Programmierbeispiele

ADVENTCODING 2017

Hello Santa!

Congratulations! If you can read this it means you have successfully managed to register, retrieve your password, log in and find the description of problem A!

Problem

Santa greets you, greet back! Write a program that writes "Hello Santa Claus!" (without the quotes) to the console.

You can use the following programming languages:

C
: use extension ".c" to indicate a program in C. The main function must return 0. Don't forget to output a newline.

C++
: use extension ".cpp" to indicate a program in C++. Don't forget to output a newline.

Java
: use extension ".java" to indicate a program in Java. Everything must in one file, the classname must match the filename. Do not put your class into a package.

C#
: use extension ".cs" to indicate a program in C#.

Python 2.x
: use extension ".py" to indicate a program in Python 2.x

Python 3.x
: use extension ".py3" to indicate a program in Python 3.x

Perl
: use extension ".pl" to indicate a program in Perl

Sample Output

```
Hello Santa Claus!
```

Weekdays in December (easy)

Sometimes, advent can be a terribly stressy time. One has to finish things that need to be done this year, get presents for you loved ones and organize christmas dinner. Eventually, one does not know anymore what weekday it is.

Problem

Implement a program that tells you the current weekday based on the date in December for this year. An input line will have the format "December n" where n is an integer number between 1 and 31. Calculate the respective weekday and print it (English language, capitalized first letter) to standard output followed by a newline. If the input consists of the word "end", the program should terminate. There will be no other input types except from the ones described.

Sample Input

```
December 1
December 16
December 7
end
```

Sample Output

```
Friday
Saturday
Thursday
```

Reusing candles (hard)

On each of the four sundays before Christmas, Mr. MacThrifty is lighting candles. On the first Sunday it is one candle, on the second it is two candles, on the third three candles and on the fourth all four candles. Each sunday, any of four candles can be used, but it is not possible to switch between candles during the day. The candles will burn for a fixed time where Mr. MacThrifty holds a session of contemplation. The session time is the same on each sunday. Since Mr. MacThrifty is a bit... thrifty ..., he is going to reuse last years candles, that is four candles with various lengths.

Problem

Write a program that calculates the maximum session time for a given set of four candles. Your program shall read lines with four space-separated integer values corresponding to the candle lengths in cm. For each cm a candle will burn for one hour. Calculate and print the maximum possible session length with these candles in minutes, then read the next line. When a line contains the word "end", the program shall terminate.

Sample Input

```
1 2 3 4
1 1 1 1
end
```

Sample Output

```
60
20
```

There's snow outside (easy)

Alice lives in an area with quite unstable weather. Sometimes it snows heavily, then the next day is sunshine, followed by rain. Alice often gets confused if there is snow outside (which means she has to get up earlier and shovel snow) or not (which means Alice can sleep in).

Problem

Implement a program that tells Alice the estimated snow height based on the metereological history. Read several lines with the weather, which can be (their effects are given in brackets):

```
heavy snowing (snow +10 cm)
snowing (snow +5 cm)
light rain (snow -3 cm)
heavy rain (snow -8 cm)
cloudy (+- 0 cm)
sunshine (snow melts down by 1cm)
```

The initial snow height is 0 cm. Your program shall output the snow height in cm after each day. When reading the word "end" the program shall terminate.

Sample Input

```
heavy snowing
heavy rain
snowing
light rain
cloudy
sunshine
end
```

Sample Output

```
10
2
7
4
4
3
```

Santagram Finder (medium)

Santa is a big fan of anagrams. An anagram is a word or phrase that exactly reproduces the letters of a given phrase or word in another order. Santa even invented his own version of anagrams, the so called Santagrams, which are build by rearranging the letters of "SANTA CLAUS", for example into "LUCAS SATAN".

Problem

Implement a program that finds Santagrams in a line of text. Whenever a sequence of words forms a Santagram, a line containing these words shall be written to the console in uppercase letters separated by spaces. Spaces, periods or commas separate words to be analyzed but should be ignored in the Santagram. Also the case should be ignored. In order to be a Santagram, the order of the letters must change, so the original "SANTA CLAUS" is not a Santagram. After processing one line of input, the programm shall terminate.

Example

Input

```
Santa Claus thinks Satan Lucas is a class aunt.
```

Output

```
SATAN LUCAS
A CLASS AUNT
```

Missed icehockey game (hard)

Santa loves to attend the games of the elvish icehockey league, but in December he is often busy with collecting wishlists so that he misses most of the games. Santa still loves to check the results and tries to imagine in what order the goals of a game might have fallen.

Problem

Write a program that calculates the number of different possibilities how a given score might have developed. For each game the programm shall read lines containing the score after the first period, the second perion and the final score. For each line, the number of possibilities should be written to standard output. When a line contains the word "end", the program shall terminate.

