

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## DOMAIN WINTER WINNING CAMP(Day-8)

Student Name: SHREYA BISWAS

Branch: BE-CSE

Semester: 6th

UID: 22BCS15325

Section/Group: 620-A

Date of Performance: 29/12/24

### 1) N-th Tribonacci Number

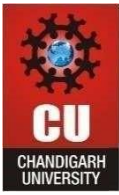
```
#include <iostream> #include  
<vector>  
using namespace std;
```

```
int tribonacci(int n) {  
    vector<int> dp(max(3, n + 1));  
    dp[0] = 0;    dp[1] = dp[2] = 1;    for  
(int i = 3; i <= n; i++) {        dp[i] = dp[i -  
1] + dp[i - 2] + dp[i - 3];  
    }  
    return dp[n];  
}
```

```
int main() {    int n = 4;    cout << "Tribonacci(" << n << ") = "  
<< tribonacci(n) << endl; return 0; }
```

main.cpp	Run	Output
<pre>1  #include &lt;iostream&gt; 2  #include &lt;vector&gt; 3  using namespace std; 4 5  int tribonacci(int n) { 6      vector&lt;int&gt; dp(max(3, n + 1)); 7      dp[0] = 0;</pre>		<pre>Tribonacci(4) = 4  === Code Execution Suc</pre>

### 2) Divisor Game



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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

bool divisorGame(int n) {
    return n % 2 == 0;
}

int main() {    int n;    cout <<
"Enter the value of n: ";
    cin >> n;    cout << "Alice wins: " << (divisorGame(n) ? "Yes" :
"No") << endl;    return 0;
}
```

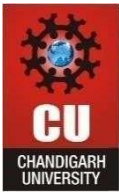
main.cpp	Run	Output
<pre>1 #include &lt;iostream&gt; 2 using namespace std; 3 4 bool divisorGame(int n) { 5     return n % 2 == 0; 6 }</pre>		<pre>Enter the value of n: 5 Alice wins: No  === Code Execution Success</pre>

### 3) Climbing Stairs

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

int climbStairs(int n) {    if
(n <= 2) return n;    int a = 1,
b = 2;    for (int i = 3; i <= n;
i++) {        int temp = a + b;
        a = b;
        b = temp;
    }
    return b;
}

int main() {
    int n;
    cout << "Enter the number of steps: ";
    cin >> n;    cout << "Ways to climb " << n << " stairs: " <<
climbStairs(n) << endl;    return 0;
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output :

```
main.cpp  [Icons]  Share  Run  Output
1 #include <iostream>
2 using namespace std;
3
4 int climbStairs(int n) {
5     if (n <= 2) return n;
6     int a = 1, b = 2;
7     for (int i = 3; i <= n; i++) {
8         int temp = a + b;
9         a = b;
10        b = temp;
11    }
12    return b;
13 }
```

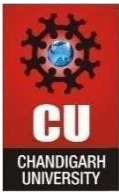
Enter the number of steps: 2  
Ways to climb 2 stairs: 2  
=== Code Execution Successful

## 4) Best Time to Buy and Sell Stock

```
#include <iostream>
#include <vector>
#include <climits> // For INT_MAX using
namespace std;

int maxProfit(vector<int>& prices) {
    int minPrice = INT_MAX, maxProfit = 0;
    for (int price : prices) {
        minPrice = min(minPrice, price); // Update the minimum price seen so far
        maxProfit = max(maxProfit, price - minPrice); // Calculate the maximum profit
    }
    return maxProfit;
}

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



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

*Discover. Learn. Empower.*

```
vector<int> prices(n);    cout <<  
"Enter the stock prices: ";    for (int  
i = 0; i < n; i++) {
```

# DEPARTMENT OF

Discover. Learn. Empower.

