Experiment 7

Student Name: Sejjwal Yadav **UID:** 22BET10241
Branch: Information Technology **Section/Group:** 22BET_IOT-702/A
Semester: 6th **Subject Code:** 22ITP-351

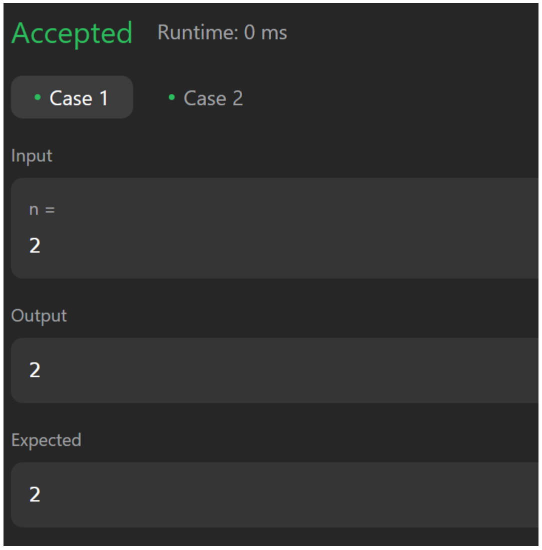
Problem: 1

Aim: Climbing Stairs

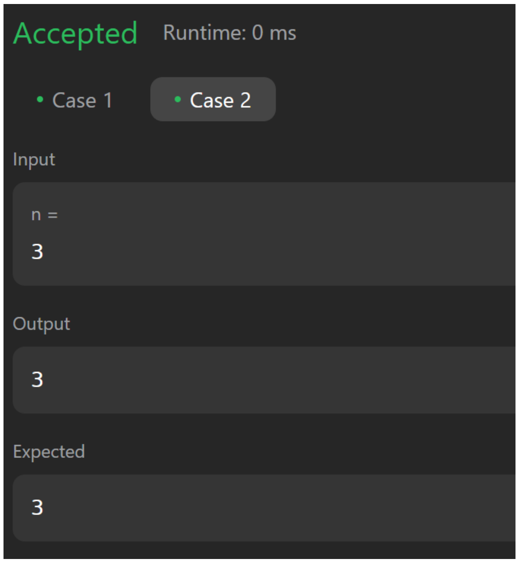
Code:

```
class Solution {
public:
    int climbStairs(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return climbStairs(n-1) +
climbStairs(n-2);
    }
};
```

Output:



```
Accepted   Runtime: 0 ms

• Case 1   • Case 2

Input
n =
2

Output
2

Expected
2
```

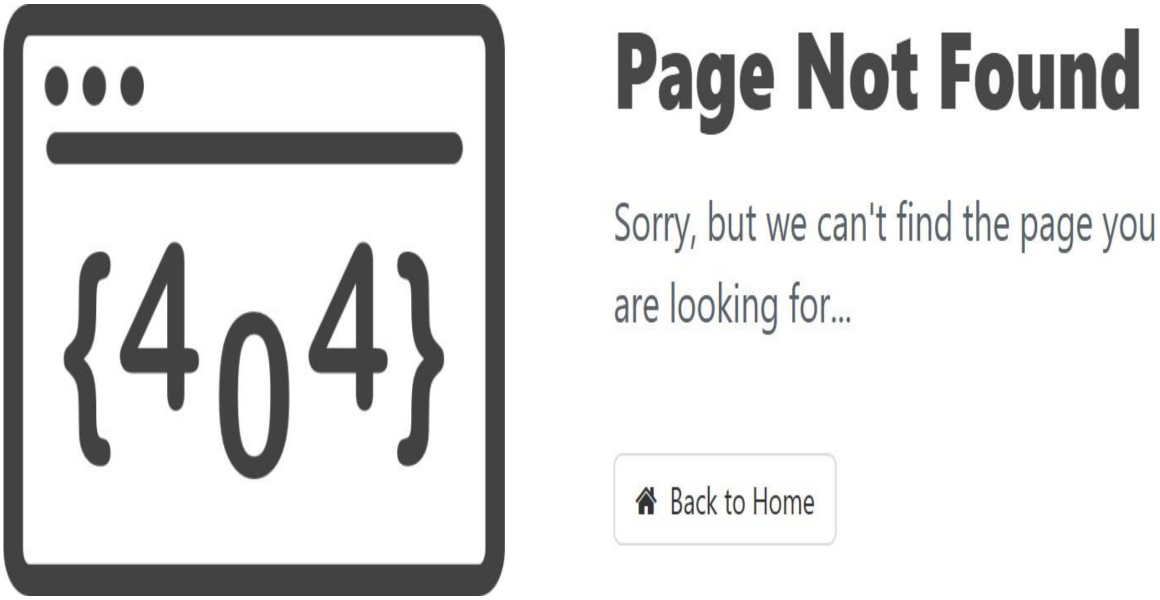


```
Accepted   Runtime: 0 ms

• Case 1   • Case 2

Input
n =
3

Output
3

Expected
3
```

<u>**Problem:**</u> **2**

Aim: Best Time to Buy and Sell a Stock

Code:



Page Not Found

Sorry, but we can't find the page you are looking for...

