Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Ernst W. Mayr
Christian Müller, Philipp Hoffmann, Chris Pinkau, Stefan Toman

Summer Term 2015
Problem Set 1
16.04.2015

Algorithms for Programming Contests

This problem set is due by

Thursday, 23.04.2015, 6:00 a.m.

Try to solve all the problems and submit them at

`http://judge.informatik.tu-muenchen.de/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	SS15N01A	SS15N01B	SS15N01C	SS15N01D	SS15N01E
Difficulty	very easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

SS15N01A Hello World!

Author: Stefan Toman

This is probably the first problem you will solve and it should help you set up and test your system. Solve this problem first to make sure everything is in place.

We would like to introduce you to Lea. You will meet her in many of the problems you will solve. After reading all of them you will know her quite well.

Lea is a very friendly person who likes to say hello to everybody, but she doesn't want to say the same thing to every person she meets. Therefore, she never knows what to say. For greeting Bob it is appropriate to say "Hello Bob!", whereas for greeting Peter it is better to say "Hello Peter!". Help her and tell her which sentence to use.

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing a name *name*.

Output

For each test case, print a line containing "Case #i: Hello *name*!" where i is its number, starting at 1. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$.
- No name will contain whitespaces.
- The names' lengths will be at most 100.

Sample Data

Input

```
2
Bob
Peter
```

Output

```
Case #1: Hello Bob!
Case #2: Hello Peter!
```

SS15N01B Relativity

Author: Chris Pinkau

After a long day at the patent office, Lea decides to dedicate her free time to something more relaxing: physics. Recently, she took the class "**P**hysics: **H**ow **Y**ou **S**urvive **I**n **C**osmos & **S**pace". During lectures, she instantly became curious about how much energy certain objects contain. To this end, she now wants to calculate the energy equivalent of specific everyday objects like lamps, apples, iguanas, swiss cheese, ... Of course, Lea has a well-known formula to calculate all this: $E = mc^2$.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a line break.

Each test case consists of an integer m, the mass of the object.

Output

For each test case, output one line containing "Case #i: E" where i is its number, starting at 1, and E is the energy equivalent of the object. Each line of the output should end with a line break. You may assume that the input is given in the same order of magnitude as c. In particular, this means that you may disregard all units and need not to convert the input.

Constraints

- $1 \leq t \leq 1000$
- $1 \leq m \leq 100$
- $c = 299792458$

Sample Data

Input

```
5
1
5
10
50
100
```

Output

```
Case #1: 89875517873681764
Case #2: 449377589368408820
Case #3: 898755178736817640
Case #4: 4493775893684088200
Case #5: 8987551787368176400
```

SS15N01C Chess Tournament

Author: Philipp Hoffmann

Just a week ago Lea read about the local school chess tournament. A lot of schools participated, but sadly Lea's former school did not win (probably due to her, master of chess, not being on the team anymore).

The tournament is held in the following fashion: Each school sends a team of five players together with an ordering of those players according to their skill. Each team plays one match against each other team. A match consists of five games, one per player, where the best players of each team play each other, the second-best players of each team play each other and so on.

Right now, a team wins a match if it wins more games than it loses (remember that in chess there are draws), but this has always bugged Lea because there are so many ways to cheat. You could send your three best players to play on position three, four and five and get easy wins! So she has thought up a new scoring system: The game between the best players is most important, so if one team wins this game, it wins the match. If this game is a draw, the games between the second-best players is the most important of the remaining games, and so on. Only if all five games are tied, the match is a draw.

Lea has estimated the skill of all players of all schools. For each school, she gives you five numbers, the skill values of the players of that school's team. If two players play each other, the one with higher skill value wins. Equal skill values will result in a draw. You are to determine a ranking of the schools, that is, you should order the schools such that the first school wins all its games, the second one loses only to the first school and so on.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of schools. n lines follow each containing five integers $a_{i,1}, \ldots, a_{i,5}$, the skill values of the team members of the i-th school.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. Output n more lines, each containing the five skill values of a school's team members such that (a) the team members' skill values are listed in decreasing order and (b) if a school wins the match against another school, it appears before that other school. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$

- $2 \leq n \leq 1000$
- $1 \leq a_{i,j} \leq 1000$ for all $1 \leq i \leq n, 1 \leq j \leq 5$

