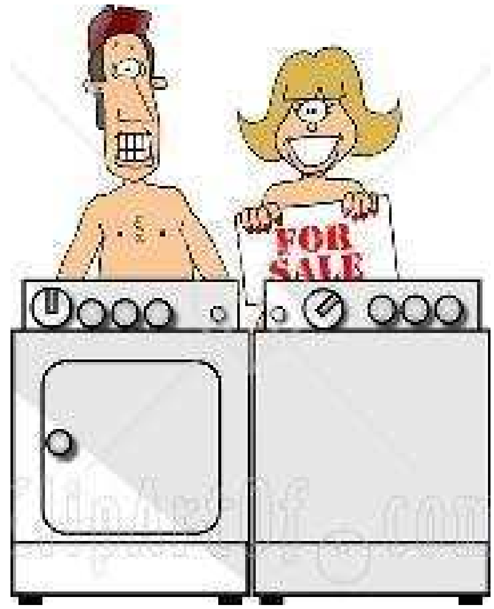
Problem A- Doing Laundry



At home, I am lucky: I have in-suite laundry. This means I can do laundry when ever I want, but that doesn't mean I want to waste my time!

Help me devise a good schedule. Doing the wash (when the clothes get wet) always takes 30 minutes no matter what I'm washing. It's the drying that varies for each load. Some loads take 60 minutes to dry, some take only 15 minutes, and some can be anywhere in between. You can assume that moving clothes into/out of the washer or dryer is an instantaneous operation (i.e., 0 minutes), and that there are enough laundry baskets around to store any wet clothes waiting for the dryer.

Input Specification:

Each test case will begin with a line with the positive integer $n \leq 30$, the number of loads of laundry for the day. Following that will be n integers in the range 15 to 60, denoting the drying time in minutes required for each load.

The input ends when $n = 0$. This is not a test case and should not be processed.

Output Specification:

For each test case, output a line with the total time required to do the laundry, assuming the best schedule. Output your answer in digital clock format, as shown in the samples.

Sample Input:

```
1
20
2
60 15
10
60 60 60 60 60 60 60 60 60 60
0
```

Sample Output:

```
0:50
1:45
10:30
```

Problem B- The Longshot

In the movie Wimbledon, wildcard entry Peter Colt wins upset after upset as he marches to the finals of this grand slam tennis tournament. Of course, that's the movies, but the news can't resist falling in love with a longshot, such as Melanie Oudin's ride to the quarter finals in the 2009 US Open.

Certainly, the top seeds in any tournament are very strong, but not unbeatable. It could be a matter of incompatible style or luck, but as they say, every dog has its day.

The problem is this. Given a group of 16 players and a 16x16 binary matrix, describing who will win if two players were to face-off, determine which players could possibly make it to the final match of a single elimination tournament, given the correct matchups along the way.

(A tournament of this type can be viewed as a complete binary tree with 16 leaves, the winner of each match proceeding to the next higher level of the tournament.)

Input Specification:

There will be up to 6 test cases presented one after the other: 16 lines of 0s and 1s, 16 characters per line. The diagonal is, naturally, all 0s, and the other matrix entries satisfy $a_{ij} + a_{ji} = 1$. An entry of 1 for a_{ij} (row i, column j) means that Player i will defeat Player j in the tournament.

The input ends on EOF.

Output Specification:

For each case, output the list of players (1-16) who could reach the finals. (They do not have to win the finals, just get there.) Output should appear in sorted order, in a comma-space separated list as shown in the sample. Output one line per test case.

Sample Input:

```
0000000000000000
1000000000000000
1100000000000000
1110000000000000
1111000000000000
1111100000000000
1111110000000000
1111111000000000
1111111100000000
1111111110000000
1111111111000000
1111111111100000
1111111111110000
1111111111111000
1111111111111100
1111111111111110
0010100100010000
1000101011010111
0101111111111001
1100101111010101
0000000000010001
1101101111111101
1000100110010000
0100100000010000
1000100100010000
1000101110010100
1101101111011101
0000000000000000
1101101111010101
1010101110010000
1011111111111101
1000001111010100
```

Sample Output:

```
8, 9, 10, 11, 12, 13, 14, 15, 16
1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16
```

Problem C- Lost in Translation

In translating from a word in Argharian to a word in Ackanese, it suffices to make exactly one lexical change. The following are a comprehensive collection of possible single lexical changes:

- An Argharian word ending in 'e' has the same meaning as the Ackanese with the exact same spelling, but with the 'e' moved to the second last character. E.g. lute → luet.
- Similarly, if there is a 'rge' combination in Argharian, you can replace it with 'che' in Ackanese. E.g. large → lache.
- Any 're' combination can be replaced by a 'rec'. E.g. pretty → prectty.
- Any 'ce' combination can be replaced by an 'ec' combination. E.g. ice → iec.

