|  |
| --- |
| ## Aim: |
|  | ## Given an array of k elements, find the maximum possible sum of a |
|  | ## 1. Contiguous subarray |
|  | ## 2. Non-contiguous (not necessarily contiguous) subarray. |
|  |  |
|  | ## Input : |
|  | ## First line of the input has an integer t. t cases follow. |
|  | ## Each test case begins with an integer k . In the next line, k integers follow representing the elements of array arr. |
|  |  |
|  | ## Output : |
|  | ## Two, space separated, integers denoting the maximum contiguous and non-contiguous subarray. |
|  | ## At least one integer should be selected and put into the subarrays |
|  | ## (this may be required in cases where all elements are negative). |
|  |  |
|  | t = int(input().strip()) |
|  | def max\_subarray(A): |
|  | max\_ending\_here = max\_so\_far = 0 |
|  | for x in A: |
|  | max\_ending\_here = max(0, max\_ending\_here + x) |
|  | max\_so\_far = max(max\_so\_far, max\_ending\_here) |
|  | return max\_so\_far |
|  |  |
|  | for i in range(2\*t): |
|  | k=int(input()) |
|  | arr = [int(arr\_temp) for arr\_temp in input().strip().split(' ')] |
|  | if(all(item>0 for item in arr)): |
|  | print(sum(arr),sum(arr)) |
|  | elif(all(item<0 for item in arr)): |
|  | print(max(arr),max(arr)) |
|  | else: |
|  | c=0 |
|  | for i in range(len(arr)): |
|  | if(c+arr[i]>c): |
|  | c+=arr[i] |
|  | print(max\_subarray(arr),c) |
|  |  |