Sample Data

Input

```
2
3
2 5 3 5 8
1 1 1 1 1
1 2 5 4 3

2
2 5 3 5 8
2 3 4 5 8
```

Output

```
Case #1:
8 5 5 3 2
5 4 3 2 1
1 1 1 1 1
Case #2:
8 5 5 3 2
8 5 4 3 2
```

SS15N01D Scrambled Chatlog

Author: Christian Müller

Lea is in tears. Just one or two hours ago, everything was still fine. She was surfing the internet on her phone and spent several hours looking at particularly cute pictures of kittens. Then, just as she got bored, she got a tweet from a seemingly very friendly person that said she should totally check out this new app called "KittenTube".

Totally hyped, she installed the app and was instantly greeted by heaps of pictures of cute kittens. Unfortunately however, she did not notice that the app had been developed by **TROLL Inc** (**T**rolls, **R**uffians, **O**bvious **L**amers and **L**owbobs) where all employees are maliciously evil (but not very bright) hackers. So when Lea installed the app, it totally scrambled her complete chat history. Her messages now all look like "11#ma tonight?Cine".

Now, the hackers want her to pay an enormous sum of money to unscramble her chat history for her. But as soon as Lea settled down, she thought she might be able to write a program to fix her history herself. Can you help her with that?

Input

The input starts with a line containing an integer t. t lines follow. Each of the following lines contains a string in the format "x#str" where each x is a number and str is the scrambled content of the message. To restore the original message, you have to split the string after x characters and swap the two halves.

Output

For each test case, output one line containing "Case #i: msg" where i is its number, starting at 1, and msg is the unscrambled message. Beware: as Lea has had her phone for quite a while, her chat log has grown quite large.

Constraints

- $1 \leq t \leq 10^5$
- $2 \leq |str| = |msg| \leq 160$
- $1 \leq x < |str|$
- str consists of letter from "a" to "z", "A" to "Z", spaces, any of the following: ".,?!*;:-/" in addition to round and square brackets

Sample Data

Input

```
4
27#o the programming homework?Hey - did you d
13#/ Is it hard?Not yet :
25#ly. Just a little coding.Well ... not real
9#ght then.Ah - alri
```

Output

```
Case #1: Hey - did you do the programming homework?
Case #2: Not yet :/ Is it hard?
Case #3: Well ... not really. Just a little coding.
Case #4: Ah - alright then.
```

SS15N01E Watson

Author: Stefan Toman

Lea is amazed by Watson, the artificially intelligent computer system that won Jeopardy! in 2011. Watson's victory proves how fast the field of artificial intelligence evolves and showcases the impressive capabilities of current systems.

Lea wanted to understand how these systems work and therefore decided to build her own Watson which she called "LEAtron 3000"! After some days spent on the project she realized that she will not be able to beat Watson. This was, of course, not due to her knowledge, but due to the limited amount of computing power she had access to (namely, her old Strawberry Tau).

Since she still wanted to build something, Lea changed the game to an easier one. Instead of answering general questions, "LEAtron 3000" should solve easy equations, which were still given in natural language. She wrote the code and tested it several times, the machine always won against her!

Before showing it to the world, she applied some last performance tweaks and somehow managed to crash the machine. As Lea is lazy, she never made any backups and does not want to write everything from scratch again. Still, she wants to take her machine to several quiz shows, defeat all the human beings and win all the prize pools. Can you help her and rewrite the code?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing a string s. The string will consist of digits 0 to 9 and the words "plus", "minus", "times" and "tothepowerof". There will always be at least one digit between operator strings. No number other than 0 will begin with 0.

Output

For each test case, print a line containing "Case #i: y" where y is the result of the computation given in x. In contrast to usual notation, all operations should be evaluated from left to right.

Constraints

- $1 \leq t \leq 20$.
- x will have at least 1 and at most 1000 characters.
- y and all intermediate results will be between -10^9 and 10^9.
- All numbers appearing in x will be between 0 and 10^9 (inclusive).

Sample Data

Input

```
3
1plus12minus3
5tothepowerof5minus3
1minus8times5tothepowerof3
```

Output

```
Case #1: 10
Case #2: 3122
Case #3: -42875
```