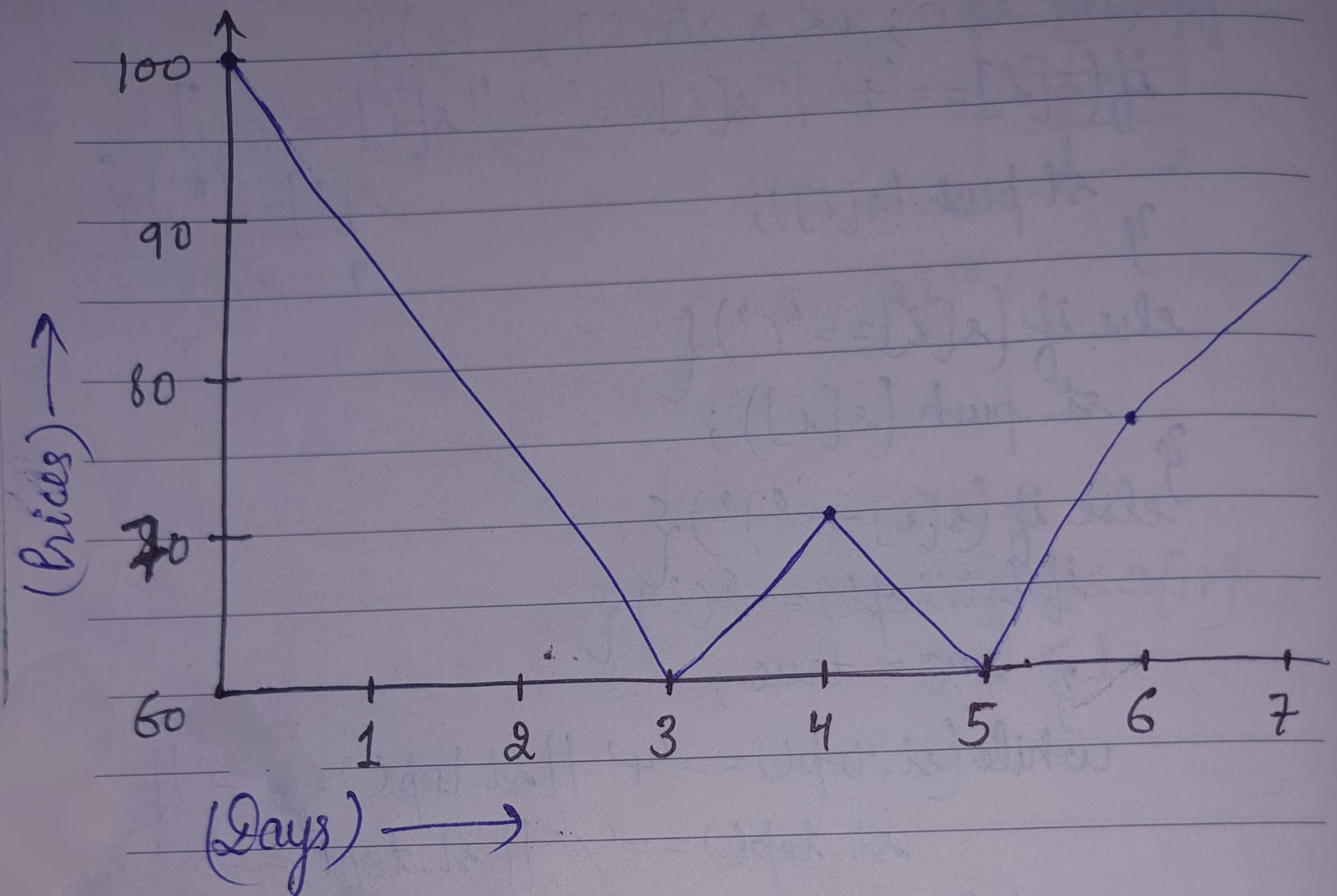
Stock Span Problem

Que The span of the stock's price today is defined as the maximum number of consecutive days (starting from today and going backwards) for which the price of the stocks was less than or equal to today's price. Find the span of the stocks for all the days.

Input : Array : [100, 80, 60, 70, 60, 75, 85]



