This year chandan has planted **N** trees (indexed **1** to **N**) and has decided that he will pluck apples on **K** days (indexed **1** to**K**). On each day, he selects a group of trees that he had not chosen previously, and plucks all apples from them. The amount of work he has done for that day is the *total number of apples he picked that day*.

Chandan wants to *minimize the average work* he does. For that he thinks it will be optimal if the difference between the largest work on any day (lets call it **A**) and the least work on some other day(lets call it **B**) is minimum. That is, mathematically writing, he wants to minimize (A - B) where *A is the largest work* and *B is the smallest*. But still he is unhappy because he feels it is also useful to take into account the total number of tree plucked on each day (irrespective of the apples it contains). So he finds the maximum trees(say **C**) and minimum trees(say **D**) he plucked on any day. He wants to minimize (C - D) as well.

Due to increasing complications in the problem, he has approached you again. He wants to you divide the **N** trees into **K**days so that the product **(C-D)2 \* (A-B)** is as small as possible.

[Input]  
The first line of each file contains a single integer **T**, the number of inputs. The first line of every test case, contains two space separated integers **N** and **K**, number of trees and number of days. Next Line contains **N** space separated integers, which are the number of apples on each tree.

[Output]  
For each test case, you need to write exactly **N** space separated integers in single line, where ith integer represents the day index when apples of ith tree were picked. All the numbers should be in the range [**1**, **K**]. All 1's denote the trees he will pick on the first day, 2 on the second day, and similarly K on the Kth day. *It is possible that he might not pick any tree on a particular day.*

[Constraints]  
T ≤ 50  
1 ≤ N, K ≤ 500  
1 ≤ apples on a tree ≤ 109

[Scoring]  
Sum of expression, **((C-D)2 \* (A-B)) / (K+N)2** over all the test cases will be score of a test file. Total score will be sum of scores of all the test files.  
Since it is minimization problem, lesser the score higher points you will get. submissions with lowest score will set highest points (100).

Sample Input [(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/apple-picking-2/sample-inputtxt.txt?Signature=re0W6bzwRaVxkoHqLF%2B%2FVLTZibs%3D&Expires=1422797269&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ)

1

5 3

10 8 2 1 7

Sample Output [(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/anonymous-event/problems/apple-picking-2/sample_outputtxt.txt?Signature=PtAMT8AYWfpy8MQHmCpV1AEzle8%3D&Expires=1422797269&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ)

1 2 3 3 3

Explanation

He decides to pick 1st tree on day 1, second on day 2 and all others on day 3.  
So A is 10, B is 8. C = 3 and D = 1.  
So the score for this test case becomes ((3-1)^2 \* (10-8))/(5+3)^2 which is 0.125.  
Note that there could possibly be better solutions, nevertheless this solution is also correct.