**Question 1 : Leader in the Array Given a unsorted array, kindly find the leader in array . An element is called the leader of an array if there is no element greater than it on the right side.**

**ANSWER:-**

**#include <iostream>**

**#include <vector>**

**#include <unordered\_set>**

**template<size\_t n>**

**void leader(int (&arr)[n]) {**

**std::vector<int> leaders;**

**std::unordered\_set<int> start;**

**int right = arr[n - 1];**

**for (int i = n - 1; i >= 0; i--) {**

**if (start.find(arr[i]) == start.end()) {**

**if (arr[i] >= right) {**

**leaders.push\_back(arr[i]);**

**right = arr[i];**

**}**

**start.insert(arr[i]);**

**}**

**}**

**for (int i = leaders.size() - 1; i >= 0; --i) {**

**std::cout << leaders[i] << " ";**

**}**

**std::cout << std::endl;**

**}**

**int main() {**

**int arr[] = {7, 10, 4, 10, 6, 5, 2};**

**leader(arr);**

**return 0;**

**}**

**Question 2 : Best Time to Buy and Sell Stock You are given an array prices where prices[i] is the price of a given stock on the ith day. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.**

**ANSWER:-**

**#include <iostream>**

**using namespace std;**

**template<size\_t n>**

**int Stocks(int (&prices)[n])**

**{**

**int buy\_stocks = prices[0], Profits = 0;**

**for (int i = 1; i < n; i++) {**

**if (buy\_stocks > prices[i])**

**{**

**buy\_stocks = prices[i];**

**}**

**else if (prices[i] - buy\_stocks > Profits)**

**{**

**Profits = prices[i] - buy\_stocks;**

**}**

**}**

**return Profits;**

**}**

**int main()**

**{**

**int prices[] = { 7, 6, 4, 3, 1 };**

**int max\_profit = Stocks(prices);**

**cout << max\_profit << endl;**

**return 0;**

**}**

**Question 3:Sum of All Subset XOR Totals The XOR total of an array is defined as the bitwise XOR of all its elements, or 0 if the array is empty. For example, the XOR total of the array [2,5,6] is 2 XOR 5 XOR 6 = 1. Given an array nums, return the sum of all XOR totals for every subset of nums. Note: Subsets with the same elements should be counted multiple times. An array a is a subset of an array b if a can be obtained from b by deleting some (possibly zero) elements of b.**

**ANSWER:-**

**#include <bits/stdc++.h>**

**using namespace std;**

**template<size\_t n>**

**int rec(int(& arr)[n], int i = 0, int x = 0)**

**{**

**if (i == n)**

**return x;**

**int choice1 = rec(arr, i + 1, x ^ arr[i]);**

**int choice2 = rec(arr, i + 1, x);**

**return choice1 + choice2;**

**}**

**int main()**

**{**

**int arr[] = { 1, 5, 6 };**

**cout << rec(arr);**

**return 0;**

**}**