**ZOJ Problem Set - 2962**

Stack By Stack

Time Limit: 5 Seconds      Memory Limit: 32768 KB

There are n stacks, named in order S[1], S[2], S[3], ... ,S[n]. Initially, all of them are empty. The following process ends when S[n] is full.

If non of the stacks is full, S[1] is filled by numbers 1, 2, 3, ... in order until it's full. Else, there is a full stack S[i], it is poped and pushed into S[i+1] until S[i] is empty or S[i+1] is full, whichever comes first. If S[i+1] is full and there are numbers in S[i], the numbers in S[i] are poped away.

**Input**

There are multiple test cases. There are three lines for each case. The first line is an integer *N* (1 <= *N* <= 1,000), the number of stacks. The second line is *N* integers: *C1* *C2* ... *CN* , in which *Ci* (1 <= *Ci* <= 1,000,000,000) is the size of ith stack. The third line is two integers *X* and *Y* (1 <= *X* <= *Y* <= *CN* ).

Process to the end of file.

**Output**

For each case, print a number in one line, the sum of the numbers in S[n] from index *X* to *Y*. The index starts from 1 at the bottom of a stack. The result will fit in a signed 64-bit integer.

**Sample Input**

1

100

1 100

2

2 4

1 3

3

5 3 5

3 4

**Sample Output**

5050

5

8