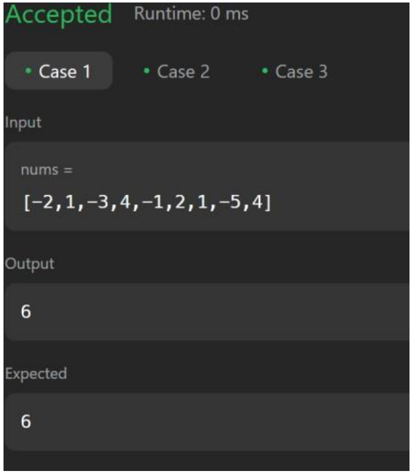
Problem: 3

Aim: Maximum Subarray

Code:

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int n = size(nums), ans = INT_MIN;
        for(int i = 0; i < n; i++)
            for(int j = i, curSum = 0; j < n ; j++)
                curSum += nums[j],
                ans = max(ans, curSum);
        return ans;
    }
};
```

Output:



Input

nums =

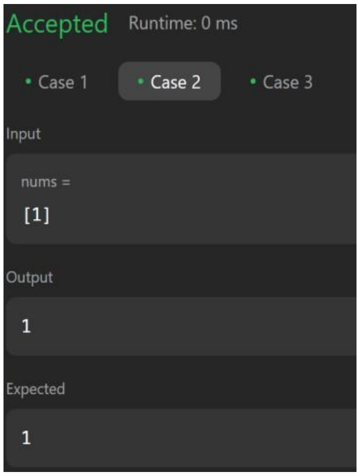
```
[-2,1,-3,4,-1,2,1,-5,4]
```

Output

```
6
```

Expected

```
6
```



Input

nums =

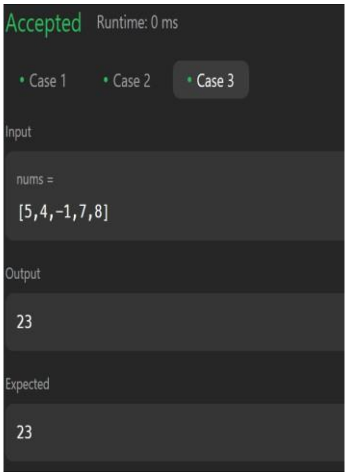
```
[1]
```

Output

```
1
```

Expected

```
1
```



Input

nums =

```
[5,4,-1,7,8]
```

Output

```
23
```

Expected

```
23
```

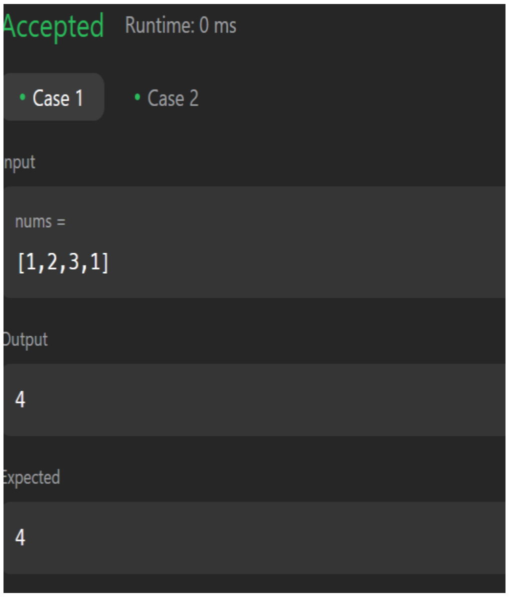
Problem: 4

Aim: House Robber

Code:

```
 class Solution {
public:
    int rob(vector<int>& nums) {
        int n = nums.size();

        if (n == 1) {
            return nums[0];
        }

        vector<int> dp(n, 0);

        dp[0] = nums[0];
        dp[1] = max(nums[0], nums[1]);

        for (int i = 2; i < n; i++) {
            dp[i] = max(dp[i - 1], nums[i] + dp[i - 2]);
        }