All words in Argharian are between 3 and 12 characters long, inclusive. If a rule applied in reverse generates a word outside of this range, it is not valid of Argharian and should be discarded.

You can use these rules to find all *synonyms* (words of the same meaning) of an Argharian word. By translating from Argharian to Ackanese and back again, you will have two Argharian words of the same meaning.

You should be reminded that when A is a synonym of B, and B is a synonym of C, then A is a synonym of C.

Write a program that finds a complete list of synonyms for Argharian words.

Input Specification:

You will be presented Argharian words: 3 to 12 lowercase characters, one per line. The input ends with EOF.

Output Specification:

For each word, output its Argharian synonyms, one per line, in alphabetical order. Head each by a case number, as per the sample. Separate successive test cases by a blank line.

Sample Input:

```
farce
merge
cererge
```

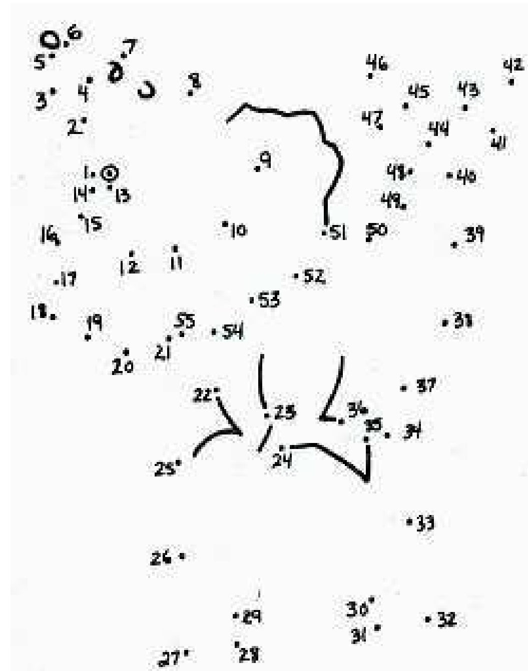
Sample Output:

```
Case 1:
farce
fare

Case 2:
mcehe
merge

Case 3:
cerccehe
cercehe
cercerge
cerehe
cererge
```

Problem D- Points on a Line



This problem is very simple- given a set of points on a plane, find a straight line that goes through the most points and print out how many points it goes through.

Input Specification:

The first line of the input contains a nonnegative integer T, the number of test cases to follow (at most 50). Each test case starts with an integer N, the number of points ($0 \leq N \leq 400$). N lines with the point coordinates follow. Each line will contain two integers X_i and Y_i. The absolute value of each coordinate will be at most 10000. All points in a single test case will be distinct.

Output Specification:

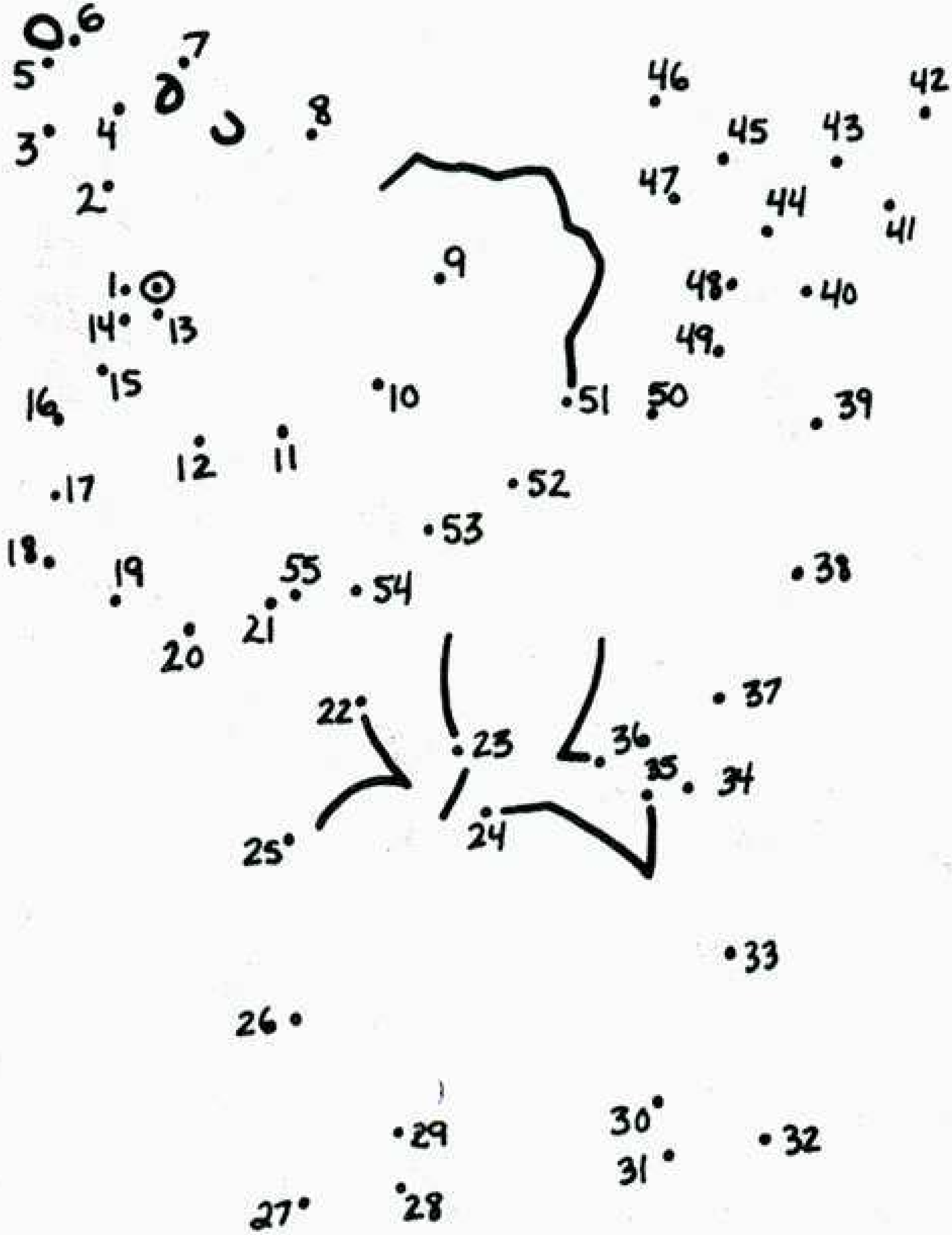
For each test case, print "`Case #k: M`", where k is the 1 or 2-digit test case number (starting from 1) and M is the number of points on the straight line that goes through most of them.

Sample Input:

```
2
1
0 0
3
0 0
1 1
-1 0
```

Sample Output:

```
Case #1: 1
Case #2: 2
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

Problem E- Canucks Psychology

Each round of the Stanley Cup playoffs is a series of 7 games, played until a team wins 4 games. The Canucks nominally play with a *strength* of 20. When they play another team with a strength of S, the chances that they win is expressed by the ratio $20 : S$.

Canuck fans are notoriously emotional: happy when their team wins, but depressed when their team loses. This can't help but affect the Canucks strength for the next game only: adding W to their strength when they win, subtracting L from their strength when they lose.

For example, say $W = 5$ and $L = 2$. In the first game of the series, the Canucks strength is 20 (no prior games). They win game 1, so their strength for game 2 is 25 $(20 + W)$. They win game 2, so their strength for game 3 is 25 (it doesn't accumulate, it's just from the previous game). They lose game 3, so their strength for game 4 is 18 $(20 - L)$. And so on until one team has 4 wins.

Input Specification:

The input begins with the positive integer $n \leq 100$, the number of test cases that follow. Following this will be n test cases, each of 2 lines. The first line will be the name of the opposing team, which will be no longer than 12 characters. The second line will contain the positive integers $S < 50$, $W < 20$ and $L < 20$.

Output Specification:

For each test case, output a line with the probability the Canucks will win the series, to 5 decimals.

Sample Input:

```
2
Flames
19 4 3
Blackhawks
25 4 4
```

Sample Output:

```
0.53254
0.37145
```