SEVEN LAKES

KICK OFF CLASSIC

SKOOL DAYZ

1. Pass Fail

Input File: passfail.in **Output Method:** Standard out

Problem Description:
Billy, your lazy friend, has come to you to ask if you can make him a program that will tell him which classes he is failing. While you don't quite understand why on Earth he needs a program to do this, you comply nevertheless because it should be no problem. Right?

Note: Below a 70 is considered failing, while a 70 or above is passing.

Input Description:
The first line will be the name of the person. The following 7 lines will have 2 parts, separated by a space: the first being the name of a class (without spaces) and the second part being the integer grade the person received in that class.

Output Description:
Output all classes which your friend failed in the order in which they appeared.

Sample Input:
```
Billy Bobby
English 80
Spanish 90
Math 143
History 64
CompSci 100
Band 53
Science 98
```

Sample Output:
```
History
Band
```

2. Sleeping Sally

Input File: sleepingsally.in **Output Method:** Standard out

Problem Description:
Mrs. Helda, the strict teacher of your Advanced Underwater Basket Weaving (AP) class, is currently out. In her absence, your friend Sally has fallen asleep! Mrs. Helda strongly disapproves of sleeping children, and so you set out on a journey to wake your friend before she suffers possible…consequences. Given a 2D character grid which outlines your classroom, there are the following symbols:

– Obstruction
. – Open Space
B – Yourself
S – Sally

Travelling 1 tile in the grid requires 1 second, and you can only travel in the 4 cardinal directions in each step. Your objective is to print out whether or not you can successfully reach Sally in a specified time limit until Mrs. Helda returns, given the outline of the grid. There will always be a valid path between you and Sally. Note that you must actually move into Sally's grid position to wake her up - not an adjacent tile.

Note: If Mrs. Helda returns in 8 seconds and it takes 8 seconds to go and wake up Sally, it is still considered a success.

Input Description:
The first integer **N** describes the number of data sets to follow. The next **N** sets each start with 2 numbers on 1 line, **R** and **C**, which denote the number of rows and columns of the grid to follow respectively. The next **R** lines each contain **C** characters that represent your map, with only the aforementioned characters. The final line contains a single integer which states the number of seconds until Mrs. Helda returns.

Output Description:
For each data set, print out "Success" if you can successfully save Sally, or "Failure" if she would suffer certain consequences.

Sample Input:
```
2
5 5
..B..
.###.
.....
.....
###S#
8
2 2
B.
.S
1
```

Sample Output:
```
Success
Failure
```

3. Boo Lee

Input File: boolee.in **Output Method:** Standard out

Problem Description:
Your very nice friend, Boo Lee, just recently threa- *ahem* asked you to do him a favor. Boo Lee is struggling in Math. He has already tried everything he can think of to "help" him get better. Despite his failures he came to the sudden realization of something that he can use to chea- *AHEM* help him with his studies. Boo Lee wants you to create a simple calculator.

These are the operations you need to implement.

ADD - Addition
SUB - Subtraction
MUL - Multiplication
DIV - Division

In the case of more than 2 operands, apply the operation on the first 2 numbers, and then on the result of the first two and the third number, and the result of that and the fourth number, and so on.

Input Description:
Your calculator will process a simple text file. There will be a number at the top of file indicating how many lines of math problems will follow. Each line will consist of a three letter operation word and then any number of space-separated operands. Operands may be decimals.

Output Description:
You are to output the operation that took place followed by the solution. Round all answers to 2 decimal places.

Sample Input:
```
3
ADD 2 2 2
SUB 2 2
DIV 4 2 3
```

Sample Output:
```
ADD 6.00
SUB 0.00
DIV 0.67
```

4. Tens

Input File: tens.in **Output Method:** Standard out

Problem Description:
You like 10. It is and has always been your favorite number. Divisible only by itself, 1, 2, and 5, it is by far the best composite number in history (you also like 2 and 5). To profess your love of this number to your beloved math teacher, you've decided to write a program to print out various powers of 10. Now go and enjoy powering your most favoritest number.

Given a positive integer **X**, print out 10^{X}.

Input Description:
The first integer **N** represents the number of data sets to follow. Each data set will be on a separate line, each with a single integer representing the value of **X** that you must bring 10 to the power of.

Output Description:
Print the resulting number for each data set. Each data set must be on a separate line.