Sample Input

```
0:0 0:0 0:1
1:2 2:2 4:3
end
```

Sample Output

```
1
9
```

Explanation: We can be sure that the first game went from 0:0 to 0:1 in the last period. The second game might have developed as 1:0->1:1->1:2 or 0:1->1:1->1:2 or 0:1->0:2->1:2 in the first period, then it went from 1:2 to 2:2 in the second period and finally might have developed as 3:2->4:2->4:3 or 3:2->3:3->4:3 or 2:3->3:3->4:3 in the last period, yielding nine different possible developments over the whole game.

Snowball Duel (medium)

Santa's elves enjoy playing in the snow when they have a break. Their favorite game is the "snowball duel". Two elves place themselves at a distance and start throwing snowballs at each other. Each throw is synchronized and can have the following outcomes: an elf is hit and the other snowball missed, both elves are hit or both snowballs missed. The game is repeated until at least one elf is hit.

Problem

Write a program that calculates the probability for a win of either elf or a draw (both elves are hit). The input to the program are lines with two probabilities, expressed as a number between 0 and 1, which state the chance that an elf will hit the other with a snowball. The program shall output the probability that the first elf wins the duel, the probability for a draw (both elves are hit) and the probability for the second elf to win. The output probabilities should be rounded to two digits after the comma and printed with these digits (even if they are 0), separated by a space. The program shall terminate when the message "Back to work!" is read.

Example

Input

```
0.2 0.9
0.5 0.5
Back to work!
```

Output

```
0.02 0.20 0.78
0.33 0.33 0.33
```

There's snow outside (hard)

As we already know from Problem D, Alice lives in an area with quite unstable weather. Sometimes it snows heavily, then the next day is sunshine, followed by rain. Alice wants you to come up with a weather prediction program.

Problem

Implement a program that predicts the weather for the next day based on previous patterns. The weather possibilities from best wo worst are:

```
sunny
cloudy
snow
rain
```

Your program should read the weather of one day and make a prediction based on the longest sequence in a stored history that matches the latest weather changes. If the are multiple possibilities for the longest match, the answer with the best weather should be chosen out of the possible answers. The weather of the day shall then be added to the history. When reading the word "end" the program shall write the precentage of successful guesses (rounded to integer percent followed by a percent sign) and terminate.

Sample Input

```
snow
rain
snow
snow
rain
snow
cloudy
rain
snow
end
```

Sample Output

```
sunny
sunny
rain
snow
snow
snow
sunny
snow
cloudy
25%
```

Explanation of the output:

We read "snow", but there is no history, so the best weather is predicted.

We read "rain", in the history there is just "snow", so after "rain" everything can happen, we predict "sunny" therefore.

We read "snow" and find it in the history, followed by "rain", so this is what we predict.

We read "snow" and find it in the history, followed by "rain" or by snow, so we predict "snow".
We read "rain" and we find a sequence of two matching days "snow"->"rain", so we predict "snow".
We read "snow" and we find "snow"->"rain"->"snow" in the history, we predict "snow".
We read "cloudy" but don't find it in the history, so we predict sunny.
We read "rain" and predict "snow".
We read "snow" and find two matching sequences "rain"->"snow"->X, we predict "cloudy" because it is the better weather.
The program was correct in 2 out of 8 guesses (the last guess could not be validated).

Feeling the music (very hard)

Kram the elf got the task to get all CDs with Santas favorite songs from a box. Unfortunately, the box contains Santa's music CDs as well as some old computer programs. For almost every person with a bit of music feeling it would be easy to determine the difference between regular music and basically randomly composed bytes, but Kram unfortunately is unable to tell the difference. Can you help Kram out with a program? Luckily, Santa does not like disharmonics in music (as it is for example common in Jazz music).

Problem

Write a program that reads lines containing a few notes which might be music or the artefact of arbitraryly assembled data. Each line contains 8 to 12 tunes encoded as notes "C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", A#", or "B". Your program shall output the line "music", when it appears to be music or "no music" when it doesn't. When a line contains the word "end", the program shall terminate.

Sample Input

```
C C D E E F E C D E
E E G G A A G A B C
F G E A F A B F D E
G G# F# D G C G C# C F#
end
```

Sample Output

```
music
music
no music
no music
```

Explanation of the output: The first line contains the first notes from "I see a star" by Hans Unterweger, the second line from "Sunny light of Bethlehem" from "There is a Light" by Lorenz Maierhofer. The third and fourth lines are random notes. While the distinction seems to be subjective, there are a number of regularities in standard music, check out harmonics and music theory for a start.

A calculator for Santa (medium)

Santa is so old that he still calculates with roman numerals. The elves are planning to give Santa a pocket calculator as a christmas gift, but they could not find one that uses roman numerals.

