Problem D

An ICPC Problem without Statement

It's a lovely day in the year of 2016.

Upon graduation from your university, the head of computer science school approached and asked you to coach ACM-ICPC teams in your university, i.e. your juniors. Of course you gladly accepted it. As usual, they want to send teams to ACM-ICPC Regional Contests. However, due to limited budget, your university can only afford to send a single team to one regional site. Through a rigorous selection process, you have formed your best team. The remaining problem is determining which regional site you should send your team to compete in. For this purpose, you decided to do some simple research on problem types, difficulties and also contest experiences on various regional sites by digging into blogs on previous year's contests.

Then you stumbled upon a contestant's blog on ACM-ICPC Regional Jakarta 2015. In this blog, the contestant described a seemingly easy problem which he called "evil".

> *"This year, I participated in ACM-ICPC Jakarta Regional 2015. The problem set was interesting. My favorite problem is An ICPC Problem without Statement. The problem has very long description. Here, I will just simplify what the problem is really asking:*
>
> *You are given N (1 ≤ N ≤ 100,000) integers. Each integer is between -2 and 2, inclusive. You are also given two integers A and B. Pick K integers, where K is between A and B inclusive, such that their product is as large as possible!*
>
> *The problem looks kind of easy, right? However, many teams got many wrong answer verdicts on this problem during the contest. The scoreboard was full of red color on this problem's column. My team finally managed to get accepted on this problem in the 7th try. We got very bad penalty :(It's so evil!*
>
> *Anyway, can you solve this problem? Post your solution in the comment section below."*

What's so difficult about this problem? The integers are only on the range of [-2 ... 2]; the solution should be easy. Then you started to code the solution right away. After you've finished, you submitted it to the Live Archive (a website that hosts all past ACM-ICPC Regional problems) confidently. Uh oh… a Wrong Answer verdict. After fixing and receiving so many Wrong Answer verdicts for this problem, you then realized one thing: the blog author was correct, this problem is evil…

Input

The first line of input contains an integer T ($T \leq 20$) denoting the number of cases. Each case begins with three integers: N, A, and B ($1 \leq A \leq B \leq N \leq 100{,}000$). The next line contains N space-separated integers. Each integer is between -2 and 2, inclusive.

Output

The output for each case corresponds to the problem mentioned in the problem statement. For each case, output "`Case #X:`" (without quotes) in a line where `X` is the case number (starts from 1). In the second line, output an integer K (which complies with A and B in the input) representing the number of selected integers. In the third line of each case, output K integers, each separated by a single space and sorted in ascending order. These K integers represent the indexes of the selected integers. The used index is 1-based, which means the first integer has the index of 1. If there is more than one solution, output any of them.

Warning: large input and output file.

Sample Input	Output for Sample Input
4 7 1 4 2 1 1 -2 -1 2 0 4 1 1 1 -2 2 -1 4 3 4 1 -1 1 -1 4 3 4 1 -1 1 -1	Case #1: 4 1 4 5 6 Case #2: 1 3 Case #3: 4 1 2 3 4 Case #4: 3 2 3 4

Explanation for 1st sample case

In this sample case, the largest product you can obtain is 8, by selecting 4 integers at index 1, 4, 5, 6 which values are 2, -2, -1, and 2, respectively.

Explanation for 2nd sample case

K should be between A = 1 and B = 1, thus you can only select one integer. 2 (at index 3) is the best choice.

Explanation for 3rd and 4th sample case

The 3rd and 4th sample cases are of the same input. The largest product which can be obtained is 1. Both outputs on 3rd and 4th cases are correct (you may output any of them).