



Test\_35 > Q\_1.cpp > main()

```
1  #include <iostream>
2  using namespace std;
3
4  int maxEle(int arr[], int n) {
5      int max = arr[0];
6      for(int i=1; i<n; i++) {
7          if(arr[i] > max) {
8              max = arr[i];
9          }
10     }
11     return max;
12 }
13
14 int main() {
15     int arr[] = {2, 7, 1, 9, 5};
16     int n = 5;
17
18     cout << "maximum : " << maxEle(arr, n);
19
20     return 0;
21 }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)"
maximum : 9
PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test_35>
```

Test\_35 >  Q\_2.cpp >  main()

```
1  #include <iostream>
2  using namespace std;
3
4  int missingNo(int arr[], int n) {
5      int sum = n * (n+1)/2;
6      int arrSum = 0;
7      for(int i=0; i<n; i++) {
8          arrSum += arr[i];
9      }
10
11     return sum - arrSum;
12 }
13
14 int main() {
15     int arr[] = {0, 1, 2, 4, 5};
16     int n = 5;
17
18     cout << missingNo(arr, n) << endl;
19
20     return 0;
21 }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35"

PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35>

Test\_35 > Q3\_twoSum.cpp > main()

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  void twoSum(const vector<int>& arr, int target, int &i1, int &i2) {
6      i1 = i2 = -1;
7      for(int i = 0; i < arr.size(); i++) {
8          for(int j = i + 1; j < arr.size(); j++) {
9              if(arr[i] + arr[j] == target) {
10                 i1 = i; i2 = j;
11                 return;
12             }
13         }
14     }
15 }
16
17 int main() {
18     vector<int> arr = {3, 2, 4, 7};
19     int target = 7, idx1, idx2;
20     twoSum(arr, target, idx1, idx2);
21     if(idx1 != -1) {
22         cout << "Indices: (" << idx1 << ", " << idx2 << ")\n";
23     } else {
24         cout << "No valid pair found.\n";
25     }
26     return 0;
27 }
```



PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35" & gcc Q3\_twoSum.cpp -std=c++11 -o twoSum

No valid pair found.

PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35" & gcc Q3\_twoSum.cpp -std=c++11 -o twoSum



Indices: (0, 2)

Test\_35 >  Q4\_reverse.cpp >  main()

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  void reverseArray(vector<int>& arr) {
6      int n = arr.size();
7      for(int i = 0; i < n/2; i++) {
8          swap(arr[i], arr[n - 1 - i]);
9      }
10 }
11
12 int main() {
13     vector<int> arr = {1, 2, 3, 4};
14     reverseArray(arr);
15     for(int i = 0; i < arr.size(); i++) {
16         cout << arr[i] << ",";
17     }
18     cout << endl;
19     return 0;
20 }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\One
4_reverse }
4,3,2,1,
```

Test\_35 >  Q5\_isSorted.cpp >  main()

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  bool isSorted(vector<int>& arr) {
6      for(int i = 0; i < arr.size() - 1; i++) {
7          if(arr[i] > arr[i+1]) {
8              return false;
9          }
10     }
11     return true;
12 }
13
14 int main() {
15     vector<int> arr1 = {1, 3, 5, 7};
16     vector<int> arr2 = {3, 2, 1};
17
18     cout << "arr1 sorted? " << isSorted(arr1) << endl;
19     cout << "arr2 sorted? " << isSorted(arr2) << endl;
20
21     return 0;
22 }
23
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA\Q5_isSorted"
arr1 sorted? 1
arr2 sorted? 0
```

Test\_35 > Q6\_maxGold.cpp > main()

```
6  int rowWithMaxGold(int mat[][3], int numRows, int numCols, int &maxSum) {
7      int maxRow = -1;
8      maxSum = INT_MIN;
9
10     for(int row = 0; row < numRows; row++) {
11         int sum = 0;
12         for(int col = 0; col < numCols; col++)
13             sum += mat[row][col];
14         if(sum > maxSum) {
15             maxSum = sum;
16             maxRow = row;
17         }
18     }
19
20     return maxRow;
21 }
22
23 int main() {
24     int mat[][3] = {
25         {1, 2, 3},
26         {4, 5, 6},
27         {7, 8, 9}
28     };
29
30     int maxSum;
31     int maxRow = rowWithMaxGold(mat, 3, 3, maxSum);
32     cout << "Row " << maxRow << " (sum=" << maxSum << ")\n";
33 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)\6\_maxGold"

Row 2 (sum=24)

PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test\_35>