# COMPUTER SCIENCE & ENGINEERING



```
        cin >> prices[i];
    }

    cout << "Max profit: " << maxProfit(prices) << endl;
    return 0;
}
```

A screenshot of a web browser displaying an online C++ compiler interface. The browser tabs include WhatsApp, ChatGPT, and the current tab 'Online C++ Compiler - Program'. The address bar shows 'programiz.com/cpp-programming/online-compiler/'. The interface has a dark theme. At the top, there's a 'Programiz' logo and a banner for 'Premium Coding Courses by Programiz'. Below this, the code editor shows a C++ program for finding the maximum profit from stock prices. The code includes headers for <iostream>, <vector>, and <climits>, uses the std namespace, and defines a maxProfit function and a main function. The output window on the right shows the program's execution: 'Enter the number of days: 6', 'Enter the stock prices: 7', followed by a list of prices (1, 5, 3, 6, 4) and the result 'Max profit: 5'. A status message at the bottom of the output window says '=== Code Execution Successful ==='. The Windows taskbar is visible at the bottom of the screen, showing the time as 7:45 PM on 12/28/2024.

## 5) Generate Parentheses

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

## DEPARTMENT OF

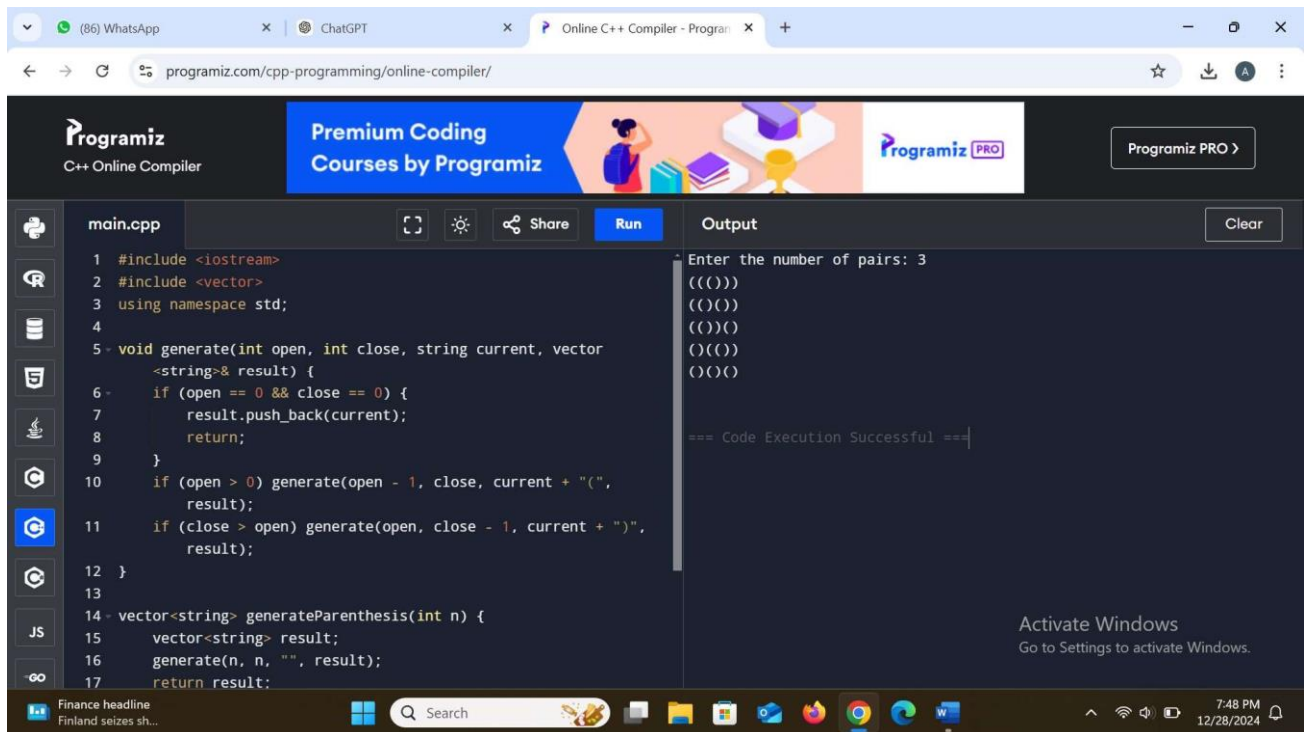
## COMPUTER SCIENCE & ENGINEERING

```
void generate(int open, int close, string current, vector<string>& result) {  
    if (open == 0 && close == 0) {        result.push_back(current);  
        return;  
    }  
    if (open > 0) generate(open - 1, close, current + "(", result);  
    if (close > open) generate(open, close - 1, current + ")", result);  
}
```

```
vector<string> generateParenthesis(int n) {  
    vector<string> result;  
    generate(n, n, "", result);    return  
    result;  
}
```

```
int main() {  
    int n;  
    cout << "Enter the number of pairs: ";  
    cin >> n;    vector<string> result =  
    generateParenthesis(n);  
    for (const string& s : result) {  
        cout << s << endl;  
    }  
    return 0; }
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



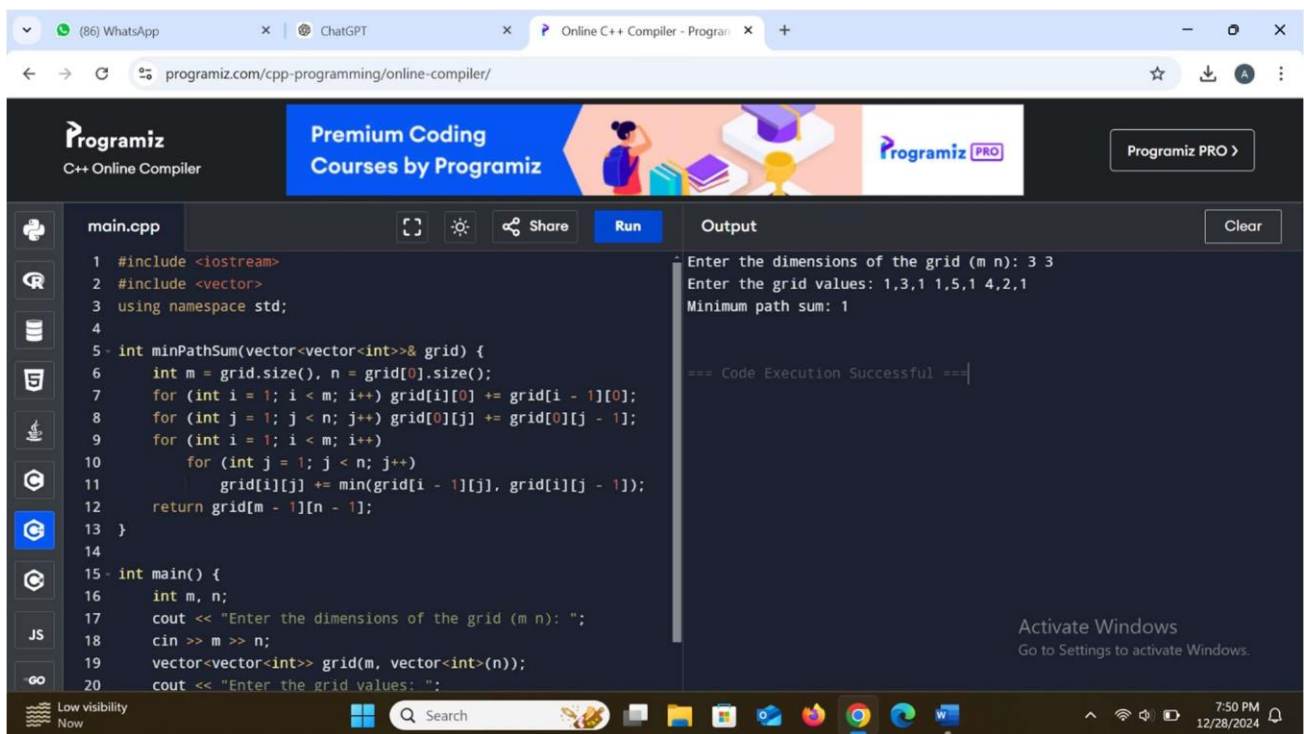
## 6) Minimum Path Sum

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