Sample Input:
```
3
1
10
2
```

Sample Output:
```
10
10000000000
100
```

5. Chess

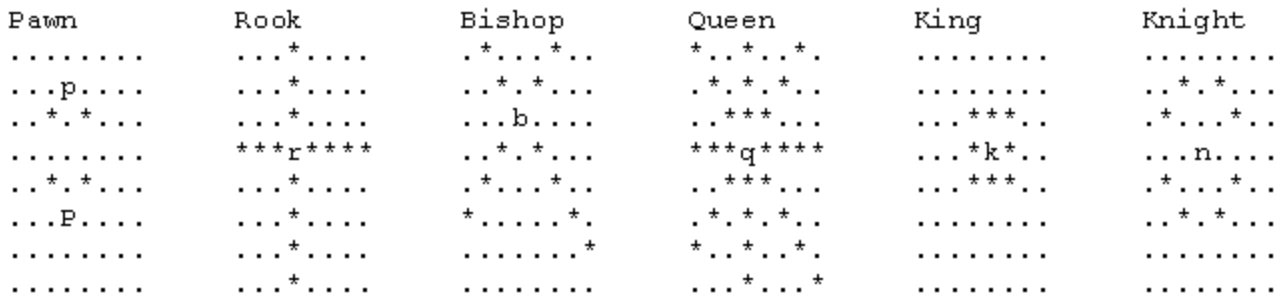
Input File: chess.in **Output Method:** Standard out

Problem Description:
You are the captain of the school chess team and, conveniently, also happen to be handy with computers. At the beginning of the school year, your club always receives a good number of new participants, but some are not so keen to the rules of the game. To help curb this, you've decided to write a few programs to help teach the new players. The first will demonstrate the concept of being "in check."

Your task is to write a program that reads a chessboard configuration and identifies whether a king is in check. A king is in check if it is on a square that can be taken by the opponent on his next move.

White pieces will be represented by uppercase letters, and black pieces by lowercase letters. The white side will always be on the bottom of the board, with the black side always on the top.

Here are the ranges of capturing for each piece.



Each piece is denoted as such (black and white, respectively):
Pawns are denoted as 'p' or 'P'
Rooks are denoted as 'r' or 'R'
Bishops are denoted as 'b' or 'B'
Queens are denoted as 'q' or 'Q'
Kings are denoted as 'k' or 'K'
Knights are denoted as 'n' or 'N'

The capturing spaces for each piece are defined above. A '*' denotes a space that can be captured by the piece in its next move. Remember that the range for pawns differs for each player and that the knight is the only piece that may jump over other pieces. Other pieces may NOT move through obstructions when capturing (see sample data).

Input Description:
There will be an unknown number of boards. Each board will be made up of an 8 by 8 grid. A '.' denotes an empty space. The denotations for each piece can be found in the problem description. **Note**: There will be no cases in the input where both kings are in check.

Output Description:
For each board, output "*** king is in check" where *** is either "black" or "white" depending on which king is in check. If no king is in check, output "no king is in check"

Sample Input:

```
..k.....
ppp.pppp
........
.R...B..
........
........
PPPPPPPP
K.......

rnbqk.nr
ppp..ppp
....p...
...p....
.bPP....
.....N..
PP..PPPP
RNBQKB.R

........
.b......
..k.....
........
....K...
.....B..
........
........
```

Sample Output:

```
black king is in check
white king is in check
no king is in check
```

6. Average

Input File: average.in **Output Method:** Standard out

Problem Description:
Your harried Math teacher has 180 students she has to deal with, and the grading program the school provides is currently not working! To help your teacher, you have so graciously decided to make an averaging program to score students. For each set, the teacher will provide the name of the student, followed by the grades (and what category they belong to) the student received. There are three categories: Homework (10%), Minor (20%), and Test (70%). Weighted scoring works on the following basis:

1. Find the average of each category. If no are in a specified category, the average is assumed to be a 100.
2. Multiply the average of each category by the worth of that category (the worth is given as a percentage, but is really a decimal like .70).
3. Add all of the averages multiplied by each worth together togrades get the end grade.

For example, Billy is entered into the grading program. He has 2 100 Homework grades, an 80 for a quiz grade, and a 90 for a test grade. His final grade is (100 * 0.1) + (80 * 0.2) + (90 * 0.7) = 89.

Input Description:
The first line will have a number **N** which is the number of data sets to follow. The next **N** sets each consist of the student's first and then last name on the first line. Following this line is another number **X** which denotes the number of grades for this student. The next **X** lines have the grades the student earned, in the format "[Category] [Grade]", where Category is one of the grade categories Homework, Minor, and Test and Grade is the integer grade the student received.

