```
def leader_in_the_array(n : int,arr: [] )->[]:
  ans = []
  for i in range(n-1):
    currentmax = max(arr[i+1:])
    if arr[i] > currentmax:
      ans.append(arr[i])
  ans.append(arr[-1])
  return ans
testcases = int(input())
for i in range(testcases):
  n = int(input())
  arr = list(map(int,input().split()))
  print(leader_in_the_array(n,arr))
```

```
1
7
7 10 4 10 6 5 2
[10, 6, 5, 2]
```

```
def maxProfit(prices:[]) -> int:
        minimum = float('inf')
        maximum = float('-inf')
        price = []
        for i in prices:
            if i < minimum:
                minimum = i
                maximum = minimum
            elif i > maximum:
                maximum = i
            price.append(maximum - minimum)
        return max(price)
testcases = int(input())
for i in range(testcases):
  prices = list(map(int,input().split()))
  print(maxProfit(prices))
```

```
2
7 1 5 3 6 4
5
7 6 4 3 1
0
```

```
from itertools import combinations
def subsetXORSum(nums: []) -> int:
        total = 0
        for i in range(1,len(nums)+1):
            for j in combinations(nums,i):
                xor = 0
                for element in j:
                    xor ^= element
                total += xor
```

```
            return total
    testcases = int(input())
    for i in range(testcases):
      nums = list(map(int,input().split()))
      print(subsetXORSum(nums))
```

```
2
1 3
6
5 1 6
28
```

✓ 38s completed at 4:05 PM