```
int minPathSum(vector<vector<int>>& grid) {    int  
m = grid.size(), n = grid[0].size();    for (int i = 1; i <  
m; i++) grid[i][0] += grid[i - 1][0];    for (int j = 1; j <  
n; j++) grid[0][j] += grid[0][j - 1];    for (int i = 1; i <  
m; i++)        for (int j = 1; j < n; j++)            grid[i][j]  
+= min(grid[i - 1][j], grid[i][j - 1]);    return grid[m -  
1][n - 1];  
}
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
int main() {    int m, n;    cout << "Enter the
dimensions of the grid (m n): ";    cin >> m >> n;
    vector<vector<int>> grid(m, vector<int>(n));
    cout << "Enter the grid values: ";
    for (int i = 0; i < m; i++)
for (int j = 0; j < n; j++)
cin >> grid[i][j];
    cout << "Minimum path sum: " << minPathSum(grid) << endl;
return 0;
}
```



The screenshot shows the Programiz C++ Online Compiler interface. The code in main.cpp is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int minPathSum(vector<vector<int>>& grid) {
6     int m = grid.size(), n = grid[0].size();
7     for (int i = 1; i < m; i++) grid[i][0] += grid[i - 1][0];
8     for (int j = 1; j < n; j++) grid[0][j] += grid[0][j - 1];
9     for (int i = 1; i < m; i++)
10         for (int j = 1; j < n; j++)
11             grid[i][j] += min(grid[i - 1][j], grid[i][j - 1]);
12     return grid[m - 1][n - 1];
13 }
14
15 int main() {
16     int m, n;
17     cout << "Enter the dimensions of the grid (m n): ";
18     cin >> m >> n;
19     vector<vector<int>> grid(m, vector<int>(n));
20     cout << "Enter the grid values: ";
```

