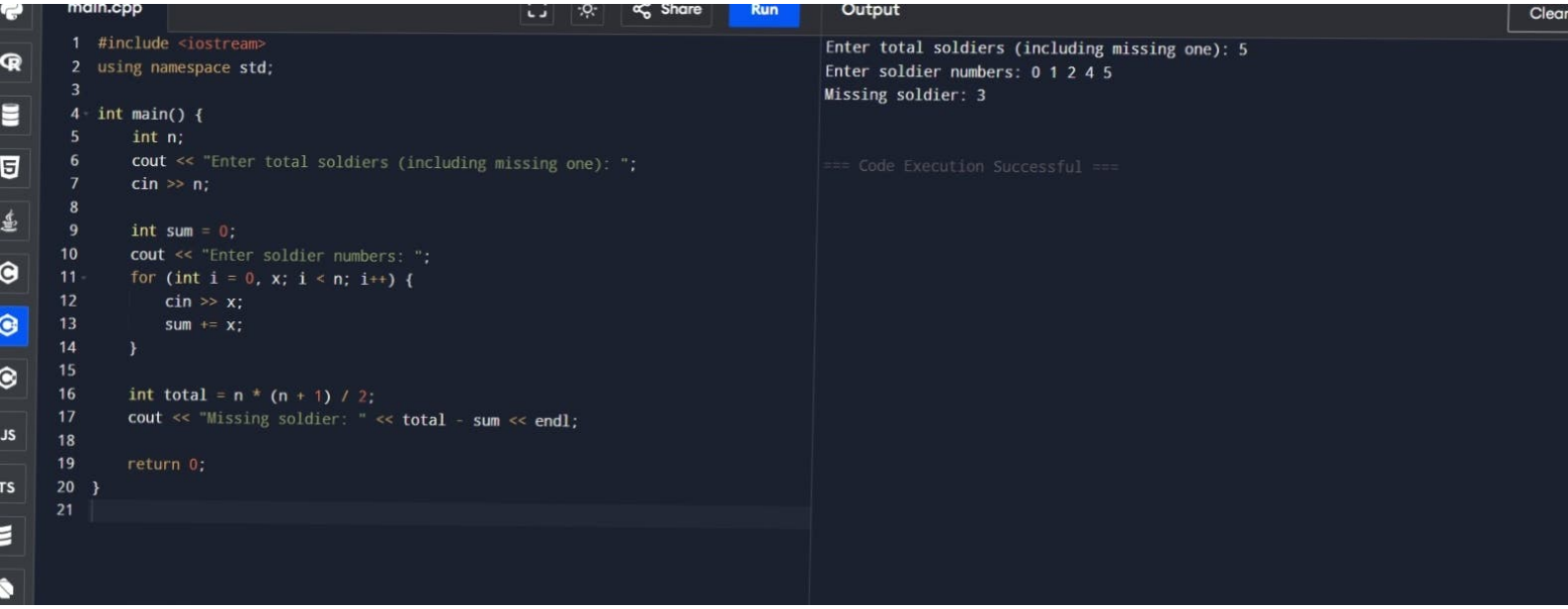
main.cpp
Share
Run
Output
Clear
#include <iostream>
using namespace std;
int main() {
int n;
cout << "Enter total soldiers (including missing one): ";
cin >> n;
int sum = 0;
cout << "Enter soldier numbers: ";
for (int i = 0, x; i < n; i++) {
cin >> x;
sum += x;
}
int total = n * (n + 1) / 2;
cout << "Missing soldier: " << total - sum << endl;
return 0;
}
Enter total soldiers (including missing one): 5
Enter soldier numbers: 0 1 2 4 5
Missing soldier: 3
=== Code Execution Successful ===

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter the secret numbers: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Decoded message: [ ";
    for (int i = n - 1; i >= 0; i--) cout << arr[i] << " ";
    cout << "]" << endl;

    return 0;
}
```

```
Enter number of elements: 4
Enter the secret numbers: 1 2 3 4
Decoded message: [ 4 3 2 1 ]


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <vector>
using namespace std;

int main() {
    int n;
    cout << "Enter number of soldiers: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter their heights: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    bool sorted = true;
    for (int i = 1; i < n; i++) {
        if (arr[i] < arr[i - 1]) {
            sorted = false;
            break;
        }
    }

    cout << (sorted ? "true" : "false") << endl;
    return 0;
}
```

Output Clear

```
Enter number of soldiers: 3
Enter their heights: 8 7 4
false


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int rows, cols;
    cout << "Enter number of rows and columns: ";
    cin >> rows >> cols;

    vector<vector<int>> gold(rows, vector<int>(cols));
    cout << "Enter gold values for each island:\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            cin >> gold[i][j];

    int maxSum = 0, richRow = 0;
    for (int i = 0; i < rows; i++) {
        int sum = 0;
        for (int j = 0; j < cols; j++)
            sum += gold[i][j];
        if (sum > maxSum) {
            maxSum = sum;
            richRow = i;
```

Output Clear

```
Enter number of rows and columns: 3 3

1 2 3

4 5 6

7 8 9


Enter gold values for each island:
Row 3 has the maximum gold (sum = 24)


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;


int moveDisks(int n, char source, char helper, char dest) {
    if (n == 0) return 0;

    int moves = 0;


    moves += moveDisks(n - 1, source, dest, helper);


    cout << "Move disk " << n << " from " << source << " to " << dest << endl;
    moves++;


    moves += moveDisks(n - 1, helper, source, dest);

    return moves;
}

int main() {
```

Output Clear

```
Enter number of golden disks: 3
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total moves required: 7


