A+ Computer Science
Computer Science Competition
Hands-On Programming Set

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.

2. All problems have a value of 60 points.

3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

II. Point Values and Names of Problems

Number	Name
Problem 1	X-Men
Problem 2	Danger Room
Problem 3	Colossus
Problem 4	News Alert
Problem 5	Wolverine
Problem 6	Arcade
Problem 7	Mother Mold
Problem 8	Cerebro
Problem 9	Missions
Problem 10	Cyclops
Problem 11	Grading
Problem 12	Nightcrawler

For more Computer Science practice tests and materials, go to www.apluscompsci.com

1. X-Men

Program Name: XMen.java Input File: none

The X-men have decided to open a booth at the “Superhero group Bonanza 2025” and they need to you print a banner for them. The one below is really amazing like Spiderman, but way to complicated for a bunch of rookies like you guys. You need a more basic banner like the one shown in the output below.

```
=ccccc,      ,cccc       ccccc      ,cccc,  ?$$$$$$$$,  ,ccc,   -ccc
:::"$$$$bc    $$$$$     ::`$$$$$c,  : $$$$$c`:"$$$$???'`."$$$$c,:`?$$c
`::::"?$$$$c,z$$$$F     `:: ?$$$$$c,`:`$$$$$h`:`?$$$,` :::`$$$$$$c,"$$h,
  `::::."$$$$$$$$$'    ..,,,:"$$$$$$h, ?$$$$$$c`:"$$$$$$$b':"$$$$$$$$$$$c
     `::::"?$$$$$$    :"$$$$c:`$$$$$$$$d$$$P$$$b`:`?$$$c : ::`?$$c "?$$$$h,
       `:::.$$$$$$$c,`::`????":`?$$$E"?$$$$h ?$$$.`:?$$$h..,,,:"$$$,:."?$$$c
         `: $$$$$$$$$c, ::``  :::"$$$b `"$$$ :"$$$b`:`?$$$$$$$c``?$F `:: "::
          .,$$$$$"?$$$$$c,    `:::"$$$$.::"$.:: ?$$$.:.???????" `:::  ` ```
          'J$$$$P'::"?$$$$h,   `:::`?$$$c`::``:: .:: : :::::''   `
         :,$$$$$':::::`?$$$$$c,  ::: "::  ::  ` ::'   ``
        .'J$$$$F  `::::: .::::    ` :::'  `
       .: ???):     `:: :::::
       : :::::'        `
        ``
```

Input

none

Output

Output #-`X-MEN` 10 times as shown below. You must have a number before `X-MEN`.

Example Input File

```
none
```

Example Output to Screen

```
1-X-MEN
2-X-MEN
3-X-MEN
4-X-MEN
5-X-MEN
6-X-MEN
7-X-MEN
8-X-MEN
9-X-MEN
0-X-MEN
```

2. Danger Room

Program Name: Danger.java Input File: danger.dat

You did such a good job with the banner for the X men that they have now hired you to work for them full time! Your first job is to help with their danger room scheduling, as it has gotten a little out of control lately, as everyone is booking their time slots out of order. You will be given a list of time and name pairs, denoting a training start time and the name of the person who booked the Danger Room. You need to sort these based on start time and output the pairs in the correct order.

Input

The input will begin with an integer n denoting the number pairs to be sorted. Each of the following n lines will contain a time stamp in the format HH:MM, followed by a space, followed by the name of the person who booked the danger room. All times will be in 24 hours format (00:00 – 23:59), and no two people will be booked for the same time.

Output

Output the Time-Person pairs, ordered by time, with the earliest time coming first, and everyone else following in order of who's earliest.

Example Input File

```
5
15:30 Colossus
04:30 Rogue
08:00 Wolverine
17:00 Storm
11:00 Cyclops
```

Example Output to Screen

```
04:30 Rogue
08:00 Wolverine
11:00 Cyclops
15:30 Colossus
17:00 Storm
```

3. Colossus

Program Name: Colossus.java **Input File: colossus.dat**

Colossus has been training a lot, and he wants you to be his spotter! Of course, you're not strong enough to spot for him, so your job is to watch him train and determine when he adds too much weight. Colossus is known to only be able to lift 10000 lbs of weight at a time, so if he ever tries to lift more, you must tell him to stop and pull the lunk alarm so he does not get hurt.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will consist of a single integer that Colossus will try to lift. Note that the numbers Colossus lifts can be quite large.