The output on the right shows the execution results:

```
Enter the dimensions of the grid (m n): 3 3
Enter the grid values: 1,3,1 1,5,1 4,2,1
Minimum path sum: 1

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

The Windows taskbar at the bottom shows the date and time as 7:50 PM on 12/28/2024.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## 7) Maximal Rectangle

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

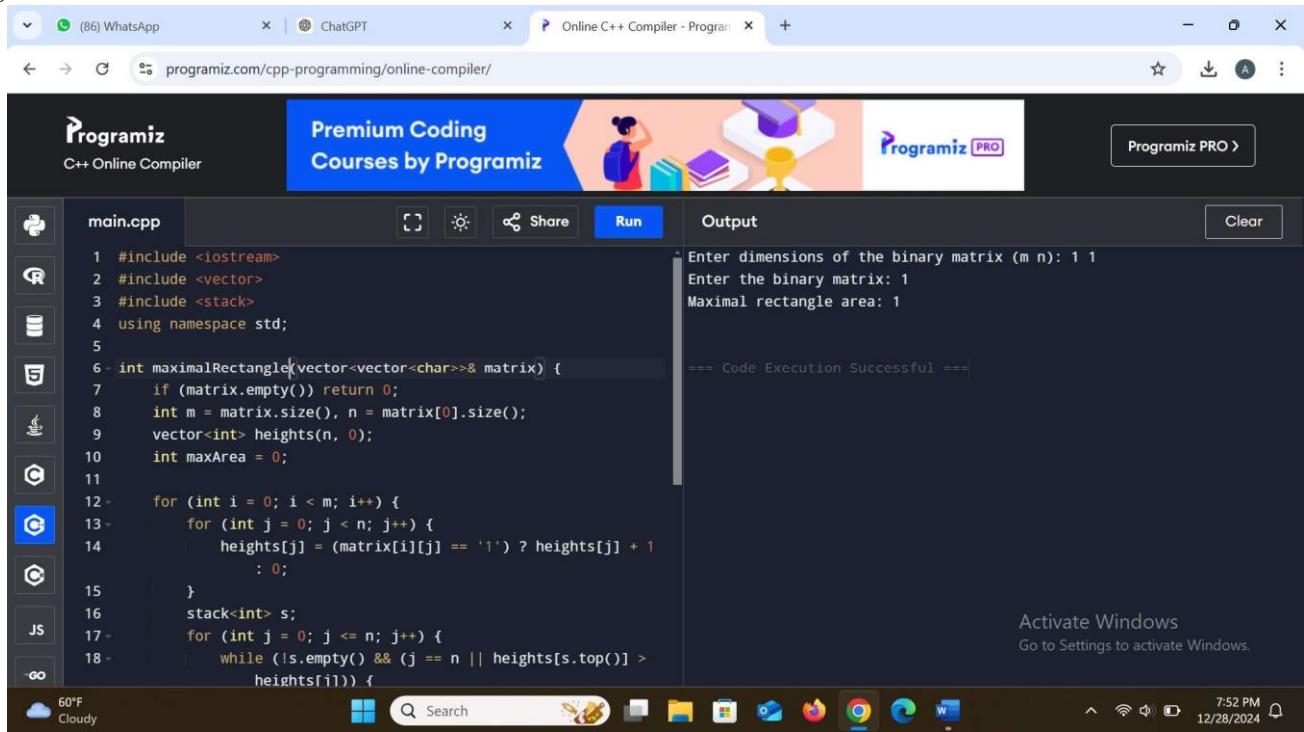
int maximalRectangle(vector<vector<char>>& matrix) {
    if (matrix.empty()) return 0;    int m =
matrix.size(), n = matrix[0].size();
    vector<int> heights(n, 0);    int maxArea =
0;

    for (int i = 0; i < m; i++) {        for (int j = 0; j < n; j++)
    {            heights[j] = (matrix[i][j] == '1') ? heights[j] + 1 :
0;
        }
    stack<int> s;
        for (int j = 0; j <= n; j++) {            while (!s.empty() && (j == n
|| heights[s.top()] > heights[j])) {                int h = heights[s.top()];
s.pop();                int width = s.empty() ? j : j - s.top() - 1;
                maxArea = max(maxArea, h * width);
            }
            s.push(j);
        }
    }
    return maxArea;
}

int main() {    int m, n;    cout << "Enter dimensions of
the binary matrix (m n): ";    cin >> m >> n;
    vector<vector<char>> matrix(m, vector<char>(n));
    cout << "Enter the binary matrix: ";
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
for (int i = 0; i < m; i++)    for (int j = 0; j < n; j++)    cin >>
matrix[i][j];    cout << "Maximal rectangle area: " <<
maximalRectangle(matrix) << endl;    return 0;
}
```