        return dp[n - 1];
    }
};
```

Output:



Accepted Runtime: 0 ms

nput

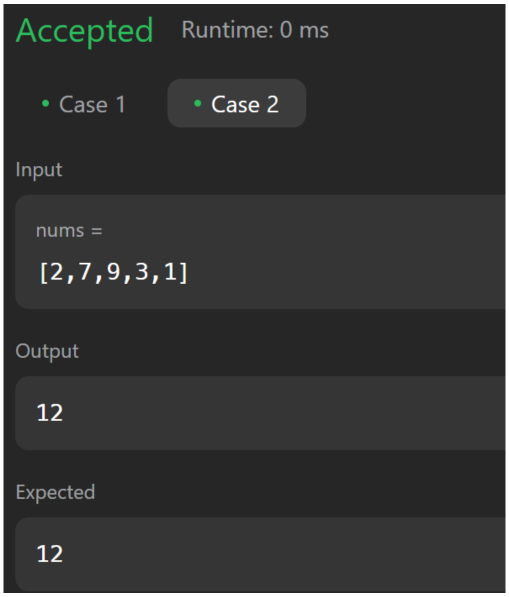
nums =
[1,2,3,1]

Dutput

4

Expected

4



Accepted Runtime: 0 ms

Input

nums =
[2,7,9,3,1]

Output

12

Expected

12

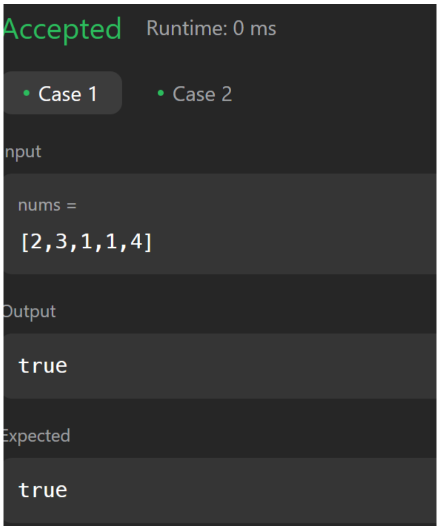
Problem :5

Aim: Jump Game

Code:

```
#include <vector> class Solution {
public:
    bool canJump(vector<int>& nums) {
        int goal = nums.size() - 1;

        for (int i = nums.size() - 2; i >= 0; i--) {
            if (i + nums[i] >= goal) {
                goal = i;
            }
        }

        return goal == 0;
    }
};
```

Output:



Accepted Runtime: 0 ms

nput

nums =

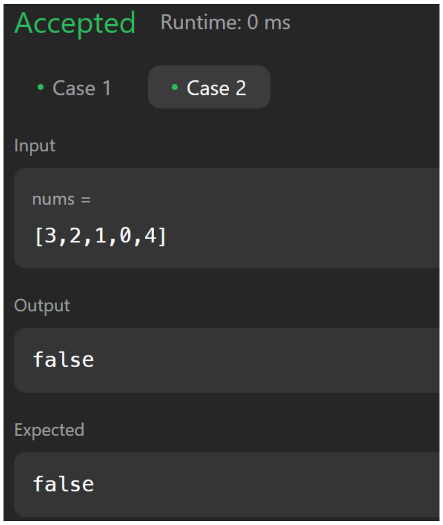
[2,3,1,1,4]

Output

true

Expected

true



Accepted Runtime: 0 ms

Input

nums =

[3,2,1,0,4]

Output

false

Expected

false

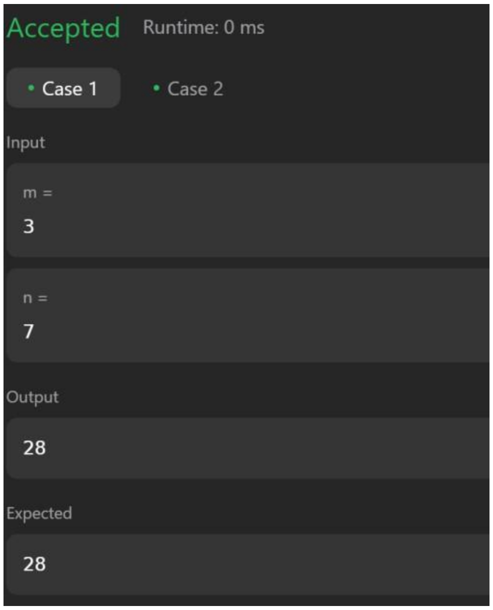
Problem:6

Aim: Unique Paths

Code:

```
class Solution {
public:
    int uniquePaths(int m, int n) {
        std::vector<int> aboveRow(n, 1);

        for (int row = 1; row < m; row++) {
            std::vector<int> currentRow(n, 1);
            for (int col = 1; col < n; col++) {
                currentRow[col] = currentRow[col - 1] + aboveRow[col];
            }
            aboveRow = currentRow;
        }

