

## Q1. The King's Feast

The King has  $n$  plates of food, each with a certain quantity. He wants to know the maximum food plate.

**Input:**  $n=5$ ,  $arr=[2,7,1,9,5]$

**Output:** 9

**Constraints:**  $1 \leq n \leq 10^5$ ,  
 $-10^9 \leq arr[i] \leq 10^9$

```
01.cpp > main()
1 //kings feast
2 #include <iostream>
3 using namespace std;
4 int maxPlate(int arr[],int n){
5     int max=arr[0];
6     for(int i=0; i<n; i++){
7         if(arr[i]>max){
8             max=arr[i];
9         }
10    }
11    return max;
12 }
13
14 int main() {
15     int n;
16     cin >> n;
17     int arr[n];
18     for(int i=0; i<n; i++){
19         cin >> arr[i];
20     }
21     cout << maxPlate(arr, n) << endl;
22
23 }
```

Debug Console (⌘⌘Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● anshikathakur@Anshikas-Laptop myCproject % g++ 01.cpp && ./a.out  
5  
2 7 1 9 5  
9  
○ anshikathakur@Anshikas-Laptop myCproject %

## Q2. The Lost Soldier

In the battlefield, soldiers are numbered  $0 \dots n$ . One soldier is missing. Find him.

**Input:**  $n=5$ ,  $\text{arr}=[0,1,2,4,5]$

**Output:** 3

**Constraints:**  $O(n)$  or  $O(\log n)$  solution required.

```
02.cpp > main()
1 //lost soldier
2 #include <iostream>
3 using namespace std;
4 int findSoldier(int arr[], int n){
5     int xorAll = 0;
6     int xorArr = 0;
7
8     for(int i=1; i<=n; i++){
9         xorAll = xorAll ^ i;
10    }
11    for(int i=0; i<n; i++){
12        xorArr = xorArr ^ arr[i];
13    }
14    return xorAll ^ xorArr;
15 }
16 int main() {
17     int n;
18     cin >> n;
19     int arr[n];
20     for(int i=0; i<n; i++){
21         cin >> arr[i];
22     }
23     cout << findSoldier(arr, n)<< endl;
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

anshikathakur@Anshikas-Laptop myCproject % g++ 02.cpp && ./a.out  
5  
0 1 2 4 5  
3  
anshikathakur@Anshikas-Laptop myCproject %

### Q3. Potion Mixing (Two Sum)

A wizard wants to mix two potions whose strengths add up to **target**.

**Input:** n=4, arr=[3,2,4,7], target=6

**Output:** Indices (1,2)

**Constraints:**  $1 \leq n \leq 10^5$ ,

$$-10^9 \leq \text{arr}[i] \leq 10^9$$

```
03.cpp > main()
1 //potion mixing
2 #include <iostream>
3 using namespace std;
4 void potionSum(int arr[], int n, int target){
5     for (int i = 0; i < n; i++) {
6         for (int j = i + 1; j < n; j++) {
7             if (arr[i] + arr[j] == target) {
8                 cout << "Indices (" << i << ", " << j << ")" << endl;
9                 return;
10            }
11        }
12    }
13 }
14 int main() {
15     int n;
16     cin >> n;
17     int arr[n];
18     for(int i=0; i<n; i++){
19         cin >> arr[i];
20     }
21     int target;
22     cin >> target;
23     potionSum(arr, n, target);
24 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
● anshikathakur@Anshikas-Laptop myCproject % g++ 03.cpp && ./a.out
4
3 2 4 7
6
Indices (1, 2)
○ anshikathakur@Anshikas-Laptop myCproject %
```

#### Q4. The Secret Message

A spy wrote a secret message as numbers. To decode it, reverse the array.

**Input:** arr=[1,2,3,4]

**Output:** [4,3,2,1]

```
04.cpp > main()
1 //secret message - reverse array
2 #include <iostream>
3 using namespace std;
4 void decodeArr(int arr[], int n){
5     for(int i=0; i<n/2; i++){
6         swap(arr[i], arr[n-i-1]);
7     }
8 }
9 int main() {
10     int n;
11     cin >> n;
12     int arr[n];
13     for(int i=0; i<n; i++){
14         cin >> arr[i];
15     }
16     decodeArr(arr, n);
17     for(int i=0; i<n; i++){
18         cout << arr[i] << " ";
19     }
20     return 0;
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● anshikathakur@Anshikas-Laptop myCproject % g++ 04.cpp && ./a.out

```
4
1 2 3 4
4 3 2 1
```

○ anshikathakur@Anshikas-Laptop myCproject %

### Q5. The King's Parade

Soldiers stand in line. Check if their heights are **sorted in non-decreasing order**.