```

4 void spiralMatrix(int mat[][4], int n, int m) {
5     int srow = 0, scol = 0;
6     int erow = n-1, ecol = m-1;
7
8     while(srow <= erow && scol <= ecol) {
9
10        //top
11        for(int j=scol; j<=ecol; j++) {
12            cout << mat[srow][j] << " ";
13        }
14
15        //right
16        for(int i=srow+1; i<=erow; i++) {
17            cout << mat[i][ecol] << " ";
18        }
19
20        //bottom
21        for(int j=ecol-1; j>=scol; j--) {
22            if(srow == erow) { // middle
23                break;
24            }
25            cout << mat[erow][j] << " ";
26        }
27
28        //left
29        for(int i=erow-1; i>=srow+1; i--) {
30            if(srow == erow) { // middle
31                break;
32            }
33            cout << mat[i][scol] << " ";
34        }
35
36        srow++; scol++;
37        erow--; ecol--;
38
39    }
40    cout << endl;
41 }
42
43 int main() {
44     int mat[3][4] = { {1, 2, 3, 4},
45                       {5, 6, 7, 8},
46                       {9, 10, 11, 12},};
47     spiralMatrix(mat, 3, 4);



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\sumit\OneDrive\Documents\c++ DSA (Course)> cd "c:\Users\sumit\OneDrive\Documents\c++ DSA (Course)"
spiralMatrix }
1 2 3 4 8 12 11 10 9 5 6 7

```

Test\_35 >  Q8\_digonalSum.cpp >  main()

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n = 3;
6      int mat[3][3] = {
7          {1, 2, 3},
8          {4, 5, 6},
9          {7, 8, 9}
10     };
11
12     int sumPrimary = 0, sumSecondary = 0;
13
14     for(int i = 0; i < n; i++) {
15         sumPrimary += mat[i][i]; //primary diagonal
16         sumSecondary += mat[i][n - 1 - i]; //secondary diagonal
17     }
18
19     cout << "Primary diagonal sum: " << sumPrimary << endl;
20     cout << "Secondary diagonal sum: " << sumSecondary << endl;
21
22     return 0;
23 }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents
{ .\Q8_digonalSum }
Primary diagonal sum: 15
Secondary diagonal sum: 15
PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test_35>
```



Test\_35 > Q9\_findPath.cpp > main()

```
4 bool canReach(int mat[][3], int n, int m, int i, int j) {
5     // Check bounds and blocked cells
6     if(i < 0 || j < 0 || i >= n || j >= m || mat[i][j] == 1)
7         return false;
8
9     // Reached destination
10    if(i == n-1 && j == m-1) return true;
11
12    mat[i][j] = 1; // Mark as visited
13
14    // Move in all 4 directions
15    if(canReach(mat, n, m, i+1, j)) return true;
16    if(canReach(mat, n, m, i-1, j)) return true;
17    if(canReach(mat, n, m, i, j+1)) return true;
18    if(canReach(mat, n, m, i, j-1)) return true;
19
20    return false;
21 }
22
23 int main() {
24     int mat[3][3] = {
25         {0, 0, 0},
26         {0, 1, 0},
27         {0, 0, 0}
28     };
29
30     cout << "Can reach destination: " << (canReach(mat, 3, 3, 0, 0) ? "true" : "false") << endl;
31     return 0;
32 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sumit\OneDrive\Documents\DSA(college)> cd "c:\Users\sumit\OneDrive\Documents\DSA(college)\Test_35\
Q9_findPath"
Can reach destination: true
PS C:\Users\sumit\OneDrive\Documents\DSA(college)\Test_35>
```