Boring job

One day, Rick got a sequence of N magic numbers. In order to find the latent code in the sequence, he decides to do the following operations with a magic parameter K:

Step (1) Take out the first K numbers from current sequence S into a temporary sequence n_1, n_2, …, n_k, and pick the largest element Vmax out of these K numbers, then subtract 1 from the remaining K-1 elements. If there exist more than one largest element, choose the one with smaller index in **temporary** sequence.

Step (2) Print the **index** (count from 1) of Vmax in the **initial** sequence, append the rest K-1 elements (with subtraction done) in temporary sequence back to the end of S.

Repeat (1)(2) until S become empty. If the size of S is smaller than K, in step (1) just take out all the elements.

Since Rick is too smart to be interested in doing this on his own, he asked his grandchild Morty to do this job for him. However, Morty is just not clever enough to do this job …

Can you use your programming skill to help poor Morty?

Input

The first line of input is an integer T (1 <= T <= 10) indicating the number of test cases.

Each test case will follow the format shown below:

The first line: Two integers N and K showing the length of sequence and magic parameter in this test case.
The second line: N integers m_1, m_2, …, m_N showing magic numbers in the sequence.
(1 <= N<= 1000, 1 <= mi <= 100000, 1 <= k <= 20 and k <= N)

Output

Print a single line containing the desired sequence of indices with each index separated by a single space.

Sample Input

```
2
4 2
2 3 1 4
5 3
5 3 2 4 10
```

Sample Output

```
2 4 1 3
1 5 4 3 2
```

Explanation for the first sample:

Initially we have sequence S = [2(1),3(2),1(3),4(4)]. (Numbers in the parenthesis is indices for clarity)

(1) The first K (=2) magic numbers are taken out as tmp[2(1),3(2)] and S becomes [1(3),4(4)]. The largest element in tmp[] is 3 whose index in original S is 2.
Print “2”
Then the rest elements in tmp[] are decreased by 1, getting [1(1)]. Afterwards [1(1)] is appended to S, making S = [1(3),4(4),1(1)].

(2) The first K magic numbers are taken out as tmp[1(3),4(4)] and S becomes [1(1)].
The largest element in tmp[] is 4 whose index in original S is 4.
Print “4”
Then the rest elements in tmp[] are decreased by 1, getting [0(3)].
[0(3)] is appended to S, making S = [1(1),0(3)].

(3) The size of S is equal to K (=2). All elements are taken out as tmp[1(1),0(3)] and S becomes [].
The largest element in tmp[] is 1 whose index in original S is 1.
Print “1”
Then the rest elements in tmp[] are decreased by 1, getting [-1(3)].
[-1] is appended to S, making S = [-1(3)].

(4) Since the only element in S is -1(3), it is the largest one without any doubt.
Print “3”
And S becomes empty.

So the answer is “2 4 1 3”.