Output

If the input number is too heavy for colossus (>10000 lbs), output the string "`Lunk Alarm!!`", otherwise print "`Hulk here we come.`".

Example Input File

```
5
9000
11000
5
10001
9999
```

Example Output to Screen

```
Hulk here we come.
Lunk Alarm!!
Hulk here we come.
Lunk Alarm!!
Hulk here we come.
```

4. News Alert

Program Name: News.java **Input File: news.dat**

You are now responsible for listening to the news while you work, and if you hear mention of any of the top X Men villains, you must sound the alarm to awaken the X Men to go save the world! Additionally, if you hear any world threats, you must sound the world wide hero alarm, as all possible help will be needed.
The names of the top X Men villains are as follows:

- Magneto
- Juggernaut
- Apocalypse
- Mystique
- Sentinels

The names of the world threats are as follows:

- Thanos
- Galactus
- Kang
- Ultron
- Graviton

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will consist of a line of words of unknown length denoting the news alerts. Any villain names will appear exactly as they are listed above, with only the first letter capitalized. Each line of words will contain only spaces and letters, no other characters will be present. Villain names appearing within other words should **not** be counted (ie Gravitonium).

Output

If any of the words in the news alert are the name of one of the world threats, output the string "Calling All Heroes.". If there are no world threats, but one of the names of one of the X Men villains are present in the news alert, output "Sharpen Your Claws Wolverine.". If there are no appearances of any of the listed threats, output "Business as Usual.".

Example Input File

```
4
The bank downtown is being attacked by Magneto
Galactus is approaching the earth
Mysterio is attacking a hockey game
Watch out for Apocalypse on a rampage in midtown today
```

Example Output to Screen

```
Sharpen Your Claws Wolverine.
Calling All Heroes.
Business as Usual.
Sharpen Your Claws Wolverine.
```

5. Wolverine

Program Name: Wolverine.java **Input File: wolverine.dat**

Wolverine is extremely old. He has been around for a while, and is having trouble with dates after his last time travelling fiasco (see Days of Future Past). He needs your help determining what dates are certain amounts of time after other dates.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will consist of 2 lines. The first will be a date in the format `MM/DD/YYYY`. The following line will contain a character, denoting the unit of time we are using, and an integer, denoting how much of that unit we are jumping ahead.
The characters can be any of the following 3:

- `D` – days
- `M` – months
- `Y` – years

Output

For each test case, output the date in the format `MMMM DD, YYYY`.

Example Input File

```
3
05/23/1987
D 12345
03/03/2003
M 153
01/03/1876
Y 369
```

Example Output to Screen

```
March 10, 2021
December 03, 2015
January 03, 2245
```

6. Arcade

Program Name: Arcade.java **Input File: arcade.dat**

Oh No! You have been trapped in the rotating claw machine jelly block maze of the villainous arcade! Quick, you need to see if you can get out before his knockout gas deploys and he traps you forever.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will begin with a line containing 3 integers, `r` and `c`, denoting the number of rows and columns in the maze respectively, and `t`, denoting the amount of time you have until the knockout gas deploys. Each of the following `r` rows will contain `c` characters made up of the following characters:

- `'J'` – denotes a jelly block. If the maze rotates and puts you into a jelly block, you are stuck until the maze rotates again.
- `'.'` – denotes an empty space.
- `'S'` – denotes your starting position in the maze.
- `'E'` – denotes the exit from the maze.

Note: This is a rotating maze. Every second that passes this maze will rotate 90 degrees clockwise. When it rotates, you will be lifted out of the maze via claw machine, and put back in the exact row and column you were in before the rotation, then you will have one second to make a move before the next rotation, unless you are stuck in a jelly block. The maze is run by a malevolent AI who can read your mind (its based on the one and only Charles Xavier), so that it will rotate right before you make your move (basically the maze rotates as you're taking a step, or as you spend a second stuck in a jelly block).

Every second in the maze you can take one step in any of the 4 main directions (up, down, left, right), no diagonals. Every maze will also be a square (r = c).

Output

If it is possible to escape the maze in time, output "`Escaped the Arcade in n steps.`", otherwise output "`Total Knockout.`".