Problem

Write the software for a roman numeral pocket calculator. Entries can be a number, defined with the symbols I,V,X,L,C,D,M followed by an operator '+','-','*','/' and another number. The operators + and * work normally, but for the minus operator the result should be given as zero instead of a negative number and the '/' operator works as an integer division. A zero should be printed as an empty line. After a calculation, Santa should be able to enter another operator, which the uses the last result as first argument or to enter a new number which starts a new calculation. When a line contains the word "end", the program shall terminate.

Example

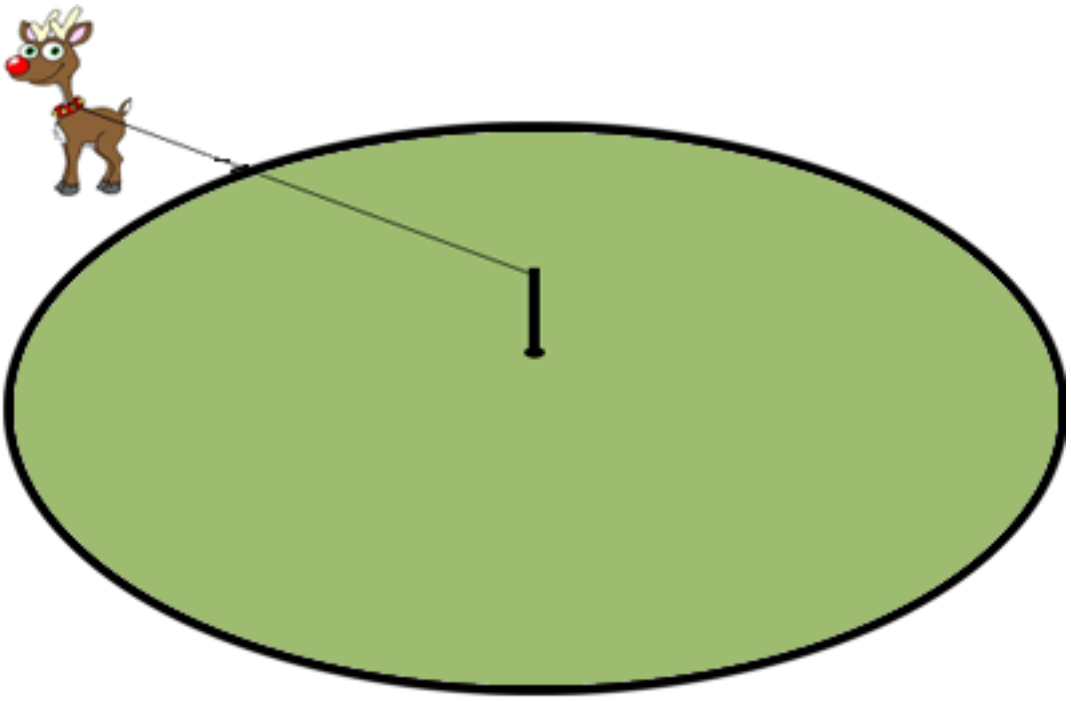
Input

```
I
+
I
*
II
M
-
V
/
VI
end
```

Output

```
II
IV
CMXCV
CLXV
```

Grazing Reindeer (easy)



In December, Santa Claus wants to keep his reindeer close. There is only one problem: his reindeer can fly, so a fence would not work. Therefore he attaches them to a stake using a rope. Your task is to calculate the area that the reindeer can access for grazing.

Problem

Implement a program that reads lines with two space-seperated numbers containing the length of the rope and the height that the stake is standing out of the ground. After reading each line, the program shall output the area that can be grazed by the deer. You can assume that the rope is connected exactly at the point where the reindeer grazes and that the area is absolutley flat. Output the area as a rounded number with two digits behind the comma (including zeros, so for example 4.00). When a line contains the word "end", the program shall terminate.

Sample Input

```
5 0
5 5
5 2.5
end
```

Sample Output

```
78.54
0.00
58.90
```

Snowball War (very hard)

The snowball duel described in Problem G has gained more and more popularity among the elves so that a new type of game has been invented: the "snowball war". In snowball war, a number of elves place themselves at a field and start throwing snowballs at another. Each throw is synchronized. If one or more elves are hit by (a) snowball(s), these elves have lost the game and stop throwing snowballs. All remaining elves then throw in the next round. The game is over if less than two elves remain. If exactly one elf remains, this elf is the winner of the game, if no elf remains (all remaining elves got hit synchronously in the last round), the game is a draw.