Output Description:
For each set, output the name of the student in the format "Last, First" followed by a space and the grade the student receives, rounded to two decimal places.

Sample Input:
```
1
Samuel Schniggles
4
Homework 100
Homework 100
Minor 80
Test 90
```

Sample Output:
```
Schniggles, Samuel 89.00
```

7. Hipster

Input File: hipster.in **Output Method:** Standard out

Problem Description:
The Seven Lakes team has a hipster on it; Hipster Colby. Hipster Colby types on a dvorak keyboard, which causes problem when other people try to type on his computer. His keyboard looks like a normal keyboard, so no one realizes the difference until they've already typed some gibberish. Somebody has just been on his computer, typing something. Convert the gibberish back to plain text! It can be assumed that the person typed as though they were on a standard QWERTY keyboard. A picture of both layouts have been provided.

Qwerty

Dvorak

Input Description:
The first line will contain a single integer **N** representing the number of lines to follow. For each of the next **N** lines, convert the text to the intended message, using the references above (the line was typed on a dvorak keyboard as though it were a qwerty keyboard). Only lowercase letters and the punctuation marks ‘ , and . will be used in input; the output will only contain lowercase letters.

Output Description:
Output the decoded messages, one per line. There may be a blank line at the end of the output.

Sample Input:

```
2
ekrpat co k.pf ocnnt
yd. ‘gcjt xpr,b urq hgmlo rk.p yd. na;f eri
```

Sample Output:

```
dvorak is very silly
the quick brown fox jumps over the lazy dog
```

8. Weird Sort

Input File: weirdsort.in **Output Method:** Standard out

Problem Description:
Your English teacher is quite progressive, and has decided that the old form of alphabetizing is too old-fashioned. As such, your teacher is devising a new way to order words, and has given you the prototype for sorting 3 character words.

Given a list of 3 character words, sort the words by this priority of characters:

2nd, 3rd, 1st

Meaning that instead of the 1st letter, the 2nd letter has the highest priority when sorting (so the word with the 2nd letter that is nearer the beginning of the alphabet goes first). If two words have the same 2nd letter, then the 3rd is used to compare the words. If the 3rd is the same, the 1st character is used. Only if all of the characters are the same can the words be declared the same/equal.

Input Description:
The first integer **N** represents the number of data sets to follow. The next **N** lines each contain a single data set with an unknown number of words, separated by spaces.

Output Description:
Output the list of words sorted alphabetically, but based on the given character priorities.

Sample Input:
```
3
abc bcd hgf hgf bcd
djk aaa zzz ehc ehc ehc chc
gdc gdc gfe gac aac
```

Sample Output:
```
abc bcd bcd hgf hgf
aaa chc ehc ehc ehc djk zzz
aac gac gdc gdc gfe
```

9. Lithp

Input File: lithp.in **Output Method:** Thtandard out

Problem Dethcription:
It’th the firth day of thchool, tho it’th very important that you make a good firtht imprethion. Unfortunately, you’ve jutht bitten your tongue and now talk with a thlight lithp and can’t pronounce “s” properly. You probably jutht got nervouth or thomething. Bethideth, aren’t lithpth thupothed to be cool or thomething?

Input Dethcription:
The firtht line of the input dethrciption will be the number of data theth (sets) to follow. For each data thet, a thingle line will be given. Thome words in the thentence may begin with a capital letter, but there will be no thentenceth or wordth that are in all capth.

Output Dethcription:
For each data thet, output the line with all “s’’th replaced with “th”. If there is an “sh”, then convert both letterth to “th” (*not* to “thh”). Proper grammar ith important, even with a lithp, so make thure that any “S” that were capitalized to begin with tranthlate to a capitalized “Th” and any punctuation thould remain in place.

Thample Input:

```
3
I have a lisp.
These problems are silly.
She said Shanna did it.
```

Thample Output:

```
I have a lithp.
Thethe problemth are thilly.
The thaid Thanna did it.
```

10. Ms. Schniggles

Input File: msschniggles.in **Output Method:** Standard out

Problem Description:
Your AP underwater basket weaving teacher is absent today, and in her place is the less than desirable Ms. Schniggles. Ms. Schniggles has a reputation for being eccentric and harsh in her teaching methods. She strongly believes in essays as a form of punishment. Because one of your friends mispronounced her name (it's Schni*gge*s not Schnigg*als*), your whole class has been assigned an essay of sorts. Ms. Schniggles will give you a sentence, word, or phrase, and you must write that a certain number of times.
If you talk while solving this problem, Ms. Schniggles **will** leave a note for your teacher!