Example Input File

```
2
5 5 10
SJ.JJ
JJ..J
EJJJ.
.J.JJ
JJJ..
6 6 15
JJ.SJJ
J.J.JJ
J.J.JJ
.J.J.J
JJJJ.J
EJJ.J.
```

Example Output to Screen

```
Escaped the Arcade in 5 steps.
Total Knockout.
```

7. Mother Mold

Program Name: Mother.java **Input File: mother.dat**

You have escaped Arcade's Maze only to be captured again, this time by Bolivar Trask and his dreaded Mother Mold. They have heard about all the work you were doing for the X-Men, and they'd like you to do some calculations about the rate at which his Mother Mold can make Master Molds, which in turn can make Sentinels. They would like a simulation of how many Sentinels will be made after a given n cycles of the Mother Mold running. The Mother Mold and subsequent Master Molds behave as follows:

- The Mother Mold will create exactly 1 Master Mold every cycle.
- Upon creation, a Master Mold will take 2 cycles to create another Master Mold.
- After creating the other Master Mold, the original Master Mold will take 1 cycles to recuperate.
- For every subsequent cycle, the Master Mold will create a Sentinel.

Input

The first line will contain a single integer `n` (0 <= `n` <=1000000) that indicates the number of data sets that follow. Each data set will consist of a single integer denoting the amount of time which has passed in the simulation of the Mother Mold's output.

Output

Output the number of sentinels created by the Mother Mold after `n` cycles have passed.

Example Input File

```
3
7
10
21
```

Example Output to Screen

```
13
50
615
```

8. Cerebro

Program Name: Cerebro.java **Input File: cerebro.dat**

You have been called down to the Cerebro room to help fix the professor's machine. You think you're done but he now needs your help testing it, and wants you to write a program to test against his superior mind. You need to determine if is possible for the professor to access someone, given that he can only access the minds of people he knows, and then from there only people they know, and so on. Connections only go one way, so just you knowing Johnny only mean Xavier could go from you to Johnny, not the other way around.

Input

The first line will contain 2 integers `n` and `m`, that indicate the number of data sets that follow, and the number of connections we know about, respectively. Each of the following `m` lines will consist of 2 space separated names, which are two people who know each other. Xavier will always be denoted as `"Xavier"`, and all other people can be named with any combination of alphabetic characters. The first person will be connected to the second person, but not vice versa. After `m` lines of connections, the next `n` lines will each contain a name of a person who Xavier will try to connect to.

Output

For each test case, if it is possible to Xavier to get to the given person, output "`Mutant Located.`", otherwise output "`Nowhere to be found.`".

Example Input File

```
3 10
Xavier Jean
Xavier Erik
Erik Raven
Raven Xavier
Jean Erik
Logan Erik
Erik Xavier
Xavier Logan
Bastion Xavier
Bastion Erik
Raven
Jean
Bastion
```

Example Output to Screen

```
Mutant Located.
Mutant Located.
Nowhere to be found.
```

9. Missions

Program Name: Missions.java Input File: missions.dat

You are being given a lot of authority in the mansion these days. Your latest job has to do with X Men missions. You will be given a mission, and a list of skills needed for this mission. Given the list of X men and all of their talents, find out how many different combinations of X Men can go on this specific mission. If an X Man is used to one skill, they cannot be re used for another. The X Men and their skills are as follows:

X Man Name:	**Skills:**
Colossus	Strength, Invulnerability
Wolverine	Hand-to-Hand, Invulnerability
Nightcrawler	Teleportation, Swords, Hand-to-Hand, Agility
Storm	Flight, Electric, Cold
Banshee	Flight, Energy-Blasts
Cyclops	Hand-to-Hand, Energy-Blasts
Phoenix	Flight, Telekinesis, Telepathy, Energy-Blasts
Iceman	Cold, Invulnerability
Beast	Strength, Agility
Angel	Flight, Swords

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will consist of a list of space separated words denoting all the skills required for this particular mission.

Output

Output how many unique combinations of X Men can complete the mission as listed. Then on the following line, output the alphabetically first combination of X Men for the mission (Xavier has this thing with alphabetizing). If there are no possible combinations, just output 0, and then an empty line after 0 (the test case should be two lines, same as all the others). The combination should be listed with the X Men in the same order as the abilities (if Electric is the first skill, then Storm has to be first).