**Input:** arr=[1,3,5,7] → **Output:** true

**Input:** arr=[3,2,1] → **Output:** false

```
05.cpp > main()
1 // king's parade
2 #include <iostream>
3 using namespace std;
4 bool sortHeight(int arr[], int n){
5     for(int i=0; i <= n-1; i++){
6         for(int j=0; j<n-i-1; j++){
7             if(arr[j] > arr[j+1]){
8                 return false;
9             }
10        }
11    }
12 }
13 return true;
14 }
15 int main() {
16     int n;
17     cin >> n;
18     int arr[n];
19     for(int i=0; i<n; i++){
20         cin >> arr[i];
21     }
22     sortHeight(arr, n);
23     cout << (sortHeight(arr, n) ? "true" : "false") << endl;
24     return 0;
25 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● anshikathakur@Anshikas-Laptop myCproject % g++ 05.cpp && ./a.out
4
1 3 5 7
true
```

```

true
● anshikathakur@Anshikas-Laptop myCproject % g++ 05.cpp && ./a.out
3
3 2 1
false

```

```
○ anshikathakur@Anshikas-Laptop myCproject %
```

## Q6. The Treasure Island

Each island grid has gold. Find the island row with **maximum gold**.

**Input:**

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

**Output:** Row 2 (sum=24)

```
06.cpp > maxRowSum(vector<vector<int>>&, int, int)
1 //island gold grid
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5 void maxRowSum(vector<vector<int>>& arr, int n, int m){
6     int maxSum = 0;
7     int rowIdx = -1;
8     for (int i = 0; i < n; i++) {
9         int rowSum = 0;
10        for (int j = 0; j < m; j++) {
11            rowSum += arr[i][j];
12        }
13        if (rowSum > maxSum) {
14            maxSum = rowSum;
15            rowIdx = i;
16        }
17    }
18    cout << "Row " << rowIdx << " (sum=" << maxSum << ")" << endl;
19 }
20 int main() {
21     int n, m;
22     cin >> n >> m;
23     vector<vector<int>> arr(n, vector<int>(m));
24     for(int i=0; i<n; i++){
25         for(int j=0; j<m; j++){
26             cin >> arr[i][j];
27         }
28     }
29     maxRowSum(arr, n, m);
30     return 0;
31 }
```

```
j/Users/anshikathakur/.zshrc:1: job table full
● anshikathakur@Anshikas-Laptop myCproject % g++ 06.cpp && ./a.out
3 3
1 2 3
4 5 6
7 8 9
Row 2 (sum=24)
○ anshikathakur@Anshikas-Laptop myCproject %
```

## Q7. The Spiral Library

The King built a library where books are kept in spiral shelves. Print them in **spiral order**.

**Input:**

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

**Output:** [1,2,3,6,9,8,7,4,5]

```
03.cpp 04.cpp 05.cpp 06.cpp 07.cpp x ... 07.cpp x
1 //spirallib
2 #include <iostream>
3 using namespace std;
4 vector<int> spirallib(vector<vector<int>>& arr, int n,
5 int m) {
6     vector<int> output;
7     int t = 0, b = n - 1, l = 0, r = m - 1;
8     while (t <= b && l <= r) {
9         // Move l -> r
10        for (int i = l; i <= r; i++)
11            output.push_back(arr[t][i]);
12        t++;
13
14        // Move t -> b
15        for (int i = t; i <= b; i++)
16            output.push_back(arr[i][r]);
17        r--;
18
19        // Move r -> l
20        if (t <= b) {
21            for (int i = r; i >= l; i--)
22                output.push_back(arr[b][i]);
23            b--;
24        }
25
26        // Move b -> t
27        if (l <= r) {
28            for (int i = b; i >= t; i--)
29                output.push_back(arr[i][l]);
30            l++;
31        }
32    }
33    return output;
34 }
35
36 int main() {
37     int n, m;
38     cin >> n >> m;
39
40     vector<vector<int>> arr(n, vector<int>(m));
41     for (int i = 0; i < n; i++)
42         for (int j = 0; j < m; j++)
43             cin >> arr[i][j];
44
45     vector<int> res = spirallib(arr, n, m);
46
47     cout << "[";
48     for (int i = 0; i < res.size(); i++) {
49         cout << res[i];
50         if (i != res.size() - 1) cout << ",";
51     }
52     cout << "]" << endl;
53     return 0;
54 }
```

```

3 3
1 2 3
4 5 6
7 8 9
[1,2,3,6,9,8,7,4,5]
```

