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()

```cpp
//kings feast
#include <iostream>
using namespace std;
int maxPlate(int arr[],int n){
    int max=arr[0];
    for(int i=0; i<n; i++){
        if(arr[i]>max){
            max=arr[i];
        }
    }
    return max;
}

int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    cout << maxPlate(arr, n) << endl;

}
```

Debug Console (⇧⌘Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

```
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…n. One soldier is missing. Find him.

Input: n=5, arr=[0,1,2,4,5]
Output: 3
Constraints: O(n) or O(log n) solution required.

02.cpp > main()

```cpp
//lost soldier
#include <iostream>
using namespace std;
int findSoldier(int arr[], int n){
    int xorAll = 0;
    int xorArr = 0;

    for(int i=1; i<=n; i++){
        xorAll = xorAll ^ i;
    }
    for(int i=0; i<n; i++){
        xorArr = xorArr ^ arr[i];
    }
    return xorAll ^ xorArr;
}
int main() {
int n;
    cin >> n;
    int arr[n];
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    cout << findSoldier(arr, n)<< endl;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

```
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 ≤ n ≤ 10^5,
-10^9 ≤ arr[i] ≤ 10^9

03.cpp > main()

```cpp
//potion mixing
#include <iostream>
using namespace std;
void potionSum(int arr[], int n, int target){
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] + arr[j] == target) {
                cout << "Indices (" << i << ", " << j << ")" << endl;
                return;
            }
        }
    }
}
int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    int target;
    cin >> target;
    potionSum(arr, n, target);
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

```
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()

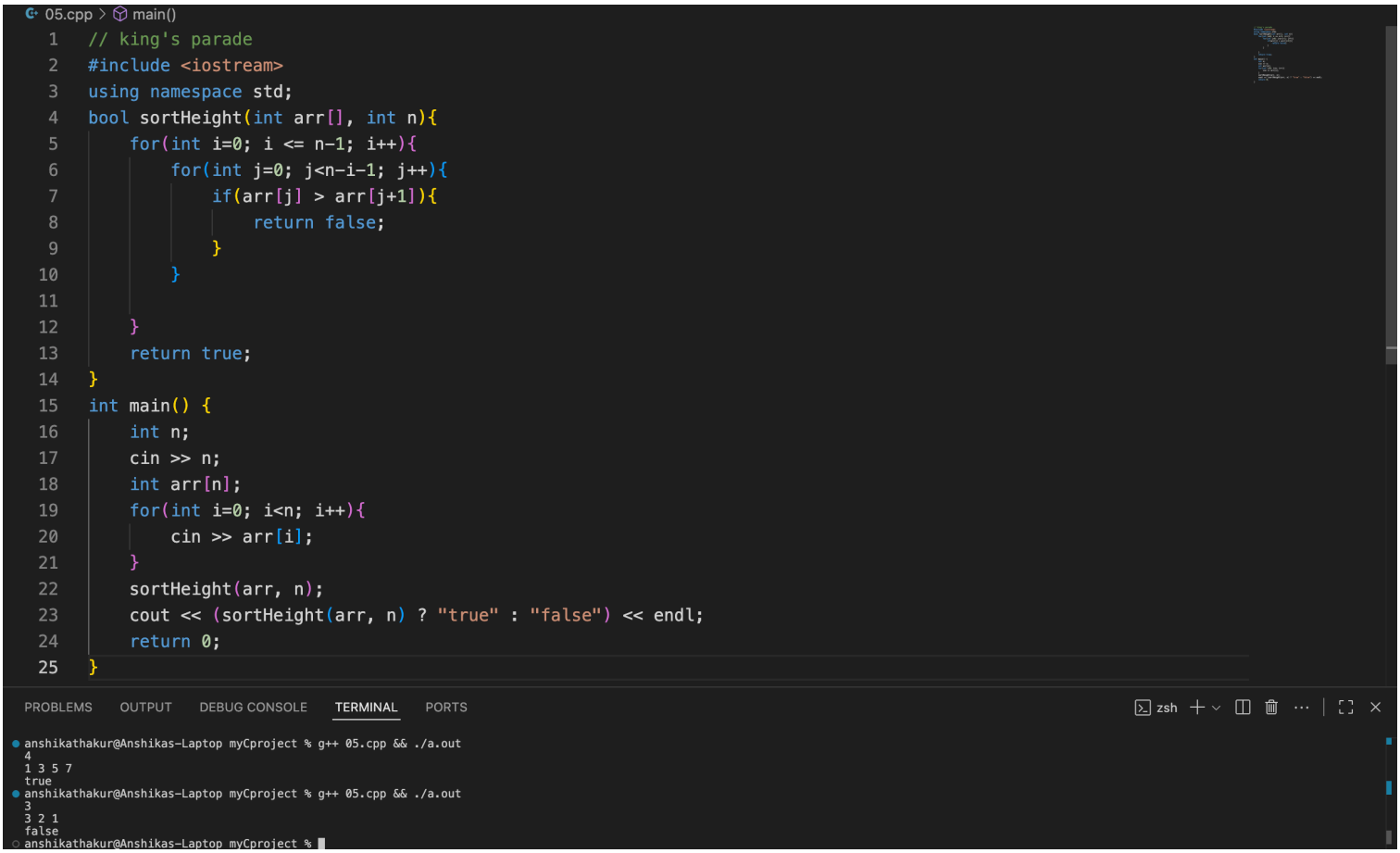
```cpp
//secret message - reverse array
#include <iostream>
using namespace std;
void decodeArr(int arr[], int n){
    for(int i=0; i<n/2; i++){
        swap(arr[i], arr[n-i-1]);
    }
}
int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    decodeArr(arr, n);
    for(int i=0; i<n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    zsh