Example Input File

```
3
Strength Flight Agility
Electric Flight Swords Invulnerability
Teleportation Telekinesis Electric
```

Example Output to Screen

```
12
Beast Angel Nightcrawler
15
Storm Angel Nightcrawler Colossus
1
Nightcrawler Phoenix Storm
```

10. Cyclops

Program Name: Cyclops.java **Input File: cyclops.dat**

Cyclops is trapped in a burning building! He needs you to take a structural readout of the wall in front of him, and tell him if he has enough power to destroy at least half of it, which would allow him to escape, otherwise he needs to call Iceman.

Input

The first line will contain a single integer `n` that indicates the number of test cases to follow. Each test case will begin with 3 space separated integers, r, c, and p, denoting the number of rows and columns in the structural readout of the wall, and how much power cyclops has left. The following r lines will contain c characters denoting the structural readout of the wall. Each character in the readout will be a digit from 0-9, denoting the amount of power it will take to destroy that section of the wall.

Output

If cyclops can destroy the wall in front of him, output the string "`Made it with n power to spare.`", where `n` is the amount of power cyclops has left after escaping. If cyclops does not have enough power to destroy at least half the wall, output "`Better Call Iceman.`". If cyclops uses all of his power to destroy exactly half the wall, he makes it out with 0 power to spare. Cyclops cannot destroy parts of wall sections, so if he cannot FULLY destroy at least half the wall sections, he cannot get out.

Example Input File

```
2
5 5 15
34857
95783
12854
42856
24819
3 6 20
123456
654321
123321
```

Example Output to Screen

```
Better Call Iceman.
Made it with 5 power to spare.
```

11. Grading

Program Name: Grading.java **Input File: grading.dat**

Many do not know this, but Xavier's school for gifted mutants is in fact a school. You are well aware of this, as they like you so much that you are now responsible for grading all the final exams and determining class rank. You will be given keys and answer sheets for all students for each of their classes, and then you must grade the exams, and determine class ranks based on the average of all these exam grades (Xavier's school is weird and the only grade anyone gets all semester is the final exam).

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will begin with a string of characters (`A-E`) denoting the key for this exam, followed by a space, followed by an integer, `s`, denoting the number of students who took the exam. Each of the following `s` lines will contain two space separated strings, the first name of the student (will be unique), followed a string of characters (`A-E`) denoting their answers. Each exam will be graded out of 100 points, with all questions being worth the same amount across a single exam. Each answer sheet will always have the same number of answers as the key

Output

Output the students in order by class rank, in the following format: `#N: Name, Grade1 Grade2 Grade3 …` . Each of the grades is one of the exam grades, and these should be order from highest to lowest, with each formatted to two decimal places. Class rank is determined by the average of all the exam scores for each student, and the students will be in order from highest to lowest average.

Example Input File

```
3
AEBCDBEACD 4
Jean AEBCBBAACD
Bobby AAECDBEADD
Warren AEBCDBEADD
Rogue AEBCCBEACD
AABBDDCCEEBDCADEA 3
Jean AABBDECCEEBCCCDEA
Warren AABBDDBBBEBDCADEA
Bobby AABCDECCEEBDCADAA
ABCDE 3
Jean ABCDE
Rogue ABBDE
Bobby ACBDE
```

Example Output to Screen

```
#1: Jean, 100.00 82.35 80.00
#2: Warren, 90.00 82.35
#3: Rogue, 90.00 80.00
#4: Bobby, 82.35 70.00 60.00
```

12. Nightcrawler

Program Name: Nightcrawler.java **Input File: nightcrawler.dat**

You're helping Nightcrawler with his teleportation in the Danger Room and he wants to test how far he'd be able to teleport. He can only teleport to a place he's been to before. You will be given a list of points (X,Y) where Nightcrawler can teleport to and from, and you need to find the maximum distance he can teleport between two of these points.

Input

The first line will contain a single integer `n` (2 <= `n` <= 100)that indicates the number of data sets that follow. Each data set will consist of one line with an unknown number of points in the format (`X`,`Y`), where `X` and `Y` are floating point numbers formatted to two decimal points.

Output

For each test case, output the maximum distance Nightcrawler can possibly teleport between two points, formatted to 2 decimal points.

Example Input File

```
3
(3.00,3.00) (4.00,4.00) (5.00,5.00)
(0.00,0.00) (-5.00,-5.00) (5.00,-5.00)
(4.32,6.12) (5.40,-8.59) (10.43,-4.32) (-3.24,9.00) (-4.32,-8.54)
```

Example Output to Screen

```
2.83
10.00
19.60
```