Orange Collection

There are N boxes in a row. Each box has one or more oranges in it. Consider these boxes as an array of integers **{b1, b2, b3, ..., bN}**, where bi is the number of oranges in the i-th box.
Mr. X is picking up some oranges. He is currently standing in front of the X-th box where **1 <= X <= N**.
Mr. X likes to move only towards one direction, either right or left. So if he chooses to go right, i.e. towards the N-th box, he can pick up all the oranges from **X-th, (X+1)-th, (X+2)-th, ... N-th** boxes.
And if he chooses to go left, i.e. towards the 1st box, he can pick up all the oranges from **X-th, (X-1)-th, (X-2)-th, ... 3rd, 2nd and 1st** boxes.
Mr. X is wondering what could be the maximum number of oranges he could collect if he chooses to go left or right optimally.

Input Format

The first line will contain an integer **T**, representing the number of the test cases. On each of the T followed by three lines. First line of each test case you will take an integer **N (1 <= N <= 1000)**, representing the number of boxes. The second line contains N space-separated integers **b1, b2, ..., bN (1 <= bi <= 100)** denoting the number of oranges in the i-th box. The third line contains an integer **X (1 <= X <= N)**, representing the position Mr. X is currently in front of.

Constraints

empth

Output Format

For each case, print the maximum number of oranges Mr. X can pick up.

Sample Input 0

```
2
5
5 6 9 3 4
3
3
1 2 2
1
```

Sample Output 0

```
20
5
```