Dynamic Programming Assignment 2 done by N S K K K Naga Jayanth

https://www.interviewbit.com/problems/unique-paths-in-a-grid/

```
public class Solution {
    public int uniquePathsWithObstacles(ArrayList<ArrayList<Integer>> A) {
        int m = A.size();
        int n = A.get(0).size();

        if (A.get(0).get(0) == 1) {
            return 0;
        }



        int[][] dp = new int[m][n];

        dp[0][0] = 1;

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (A.get(i).get(j) == 1) {
                    dp[i][j] = 0;
                } else {

                    if (i > 0) {
                        dp[i][j] += dp[i - 1][j];
                    }
                    if (j > 0) {
                        dp[i][j] += dp[i][j - 1];
                    }
                }
            }
        }
```

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

https://leetcode.com/problems/house-robber/

```
class Solution {
  public int rob(int[] nums) {
    int n = nums.length;
    if (n == 0)
      return 0;
    if (n == 1)
      return nums[0];
    int[] max_val = new int[n];
    max_val[0] = nums[0];
    max_val[1] = Math.max(nums[0], nums[1]);

    for (int i = 2; i < n; i++)
      max_val[i] = Math.max(max_val[i - 1], max_val[i - 2] + nums[i]);

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

https://practice.geeksforgeeks.org/problems/stickler-theif-1587115621/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

```
class Solution
{
    //Function to find the maximum money the thief can get.
    public int FindMaxSum(int arr[], int n)
    {
    return DFS(arr, arr.length-1);
```

```
    }
    public int DFS(int [] nums, int index)
    {
        if(index  < 0) return 0;
        return Math.max(DFS(nums, index - 2) + nums[index], DFS(nums, index -1));
    }
}
```