



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment 07

**Student Name:** Harshit Shandilya

**Branch:** BE-IT

**Semester:** 06<sup>th</sup>

**Subject Name:** Advanced Programming-II

**UID:** 22BET10274

**Section/Group:** BET\_701/A

**Date of Performance:** 21-03-2025

**Subject Code:** 22ITP-351

### 1. Problem: Climbing Stairs

```
class Solution {  
public:  
    int climbStairs(int n) {  
        if (n == 1) return 1;  
        if (n == 2) return 2;  
  
        int prev1 = 1, prev2 = 2;  
        for (int i = 3; i <= n; i++) {  
            int curr = prev1 + prev2;  
            prev1 = prev2;  
            prev2 = curr;  
        }  
        return prev2;  
    }  
};
```

The screenshot shows a C++ IDE with the following code in the editor:

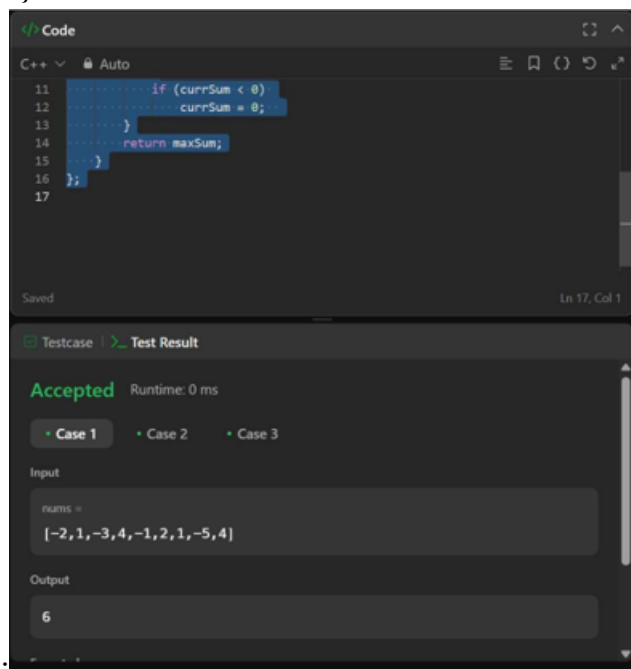
```
11     prev2 = curr;  
12 }  
13 return prev2;  
14 }  
15 };  
16
```

Below the editor, the 'Test Result' tab is active, showing:

- Status: **Accepted** (Runtime: 0 ms)
- Test Cases: Case 1, Case 2
- Input: n = 2
- Output: 2

## 2. Problem: Maximum Subarray

```
class Solution {  
public:  
    int maxSubArray(vector<int>& nums) {  
        int maxSum = nums[0];  
        int currSum = 0;  
        for (int num : nums) {  
            currSum += num;  
            maxSum = max(maxSum, currSum);  
            if (currSum < 0)  
                currSum = 0;  
        }  
        return maxSum;  
    }  
};
```



The screenshot shows a C++ IDE with the following code in the editor:

```
11         if (currSum < 0)  
12             currSum = 0;  
13     }  
14     return maxSum;  
15 }  
16 }  
17
```

Below the code editor, the 'Test Result' section shows:

- Status: Accepted (Runtime: 0 ms)
- Test Cases: Case 1 (selected), Case 2, Case 3
- Input: nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]
- Output: 6