Problem

Write a program that calculates the probability for a win of each elf and the probability for a draw. The input to the program are lines with space-separated probabilities, expressed as a number between 0 and 1, which state the chance that an elf will hit another with a snowball. Assume that each elf will always aim at the elf with the highest accuracy (or the second highest if the repective elf itself is the single best shooter). If there are multiple elves with highest accuracy, the target is chosen randomly from the highest accuracy group. The program shall output the probability for each elf to win the snowball war and the probability for a draw. The output probabilities should be rounded to two digits after the decimal point and printed with these digits (even if they are 0), separated by spaces. The program shall terminate when the message "Back to work!" is read.

Example

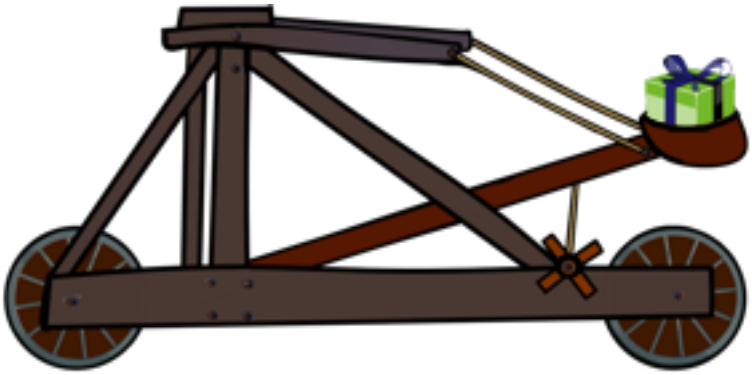
Input

```
0.2 0.9
0.5 0.5 0.5
0.3
Back to work!
```

Output

```
0.02 0.78 0.20
0.26 0.26 0.26 0.21
1.00 0.00
```

Gift-A-Pult(medium)



Santa has a lot of stress delivering all the presents personally. Therefore, he is looking for easier solutions, and currently he is absolutely fond of catapults and plans to deliver presents with them! Santa's test catapult is mounted on a 10 meter high hill (including the catapult height) and can eject objects at a defined angle and a given speed. Santa uses his magic to make the presents unaffected by air drag, but he still needs to develop the skill to set the parameters right for a catapult delivery.

Problem

Implement a program that helps Santa to determine how far the catapulted presents would go. The program shall read lines with the eject **speed** in meter and the **angle** in degrees over the horizontal, so 0 means a horizontal shot and 90 means a vertical shot. You can assume that the area around the catapult hill is absolutely flat, the gravitational acceleration at Santa's test site is exactly 9.81 m/s^2 and that the parcels stop traveling once they touched the ground. The program shall output the distance of the catapulted parcel rounded to two digits after the decimal point and printed with these digits (even if they are 0), separated by spaces. If a parcel is catapulted backwards, the distance should be given as a negative value. When a line contains the word "end", the program shall terminate.

Sample Input

```
10 45
10 90
0 30
end
```

Sample Output

```
16.41
0.00
0.00
```

Candle Burn Time (easy)

Bob likes to make candles. He has made a number of identical candles, but it is hard for him to determine the burn time of his candles. Therefore, he decides to measure the burn time with a test burn where he burns one of his candles for a short time. Before and after the test burn he measures the weight of candle. Assuming that the candle will burn uniformly and that all candle material will eventually burn up it is possible to determine the overall burn time.

Problem

Implement a program that estimates the overall burntime based on the test burn results. Your programm shall read lines with three space-seperated values: the length of the test burn in minutes and the weight in grams before and after the test burn. For each input the program shall output the estimated overall burntime in hours and minutes in the form of "x hours y minutes" with x and y being the respective numbers for hours and minutes (rounded to the nearest minute). If the overall burn time is below one hour, only the minutes should be printed. One hour or one minute shall be print in the singular form, for example "1 hour 1 minute". If the input consists of the word "end", the program should terminate.

Sample Input

```
23 990 983
10 100 80
10 120 100
end
```

Sample Output

```
54 hours 13 minutes
50 minutes
1 hour
```

There is something wrong with my Sudoko! (hard)

Whenever Santa has some leisure time, he loves to do Sudokos. In case you don't know it, a Sudoku is a puzzle where you have to replace the blanks in a 9 times 9 grid so that each row, column, and 3 by 3 box contains each of the digits 1 to 9. An example is the following Sudoku:

```
9 5 . . . . 6 7 .
. . . . . 7 3 1 .
8 . . 5 6 1 . . .
2 3 . . . . . . .
. . 9 . . . 1 . .
. . . . . . . 3 7
. . . 1 4 5 . . 9
. 9 4 2 . . . . .
. 8 5 . . . . 2 1
```

After solving it, the solution is:

```
9 5 1 3 2 4 6 7 8
6 4 2 8 9 7 3 1 5
8 7 3 5 6 1 2 9 4
2 3 7 4 1 9 8 5 6
5 6 9 7 8 3 1 4 2
4 1 8 6 5 2 9 3 7
3 2 6 1 4 5 7 8 9
1 9 4 2 7 8 5 6 3
7 8 5 9 3 6 4 2 1
```

However the elves sometimes give Santa an unsolvable Sudoko or one that has more than one solution.

