

Oversized Pancake Choppers

Problem

You just showed up to your job as the head chef of the Infinite House of Pancakes, and as usual, you found a disaster in progress! The other chefs accidentally created some enormous circular pancakes, all of the same size. These pancakes are too large to serve whole, so they have already started to chop them up into *slices* (which, in this problem, are [circular sectors](#)). You currently have N slices, the i -th of which is a sector with an internal (central) angle of A_i nanodegrees (a nanodegree is 10^{-9} degrees).

You have D diners waiting for their food. Each diner wants a single slice that is the same size as every other diner's slice, although they do not care what that size is. But it may not be possible to do this using the current slices, so you may need to make one or more radial *cuts*.

A cut changes an existing slice with internal angle X into two new slices with internal angles Y and $X - Y$. You can do this for any $0 < Y < X$, and these values do not need to be integers. You may apply further cuts to either or both of these new slices, and so on.

It is OK to have one or more leftover slices (of any size) that are not given to the diners; you can eat those later, since this disaster is making you miss your own breakfast!

Determine the smallest total number of cuts you need to make to satisfy the diners.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each case begins with one line containing two integers N and D : the number of slices you currently have and the number of diners. Then, there is one more line containing N integers A_1, A_2, \dots, A_N ; the i -th of these represents the internal angle (in nanodegrees) of the i -th slice.

Output

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the smallest number of cuts you need, as described above.

Limits

Memory limit: 1GB.

$1 \leq T \leq 100$.

$1 \leq A_i < 360 \times 10^9$, for all i .

Test Set 1 (Visible Verdict)

Time limit: 20 seconds.

$1 \leq N \leq 300$.

$2 \leq D \leq 3$.

Test Set 2 (Visible Verdict)

Time limit: 20 seconds.

$1 \leq \mathbf{N} \leq 300$.

$2 \leq \mathbf{D} \leq 50$.

Test Set 3 (Hidden Verdict)

Time limit: 60 seconds.

For exactly 21 cases, $9000 \leq \mathbf{N} \leq 10000$.

For exactly $\mathbf{T}-21$ cases, $1 \leq \mathbf{N} \leq 1000$.

$2 \leq \mathbf{D} \leq 50$.

Sample

Sample Input

```
4
1 3
1
5 2
10 5 3599999999999 123456789 10
2 3
8 4
3 2
1 2 3
```

Sample Output

```
Case #1: 2
Case #2: 0
Case #3: 1
Case #4: 1
```

In Sample Case #1, you only have one tiny slice to start with. The optimal solution is to use one cut to change it into two slices with angles of $1/3$ nanodegree and $2/3$ nanodegrees, and then further cut the latter slice into two more slices with angles of $1/3$ nanodegree.

In Sample Case #2, you already have two slices of the same size, so you can give those to the two diners, and you do not need to make any cuts.

In Sample Case #3, the optimal solution is to cut the slice with internal angle 8 nanodegrees in half. After that operation, you have exactly 3 slices of internal angle 4 nanodegrees, with no leftovers.

In Sample Case #4, remember that every diner must receive a single slice. You cannot give one diner the "3" slice and the other diner the "1" and "2" slices, even though the total areas are the same. You must make at least one cut in this case to satisfy the requirements.