Input Description:
The first integer **N** will represent the number of data sets to follow.
Each data set will be two lines: The first representing what must be written and the second being an integer representing the number of times it must be written.

Output Description:
For each data set, output the phrase written the correct number of times, each time on a new line. Output a blank line between test cases. A blank line at the end of the file is optional.

Sample Input:
```
3
I will not mispronounce Ms. Schniggles.
3
I will be quiet in class.
1
I will not make fun of Ms. Schniggles' hat or funny name.
5
```

Sample Output:
```
I will not mispronounce Ms. Schniggles.
I will not mispronounce Ms. Schniggles.
I will not mispronounce Ms. Schniggles.

I will be quiet in class.

I will not make fun of Ms. Schniggles' hat or funny name.
I will not make fun of Ms. Schniggles' hat or funny name.
I will not make fun of Ms. Schniggles' hat or funny name.
I will not make fun of Ms. Schniggles' hat or funny name.
I will not make fun of Ms. Schniggles' hat or funny name.
```

11. The Easy Things in Life

Input File: None **Output Method:** Standard out

Problem Description:
There can be many challenging tests and problems in school, which can cause a great deal of stress and discouragement. However, buried in the pile of annoying homework and difficult classes always lies a blow-off class, like Physics or BC Calculus (whatever floats your boat).

Input Description:
Nothing. You're literally just printing something out.

Output Description:
Print out the line exactly as shown in the Sample Output.

Sample Output:

```
This was unexpectedly easy.
```

12. Meta Numbers

Input File: metanumbers.in **Output Method:** Standard out

Problem Description:
Ms. Schniggles is at it again, this time as your math substitute. Because she thinks the class is far too loud, Ms. Schniggles has given you an absolutely ridiculous problem to solve. She wants you to find all meta numbers within certain ranges (very large ranges, too). Luckily, you know how to program, which simplifies things greatly.

Given a range, return the number of Meta numbers within that range, inclusive. A Meta number is a positive integer that can be defined as x^x where x is a positive integer. For example, 27 is a Meta number because it can be written as 3^3.
The first few Meta numbers are 1, 4, 27, 256…

Input Description:
The first integer **N** represents the number of data sets to follow. Each data set will be on a separate line, containing two numbers (**A** and **B**) representing the range, inclusive.

Output Description:
For each data set, output the number of meta numbers within the range. Each data set must be on a separate line.

Assumptions:
$1 <= \mathbf{A} <= 10^{1000}$
$1 <= \mathbf{B} <= 10^{1000}$
A <= **B**

Sample Input:
```
3
1 9
2 8
27 1000
```

Sample Output:
```
2
1
2
```

13. Multiplication Square

Input File: multiplicationsquare.in **Output Method:** Standard out

Problem Description:
You and your calculus class have decided that all of these integrals and derivatives are far too easy for your vast intellect. Instead, you've decided to practice your valuable multiplication skills. And to make things just a tad interesting, you've come up with a creative task to test your proficiency.

Given a number **N**, write the multiplication square from 1 to **N**.
A multiplication square is an **N** by **N** grid of numbers where the first row and first column show the numbers from 1 to **N** in sequential order. For all other coordinates, the value can be found by multiplying (row+1)*(column+1). Note that the first row and first column are both noted as row 0 and column 0 respectively.

Input Description:
There will be only one number, **N**, in the input.

Output Description:
Output the multiplication square from 1 to **N**. Each number in the square must be separated by a single space.

Sample Input:
```
5
```

Sample Output:
```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

14. Mirror esreveR

Input eliF: esrever.in **Output dohteM:** Standard out

melborP Description:
Mirrors era rather gniyonna, they tcartsid you htiw their revelc reflections. As uoy can ylbaborp see ew have etiuq the tnemaciderp. We deen you ot help su fix ruo sentences. Wait, it seems to be gniraelc up... ro not.

Note: The input file is actually esrever.in; it was not just affected by the strange mirror disease like the rest of this melborP description.

Input noitpircseD:
The first line will consist of a single integer **N**. There will be **N** data sets. For every **N**th data set, there will be a single line which has every second word reversed.

tuptuO Description:
For every data set, output the corrected sentence where every second word has been un-reversed - remember to maintain punctuation if present.

Sample tupnI:

```
3
Did uoy know?
Leonardo ad Vinci etorw his seton backwards ot protect sih findings.
Mirrors era rather lufesu!
```

elpmaS Output:

```
Did you know?
Leonardo da Vinci wrote his notes backwards to protect his findings.
Mirrors are rather useful!
```

15. Hills

Input File: hills.in **Output Method:** Standard out

Problem Description:
You live in the wonderful land of the hills. There are so many distinct hills that your house is on one. Your school is on another. And everything else has its own hill as well. The hill under your school is so striking that it made you want to pay homage to it. You have decided to write a program that will make ASCII hills of any size. Unfortunately you programing skills aren't the best so the hills look like triangles.

Input Description:
The first number represents how many numbers will follow. For each number **N** to follow, create a hill of height **N**. **N** will always be at least 3.

Output Description:
Each hill should be separated by a single line. The format of each hill should match that of the hills in the sample output.

Sample Input:

```
2
3
4
```

Sample Output:

```
____*____
__*_*_*__
*_*_*_*_*

______*______
____*_*_*____
__*_*_*_*_*__
*_*_*_*_*_*_*
```

16. Histo's Secrets

Input File: histo.in **Output Method:** Standard out

Problem Description:
Mr. Histo was the sponsor of the school's literature club until recently. He sadly passed away due to mysterious causes (the police say "natural causes"). To discover if he really wasn't murdered a detective was contracted to research him. He discovered many things but still hasn't got anything solid. However, just recently a stash of letters was found, all addressed to Mr. Histo. They are, however, not normal. They solely consist of words repeated over and over again. The wonderful detective has come to you to create a program that creates frequency charts, as he suspects that the frequency of the words is key to understanding their true meaning.

Input Description:
The first number **N** represents the number of data sets to follow. The next **N** data sets, each on a single line, will consist of any number of words. All words will only consist of alphabetical letters (eg. A-Z).

Output Description:
For each data set, print out the frequency chart of the words in alphabetical order, in the format shown in the sample output. Case does not matter when determining frequency. An extra new line at the end of the output is acceptable.

Sample Input:
```
2
How How Are Are You Are How You
L L L H E L E L H E E E E LL L L LL LL H H O
```

Sample Output:
```
ARE 3
HOW 3
YOU 2

E 6
H 4
L 7
LL 3
O 1
```

17. Free Response

Input File: freeresponse.in **Output Method:** Standard out

Problem Description:
You, being the wonderful person that you are, have decided to write a free response grading program to ease the life of your Chemistry teacher. Your teacher will provide you with a key as well as all of the free response answers; it is your job to output the number of correct answers while refraining from laughing at some of the less intelligent answers.

Input Description:
The file will start with a number **N**, which denotes the number of answers in the key. The next **N** lines will contain the key for each question, one answer per line. Following the key will be a number **M** on its own line, which denotes the number of sets to follow. Each set contains **N** lines, each corresponding to an answer in the key. When grading, case matters (it's Chemistry class, after all), so answering "no2" when the correct answer is "NO2" would be incorrect.

Output Description:
For each set, print out the number of correct answers followed by a forward-slash and the number of total answers (aka the number of answers in the key) as in the following format (ignoring the bolding, of course): "Correct/**N**".

Sample Input:
```
3
hello
maybe
Charles Darwin
2
maybe
no
Charles Darwin
hello
maybe
einstein
```

Sample Output:
```
1/3
2/3
```

18. Inverse Hills

Input File: inversehills.in **Output Method:** Standard out

Problem Description:
Your friend lives in the wonderful land of hills – so wonderful, in fact, that his Computer Science teacher assigned a lab to determine the height of a hill given its ASCII representation (provided by a lab from the previous year of Computer Science students). Your friend is not particularly good at computer science, and has come begging/asking you to help him out.

Input Description:
The first line represents the number of ASCII hills that will follow. Each hill will be an unknown number of lines long, and will terminate with a blank line. The last hill will terminate with a blank line as well.

Output Description:
Output the height of the hill, given that each hill ends with a blank line (the height does not include the blank line).

Sample Input:

```
2
____*____
__*_*_*__
*_*_*_*_*

______*______
____*_*_*____
__*_*_*_*_*__
*_*_*_*_*_*_*
```

Sample Output:

```
3
4
```