Problem

Implement a program that tests Sudokos for their solvability. The program shall read a Sudoku as 9 lines, where each line contains 9 times a space followed by a dot or a digit. The digits denote the fixed numbers in the grids, the dots the blank field. Your program shall find out if the Sudoku is "correct", "unsolvable" or "has multiple solutions". After reading and processing a Sudoku the program shall read the next Sudoku. Whenever a line contains the word 'end' the program shall terminate.

Sample Input (3 Sudokus)

```
9 5 . . . . 6 7 .
. . . . . 7 3 1 .
8 . . 5 6 1 . . .
2 3 . . . . . . .
. . 9 . . . 1 . .
. . . . . . . 3 7
. . . 1 4 5 . . 9
. 9 4 2 . . . . .
. 8 5 . . . . 2 1
9 5 . . . . 6 7 .
. . . . . 7 3 1 .
8 . . 5 6 1 . . .
2 3 . . 8 . . . .
```

```
 . . 9 . . . 1 . .
 . . . . . . . 3 7
 . . . 1 4 5 . . 9
 . 9 4 2 . . . . .
 . 8 5 . . . . 2 1
 9 5 . . . . 6 7 .
 . . . . . 7 3 1 .
 8 . . 5 6 . . . .
 2 3 . . . . . . .
 . . 9 . . . . . .
 . . . . . . . 3 7
 . . . 1 4 5 . . 9
 . 9 4 2 . . . . .
 . 8 5 . . . . 2 1
end
```

Sample Output

```
correct
unsolvable
has multiple solutions
```

Baking cookies (very hard)

It is the time of the year to bake some cookies! The cookies are made on a square baking tray and since we like cookies, we want to put as many of them on the tray as possible. This is more difficult than you might think.

Problem

Your program should read the size of one side of the baking tray as a float value in each line and then output the maximum number of cookies that can be fit onto the tray. When reading the word "end" the program shall terminate. Every cookie has a diameter of exactly on unit, the maximum size of the baking tray is 20x20 units.

Example

Input

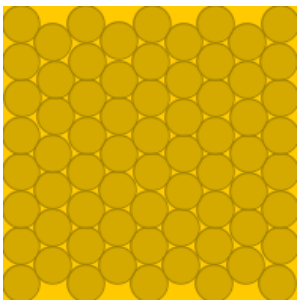
```
1
1.75
8
end
```

Output

```
1
2
68
```

Explanation

A tray with a side length of 1 can fit exactly one cookie. It is possible to fit two cookies diagonally on a tray with a sidelength of 1.5. It is possible to place up to 68 cookies on a tray with a sidelength of 8 by using the following layout:



Tree Cutting (medium)

Sam is selling Christmas trees. As can be expected, she is making most of her business in the last week before Christmas. Therefore, she needs to optimize her path to cut the required amount of trees in mimimum time.

Problem

Write a program that helps Sam optimizing her path. The program shall first read a line with three space-seperated integer numbers containing the width and length of her forest and the number of required trees. Followed by this are *length* lines of *width* characters containing either a "t" for a tree, a "." for a free space, or an "S" for Sam's shop. Sam starts at the shop position. At each time step, Sam can either walk 1 field north, south, east or west or cut down a tree she is standing on. A cut down tree is carried by Sam from the next time step on. When carrying a tree, Sam can walk as before, but cannot chop down further trees until she reaches the shop with the tree she carries. There she loads it off in zero time and is ready for going for the next tree until she has gathered the number of required trees. Since Sam has very picky neighbours, she is not allowed to take a step out of her forest. Your program shall output the minimum steps to bring all required trees to her shop. In case it is not possible to get enough trees, write "Until next year..." as answer. After writing the output the program shall process the next input or terminate when reading the line "end".

Sample Input

```
5 6 3
.....
..t..
....t
.t...
.....
..t.S
40 10 3
........................................
........................................
................................t.......
.....t...................................
........................................
...................S....................
........................................
........................................
........................................
........................................
end
```

Sample Output

```
23
Until next year...
```

Which town first? (easy)

Santa decided to optimize the towns for his visits based on the number of kids. However, Santa has troubles estimating these numbers, since usually this information is published in the form of *overall number of citizens* and *percentage of children* in this town.

Problem

Implement a program that reads census information consisting of *town name*, *overall number of citizens* and *percentage of children* in this town. The fields are separated by spaces but there may be extra spaces in the town names. When reading a line with the word "end" in it, the program shall output the towns sorted from the one with the highest absolute number of children until the one with the lowest number. Towns with the same amount should be printed in the order they were fed into the program. Towns with no children should not be printed at all. After printing the list of towns, the program shall terminate.