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Enter number of steps: ";
    cin >> n;

    if (n == 0) {
        cout << "Number of ways: 0" << endl;
        return 0;
    }

    int prev2 = 1;
    int prev1 = 1;
    int ways = 0;

    for (int i = 2; i <= n; i++) {
        ways = prev1 + prev2;
        prev2 = prev1;
        prev1 = ways;
    }

    cout << "Number of ways: " << ways << endl;
```

Output Clear

```
Enter number of steps: 4
Number of ways: 5


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;


void reverseString(string &str, int start, int end) {
    if (start >= end) return;
    swap(str[start], str[end]);
    reverseString(str, start + 1, end - 1);
}

int main() {
    string s;
    cout << "Enter the spell: ";
    cin >> s;

    reverseString(s, 0, s.length() - 1);

    cout << "Reversed spell: " << s << endl;
    return 0;
}

```

Output Clear

```
Enter the spell: abcdef
Reversed spell: fedcba


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;


void printNumbers(int n) {
    if (n == 0) return;
    printNumbers(n - 1);
    cout << n << " ";
}

int main() {
    int n;
    cout << "Enter a number: ";
    cin >> n;

    cout << "Dragon's roar numbers: ";
    printNumbers(n);
    cout << endl;

    return 0;
}

```

Output Clear

```
Enter a number: 5
Dragon's roar numbers: 1 2 3 4 5


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;


int sumArray(vector<int> &arr, int n) {
    if (n == 0) return 0;
    return arr[n - 1] + sumArray(arr, n - 1);
}

int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter array elements: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    int total = sumArray(arr, n);
    cout << "Sum of elements: " << total << endl;

    return 0;
}
```

Output

Clear

```
Enter number of elements: 5
Enter array elements: 5 6 7 8 9
Sum of elements: 35


=== Code Execution Successful ===
```


main.cpp
Share
Run
Output
Clear
#include <iostream>
#include <vector>
using namespace std;
int searchScroll(vector<int> &arr, int n, int key) {
for (int i = 0; i < n; i++) {
if (arr[i] == key)
return i;
}
return -1;
}
int main() {
int n, key;
cout << "Enter number of scrolls: ";
cin >> n;
vector<int> arr(n);
cout << "Enter scroll IDs: ";
for (int i = 0; i < n; i++) cin >> arr[i];
cout << "Enter the scroll ID to search: ";
cin >> key;
Enter number of scrolls: 4
Enter scroll IDs: 2 5 7 8
Enter the scroll ID to search: 7
Scroll found at index: 2
=== Code Execution Successful ===

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, key;
    cout << "Enter number of fruits: ";
    cin >> n;

    vector<int> basket(n);
    cout << "Enter the fruit numbers: ";
    for (int i = 0; i < n; i++) cin >> basket[i];

    cout << "Enter the fruit to search: ";
    cin >> key;

    int index = -1;
    for (int i = 0; i < n; i++) {
        if (basket[i] == key) {
            index = i;
            break;
        }
    }
```

Output Clear

```
Enter number of fruits: 3
Enter the fruit numbers: 10 20 30
Enter the fruit to search: 25
-1


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;


int findDoor(const vector<int>& doors, int key) {
    int left = 0, right = doors.size() - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (doors[mid] == key)
            return mid;
        else if (doors[mid] < key)
            left = mid + 1;
        else
            right = mid - 1;
    }

    return -1;
}

int main() {
    int n, key;
```

Output Clear

```
Enter number of doors: 3
Enter door numbers in increasing order: 1 5 7
Enter the door to find: 7
2


=== Code Execution Successful ===
```

main.cpp

Share Run Output Clear

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, key;
    cout << "Enter number of arrows: ";
    cin >> n;

    vector<int> distances(n);
    cout << "Enter arrow distances: ";
    for (int i = 0; i < n; i++) cin >> distances[i];

    cout << "Enter the target distance: ";
    cin >> key;

    int firstIndex = -1;
    for (int i = 0; i < n; i++) {
        if (distances[i] == key) {
            firstIndex = i;
            break;
        }
    }

```

```
Enter number of arrows: 4
Enter arrow distances: 1 2 2 2 3
Enter the target distance: -1


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, target;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter elements in sorted order: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Enter target: ";
    cin >> target;

    int left = 0, right = n - 1;
    int result = n;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] >= target) {
            result = mid;
```

Output Clear

```
Enter number of elements: 5
Enter elements in sorted order: 1 3 5 7 9
Enter target: 7
First index ≥ target: 3


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, target;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter elements in sorted order: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Enter target: ";
    cin >> target;

    int left = 0, right = n - 1;
    int result = n;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] > target) {
            result = mid;
```

Output Clear

```
Enter number of elements: 6
Enter elements in sorted order: [1, 2, 4, 6, 6, 8]
Enter target: Upper bound index: 6


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, target;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter elements in sorted order: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Enter target: ";
    cin >> target;

    int left = 0, right = n - 1;
    int ceilValue = -1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] >= target) {
            ceilValue = arr[mid];
```

Output Clear

```
Enter number of elements: 6
Enter elements in sorted order: 1 2 4 6 6 8
Enter target: 4
Ceil of target: 4


=== Code Execution Successful ===
```

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, target;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter elements in sorted order: ";
    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Enter target: ";
    cin >> target;

    int left = 0, right = n - 1;
    int floorValue = -1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] <= target) {
            floorValue = arr[mid];
```

Output Clear

```
Enter number of elements: 4
Enter elements in sorted order: 2 4 6 8
Enter target: 4
Floor of target: 4


=== Code Execution Successful ===
```