

# CougarCS Coding Competition

## **First Prize:**

The winning team members are offered an internship over the summer with the:

**Computational Biomedicine Lab (\$1000/month).**

Thank you Professor Kakadiaris and Professor Shaw for this generous offer and support!



## Contest Rules:

1. You may not communicate with other teams about the contest packet or any coding/programming terminology in general. Any communications between teams should be limited to angry gestures, jeering and other forms of contempt.
2. You may not communicate with any persons outside of the contest domain whatsoever. This includes verbal communication as well as any electronic means of communication. Cell phone use, instant messaging, emailing, and any other forms of communication is strictly prohibited.
3. If you have questions about the problems or the contest while the contest is running you may only communicate with contest officials via the PC2 query system. Contest officials can communicate with you verbally at their own discretion.
4. You may use any written resources that you have brought along which includes programming books and other related supplemental material.
5. You may not use any code generated before the contest begins. All code submitted as a solution must be created during the contest.
6. You may not in any way interfere with the operation and continuation of the contest. This includes the following.
  - a. Tampering or crippling other teams computers or preventing them from submitting problems or effectively using their computers.
  - b. Tampering or interfering with the computer network used to run the contest.
  - c. Tampering or interfering with the computer running the PC2 server.
  - d. Tampering or interfering with the judges computers
7. Your program solutions should be programmed only in the manner to solve the problem as described. Do not try to trick the judging system, "hard-code" solutions, or perform other shenanigans.

## Data Input:

All input shall be through data files. The files will be named in the form of "pr##.dat". An example is the input of problem 5.5 would be "pr55.dat".

## Data Output:

Output all text to the standard output stream of your language. In C++ this is "cout", Java is "System.out", C# is "System.Console", C is "printf" etc. If you have any specific questions please verify before the contest. Unless otherwise specified in the problem your program has exactly 12 seconds to finish its execution and output the solution.

## Assumptions:

You may assume that all data files and input will be formed exactly as described in the problem. It is not necessary to check for bad input or malformed files. Just assume the correct data file will be used for your program formatted exactly as described in the problem. It is also important for you to realize that the judges will use much more comprehensive data sets to test your program to test and make sure you have fully solved the problem

Thank you for participating in the first CougarCS programming competition. I hope you had fun coding with other CS students. We will be hosting another contest in Spring 2010. Depending on the results from this contest we will make the problems easier or harder. Please plan on competing again!

*Michael Slater*  
Competition Director

*Eric Siebeneich*  
Assistant Director

*Zakj Mughal*  
Our brave Knight...

*CougarCS Officers*

Thanks to the Computer Science department, TLC<sup>2</sup>, and the CBL for your support