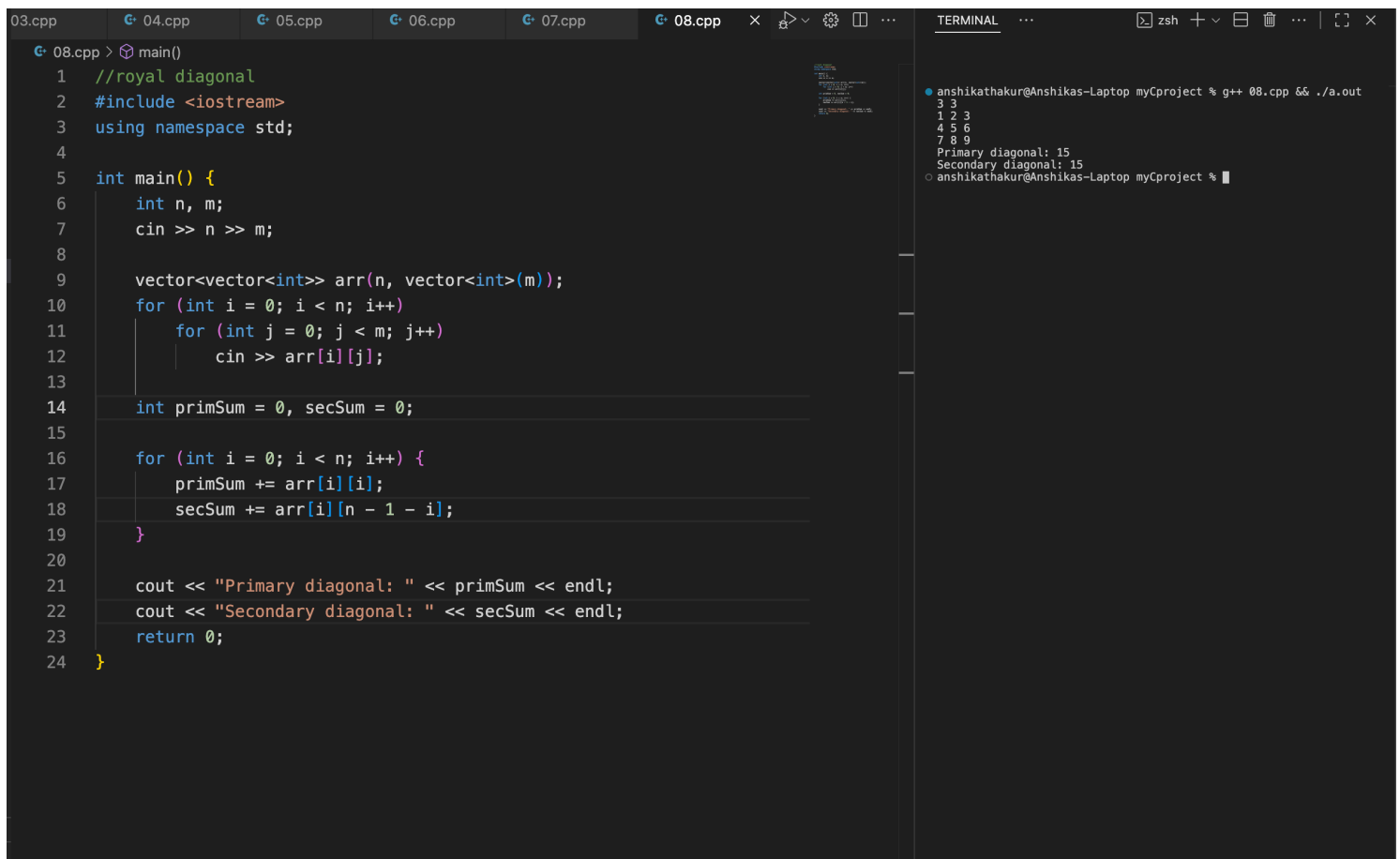
## Q8. The Royal Diagonal

In a royal hall represented as a square, find **sum of both diagonals**.

**Input:**

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

**Output:**  $1+5+9=15$ ,  $3+5+7=15$



```
03.cpp 04.cpp 05.cpp 06.cpp 07.cpp 08.cpp x [icons] ...
08.cpp > main()
1 //royal diagonal
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8
9     vector<vector<int>> arr(n, vector<int>(m));
10    for (int i = 0; i < n; i++)
11        for (int j = 0; j < m; j++)
12            cin >> arr[i][j];
13
14    int primSum = 0, secSum = 0;
15
16    for (int i = 0; i < n; i++) {
17        primSum += arr[i][i];
18        secSum += arr[i][n - 1 - i];
19    }
20
21    cout << "Primary diagonal: " << primSum << endl;
22    cout << "Secondary diagonal: " << secSum << endl;
23    return 0;
24 }
```

```
anshikathakur@Anshikas-Laptop myCproject % g++ 08.cpp && ./a.out
3 3
1 2 3
4 5 6
7 8 9
Primary diagonal: 15
Secondary diagonal: 15
anshikathakur@Anshikas-Laptop myCproject %
```