        return aboveRow[n - 1];
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

m =
3

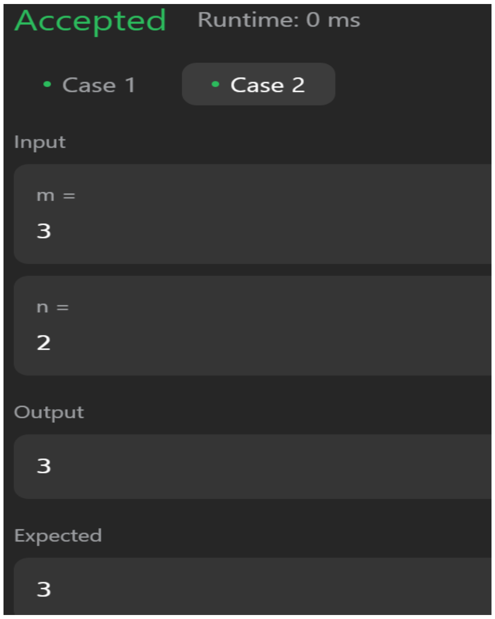
n =
7

Output

28

Expected

28



Accepted Runtime: 0 ms

Input

m =
3

n =
2

Output

3

Expected

3

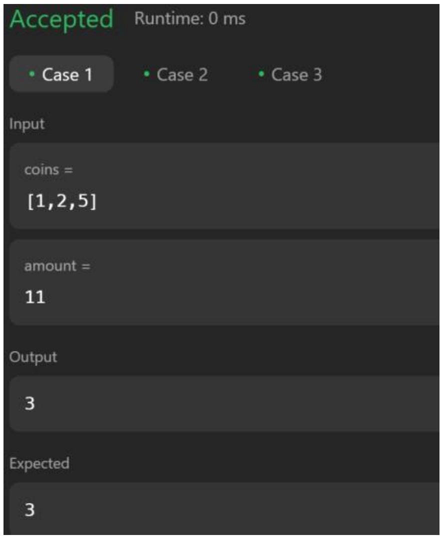
Problem:7

Aim: Coin Change

Code:

```
class Solution {
public:
    int coinChange(vector<int>& coins, int amount) {
        vector<int> minCoins(amount + 1, amount + 1);
        minCoins[0] = 0;

        for (int i = 1; i <= amount; i++) {
            for (int j = 0; j < coins.size(); j++) {
                if (i - coins[j] >= 0) {
                    minCoins[i] = min(minCoins[i], 1 +
minCoins[i - coins[j]]);
                }
            }
        }

        return minCoins[amount] != amount + 1 ?
minCoins[amount] : -1;
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

coins =
[1,2,5]

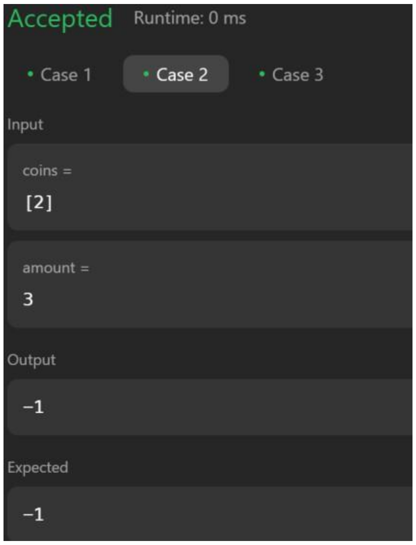
amount =
11

Output

3

Expected

3



Accepted Runtime: 0 ms

Input

coins =
[2]

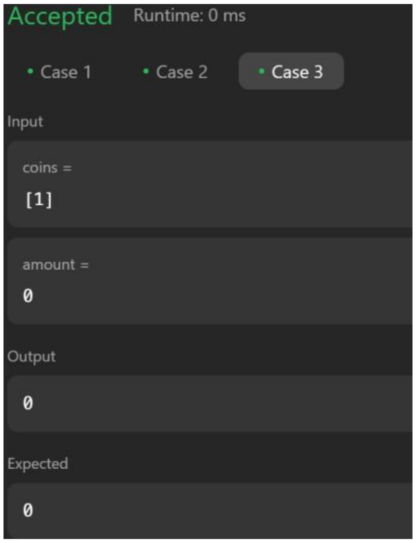
amount =
3

Output

-1

Expected

-1



Accepted Runtime: 0 ms

Input

coins =
[1]

amount =
0

Output

0

Expected

0

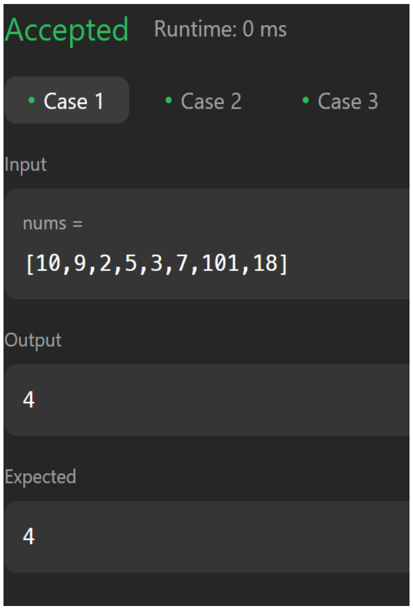
Problem:8

Aim: Longest Increasing Subsequence

Code:

```
class Solution {
public:
    int lengthOfLIS(vector<int>& nums) {
        vector<int> res;

        for (int n : nums) {
            if (res.empty() || res.back() < n) {
                res.push_back(n);
            } else {
                int idx = binarySearch(res, n);
                res[idx] = n;
            }
        }

        return res.size();
    }

private:
    int binarySearch(const vector<int>& arr, int target) {
        int left = 0;
        int right = arr.size() - 1;

        while (left <= right) {
            int mid = (left + right) / 2;
            if (arr[mid] == target) {
                return mid;
            } else if (arr[mid] > target) {
                right = mid - 1;
            } else {
                left = mid + 1;
            }
        }

        return left;
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

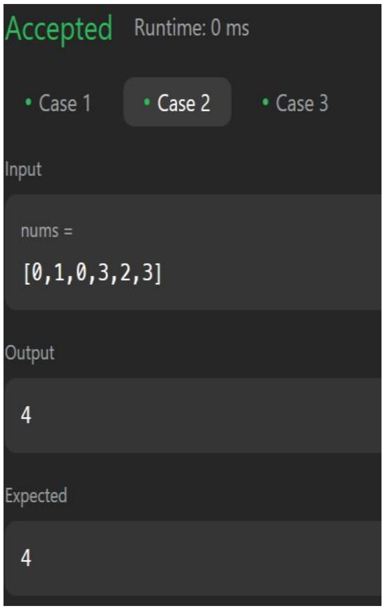
nums =
[10,9,2,5,3,7,101,18]