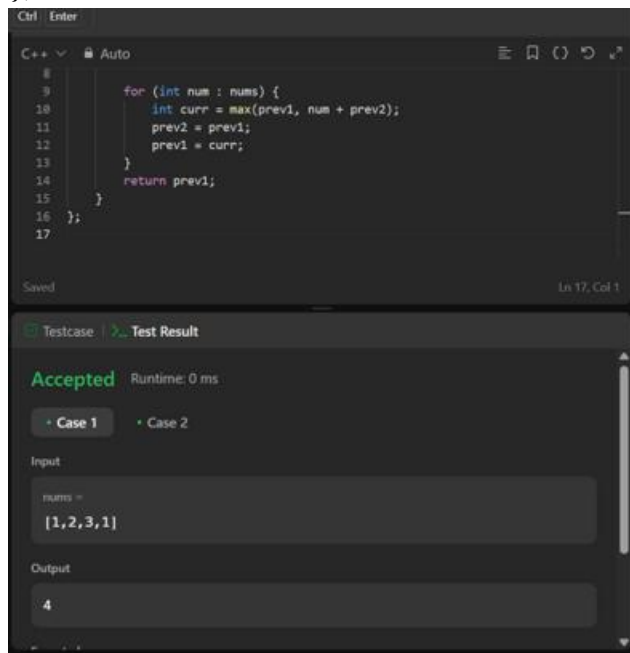
## 3. Problem: House Robber

```
class Solution {  
public:  
    int rob(vector<int>& nums) {  
        if (nums.empty()) return 0;
```

```

    if (nums.size() == 1) return nums[0];
    int prev2 = 0, prev1 = 0; // prev2 -> dp[i-2], prev1 -> dp[i-1]
    for (int num : nums) {
        int curr = max(prev1, num + prev2);
        prev2 = prev1;
        prev1 = curr;
    }
    return prev1;
}
};

```



The screenshot shows a C++ IDE with the following code:

```

8
9     for (int num : nums) {
10         int curr = max(prev1, num + prev2);
11         prev2 = prev1;
12         prev1 = curr;
13     }
14     return prev1;
15 }
16 };
17

```

Below the code editor, the test results are shown:

- Testcase: **Accepted** Runtime: 0 ms
- Case 1: **Accepted**
- Case 2: **Accepted**

Input:

```

nums =
[1,2,3,1]

```

Output:

```

4

```

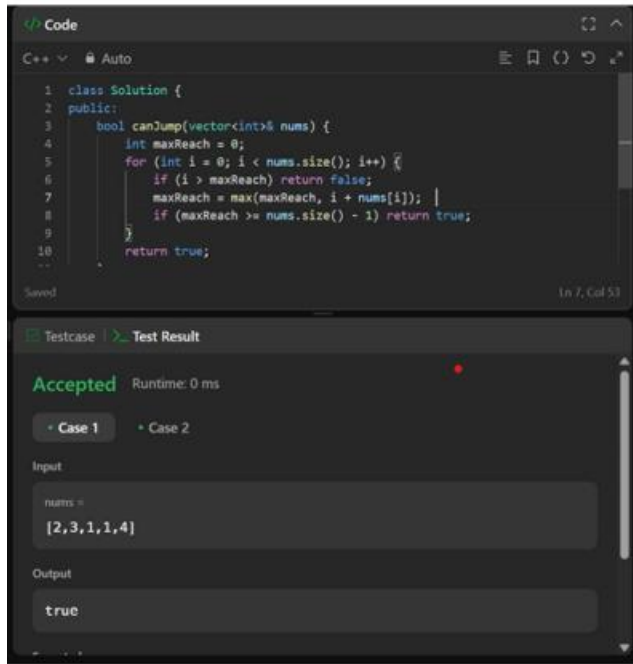
#### 4. Problem: Jump Game

```

class Solution {
public:
    bool canJump(vector<int>& nums) {
        int maxReach = 0;
        for (int i = 0; i < nums.size(); i++) {
            if (i > maxReach) return false;
            maxReach = max(maxReach, i + nums[i]);
            if (maxReach >= nums.size() - 1) return true;
        }
        return true;
    }
};

