6A2	REVERSE POLISH NOTATION	
	Input	Standard Input
	Output	Standard Output

Problem Description
Reverse Polish notation, also known as postfix notation, is a method for representing expressions in which the operator symbol is placed after the arguments being operated on. Polish notation, in which the operator comes before the operands, was invented in the 1920s by the Polish mathematician Jan Lucasiewicz. In the late 1950s, Australian philosopher and computer scientist Charles L. Hamblin suggested placing the operator after the operands and hence created reverse polish notation.

RPN has the property that brackets are not required to represent the order of evaluation or grouping of the terms. RPN expressions are simply evaluated from left to right and this greatly simplifies the computation of the expression within computer programs. As an example, the arithmetic expression 3 + 4 * 5 can be expressed in RPN as 3 4 5 * +.

RPN is used in Hewlett Packard and some Texas Instruments calculators and internally in some computer languages.

Write a program that converts arithmetic expression to RPN.

Input
The first line will contain T (T <= 100), number of test cases, followed by T lines each with a line of string representing an arithmetic expression.

Output
For each test case, output the RPN expression.

Sample Input	Sample Output
5 3 + 4 3 + 4 * 5 3 * (4 + 5) 12 + 60 / 23 (3 + 7) * (8 – 5)	3 4 + 3 4 5 * + 3 4 5 + * 12 60 23 / + 3 7 + 8 5 – *

6B MONK AND CHAMBER OF SECRETS

Input	Standard Input
Output	Standard Output

Problem Description

Hagrid says "follow the spiders" and so Harry and Ron head to the Forbidden Forest. There they meet Aragog, a giant spider who tells them about the innocence of Hagrid. But Aragog only allows Hagrid to go back. These boys have got into a serious trouble now.

The only way to escape as Aragog says is to answer a question. Aragog shows them a queue of ***N*** spiders of which only ***X*** spiders are to be selected. Each spider has some power associated with it. There are ***X*** iterations on the queue. In each iteration, ***X*** spiders are dequeued (if queue has less than X entries, all of them will be dequeued) and the one with maximum power is selected and remaining spiders are enqueued back to the queue (in the order they were dequeued) but their power is decreased by one unit. If there are multiple spiders with maximum power in those dequeued spiders, the one which comes first in the queue is selected. If at any moment, power of any spider becomes 0, it can't be decremented any further, it remains the same. Now, Aragog asks the boys to tell him the positions of all the selected spiders (positions in the initial given queue) in the order they are selected. As the boys are frightened and can't think of anything , they call Monk for the rescue. Help Monk to get the answer fast and save the boys.

Input

There will be several test cases. For each test case, the first line consists of two space separated integers ***N*** and ***X***, denoting the number of spiders in the queue and the number of spiders that have to be selected respectively. The next line consists of an array ***A***, ***A***[*i*] denoting the power of spider at position *i* ($1 \leq i \leq N$). Input terminates with both ***N*** and ***X*** = 0

Output

For each of the ***X*** iterations, output the position of the selected spider in that iteration. Position refers to the index at which the spider was present in the initial given queue (1 based indexing).

Constraints:

$1 \leq X \leq 100$

$X \leq N \leq X*X$

$1 \leq A[i] \leq X;\ 1 \leq i \leq N$

Sample Input	Sample Output
6 5 1 2 2 3 4 5 8 6 1 2 3 4 5 5 7 8 8 2 1 2 3 4 5 5 7 8 0 0	5 6 4 1 2 5 8 7 6 1 2 2 4

6C	ABC SERVICE MACHINE	
	Input	Standard Input
	Output	Standard Output

Problem Description

ABC Service Machine provides a self-service for its customer. The customer must queue up in order to get the service. The customer will immediately be served if the queue is empty. If the machine is free, then it will serve the front customer in the queue. The machine takes 13 minutes to complete one service. The machine operation starts at 8.00 am and ends at 12.00 pm. However, it will continue serve customers who arrive before or by 12.00 pm.

The owner of the ABC Service Machine would like to know some statistical information so that he can improve the machine in the future. The information are:

- Number of customer that arrives by 12.00 pm.
- Longest customer waiting time.
- Average customer waiting time. Assume that all customers are an ethical person and determined to get the service, also the machine is ideal.

Input

First line of input is an integer N (1 <= N <= 50), that represents the number of test case, followed by N lines where each line of input starts with an integer M (1 <= M <= 30) that represents the number of customer followed by M arrival time (in minutes) of the customers. The arrival time is a gap between arrivals except for the first customer, the time refer to a gap between starting time of the machine and his/her arrival.

Output

For each test case, the output contains a line in the format Case #x: A B C where x is the case number (starting from 1), A is an integer represent the number of customer that arrives by 12.00 pm, B is an integer represent the longest waiting time in minute and C is a real number with two decimal places represent an average customer waiting time.

Sample Input

```
3
22 14 10 15 12 11 14 11 12 13 10 13 10 14 10 13 10 10 10 14 13 13 10
16 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
2 0 10
```

Sample Output

```
Case #1: 20 23 10.15
Case #2: 12 0 0.00
Case #3: 2 3 1.50
```