## 8) Dungeon Game

```
#include <iostream>
#include <vector>
#include <climits> // For INT_MAX using
namespace std;
```

```
int calculateMinimumHP(vector<vector<int>>& dungeon) {
    int m = dungeon.size(), n = dungeon[0].size();
    vector<vector<int>> dp(m + 1, vector<int>(n + 1, INT_MAX));
    dp[m][n - 1] = dp[m - 1][n] = 1;

    for (int i = m - 1; i >= 0; i--) {        for (int j = n - 1; j >= 0; j--) {
        dp[i][j] = max(1, min(dp[i + 1][j], dp[i][j + 1]) - dungeon[i][j]);
    }
    }
    return dp[0][0];
}
```

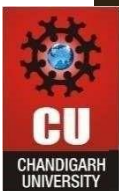
# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
int main() {    int m, n;    cout << "Enter dimensions of  
the dungeon grid (m n): ";    cin >> m >> n;  
    vector<vector<int>> dungeon(m, vector<int>(n));  
    cout << "Enter the dungeon grid values: " << endl;  
    for (int i = 0; i < m; i++) {  
    for (int j = 0; j < n; j++) {  
        cin >> dungeon[i][j];  
    }  
    }  
  
    cout << "Minimum initial health required: " << calculateMinimumHP(dungeon) <<  
endl;    return 0;  
}
```

The screenshot shows a web browser window with the URL `programiz.com/cpp-programming/online-compiler/`. The page features the Programiz logo and a navigation bar with options like "Premium Coding Courses by Programiz" and "Programiz PRO". The main area is divided into two panels: a code editor on the left and an output window on the right. The code editor contains a C++ program that defines a `calculateMinimumHP` function and a `main` function. The `main` function prompts the user to enter the dimensions of the dungeon grid (m n) and the grid values. The output window shows the execution results: "Enter dimensions of the dungeon grid (m n): 1 1", "Enter the dungeon grid values: 0", and "Minimum initial health required: 1". The output also indicates "Code Execution Successful". The bottom of the browser window shows a Windows taskbar with the date and time as 7:54 PM on 12/28/2024.

