1. Benford's Law: Who's Number One?

Program Name: Benford.java **Input File: benford.dat**

Benford's Law, also known as the first digit law, states that given a set of values based on real data or measurements the distribution of the leading digits of those values is **not** expected to be uniform. Instead it is often the case that there are more values that start with a 1 than any other. Values that start with a 2 will be the next most frequent and so forth with values starting with 9 expected to be the least frequent. Write a program that given a data set of integers determines if the data confirms Benford's Law or not. For this problem a data set confirms Benford's Law if the number of values that start with a 1 is greater than or equal to each of the number of values that start with 2 through 9. Likewise the number of values that start with 2 is greater than or equal to each of the number of values that start with 3 through 9. And so forth for the number of values that start with 3 through 8.

Input

- The first line will contain a single integer `n` that indicates the number of data sets that follow.
- The first line of each data set will contain a single integer `m` that indicates how many values are in the data set.
- The values in the data set will be on the following lines, 5 values per line, with a space between each value. The last line in the data set could have fewer than 5 values.
- All values in the data set will be integers greater than 0 and less than 1,000,000. Values will not contain leading 0s.

Output

For each data set print out `CONFIRMS` if the leading digits of the values in the data set confirm Benford's Law. Print out `DOES NOT CONFIRM` if the leading digits of the values in the data set do not follow Benford's Law.

Example Input File

```
3
36
661 59 1452 1338 1438
2923 3796 4796 1 2781
1742 20 626 2230 7553
42 2113 7450 5034 3516
5381 31 5665 101 19
11100 4028 2 6336 4683
35 8975 2908 3177 49
389111
40
76 4319 6859 286 172811
8442 32530 29 4170 2890
61 1044 4478 28310 1619
5093 2500 68 5939 5
55 6105 2841 1207 5
4207 12 2248 1967 124715
13 1613 7391 45 4855
60 4949 28 5488 1755
9
1 22 333 4444 55555
678 713 80200 9000
```

Example Output To Screen

```
CONFIRMS
DOES NOT CONFIRM
CONFIRMS
```