Output

4

Expected

4



Accepted Runtime: 0 ms

Input

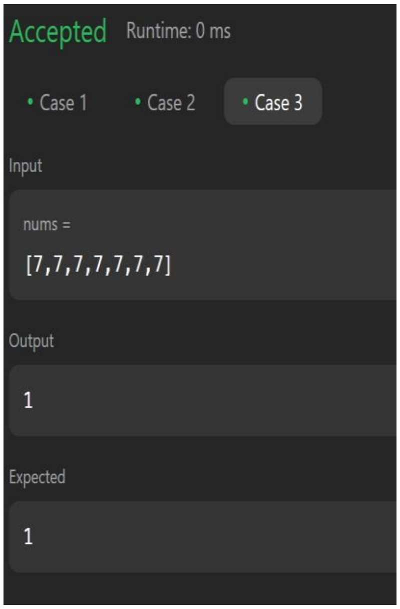
nums =
[0,1,0,3,2,3]

Output

4

Expected

4



Accepted Runtime: 0 ms

Input

nums =
[7,7,7,7,7,7,7]

Output

1

Expected

1

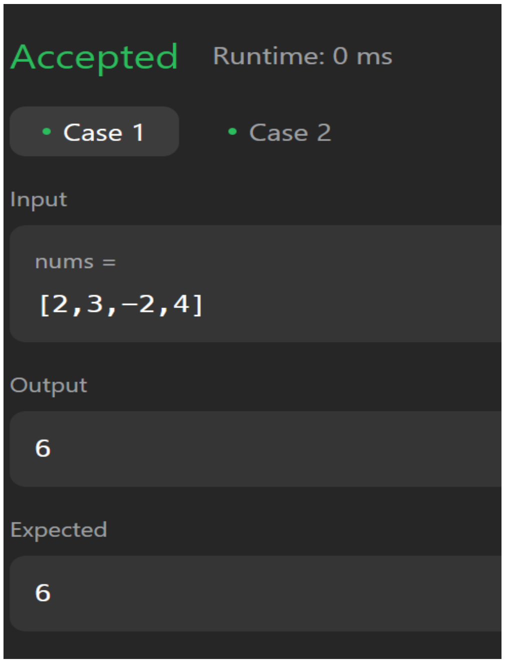
Problem:9

Aim: Maximum Product Subarray

Code:

```
class Solution {
public:
    int maxProduct(vector<int>& nums) {
        int res = *max_element(nums.begin(), nums.end());
        int curMax = 1, curMin = 1;

        for (int n : nums) {
            int temp = curMax * n;
            curMax = max({temp, curMin * n, n});
            curMin = min({temp, curMin * n, n});

            res = max(res, curMax);
        }

        return res;
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

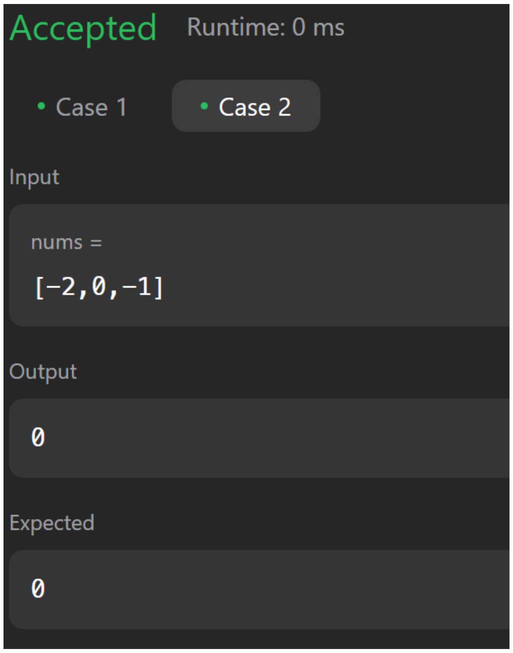
```
nums =
[2,3,-2,4]
```

Output