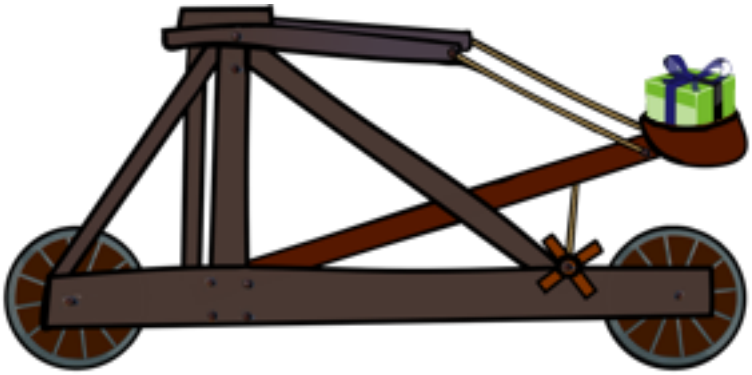
Sample Input

```
Santa Cruz 274146 20
Santana 98600 58
Santa Clara 125948 12
Antarctica 1000 0
end
```

Sample Output

```
Santana
Santa Cruz
Santa Clara
```

Gift-A-Pult (hard)



Santa was very pleased with the test results of the initial giftapult! The only thing that annoys him is the effort to wind up the catapult. This effort is the higher the more speed is required for the packet launch.

Problem

Implement a program that helps Santa to determine the angle and speed to catapult a parcel over a given distance with minimum initial speed. Santa's catapult is mounted on a 10 meter high hill (including the catapult height) and can eject objects at a defined angle and speed. Santa uses his magic to make the presents unaffected by air drag. The project shall read lines with intended catapulting distances and output lines with the eject **speed** in meter and the **angle** in degrees over the horizontal, so 0.00 means a horizontal shot and 90.00 means a vertical shot. The program shall determine the exact angle and velocity where the velocity is minimal and then output these two values rounded to two digits after the period. The numbers shall be printed with two digits after the period, including zeroes and be separated by a space. You can assume that the area around the catapult hill is absolutely flat, the gravitational acceleration at Santa's catapulting site is exactly 9.81 m/s^2 and that the parcels stop traveling once they touched the ground. When a line contains the word "end", the program shall terminate.

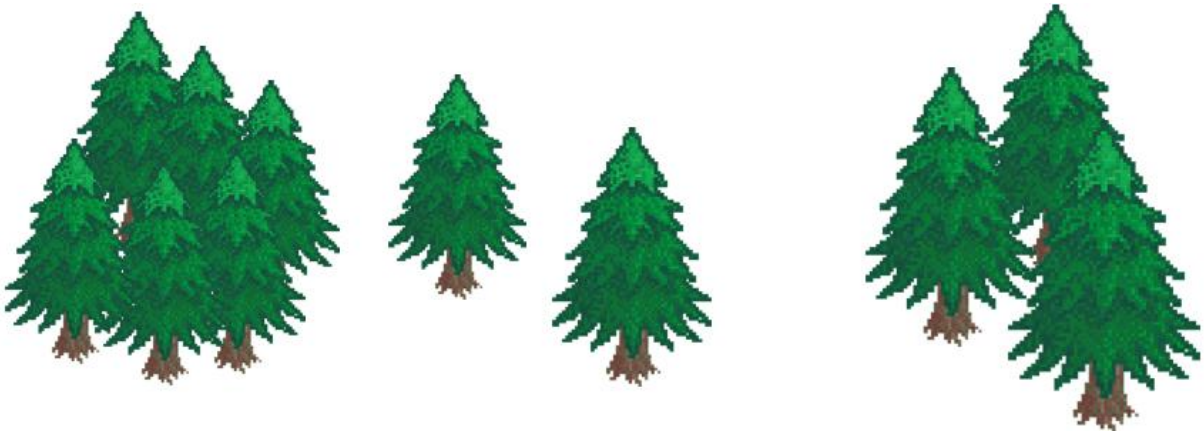
Sample Input

```
10
0
100
end
```

Sample Output

```
6.37 22.50
0.00 0.00
29.80 42.14
```

Tree Cutting (hard)



Sam (the lady who is selling Christmas trees from her forest) has an additional requirement for optimizing the paths to cut christmas trees in her forest. Some of the trees in the forest are so large, that they should not be cut as christmas trees. Moreover it is not possible to walk over a field with a large tree. Therefore, Sam asks you to make a new program to optimize her path to cut the required amount of trees in mimimum time.