```
1 #include <iostream>  
2 #include <vector>  
3 #include <limits> // For INT_MAX  
4 using namespace std;  
5  
6 int calculateMinimumHP(vector<vector<int>>& dungeon) {  
7     int m = dungeon.size(), n = dungeon[0].size();  
8     vector<vector<int>> dp(m + 1, vector<int>(n + 1, INT_MAX));  
9     dp[m][n - 1] = dp[m - 1][n] = 1;  
10  
11     for (int i = m - 1; i >= 0; i--) {  
12         for (int j = n - 1; j >= 0; j--) {  
13             dp[i][j] = max(1, min(dp[i + 1][j], dp[i][j + 1]) -  
                dungeon[i][j]);  
14         }  
15     }  
16     return dp[0][0];  
17 }  
18  
19 int main() {
```

Enter dimensions of the dungeon grid (m n): 1 1  
Enter the dungeon grid values:  
0  
Minimum initial health required: 1  
=== Code Execution Successful ===



Discover. Learn. Empower.

# DEPARTMENT OF

## COMPUTER SCIENCE & ENGINEERING

```
#include <iostream>
#include <vector>
#include <queue>
#include <unordered_set> #include
<string>
using namespace std;

int slidingPuzzle(vector<vector<int>>& board) {
    string target = "123450";
    string start = "";    for (auto&
row : board)        for (int cell :
row)            start +=
to_string(cell);

    vector<vector<int>> dirs = {{1, 3}, {0, 2, 4}, {1, 5}, {0, 4}, {1, 3, 5}, {2, 4}};
    unordered_set<string> visited;    queue<pair<string, int>> q;    q.push({start,
0});
    visited.insert(start);

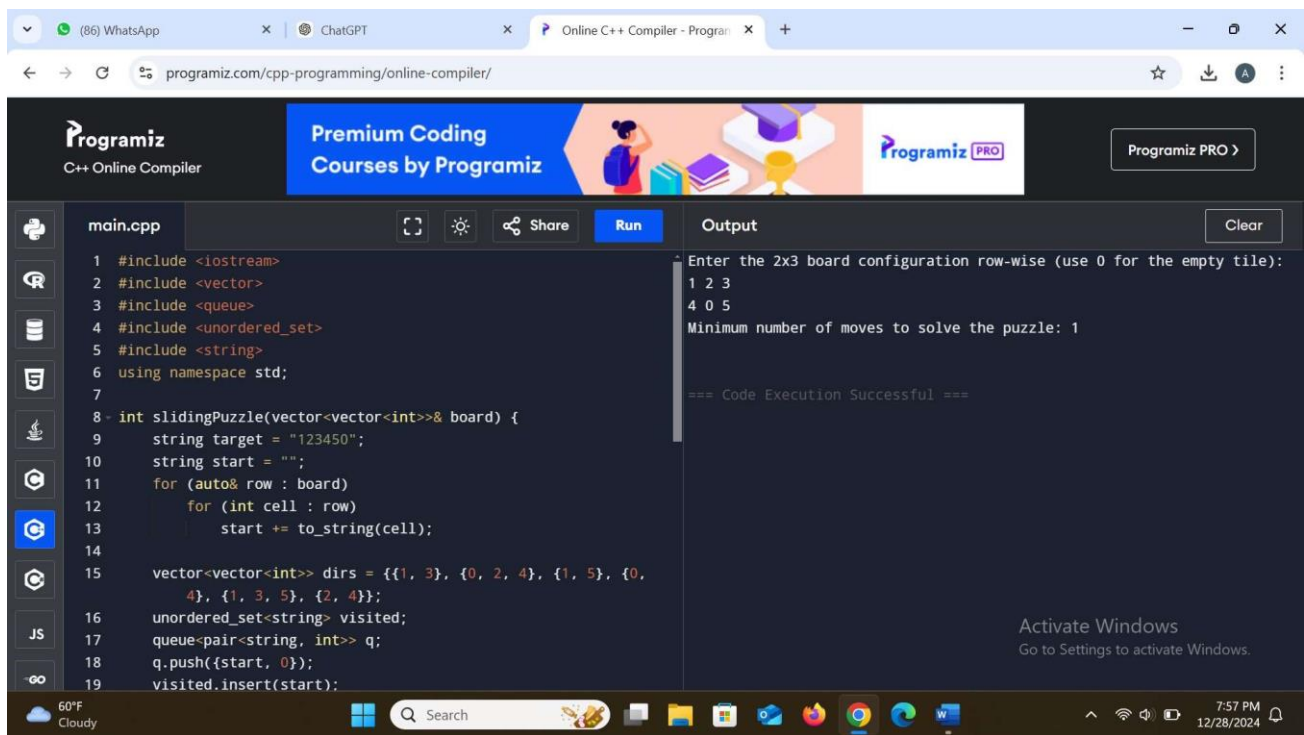
    while (!q.empty()) {        auto
[curr, steps] = q.front();
q.pop();        if (curr == target)
return steps;

