**Problema 1**

Per il primo problema viene richiesto, dato un vettore che può contenere numeri interi sia positivi che negativi, di trovare il sottoarray di numeri contigui che ha la somma più grande, e di riportare tale somma. La logica applicata per la funzione che svolge l’operazione di trovare la somma più grande è quella che la somma è valida fino a quando questa non diventa negativa, in quanto anche solo la presenza di un singolo numero positivo comporterebbe la presenza di un sottoarray con la somma massima maggiore di tutti gli altri sottoarray. Quando è presente inoltre un numero negativo che decrementa la somma senza però portarla a valori negativi, questo viene comunque considerato parte del sottoarray, poiché in seguito viene in qualsiasi caso considerato se il valore della somma attuale è maggiore di quello presente al momento del controllo. In questo modo ci si assicura di avere sempre la somma più grande ottenibile.

Di seguito vengono riportati 3 casi di test alternativi a quelli riportati nell’esempio della traccia:

**Sample Input**

3 2 -4 9

-1 5 3 -6

10 -8 6 6

END

**Sample Output**

10

8

14

Di seguito viene riportata l’analisi di complessità:

**Complessità nel caso peggiore:** O(nlogm)

La complessità varia in base al numero di testcase poiché la funzione per individuare la massima somma tra i sottoarray può essere invocata più volte. In particolare ne risulta che **n** è il numero di elementi dell’array mentre, **logm** rappresentà il numero di righe che vi sono e quindi che invocano la funzione.

Elaborato svolto da:  
Marco Dell’Isola M63001637

Raffaele Cuzzaniti M63001614