```

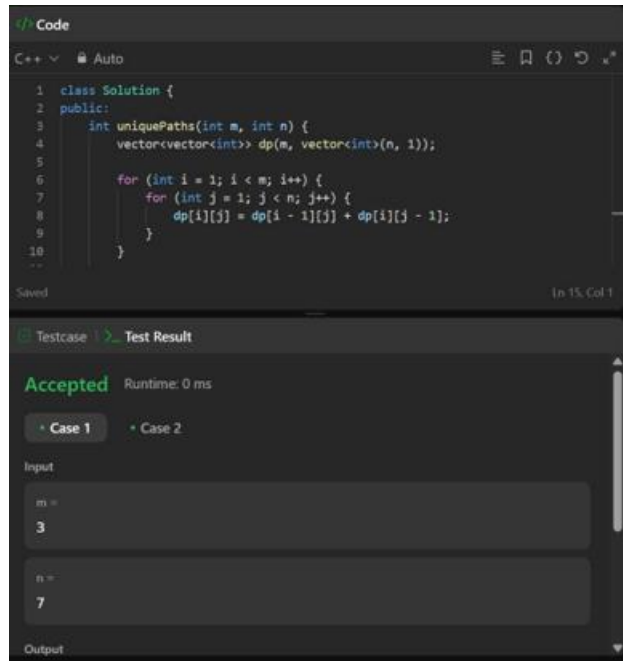
```
    }  
};
```



```
Code  
C++ v Auto  
1 class Solution {  
2 public:  
3     bool canJump(vector<int>& nums) {  
4         int maxReach = 0;  
5         for (int i = 0; i < nums.size(); i++) {  
6             if (i > maxReach) return false;  
7             maxReach = max(maxReach, i + nums[i]);  
8             if (maxReach >= nums.size() - 1) return true;  
9         }  
10        return true;  
11    }  
12};  
Saved In 7, Col 53  
Testcase Test Result  
Accepted Runtime: 0 ms  
Case 1 Case 2  
Input  
nums =  
[2,3,1,1,4]  
Output  
true
```

## 5. Problem: Unique Paths

```
class Solution {  
public:  
    int uniquePaths(int m, int n) {  
        vector<vector<int>> dp(m, vector<int>(n, 1));  
  
        for (int i = 1; i < m; i++) {  
            for (int j = 1; j < n; j++) {  
                dp[i][j] = dp[i - 1][j] + dp[i][j - 1];  
            }  
        }  
  
        return dp[m - 1][n - 1];  
    }  
};
```



The screenshot shows a C++ IDE with a code editor and a test result panel. The code editor contains the following C++ code:

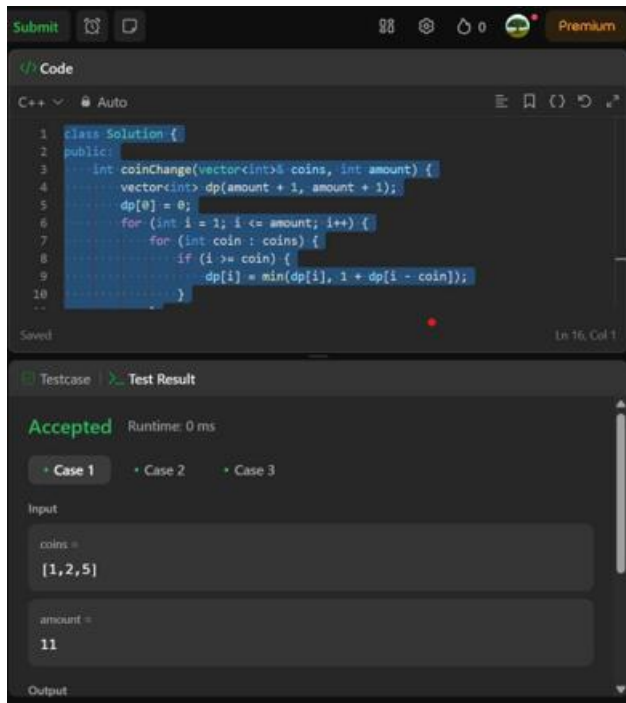
```
1 class Solution {
2 public:
3     int uniquePaths(int m, int n) {
4         vector<vector<int>> dp(m, vector<int>(n, 1));
5
6         for (int i = 1; i < m; i++) {
7             for (int j = 1; j < n; j++) {
8                 dp[i][j] = dp[i - 1][j] + dp[i][j - 1];
9             }
10        }
11    }
12 }
```

The test result panel shows the following information:

- Testcase 1: Test Result
- Accepted Runtime: 0 ms
- Case 1: Case 2
- Input: m = 3, n = 7
- Output:

## 6. Problem: Coin Change

```
class Solution {
public:
    int coinChange(vector<int>& coins, int amount) {
        vector<int> dp(amount + 1, amount + 1);
        dp[0] = 0;
        for (int i = 1; i <= amount; i++) {
            for (int coin : coins) {
                if (i >= coin) {
                    dp[i] = min(dp[i], 1 + dp[i - coin]);
                }
            }
        }
        return (dp[amount] == amount + 1) ? -1 : dp[amount];
    }
};
```



```

1 class Solution {
2 public:
3     int coinChange(vector<int>& coins, int amount) {
4         vector<int> dp(amount + 1, amount + 1);
5         dp[0] = 0;
6         for (int i = 1; i <= amount; i++) {
7             for (int coin : coins) {
8                 if (i >= coin) {
9                     dp[i] = min(dp[i], 1 + dp[i - coin]);
10                }
11            }
12        }
13        return dp[amount] < amount + 1 ? dp[amount] : -1;
14    }
15 };
  
```

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

coins =  
[1, 2, 5]

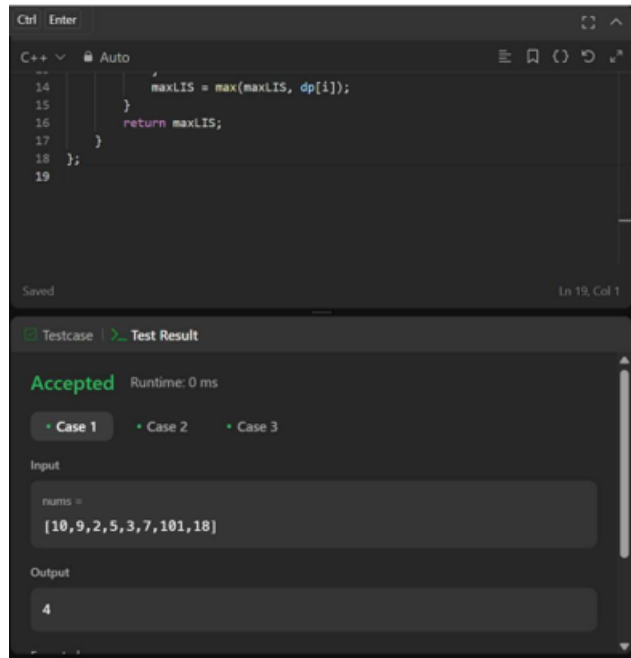
amount =  
11

Output

## 7. Problem: Longest Increasing Subsequence

```

class Solution {
public:
    int lengthOfLIS(vector<int>& nums) {
        int n = nums.size();
        vector<int> dp(n, 1);
        int maxLIS = 1;
        for (int i = 1; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (nums[j] < nums[i]) {
                    dp[i] = max(dp[i], dp[j] + 1);
                }
            }
            maxLIS = max(maxLIS, dp[i]);
        }
        return maxLIS;
    }
};
  
```



```
14         maxLIIS = max(maxLIIS, dp[i]);
15     }
16     return maxLIIS;
17 }
18 };
19
```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

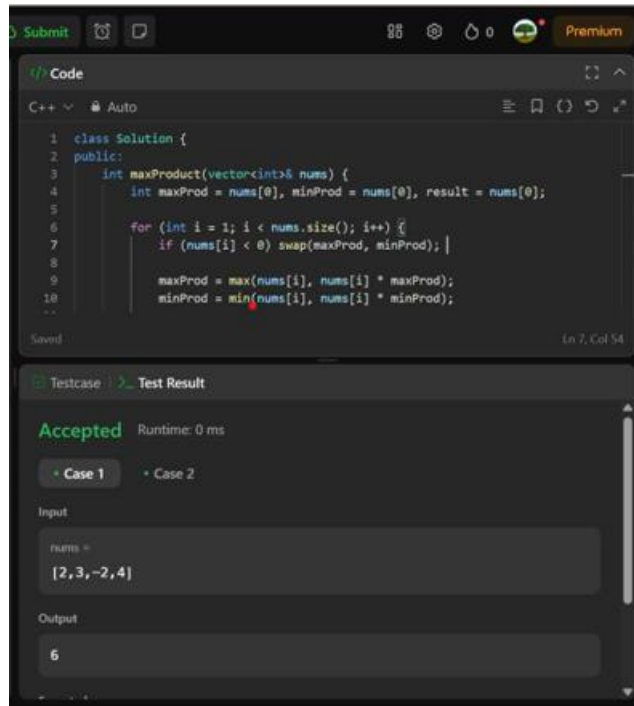
nums =  
[10,9,2,5,3,7,101,18]

