We define subsequence as any subset of an array. We define a subarray as a contiguous subsequence in an array.

Given an array, find the maximum possible sum among:

1. all nonempty subarrays.
2. all nonempty subsequences.

Print the two values as space-separated integers on one line.

Note that empty subarrays/subsequences should not be considered.

Example

$arr = [-1, 2, 3, -4, 5, 10]$

The maximum subarray sum is comprised of elements at inidices $[1 - 5]$. Their sum is $2 + 3 + -4 + 5 + 10 = 16$. The maximum subsequence sum is comprised of elements at indices $[1, 2, 4, 5]$ and their sum is $2 + 3 + 5 + 10 = 20$.

Function Description

Complete the maxSubarray function in the editor below.

maxSubarray has the following parameter(s):

- int arr[n]: an array of integers

Returns

- int[2]: the maximum subarray and subsequence sums

Input Format

The first line of input contains a single integer t, the number of test cases.

The first line of each test case contains a single integer n.

The second line contains n space-separated integers $arr[i]$ where $0 \leq i < n$.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 10^5$
- $-10^4 \leq arr[i] \leq 10^4$

The subarray and subsequences you consider should have at least one element.

Sample Input

```
2
4
1 2 3 4
6
2 -1 2 3 4 -5
```

Sample Output

```
10 10
10 11
```

Explanation

In the first case:

The max sum for both contiguous and non-contiguous elements is the sum of ALL the elements (as they are all positive).

In the second case:

[2 -1 2 3 4] --> This forms the contiguous sub-array with the maximum sum.

For the max sum of a not-necessarily-contiguous group of elements, simply add all the positive elements.

Change Theme Language: Python 3

```python
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'maxSubarray' function below.
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts INTEGER_ARRAY arr as parameter.
#

def maxSubarray(arr):
    # Write your code here
    def kadane(arr):
        dynamic_sum = arr[0]
        max_sum     = arr[0]

        for ele in arr[1:]:
            dynamic_sum = max(ele, dynamic_sum + ele)
            max_sum     = max(dynamic_sum, max_sum)

        return max_sum

    def subseq(arr):
        result = 0

        for ele in arr:
            if ele > 0: result += ele

        if result == 0: result = max(arr)

        return result

    return [kadane(arr), subseq(arr)]
```

Line: 20 Col: 28

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0 Compiler Message