```
6
```

Expected

```
6
```



Accepted Runtime: 0 ms

Input

```
nums =
[-2,0,-1]
```

Output

```
0
```

Expected

```
0
```

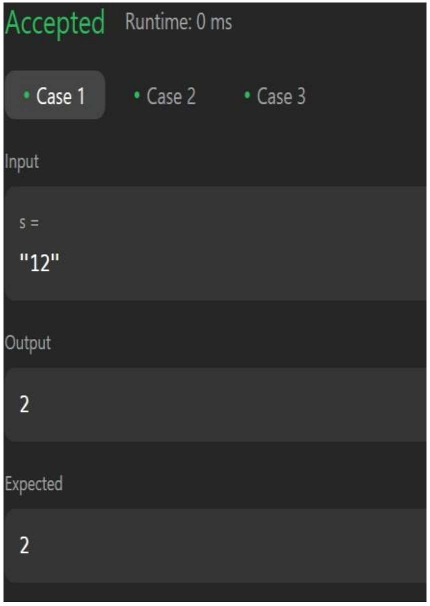
Problem: 10

Aim: Decode Ways

Code:

```
class Solution {
public:
    int numDecodings(std::string s) {
        if (s.empty() || s[0] == '0') {
            return 0;
        }

        int n = s.length();
        std::vector<int> dp(n + 1, 0);
        dp[0] = 1;
        dp[1] = 1;

        for (int i = 2; i <= n; ++i) {
            int oneDigit = s[i - 1] - '0';
            int twoDigits = std::stoi(s.substr(i - 2, 2));

            if (oneDigit != 0) {
                dp[i] += dp[i - 1];
            }

            if (10 <= twoDigits && twoDigits <= 26) {
                dp[i] += dp[i - 2];
            }
        }
        return dp[n];
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

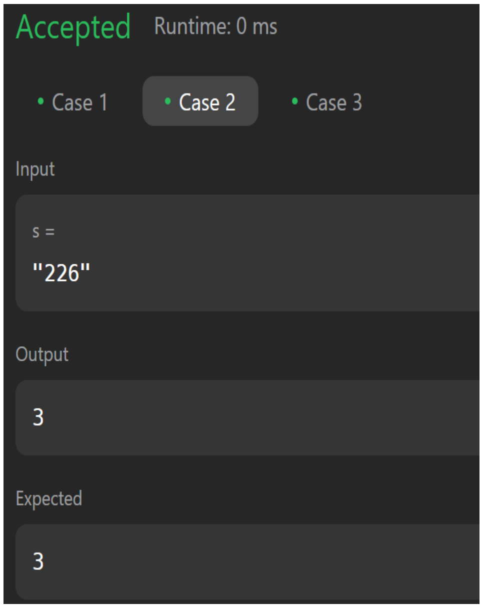
s =
"12"

Output

2

Expected

2



Accepted Runtime: 0 ms

Input

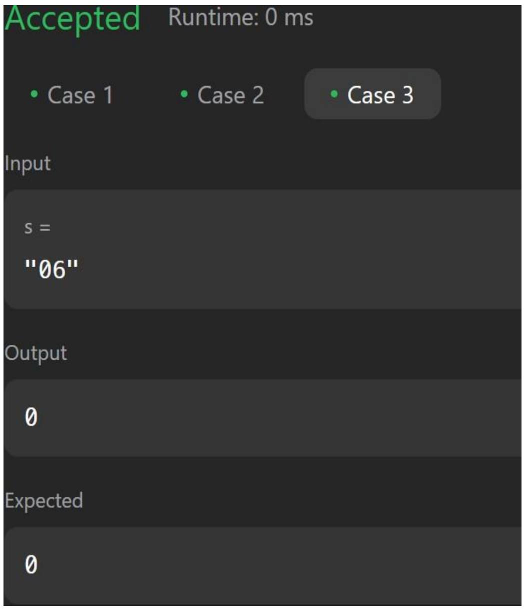
s =
"226"

Output

3

Expected

3



Accepted Runtime: 0 ms

Input

s =
"06"

Output

0

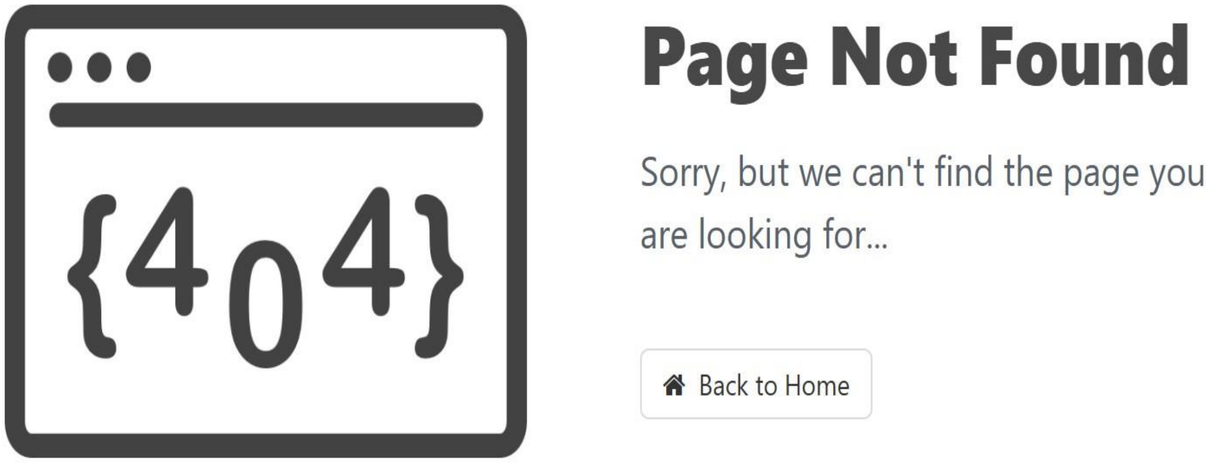
Expected

0

Problem: 11

Aim: Best time to buy and Sell a Stock with Cooldown

Code:



Page Not Found

Sorry, but we can't find the page you are looking for...