        int zeroPos = curr.find('0');
        for (int dir : dirs[zeroPos]) {
            string next = curr;
            swap(next[zeroPos], next[dir]);
            if (visited.find(next) == visited.end()) {
                visited.insert(next);
                q.push({next, steps + 1});
            }
        }
    }
    return -1; // Impossible to solve }

int main() {
    vector<vector<int>> board(2, vector<int>(3));    cout << "Enter the 2x3 board
configuration row-wise (use 0 for the empty tile):" << endl;    for (int i = 0; i < 2;
i++)        for (int j = 0; j < 3; j++)
        cin >> board[i][j];
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
int result = slidingPuzzle(board);
if (result == -1) {      cout << "The puzzle is
unsolvable." << endl;
} else {
    cout << "Minimum number of moves to solve the puzzle: " << result << endl;
}
return 0;
}
```



The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page features the Programiz logo and a banner for "Premium Coding Courses by Programiz". The main interface is divided into two panels. The left panel, titled "main.cpp", contains C++ code for a sliding puzzle solver. The right panel, titled "Output", shows the execution results. The code defines a `slidingPuzzle` function that takes a 2x3 board and returns the minimum number of moves to solve it. The output shows the input board configuration `1 2 3` and `4 0 5`, and the result `Minimum number of moves to solve the puzzle: 1`. The status "Code Execution Successful" is displayed at the bottom of the output panel. The browser's taskbar at the bottom shows the system time as 7:57 PM on 12/28/2024.

```
main.cpp
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4 #include <unordered_set>
5 #include <string>
6 using namespace std;
7
8 int slidingPuzzle(vector<vector<int>>& board) {
9     string target = "123450";
10    string start = "";
11    for (auto& row : board)
12        for (int cell : row)
13            start += to_string(cell);
14
15    vector<vector<int>> dirs = {{1, 3}, {0, 2, 4}, {1, 5}, {0,
16    4}, {1, 3, 5}, {2, 4}};
17    unordered_set<string> visited;
18    queue<pair<string, int>> q;
19    q.push({start, 0});
20    visited.insert(start);
```

Output

```
Enter the 2x3 board configuration row-wise (use 0 for the empty tile):
1 2 3
4 0 5
Minimum number of moves to solve the puzzle: 1

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

## 10) Super Egg Drop

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

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
int superEggDrop(int k, int n) {    vector<vector<int>>
dp(k + 1, vector<int>(n + 1, 0));    int moves = 0;

    while (dp[k][moves] < n) {
moves++;
        for (int i = 1; i <= k; i++) {            dp[i][moves] = dp[i -
1][moves - 1] + dp[i][moves - 1] + 1;
        }
    }
    return moves;
}

int main() {    int k, n;    cout << "Enter
the number of eggs: ";    cin >> k;
cout << "Enter the number of floors: ";
    cin >> n;

    cout << "Minimum number of moves to find the critical floor: " <<
superEggDrop(k, n) << endl;
    return 0;
}
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

The screenshot displays the Programiz C++ Online Compiler interface. The browser address bar shows the URL `programiz.com/cpp-programming/online-compiler/`. The page header includes the Programiz logo, navigation links for Premium Coding Courses, and a 'Programiz PRO' button. The main editor area is titled 'main.cpp' and contains the following C++ code:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int superEggDrop(int k, int n) {
6     vector<vector<int>>> dp(k + 1, vector<int>(n + 1, 0));
7     int moves = 0;
8
9     while (dp[k][moves] < n) {
10         moves++;
11         for (int i = 1; i <= k; i++) {
12             dp[i][moves] = dp[i - 1][moves - 1] + dp[i][moves - 1] + 1;
13         }
14     }
15     return moves;
16 }
17
18 int main() {
19     int k, n;
```

The 'Output' panel on the right shows the execution results:

```
Enter the number of eggs: 1
Enter the number of floors: 2
Minimum number of moves to find the critical floor: 2

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

The Windows taskbar at the bottom indicates a temperature of 60°F, a search bar, and the date/time 7:58 PM on 12/28/2024.