## In the beginning...

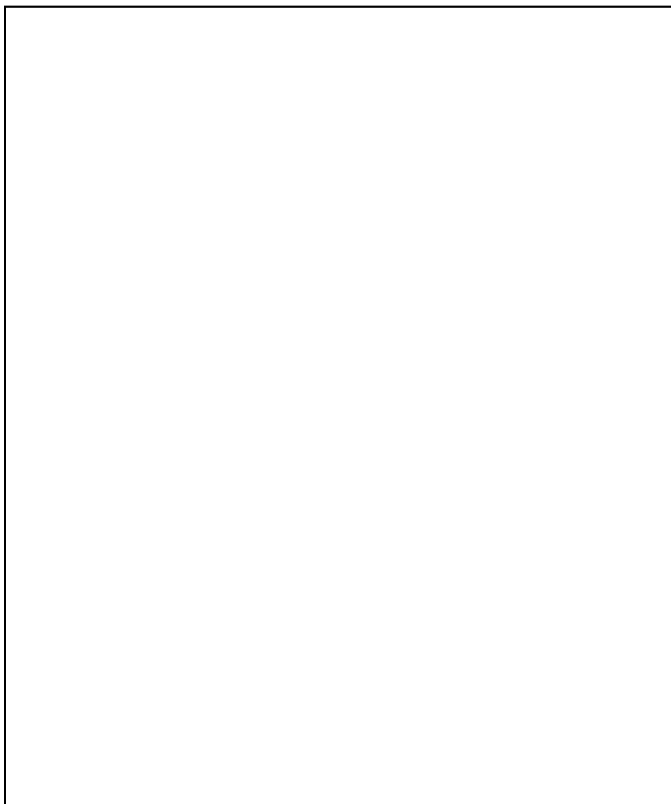
### Description

Welcome valued contestants who have so bravely decided to compete in this contest, we appreciate your aspirations to be the best at this contest, but really, this is going to be pretty difficult, so good luck! Like most great works of art, the introduction usually contains information regarding credits to the authors and anyone who has basically contributed to the creation of the work, BUT NOT THIS TIME. We did away with that because we thought (and still think) that that's just RUBBISH! Instead we bring you straight to our story, which will be very interesting, and if you don't think that it is THEN DON'T READ IT, I DON'T CARE! So back to our story (I have to cut to the chase now because my editor is yelling at me, so here you go). Squire Zaki wants to become a Knight and he's a pansy right now so give him the confidence to begin his journey!

**Input:** None!

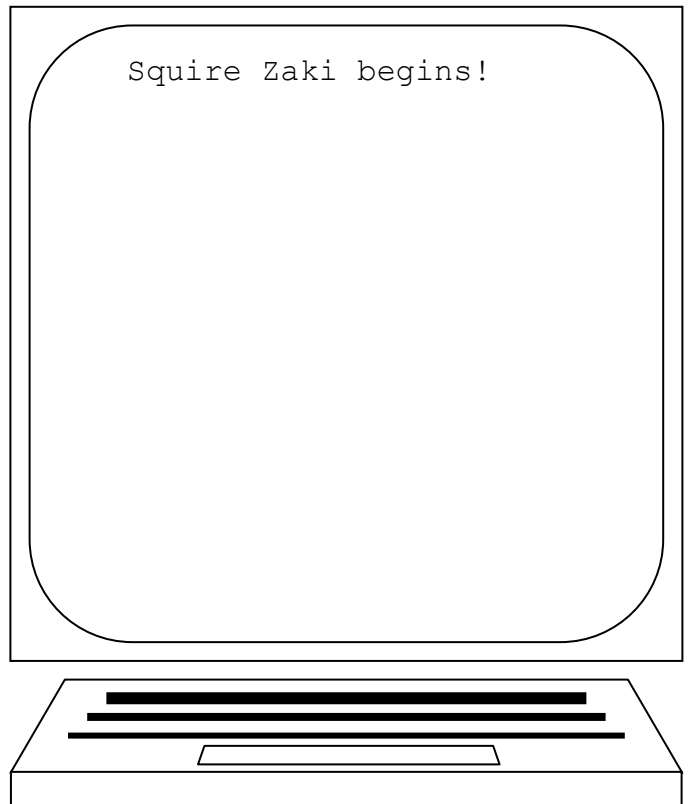
**Output:** Print "Squire Zaki begins!" to the screen.

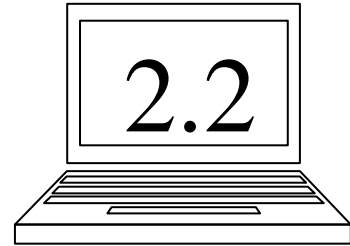
**Actual Input:** (No Input)



**Actual Output:**

Squire Zaki begins!





## The Price of a Life

### Description

Zaki sees a pirate ship full of old men sail off the edge of the world. They scream their final wishes as their ship fades into the distance. Help Squire Zaki count how many letters their final wishes include so they may be properly remembered, because that's what's important – the letters. Obituaries are expensive by the line, you know.

**Assumptions:** Only the letters A-Z are to be counted, and all numbers, spaces, and punctuations are to be ignored.

**Input:** A string consisting of a one-line sentence.

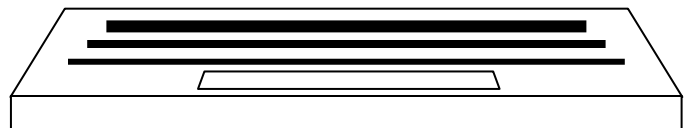
**Output:** The total number of letters in the given sentence.