<u>Brute Force :-</u>

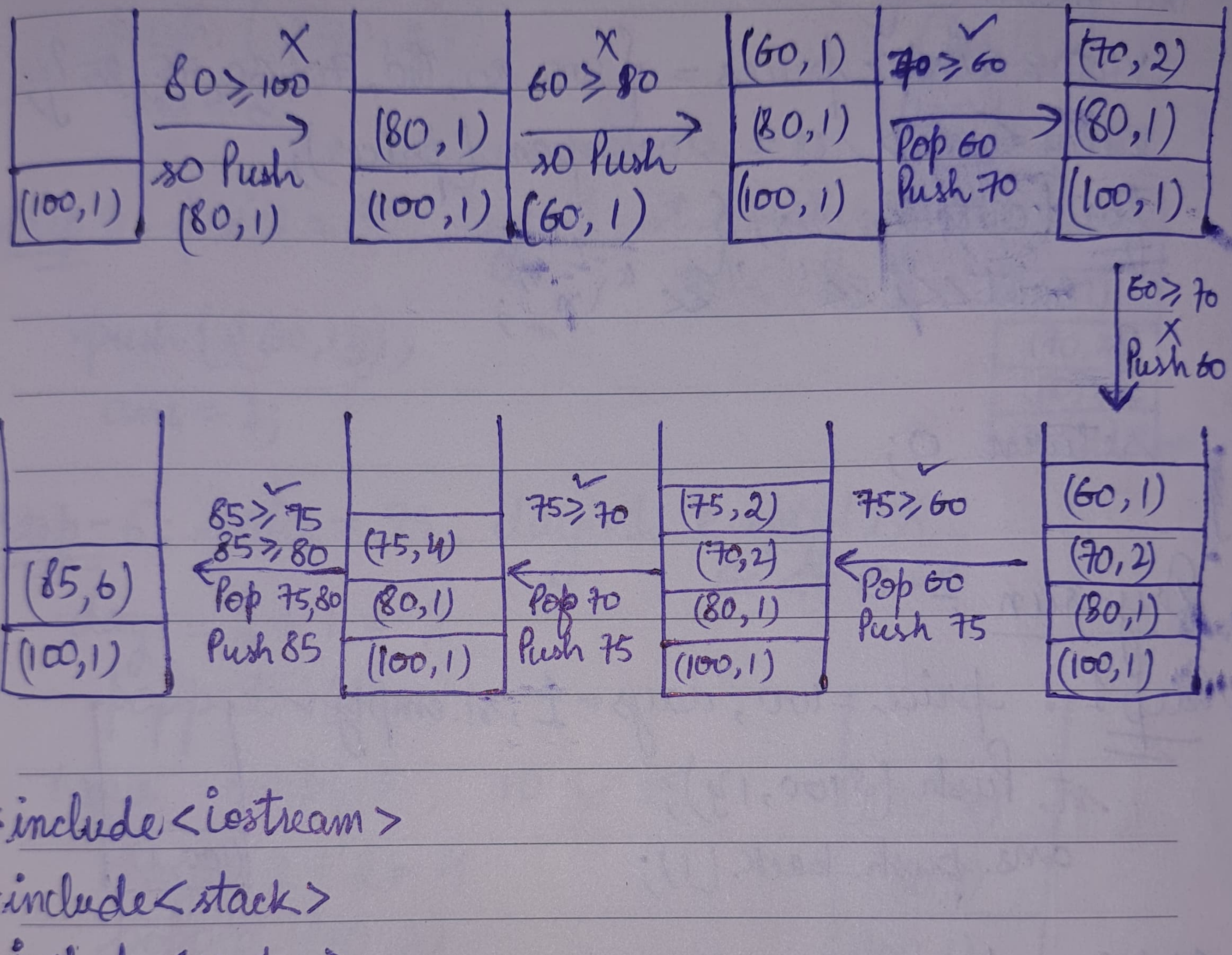
```
for (int i=0 ; i<n ; i++){
    int days = 0;
    for(int j=i ; j>=0 ; j--){
        if(prices[i] > = prices[j]){
            days++;
        }
        else{
            break;
    }   }
    //cout<< days ;
}
```

इसमें stack को pairs में बनाना है।

stack < pair <int, int> > representing {price, days}

Input : [100, 80, 60, 70, 60, 75, 85]

Ans : [1, 1, 1, 2, 1, 4, 6]



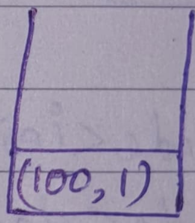
```
#include<iostream>
#include<stack>
#include<vector>
using namespace std;
vector<int> stockspan (vector <int> prices){
    vector<int > ans;
    stack<pair<int, int > > st;
    for(auto price : prices){
        int days = 1;
        while(!st.empty() && st.top(). first <= price){
                days += st.top(). second;
        }  st.pop();
        st.push({price, days});
      ans.push_back(days);
    }
    return ans;
}
int main(){
    vector <int> prices = {100, 80, 60, 70, 60, 75, 85};
    vector<int> res = stockspan(prices);
    for (auto i : res){
        cout<<i<<" "<< "\n";
    }
    return 0;
}
```

Dry run :-

Step 1 : price = 100, days = 1, st.empty ✓

st. Push ({100, 1});

ans. push_back (1);



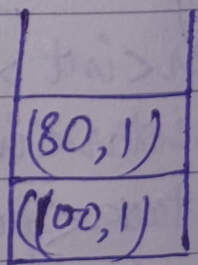
Step 2 : price = 80, days = 1, st.empty X

st.top().first <= price

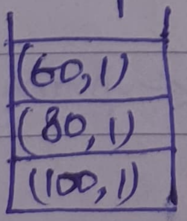
100 ≤ 80 X

so push ({80, 1});

ans. push_back (1);



Step-3 : price = 60, days = 1, st. empty() X,



~~st~~ 80 <= 60 X

push ({60, 1});

ans = 1

Step-4 : price = 70, days = 1, st.empty() X

60<=70 ✓

days = 1 + $\overset{\text{st.top.second}}{1}$ = 2

popped (60,1)

push ({70,2})

ans = 2

(70,2)
(80,1)
(100,1)

Step 5: price = 60, days = 1, st.empty() X

70<=80 X

push({60,1});

ans = 1;

(60,1)
(70,2)
(80,1)
(100,1)

Step-6: price = 75, days=1, empty() X

60 <= 75 ✓

days = 1+1 = 2

popped (60,1)

Now empty X, 70 <= 75

days = 2 + 2 = 4

popped (70,2)

Push ({75,4}); ans=4

(75, 4)
(80,1)
(100,1)

Step-7 : price = 85, days=1 empty() X and 75<=85 ✓

days = 1+4=5, popped (75,4), ~~ans = 5~~

Now, empty() X and 80<=85 ✓

days = 5+1 = 6, popped(80,1)

st. push ({85,6});

ans = 6

(85,6)
(100,1)