Problem

Write a program that helps Sam optimizing her path. The program shall first read a line with three space-seperated integer numbers containing the width and length of her forest and the number of required trees. Followed by this are *length* lines of *width* characters containing either a "t" for a small tree, a "T" for a large tree, a "." for a free space, or an "S" for Sam's shop. Sam starts at the shop position. At each time step, Sam can either walk 1 field north, south, east or west or cut down a small tree she is standing on. Sam cannot walk onto fields with large trees. A cut down tree is carried by Sam from the next time step on. When carrying a tree, Sam can walk as before, but cannot chop down further trees until she reaches the shop with the tree she carries. There she loads it off in zero time and is ready for going for the next tree until she has gathered the number of required trees. Since Sam has very picky neighbours, she is not allowed to take a step out of her forest. Your program shall output the minimum steps to bring all required trees to her shop. In case it is not possible to get enough trees, write "Until next year..." as answer. After writing the output the program shall process the next input or terminate when reading the line "end".

Sample Input

```
5 6 3
.....
..tT.
...Tt
.tT..
.T...
..t.S
40 10 3
........................................
........................................
.........T......................t.......
.....t..T..............................
```

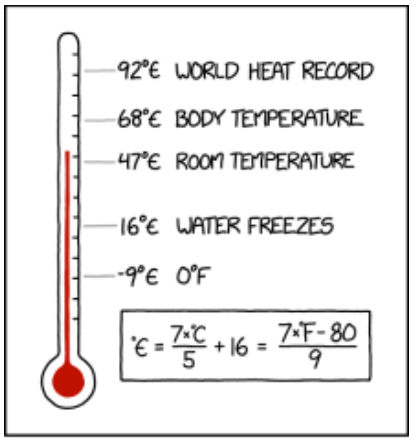
```
.....TTTT.........T..................
.................TS..................
....................................
....................................
.............T......................
....................................
end
```

Sample Output

```
27
Until next year...
```

Felsius (easy)

Measuring and communicating the current temperature is an existential skill at the north pole. Usually this would be done in degree Celcius, but there are a few countries that use the Fahrenheit scale. As a compromise, Santa decided to adopt the famous Felsius scale:



Problem

Write a program that helps converting to the superior Felsius scale. The program shall read lines with either "*n* degrees Fahrenheit" or "*n* degrees Celsius" where *n* stands for a floating point number that should be converted to the respective Felsius value *x* and be output in the form "*x* degrees Felsius". The output number shall be printed with exactly one digit after the period (zero also to be printed). When reading a line with the word "end", the program shall terminate.

Sample Input

```
100 degrees Celsius
98.6 degrees Fahrenheit
end
```

Sample Output

```
156.0 degrees Felsius
67.8 degrees Felsius
```

Sustainability (hard)

Dear advent programmers,
It is Sam again. Thank you for the program you made for day R and U, which helped me a lot in optmizing the tree cutting for my shop. Two days more of work and I can go to vacation as well. But I have to think about the future - I need to take care that I will have enough trees in my forest in the coming years as well. To be sustainable, I'm planning to plant a new small tree in the forest for every one that I cut. It does not need to be at the same place where I cut the tree, just anywhere in the forest. I should not step on the places where I planted a baby tree. So from now on, whenever I cut a tree, I will plant a new one while I'm on the way back to my shop and I won't go to the shop until I planted the new tree. Apart from this it is business as before.
So, could you provide me a program that optimizes this extra task?
Thank you very much in advance and perhaps see you at my shop one day
Sam

Problem

Write a program that helps Sam optimizing her path. The program shall first read a line with three space-separated integer numbers containing the width and length of her forest and the number of required trees. Followed by this are *length* lines of *width* characters containing either a "t" for a small tree, a "T" for a large tree, a "." for a free space, or an "S" for Sam's shop. Sam starts at the shop position. At each time step, Sam can either walk 1 field north, south, east or west or cut down a small tree she is standing on. Sam cannot walk onto fields with large trees. A cut down tree is carried by Sam from the next time step on. When carrying a tree, Sam can walk as before, but cannot chop down further trees until she reaches the shop with the tree she carries. Before entering the shop with a tree, Sam has to plant a new baby tree somewhere, which takes one time step. A baby tree is then planted on the field Sam is standing (this field must be empty, but can be one where a small tree was chopped from), once Sam leaves this field, it cannot be visited again. At the shop Sam loads the carried tree off in zero time and is ready for going for the next tree until she has gathered the number of required trees. Since Sam still has very picky neighbours, she is not allowed to take a step out of her forest. Your program shall output the minimum steps to bring all required trees to her shop. In case it is not possible to get enough trees, write "Until next year..." as answer. After writing the output the program shall process the next input or terminate when reading the line "end".