**Sample Input:** (Data file)

```
I love to program!
```

**Sample Output:**

```
The sentence has 14  
letters.
```





## Food for thought

### Description

After checking and learning that obituaries are too expensive, Zaki stops for a rabbit-sandwich. The delicacy of the day are evil rabbit patties, and he's not sure how many patties he wants on his sandwich. Help Zaki picture his sandwich by drawing it with a given number of rabbits so as to help him decide if it would hit the spot.

**Input:** An integer representing the number of rabbit patties.

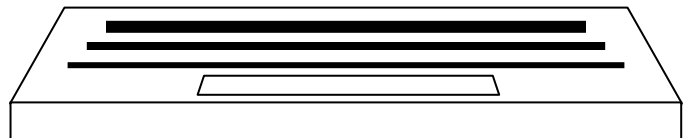
**Output:** A textual representation of the rabbit-sandwich.

**Sample Input:** (Data file)

3

**Sample Output:**

```
Bun
Rabbit
Rabbit
Rabbit
Bun
```





## The hypotomoose

### Description

As Squire Zaki is walking along, he hears a damsel in distress atop a tower. Fortunately, he has his handy dandy pocket cross-bow. He needs to know how much rope he should attach to the arrow he is going to shoot from his cross-bow to the damsel in order to have a taut rope for her to slide down.

**Input:** Two integers representing the distance to from Zaki to the tower and the height of the tower.

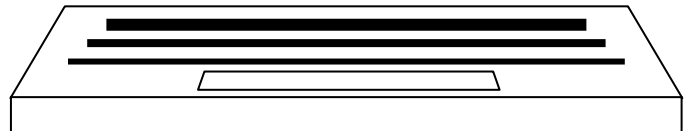
**Output:** The length of rope, rounded to the nearest foot, needed to save the dear damsel.

**Sample Input:** (Data file)

```
3 4
```

**Sample Output:**

```
Zaki should use a rope of  
length 5.
```





## 3, not 5, nor 4...

### Description

After unfortunately forgetting he was not aiming *for* the damsel, Zaki continues to quest on. Squire Zaki comes to a village where the poor citizens are being attacked by a small fuzzy rabbit with vicious teeth and claws. Squire Zaki luckily has his Holy Hand Grenade of Antioch. He must then taketh thine holy hand grenade and count to: four. No, wait, three. Not one, nor two, unless thereafter proceeding unto three. Given the coordinates of thine rabbit, helpeth Squire Zaki blow thine enemies unto smithereens.

**Assumption:** No input will be zero or lie on axis.

**Input:** Two integers representing the x-coordinate and y-coordinate of the said fuzzy rabbit on a Cartesian plane.

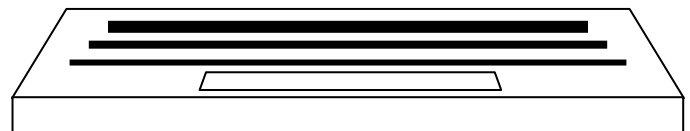
**Output:** The quadrant number of where the rabbit is indicated in Roman Numerals.

**Sample Input:** (Data file)

```
-4 3
```

**Sample Output:**

```
II
```





## Parlez vous français?

### Description

Zaki sees a man piling up the corpses of those who died along with the rabbit in the village. The man, apparently a Frenchman, talks in some strange dialect. Help Squire Zaki decode this strange man's repetitive babble so he may determine what is going on. The dialect seems to be one of the many anachronisms in the tale of Zaki. It would appear to be a gibberish known as ASCII which needs to be converted into plain English.

**Input:** A set of integers. The first of which denotes the number of "secret" numbers there are in the word.

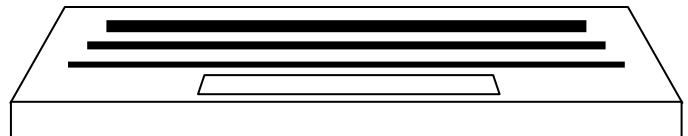
**Output:** The word the Frenchman is trying to convey so secretly.