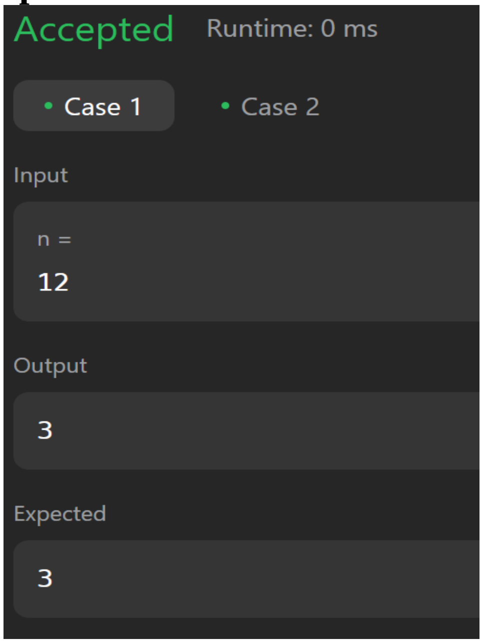
<u>Problem: 12</u>

Aim: Perfect Squares

Code:

```
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], dp[i - j * j] + 1);
            }
        }
        return dp[n];
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

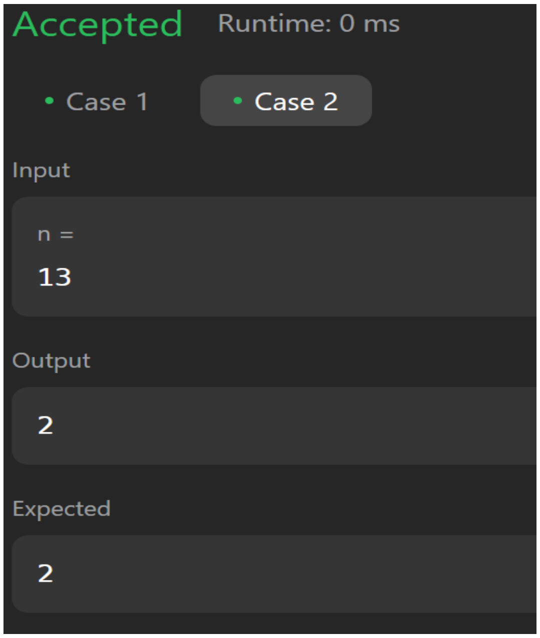
n =
12

Output

3

Expected

3



Accepted Runtime: 0 ms

Input

n =
13

Output

2

Expected

2

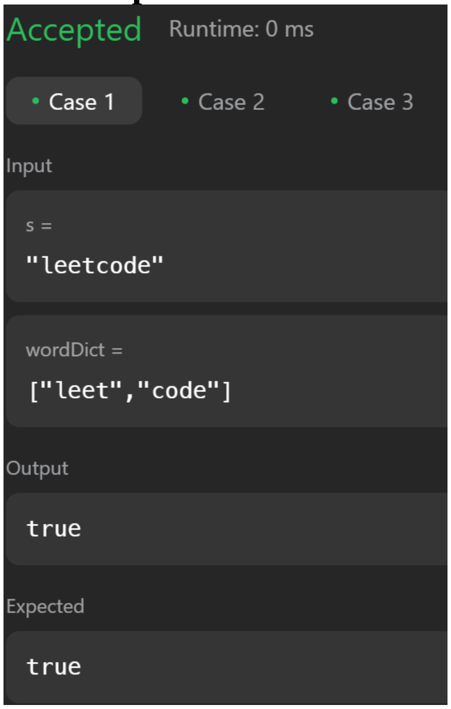
Problem: 13

Aim: Word Break

Code:

```
class Solution {
public:
    bool wordBreak(string s, vector<string>& wordDict) {
        vector<bool> dp(s.size() + 1, false);
        dp[0] = true;

