**Day-1**

**Problem1**- [**121. Best Time to Buy and Sell Stock**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock/)

Solution-

class Solution {

public:

    int maxProfit(vector<int>& prices) {

        int n=prices.size();

        int p=0;

        int mini=prices[0];

        for(int i=1;i<n;i++)

        {

            int diff=prices[i]-mini;

            p=max(diff,p);

            mini=min(mini,prices[i]);

        }

        return p;

    }

};

**Problem2**- **Chocolate Distribution Problem**

Solution-

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

long long findMinDiff(vector<long long> a, long long n, long long m){

//code

sort(a.begin(),a.end());

int i=0;

int mini=INT\_MAX;

while(i+m-1<n)

{

int diff=(a[m+i-1]-a[i]);

mini=min(mini,diff);

i++;

}

return mini;

}

};

**Problem 3**- [**1. Two Sum**](https://leetcode.com/problems/two-sum/)

class Solution {

public:

    vector<int> twoSum(vector<int>& nums, int target) {

        map<int,int>mpp;

        int n=nums.size();

        for(int i=0;i<n;i++)

        {

            int num=nums[i];

            int t=target-num;

            if(mpp.find(t)!=mpp.end())

            {

                return { mpp[t],i};

            }

            mpp[num]=i;

        }

         return {-1,-1};

    }

};

**Problem 4**- [**122. Best Time to Buy and Sell Stock II**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/)

class Solution {

public:

int f(int i,int buy,vector<int>& p,vector<vector<int>>&dp)

{

    int n=p.size();

    if(i==n)return 0;

    if(dp[i][buy]!=-1)

    return dp[i][buy];

    int gain=0;

    if(buy)

    {

        int a=-p[i]+f(i+1,0,p,dp);

        int b=0+f(i+1,1,p,dp);

        gain=max(a,b);

    }

    else

    {

         int a=p[i]+f(i+1,1,p,dp);

        int b=0+f(i+1,0,p,dp);

        gain=max(a,b);

    }

    return dp[i][buy]=gain;

}

    int maxProfit(vector<int>& prices) {

        int n=prices.size();

        vector<vector<int>>dp(n,vector<int>(2,-1));

        return f(0,1,prices,dp);

    }

};