// C++ program for the above approach

#include <bits/stdc++.h>

using namespace std;

// Function to find the maximum profit

int maxProfit(int\* prices, int n)

{

int profit = 0, currentDay = n - 1;

// Start from the last day

while (currentDay > 0) {

int day = currentDay - 1;

// Traverse and keep adding the

// profit until a day with

// price of stock higher

// than currentDay is obtained

while (day >= 0

&& (prices[currentDay]

> prices[day])) {

profit += (prices[currentDay]

- prices[day]);

day--;

}

// Set this day as currentDay

// with maximum cost of stock

// currently

currentDay = day;

}

// Return the profit

return profit;

}

// Driver Code

int main()

{

// Given array of prices

int prices[] = { 2, 3, 5 };

int N = sizeof(prices) / sizeof(prices[0]);

// Function Call

cout << maxProfit(prices, N);

return 0;

}