Output

4

## 8. Problem: Maximum Product Subarray

```
class Solution {
public:
    int maxProduct(vector<int>& nums) {
        int maxProd = nums[0], minProd = nums[0], result = nums[0];
        for (int i = 1; i < nums.size(); i++) {
            if (nums[i] < 0) swap(maxProd, minProd);
            maxProd = max(nums[i], nums[i] * maxProd);
            minProd = min(nums[i], nums[i] * minProd);
            result = max(result, maxProd);
        }
        return result;
    }
};
```



```

1 class Solution {
2 public:
3     int maxProduct(vector<int>& nums) {
4         int maxProd = nums[0], minProd = nums[0], result = nums[0];
5
6         for (int i = 1; i < nums.size(); i++) {
7             if (nums[i] < 0) swap(maxProd, minProd);
8
9             maxProd = max(nums[i], nums[i] * maxProd);
10            minProd = min(nums[i], nums[i] * minProd);
11        }
12        return maxProd;
13    }
14 };

```

Testcase: Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =

[2, 3, -2, 4]

Output

6

## 9. Problem: Decode Ways

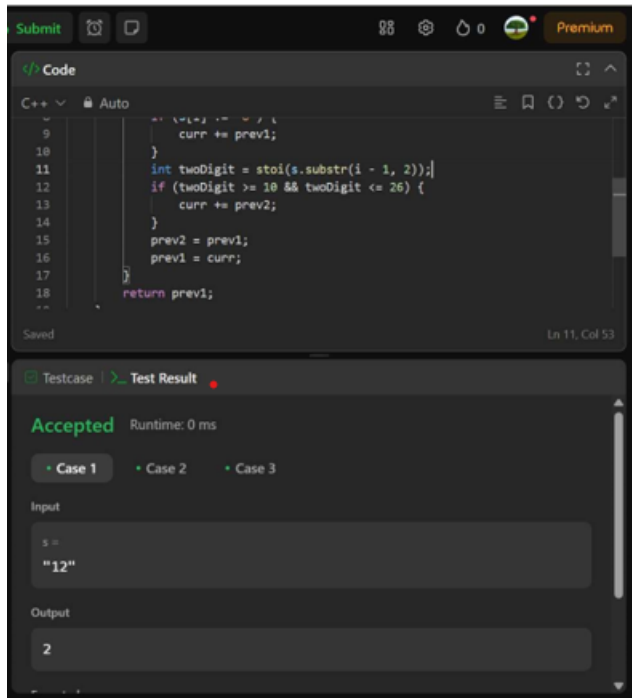
```

class Solution {
public:
    int numDecodings(string s) {
        if (s.empty() || s[0] == '0') return 0;
        int prev1 = 1, prev2 = 1;
        for (int i = 1; i < s.size(); i++) {
            int curr = 0;
            if (s[i] != '0') {
                curr += prev1;
            }
            int twoDigit = stoi(s.substr(i - 1, 2));
            if (twoDigit >= 10 && twoDigit <= 26) {
                curr += prev2;
            }
            prev2 = prev1;
            prev1 = curr;
        }
        return prev1;
    }
}

