You just saw a TV commercial for an interesting business plan that consists of a cycle of N phases. The i-th phase has a value V_i, which can be positive, negative, or zero. When you enter the i-th phase, you add V_i to the amount of money you currently have, unless this would create a negative number, in which case your amount of money becomes 0 instead. Then you move on to the $((i+1) \bmod N)$-th phase.

You start the business plan just before phase 0, with some initial amount of money (possibly 0). What is the minimum initial amount of money you need to guarantee that you will have **at least** G money after no more than P phases?

Input

The first line of the input gives the number of test cases, T; this is followed by one blank line.

T test cases follow; each consists of two lines, followed by a blank line. The first line has three integers N, G, and P, as described above. The second line has N integers; the i-th of these is V_i, representing the value of the i-th phase.

Output

For each test case, output one line containing '`Case #`x`:` y', where x is the test case number (starting from 1) and y is the minimum initial amount of money needed, as described above.

Limits:

- $1 \leq T \leq 100$.
- $1 \leq N \leq 10^5$.
- $1 \leq G, P \leq 10^{18}$.
- $|V_i| \leq 10^9$.

Note:

In Case #1, if you start with 7 money, then after one phase (phase #0), you will have 10 money. This satisfies the condition of having at least 10 money after no more than 2 phases. No smaller amount works.

In Case #2, if you start with 5 money, then you will have 8 money after one phase (phase #0), 7 money after two phases (phase #1), and 10 money after three phases (phase #0 again). No smaller amount works.

In Case #3 (which is a terrible business plan, by the way!), the only way to have at least 10 money at any point in the process is to start with at least 10 money, and 10 is the smallest amount that works.

Sample Input

```
3

2 10 2
3 -1

2 10 3
3 -1

1 10 10
-999
```

Sample Output

```
Case #1: 7
Case #2: 5
Case #3: 10
```