**Sample Input:** (Data file)

```
6
78
101
119
116
115
33
```

**Sample Output:**

```
Newts!
```





## Squabbles for a piece of pie

### Description

Meanwhile in a kingdom far, far, away Lancelot the brave, Sir Robin the Not-Quite-So-Brave-As-Sir-Lancelot and Galahad are quarreling over the last piece of pie. King Arthur steps up to bring peace and call for a height contest in which the pie shall go to the tallest! After much bickering and such measurements are taken, Arthur, at the last second, calls for the shortest to win! Determine the smallest height.

**Input:** A series of three integers.

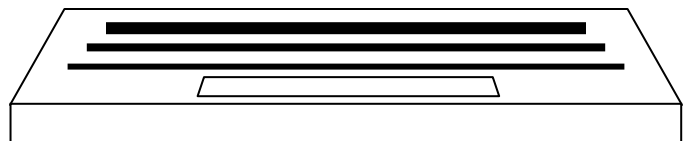
**Output:** The smallest number of the three.

**Sample Input:** (Data file)

```
89
91
27
```

**Sample Output:**

```
27 is the smallest number.
```



## Tim's Spelling

### Description

Back to Squire Zaki, and Tim the Enchanter. Tim flew in from Bermuda during our break, and we apologize for not telling you until now. He found a dangerous spell and needs a way to keep it out of the hands of *evil*. Your task is to cipher the message with the following pattern;

A - \*      E - &      I - @      O - #      U - }

Cipher the secret message's vowels with symbols.

### Assumptions

The pattern applies to both upper and lower case vowels.

**Input:** A string consisting of a dangerous spell.

**Output:** The ciphered secret spell.

"Some call me...Tim?"

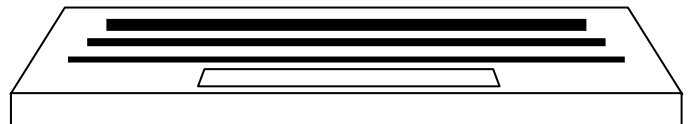


**Sample Input:** (Data file)

```
Expecto Patronum
```

**Sample Output:**

```
&xp&ct# P*tr#n}m
```





## Shift!

### Description

Our poor script writer passed away last night. Given such short notice the only writer we could find was shiftily challenged. He just seems to have trouble with his cases. They are all backwards, and the final product is an eyesore. Do fix it for the sake of the packet by writing a program to correct the case.

**Input:** An integer representing the number of inputs that shall require inversion, which will be followed by the sentences to be inverted.

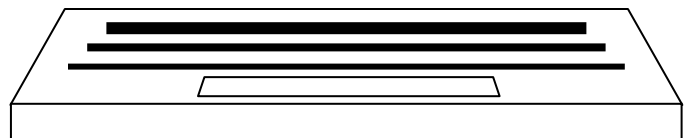
**Output:** The inverted sentences.

**Sample Input:** (Data file)

```
2
How to Do It
...with A Submarine!
```

**Sample Output:**

```
hOW TO dO iT
...WITH a SUBMARINE!
```





## A violent critique

### Description

After helping Tim decode his spell, Zaki was inadvertently transported to the Louvre of 2010. Disgusted by what had become of the great art of finger painting, he began a mad rampage, systematically destroying the pitiful art of this late age; first going and destroying every third, starting from the beginning, and then starting over from the second piece of art and destroying every other piece of art. Given the number of pieces and the number of the first piece determine the numbered pieces of art remaining.

**Input:** Two integers. The first represents the total number of “works of art”. The second is the number of the first master piece.

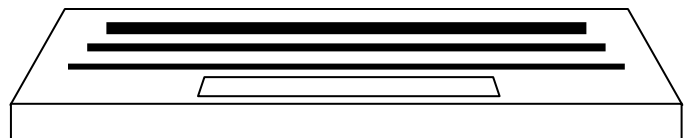
**Output:** The remaining pieces of art lined up in order.

**Sample Input:** (Data file)

```
20 45
```

**Sample Output:**

```
46 49 52 55 58 61 64
```





# IN SOVIET RUSSIA, FOUR SQUARE PLAYS YOU!

## Description

Leaving the museum in ruins, Zaki emerges to find a soldier marching up and down the square by himself. After consulting with him, it turns out that his soldiers would rather be at the movies than marching up and down the square with him. In order to make the square more “march-able,” help Zaki paint the men’s orders on it so they may be more entertained while marching.

