За якої умови елемент матриці буде розташований на головній діагоналі? На побічній діагоналі?
5. Які операції можна здійснити над рядками та стовпцями матриці?
6. Як вставити останній рядок/стовпець матриці? Як здійснити цю операцію, якщо працюєш зі звичайною матрицею `int A[n][m]`?
7. Як видалити останній рядок/стовпець матриці? Як здійснити цю операцію, якщо працюєш зі звичайною матрицею `int A[n][m]`?
8. Як вставити рядок/стовпець перед рядком/стовпцем з заданим номером? Як здійснити цю операцію, якщо працюєш зі звичайною матрицею `int A[n][m]`?
9. Як видалити рядок/стовпець з заданим номером? Як здійснити цю операцію, якщо працюєш зі звичайною матрицею `int A[n][m]`?
10. Як обміняти місцями два рядки/стовпці матриці з заданими номерами? Як здійснити цю операцію, якщо працюєш зі звичайною матрицею `int A[n][m]`?

Тестові завдання

№	Завдання
1.	<p>Задані масиви <code>X[n][m]</code> і <code>Y[n][m]</code>. Які дії виконує наступний алгоритм:</p> <pre>for (int i=0; i<n; i++) for (int j=0; j<m; j++) X[i][j] = X[i][j] + Y[i][j];</pre> <p>А. сума елементів головної діагоналі матриці; Б. сума елементів одновимірних масивів <code>X</code> і <code>Y</code>; В. кількість рівних відповідних елементів масиву <code>X</code> і масиву <code>Y</code>; Г. сума матриць <code>X</code> і <code>Y</code>.</p>
2.	<p>Що буде виведене на екран після виконання програми: <code>#include <iostream></code> <code>int main()</code></p>

	<pre> { int sum = 0; int array[][3] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}}; for (int i = 0; i < 3 ; ++i) { for (int j = 2; j < 3 ; j++) { sum += array[i][j]; } } std::cout << sum << std::endl; return 0; } </pre> <p>А – 9 Б – 15 В – 21 Г – повідомлення про помилку</p>
3.	<p>Задані масиви $X[n][m]$ і $Y[n][m]$. Які дії виконує наступний алгоритм:</p> <pre> for (int i=0; i<n; i++) for (int j=0; j<m; j++) if ((X[i][j] + Y [i][j]) % 2 == 0) T++; </pre> <p>А. сума елементів головної діагоналі матриці, що діляться на 2; Б. суму рівних відповідних елементів масиву X і масиву Y; В. кількість відповідних елементів матриць X і Y, що мають однакову парність; Г. кількість елементів X і Y, що діляться на 2.</p>
4.	<p>Задані масиви $X[n][m]$ і $Y[n][m]$. Які дії виконує наступний алгоритм:</p> <pre> for (int i=0; i < n; i++) for (int j=0; j < m; j++) if ((X[i][j]%2==0 Y[i][j])%2==0) T++; </pre> <p>А. сума елементів головної діагоналі матриці, що діляться на 2; Б. суму рівних відповідних елементів масиву X і масиву Y; В. кількість відповідних елементів матриць X і Y, що мають однакову парність; Г. кількість елементів матриць X і Y, що діляться на 2.</p>

Завдання з відповіддю у відкритій формі

Можливі варіанти відповідей:

- ✓ видалення рядка t матриці;
- ✓ видалення стовпця t матриці;
- ✓ добуток матриці та вектору (одновимірного масиву);
- ✓ добуток двох матриць;
- ✓ перестановка рядків матриці;
- ✓ перестановка стовпців матриці;
- ✓ транспонована матриця;
- ✓ матриця залишиться без змін.

№	Завдання
----------	-----------------

1.	<p>Як зміниться матриця після виконання наступного коду (запишіть відповідь)?</p> <pre>int a[10][10], t; for (int i=0; i<10; i++) for (int j=0; j<10; j++) if (i == j) continue; else { t=a[i][j]; a[i][j]=a[j][i]; a[j][i]=t; }</pre>
2.	<p>Що буде отримано в результаті роботи програми (напишіть відповідь):</p> <pre>int a[n][n], t; for (int i=0; i<=n/2; i++) for (int j=0; j<n; j++){ t=a[i][j]; a[i][j]=a[n-i-1][j]; a[n-i-1][j]=t; }</pre>
3.	<p>Що буде отримано в результаті роботи програми:</p> <pre>int a[10][10], t; for (int i=0; i<10; i++) for (int j=i; j<10; j++) if (i == j) continue; else { t=a[i][j]; a[i][j]=a[j][i]; a[j][i]=t; }</pre>
4.	<p>Яких змін зазнає матриця після виконання наступного коду (напишіть відповідь):</p> <pre>int a[n][n], t; for (int i=0; i<n; i++) for (int j=0; j<=n/2; j++){ t=a[i][j]; a[i][j]=a[i][n-j-1]; a[i][n-j-1]=t; }</pre>
5.	<p>Який результат виконання наступного коду (напишіть відповідь):</p> <pre>for(int i=0; i<n; i++){ for(int l=0; l<n; l++){ c[i][l] = 0; for(int j=0; j<n; j++){ c[i][l] += a[i][j]*b[j][l]; } } }</pre>
6.	<p>Що буде отримано в результаті роботи програми:</p> <pre>int a[n][n]; for (int i=0; i<n; i++) for (int j=t; j<n-1; j++) a[i][j]=a[i][j+1]; for (int i=0; i<n; i++) a[i][n-1]=0;</pre>

7.	Що буде отримано в результаті роботи програми: <pre>for(int j=0;j<n;j++){ d[j]=0; for(int i=0;i<m;i++) d[j]+=A[j][i]*c[i]; }</pre>
8.	Що буде отримано в результаті роботи програми: <pre>int a[n][n]; for (int i=t; i<n-1; i++) for (int j=0; j<n; j++) a[i][j]=a[i+1][j]; for (int j=0; j<n; j++) a[n-1][j]=0;</pre>