Sample Input

```
5 6 3
.....
..tT.
...Tt
.tT..
.T...
..t.S
40 10 3
```

```
.......................................
.......................................
.........T.....................t.......
.....t..T..............................
.....TTTT.........T....................
..................TS...................
.......................................
.......................................
.............T.........................
.......................................
end
```

Sample Output

```
30
Until next year...
```

Which song is it? (medium)

Santa likes Christmas carols a lot, but since he is a bit old, it is hard for him to recognize and name the song he is listening to.

Problem

Implement a program that tells Santa the name of the song based on a few consecutive words. Luckily, the number of songs Santa knows is very limited (see below). An input line will consist of a word sequence to be identified among the known songs. All songs that match the sequence should be then written in alphabetical order to standard output, one song title per matching song. A match needs to have the same whole words in the same sequence, punctuation and newlines shall be considered as a word separater, but otherwise ignored. If no song matches, the text "I don't know that song" should be printed. If the input consists of the word "end", the program should terminate.

Sample Input

```
snow
navidad
bell
end
```

Sample Output

```
Jingle Bells
Let it Snow
Feliz Navidad
I don't know that song
```

List of Songs

Feliz Navidad

```
Feliz Navidad
Feliz Navidad
Feliz Navidad
Feliz Navidad
Prospero Ano y Felicidad.

Feliz Navidad
Feliz Navidad
Feliz Navidad
Prospero Ano y Felicidad.

I wanna wish you a Merry Christmas
I wanna wish you a Merry Christmas
I wanna wish you a Merry Christmas
From the bottom of my heart.
```

Jingle Bells

Dashing through the snow
On a one horse open sleigh
O'er the fields we go,
Laughing all the way
Bells on bob tail ring,
making spirits bright
What fun it is to laugh and sing
A sleighing song tonight

Oh, jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh
Jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh

A day or two ago,
I thought I'd take a ride,
And soon Miss Fanny Bright
Was seated by my side;
The horse was lean and lank
Misfortune seemed his lot
We got into a drifted bank,
And then we got upsot.

Oh, jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh
Jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh

Jingle Bells, Jingle Bells,
Jingle all the way!
Oh, What fun it is to ride
In a one horse open sleigh.
Jingle Bells, Jingle Bells,
Jingle all the way!
Oh, What fun it is to ride
In a one horse open sleigh.

Now the ground is white
Go it while you're young
Take the girls tonight
And sing this sleighing song
Just get a bob tailed bay
two-forty as his speed
Hitch him to an open sleigh
And crack! you'll take the lead

Jingle Bells, Jingle Bells,
Jingle all the way!
Oh, What fun it is to ride
In a one horse open sleigh.
Jingle Bells, Jingle Bells,
Jingle all the way!
Oh, What fun it is to ride
In a one horse open sleigh.

Let it Snow

Oh the weather outside is frightful
But the fire is so delightful
And since we've no place to go
Let It Snow! Let It Snow! Let It Snow!

It doesn't show signs of stopping
And I've bought some corn for popping
The lights are turned way down low
Let It Snow! Let It Snow! Let It Snow!

When we finally kiss good night
How I'll hate going out in the storm!
But if you'll really hold me tight
All the way home I'll be warm

The fire is slowly dying
And, my dear, we're still goodbying
But as long as you love me so
Let It Snow! Let It Snow! Let It Snow!

O Tannenbaum

O Tannenbaum, o Tannenbaum,
wie treu sind deine Blaetter!
Du gruenst nicht nur
  zur Sommerzeit,
Nein auch im Winter, wenn es schneit.
O Tannenbaum, o Tannenbaum,
wie treu sind deine Blaetter!

O Tannenbaum, o Tannenbaum!
Du kannst mir sehr gefallen!
Wie oft hat nicht zur Weihnachtszeit
Ein Baum von dir mich hoch erfreut!
O Tannenbaum, o Tannenbaum!
Du kannst mir sehr gefallen!

O Tannenbaum, o Tannenbaum!
Dein Kleid will mich
  was lehren:
Die Hoffnung und Bestaendigkeit
Gibt Trost und Kraft
  zu jeder Zeit.
O Tannenbaum, o Tannenbaum!
Das soll dein Kleid
  mich lehren.

Silent Night

Silent night, holy night!
All is calm, all is bright.
Round yon Virgin, Mother and Child.
Holy infant so tender and mild,
Sleep in heavenly peace,
Sleep in heavenly peace

Silent night, holy night!
Shepherds quake at the sight.

Glories stream from heaven afar
Heavenly hosts sing Alleluia,
Christ the Savior is born!
Christ the Savior is born

Silent night, holy night!
Son of God love's pure light.
Radiant beams from Thy holy face
With dawn of redeeming grace,
Jesus Lord, at Thy birth
Jesus Lord, at Thy birth