

C++ Programming

Multidimensional Arrays

Practice 1

Mostafa S. Ibrahim

Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada

Bachelor / Msc from Cairo University - Egypt

Ex-(Software Engineer / ICPC World Finalist)



Practice: Max value

- Read 2 integers for the rows and columns of a matrix (≤ 100). Then read rows x cols integer value. Find the position of maximum value in the array. If there are several ones, find the last occurrence
- Input:
 - 3 4
 - 1 5 1 10
 - 2 10 3 4
 - 1 10 **10** 7
- Output
 - Max value at position 2 2 with value = 10

Practice: Max value

```
4 int main() {
5     int arr[100][100];
6
7     int rows, cols;
8     cin >> rows >> cols;
9
10    for (int row = 0; row < rows; ++row)
11        for (int col = 0; col < cols; ++col)
12            cin >> arr[row][col];
13
14    int max_i = 0, max_j = 0;
15
16    for (int i = 0; i < rows; ++i) {
17        for (int j = 0; j < cols; ++j) {
18            if (arr[i][j] >= arr[max_i][max_j])
19                max_i = i, max_j = j;
20        }
21    }
22    cout << "Max value at position " << max_i << " " << max_j
23         << " with value = " << arr[max_i][max_j];
24    return 0;
25 }
```

- Using `>=` finds last occurrence

Practice: Special print

- Read 2 integers for the rows and columns of a matrix (≤ 100). Then read rows x cols integer value.
- Print the following 4 values
 - The sum of the left diagonal & The sum of the right diagonal
 - The sum of the last row & The sum of the last column
- Input: 3 4
 - 8 16 9 52
 - 3 15 27 6
 - 14 25 2 10
- Output
 - 25 104
 - 51 68

8	16	9	52
3	15	27	6
14	25	2	10

Practice: Special print

```
5  int arr[100][100];
6
7  int rows, cols;
8  cin >> rows >> cols;
9
10 for (int i = 0; i < rows; ++i)
11     for (int j = 0; j < cols; ++j)
12         cin >> arr[i][j];
13
14 int i = 0, j = 0;
15
16 int left_diagonal = 0;
17 while (i < rows && j < cols)
18     left_diagonal += arr[i++][j++];
19
20 int right_diagonal = 0;
21 i = 0, j = cols-1;
22 while (i < rows && j >= 0)
23     right_diagonal += arr[i++][j--];
```

```
24
25 int last_row = 0;
26 j = 0;
27 while (j < cols)
28     last_row += arr[rows-1][j++];
29
30 int last_col = 0;
31 i = 0;
32 while (i < rows)
33     last_col += arr[i++][cols-1];
34
35 cout << left_diagonal << " " << right_diagonal << "\n";
36 cout << last_row << " " << last_col << "\n";
37
```

Practice: Swap 2 columns

- Read integers N, M, then Read **matrix** NxM. Then read 2 indices of columns. Swap the 2 columns together. Print the new matrix.
- Input: 3 4
 - 8 16 9 52
 - 3 15 27 6
 - 14 25 2 10
 - **0 3**
- Output
 - 52 16 9 8
 - 6 15 27 3
 - 10 25 2 14

Practice: Swap 2 columns

```
4 int main() {  
5     int arr[100][100];  
6  
7     int rows, cols;  
8     cin >> rows >> cols;  
9  
10    for (int i = 0; i < rows; ++i)  
11        for (int j = 0; j < cols; ++j)  
12            cin >> arr[i][j];  
13  
14    int c1, c2;  
15    cin >> c1 >> c2;  
16  
17    for (int i = 0; i < rows; ++i) {  
18        // swap [i][c1] with [i][c2]  
19        int tmp = arr[i][c1];  
20        arr[i][c1] = arr[i][c2];  
21        arr[i][c2] = tmp;  
22    }  
23    for (int i = 0; i < rows; ++i) {  
24        for (int j = 0; j < cols; ++j)  
25            cout << arr[i][j] << " ";  
26        cout << "\n";  
27    }  
28  
29    return 0;  
30 }  
31
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”