

A charting component is a useful tool in many applications. Typically the component takes as input a set of points and outputs a nice graph (for instance a scatter plot). One thing that is done “automagically” by the component is deciding which grid points to use on the axes. Most people would agree that an axis with grid points 0, 5, 10, ... looks nicer than an axis with the grid points 0, 3, 6, ... Your task is to write a program which selects the grid points to use based on the points to plot and the number of desired grid points. Since the x - and y -axes are independent of each other, we will only consider one axis in this problem. Hence the plot points will be in a 1-dimensional space.

The following rules should be applied when selecting the grid points. The grid points should of course always be evenly spaced, and should always include the origin (point 0). The only allowed (so it “looks nice”) spacings of the grid points are 1, 2, 2.5 or 5 - or any multiple/divisible by 10 of these values. The span between the lowest and highest grid point must always encompass (inclusive) all points and the origin. So if the points to plot are 0.029, 0.185 and -0.095, a valid set of grid points are -0.10, -0.05, 0.00, 0.05, 0.10, 0.15, 0.20.

Furthermore, there should be no unnecessary grid points after the end points. Hence adding the grid point 0.25 to the grid points above would not be allowed since the last point, 0.20, was enough to encompass the rightmost plot point. However, if a more sparse grid spacing was used this grid point would be allowed: the grid points -0.25, 0.00, 0.25 are perfectly valid for the plot points above.

The *best* choice of grid points is one that fulfills the criteria above and where the difference between the number of grid points used and the desired number of grid points (given in the input) is minimized. If there is a tie, select the set with most grid points. If still a tie, select the set with the smallest grid spacing. For instance, if the desired number of grid points for a particular set of plot points is 6, and the two best grid point sets requires 5 and 7 grid points, select the set containing 7 grid points.

Input

The first line in the input contains the number of test cases (no more than 100). Each test case contains two lines. The first of these lines contains two integers, n (the number of plot points, $1 < n < 100$) and k (the desired number of grid points, $1 < k < 20$). The second of these lines contains the n distinct plot points, separated with a blank space. These points are real numbers, whose absolute values are between 10^{-5} and 10^5 , inclusive, and they will not have more than 5 digits after the decimal point.

Output

For each test case, output a single line containing the best selection of grid points (according to the rules above) in increasing order. Separate the grid points by exactly one space, and leave no trailing spaces at the end of the line. All grid points in a test case should be printed with same number of digits after the decimal point, and at least one of the grid points should not have trailing zeros after the decimal point. Also make sure you don't print values such as '-0.0'.

Sample Input

```
4
3 8
0.029 0.185 -0.095
4 10
-0.0002 -0.00025 -0.001 0.0008
7 4
24378 -189832 488 23478 12345 10000 -9991
3 5
-0.05 -0.00001 -0.09
```

Sample Output

```
-0.10 -0.05 0.00 0.05 0.10 0.15 0.20
-0.0010 -0.0008 -0.0006 -0.0004 -0.0002 0.0000 0.0002 0.0004 0.0006 0.0008
-200000 -100000 0 100000
-0.100 -0.075 -0.050 -0.025 0.000
```