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



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)

```cpp
//island gold grid
#include <iostream>
#include <vector>
using namespace std;
void maxRowSum(vector<vector<int>>& arr, int n, int m){
    int maxSum = 0;
    int rowIdx = -1;
    for (int i = 0; i < n; i++) {
        int rowSum = 0;
        for (int j = 0; j < m; j++) {
            rowSum += arr[i][j];
        }
        if (rowSum > maxSum) {
            maxSum = rowSum;
            rowIdx = i;
        }
    }
    cout << "Row " << rowIdx << " (sum=" << maxSum << ")" << endl;
}
int main() {
    int n, m;
    cin >> n >> m;
    vector<vector<int>> arr(n, vector<int>(m));
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            cin >> arr[i][j];
        }
    }
    maxRowSum(arr, n, m);
    return 0;
}
```

```
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 %
```

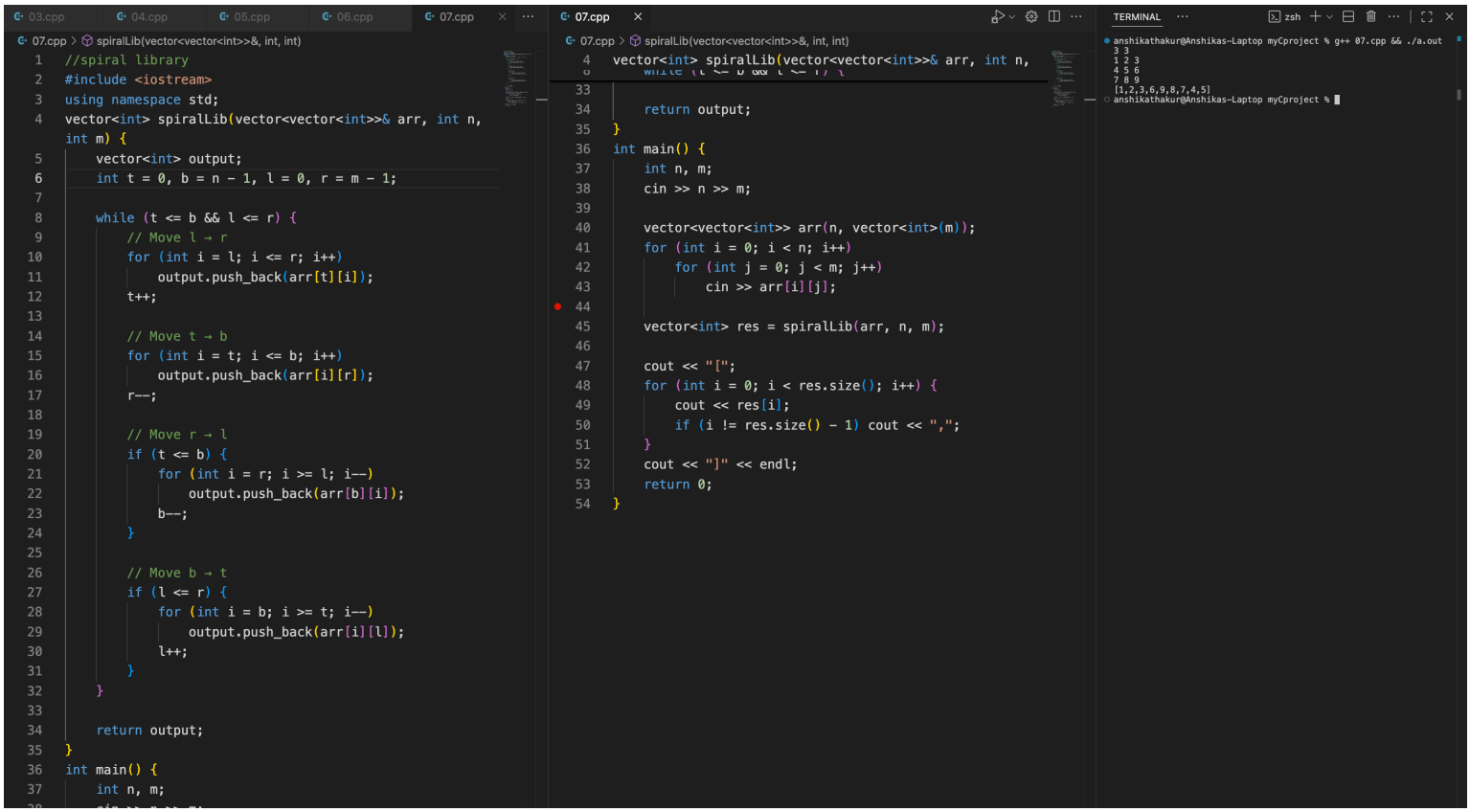
Q7. The Spiral Library

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]



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