```



```
};
```



```

C++ v Auto
9      curr += prev1;
10     }
11     int twoDigit = stoi(s.substr(i - 1, 2));
12     if (twoDigit >= 10 && twoDigit <= 26) {
13         curr += prev2;
14     }
15     prev2 = prev1;
16     prev1 = curr;
17 }
18 return prev1;
19
Saved Ln 11, Col 53

Testcase Test Result
Accepted Runtime: 0 ms
• Case 1 • Case 2 • Case 3
Input
s =
"12"
Output
2

```

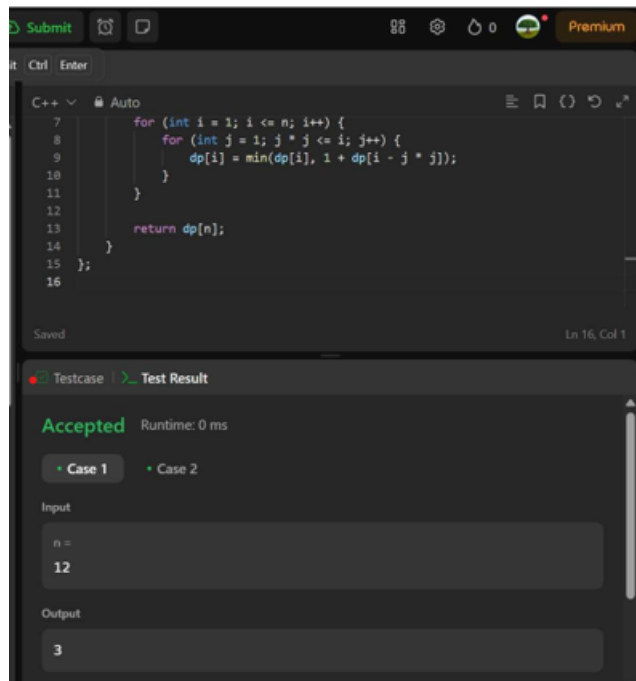
## 10. Problem: Perfect squares

```

class Solution {
public:
    int numSquares(int n) {
        vector<int> dp(n + 1, INT_MAX);
        dp[0] = 0;

        for (int i = 1; i <= n; i++) {
            for (int j = 1; j * j <= i; j++) {
                dp[i] = min(dp[i], 1 + dp[i - j * j]);
            }
        }
        return dp[n];
    }
};

```



```

7      for (int i = 1; i <= n; i++) {
8          for (int j = 1; j * j <= i; j++) {
9              dp[i] = min(dp[i], 1 + dp[i - j * j]);
10         }
11     }
12     return dp[n];
13 }
14 }
15 };
16

```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

n = 12

Output

3

## 11. Problem: Word Break

```

class Solution {
public:
    bool wordBreak(string s, vector<string>& wordDict) {
        unordered_set<string> wordSet(wordDict.begin(), wordDict.end());
        int n = s.size();
        vector<bool> dp(n + 1, false);
        dp[0] = true;
        for (int i = 1; i <= n; i++) {
            for (int j = 0; j < i; j++) {
                if (dp[j] && wordSet.find(s.substr(j, i - j)) != wordSet.end()) {
                    dp[i] = true;
                    break;
                }
            }
        }
        return dp[n];
    }
}

```



Discover. Learn. Empower.

```
Submit 88 60 100 Premium
Ctrl Enter
C++ Auto
1 class Solution {
2 public:
3     bool wordBreak(string s, vector<string>& wordDict) {
4         unordered_set<string> wordSet(wordDict.begin(), wordDict.end());
5         int n = s.size();
6         vector<bool> dp(n + 1, false);
7         dp[0] = true;
8         for (int i = 1; i <= n; i++) {
9             for (int j = 0; j < i; j++) {
10                 if (dp[j] && wordSet.find(s.substr(j, i - j)) != wordSet.end())
11                     dp[i] = true;
12             }
13         }
14         return dp[n];
15     }
16 };
Saved Ln 1, Col 1
Testcase Test Result
Accepted Runtime: 0 ms
• Case 1 • Case 2 • Case 3
Input
s =
"leetcode"
wordDict =
{"leet", "code"}
```