



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 6

Student Name: Ankita

UID: 22BCS17162

Branch: CSE

Section/Group: IOT_NTPP_602-A

Semester: 6th

Date of Performance: 20-02-25

Subject Name: AP- 2

Subject Code: 22CSP-351

Aim:

- a) Climbing Stairs
- b) Coin Change
- c) House Robbers

Objective: To learn about dynamic programming.

Code:

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

b)

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

c)

```
class Solution {
public:
int rob(vector<int>& nums) {
int n=nums.size();
if(n==0) return 0;
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],dp[i-2]+nums[i]);
}
return dp[n-1];
}
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

a)

☒ Testcase | [Test Result](#)

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

n =
2

Output

2

Expected

2

b)

☒ Testcase | [Test Result](#)

Accepted Runtime: 0 ms

- Case 1
- Case 2
- Case 3

Input

coins =
[1,2,5]

amount =
11

Output

3

Expected

3

c)

☒ Testcase | [Test Result](#)

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

nums =
[1,2,3,1]

Output

4

Expected

4

Learning Outcomes:

- Understand the concept of dynamic programming.
- Learnt about different problem like coin change, climbing stairs.
- Gain an understanding about the efficiency of dynamic programming.