        for (int i = 1; i <= s.size(); i++) {
            for (const string& w : wordDict) {
                int start = i - w.length();
                if (start >= 0 && dp[start] && s.substr(start, w.length()) == w) {
                    dp[i] = true;
                    break;
                }
            }
        }
        return dp[s.size()];
    }
};
```

Output:



Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

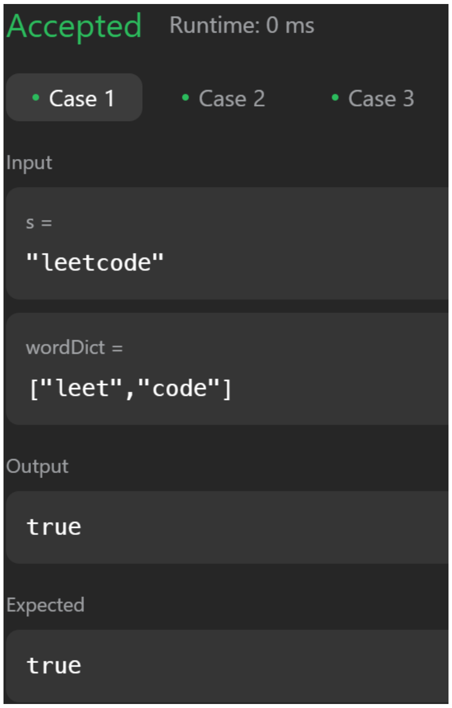
```
s =
"leetcode"
wordDict =
["leet","code"]
```

Output

```
true
```

Expected

```
true
```



Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

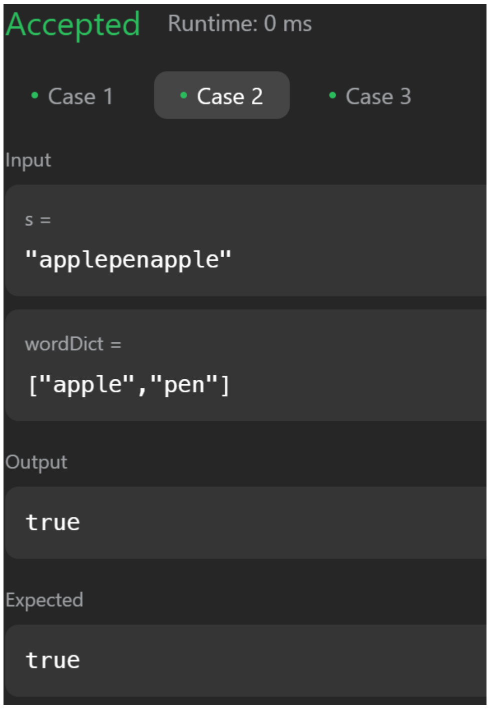
```
s =
"leetcode"
wordDict =
["leet","code"]
```

Output

```
true
```

Expected

```
true
```



Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

```
s =
"applepenapple"
wordDict =
["apple","pen"]
```

Output

```
true
```

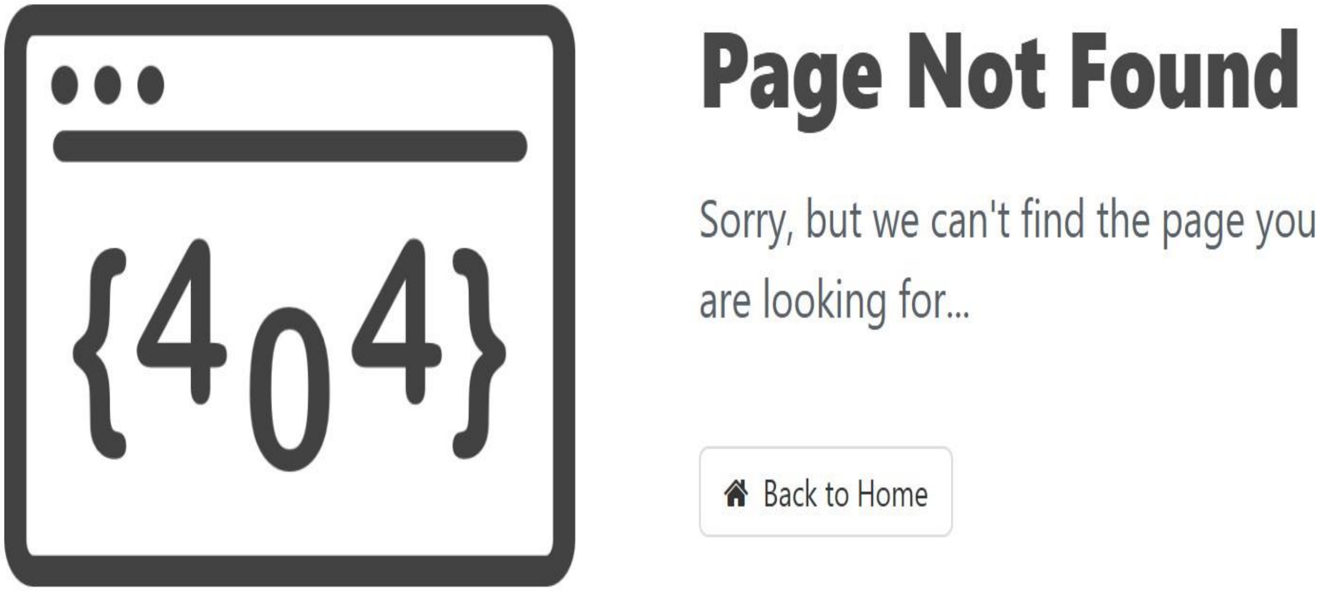
Expected

```
true
```

Problem: 14

Aim: Word Break 2

Code:



Page Not Found

Sorry, but we can't find the page you are looking for...