Game of Two Stacks

HackerRank

Alexa has two stacks of non-negative integers, stack $a[n]$ and stack $b[m]$ where index 0 denotes the top of the stack. Alexa challenges Nick to play the following game:

- In each move, Nick can remove one integer from the top of either stack a or stack b.
- Nick keeps a running sum of the integers he removes from the two stacks.
- Nick is disqualified from the game if, at any point, his running sum becomes greater than some integer $maxSum$ given at the beginning of the game.
- Nick's *final score* is the total number of integers he has removed from the two stacks.

Given a, b, and $maxSum$ for g games, find the maximum possible score Nick can achieve.

Example
$a = [1, 2, 3, 4, 5]$
$b = [6, 7, 8, 9]$

The maximum number of values Nick can remove is 4. There are two sets of choices with this result.

1. Remove $1, 2, 3, 4$ from a with a sum of 10.
2. Remove $1, 2, 3$ from a and 6 from b with a sum of 12.

Function Description
Complete the *twoStacks* function in the editor below.

twoStacks has the following parameters: - *int maxSum:* the maximum allowed sum
- *int a[n]:* the first stack
- *int b[m]:* the second stack

Returns
- *int:* the maximum number of selections Nick can make

Input Format

The first line contains an integer, g (the number of games). The $3 \cdot g$ subsequent lines describe each game in the following format:

1. The first line contains three space-separated integers describing the respective values of n (the number of integers in stack a), m (the number of integers in stack b), and $maxSum$ (the number that the sum of the integers removed from the two stacks cannot exceed).
2. The second line contains n space-separated integers, the respective values of $a[i]$.
3. The third line contains m space-separated integers, the respective values of $b[i]$.

Constraints

- $1 \leq g \leq 50$

- $1 \leq n, m \leq 10^5$
- $0 \leq a[i], b[i] \leq 10^6$
- $1 \leq maxSum \leq 10^9$

Subtasks

- $1 \leq n, m, \leq 100$ for 50% of the maximum score.

Sample Input 0

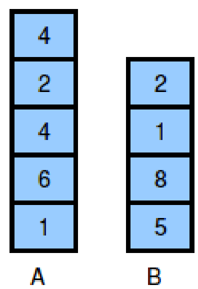
```
1
5 4 10
4 2 4 6 1
2 1 8 5
```

Sample Output 0

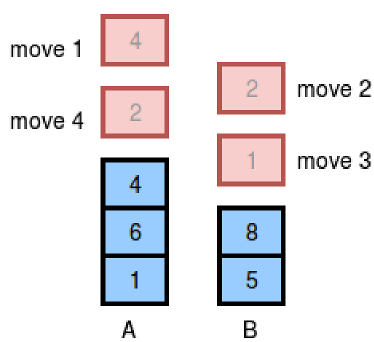
```
4
```

Explanation 0

The two stacks initially look like this:



The image below depicts the integers Nick should choose to remove from the stacks. We print 4 as our answer, because that is the maximum number of integers that can be removed from the two stacks without the sum exceeding $x = 10$.



(There can be multiple ways to remove the integers from the stack, the image shows just one of them.)