**Stickler Thief: -**

Medium Accuracy: 37.98% Submissions: 189K+ Points: 4

Stickler the thief wants to loot money from a societyhaving**n** houses in a single line. He is a weird person and follows a certain **rule**when looting the houses. According to the rule, he will **never loot two consecutive houses**. At the same time, he wants to **maximize**the amount he loots. The thief knows which house has what amount of money but is unable to come up with an optimal looting strategy. He asks for your help to find the **maximum money** he can get if he strictly follows the rule. ith house has **a[i]**amount of money present in it.

**Example 1:**

**Input:**

n = 5

a[] = {6,5,5,7,4}

**Output:**15

**Explanation:**Maximum amount he can get by looting 1st, 3rd and 5th house. Which is 6+5+4=15.

**Example 2:**

**Input:**

n = 3

a[] = {1,5,3}

**Output:**5

**Explanation:**Loot only 2nd house and get maximum amount of 5.

**Your Task:**  
Complete the function**FindMaxSum()**which takes an array **arr[]** and **n** as input which returns the maximum money he can get following the rules.

**Expected Time Complexity:**O(N).  
**Expected Space Complexity:**O(N).

**Constraints:**  
1 ≤ n ≤ 105  
1 ≤ a[i] ≤ 104

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

typedef long long int ll;

// } Driver Code Ends

class Solution

{

public:

int helper(int arr[], int n, int start, vector<int> &dp){

// base case

if(start == n-1)

return dp[start] = arr[start];

else if(start >= n)

return 0;

// dp found case

if(dp[start] != -1)

return dp[start];

// recursive case

int pick, notpick;

pick = arr[start] + helper(arr, n, start+2, dp);

notpick = helper(arr, n, start+1, dp);

// return from current state

return dp[start] = max(pick, notpick);

}

//Function to find the maximum money the thief can get.

int FindMaxSum(int arr[], int n){

vector<int> dp(n+1, -1);

return helper(arr, n, 0, dp);

}

};

//{ Driver Code Starts.

int main()

{

//taking total testcases

int t;

cin>>t;

while(t--)

{

//taking number of houses

int n;

cin>>n;

int a[n];

//inserting money of each house in the array

for(int i=0;i<n;++i)

cin>>a[i];

Solution ob;

//calling function FindMaxSum()

cout<<ob.FindMaxSum(a,n)<<endl;

}

return 0;

}

// } Driver Code Ends