**Input:** An integer representing the number of executions the program must complete, followed by terms, which are to form the four square.

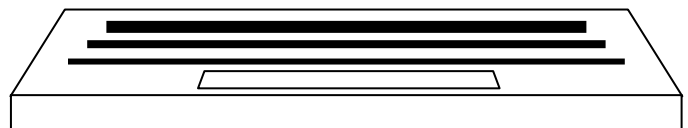
**Output:** A four square per each input.

**Sample Input:** (Data File)

```
1  
WORDS
```

**Sample Output:**

```
WORDS DROW  
O   D   O  
R   R   R  
D   O   D  
SDROW WORDS  
D   O   D  
R   R   R  
O   D   O  
WORDS DROW
```





## Words from a quack

### Description

Tim happened to be speaking to a duck, as he always does, and the duck somehow spoke back in gibberish or so it seemed; for he was, actually linked to the world where Zaki was lost and was voicing Zaki's cries for help. But the writer digresses. FIND a way to decode the babble and save dear Zaki by using the cipher provided and decoding the gibberish.

**Assumption:** All messages shall be in uppercase.

**Input:** The ciphered alphabet and the corresponding corrected alphabet. An integer representing the number of inputs the program must process, followed by the encrypted messages.

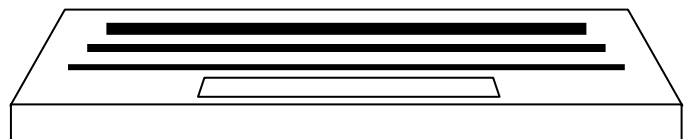
**Output:** The series of decrypted messages.

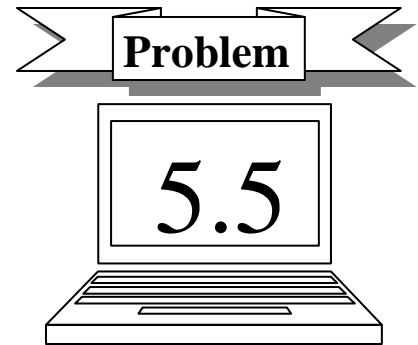
### Sample Input: (Data File)

```
QWERTYUIOPASDFGHJKLZXCVBNM  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
  
3  
VT OFZTKKXHZ ZIOL HKGUKQD  
ZG WKOFU NGX Q VGKR YKGD  
GXX LHGFLGK...
```

### Sample Output:

```
WE INTERRUPT THIS PROGRAM  
TO BRING YOU A WORD FROM  
OUR SPONSOR...
```





## One big happy family

### Description

Zaki is inspired to save people now, after having come so close to (near) death. He comes upon two whining royals, apparently King Arthur and an unimportant French Dauphin. Help them sort out their problems and discover that they are really friends!

**Assumptions:** In comparing, case does not matter. The output should be in alphabetical order.

**Input:** An integer which represents how many data sets there are to follow. Each data set contains two people and a list of their qualities.

**Output:** The qualities which they have in common.

### Sample Input: (Data File)

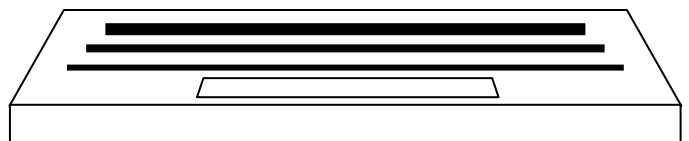
```
2
Arthur: English
The French Dauphin: French

Lance: French Bread, horses,
fishing, African swallows, duels
Galahad: Reading poetry, French
Bread, the Eiffel tower, duels,
horses
```

### Sample Output:

```
Arthur and The French Dauphin
have nothing in common.

Lance and Galahad have duels,
French Bread, and horses in
common.
```





## A Proletarian uprising

### Description

One day Squire Zaki comes upon a school house where Professor Hilford tries so hard to educate the dear children... BAM! The door flew open and a swarm of peasants rushed in and began filthifying the poor classroom. If one comes too close to her she wails hysterically. Write a program to illustrate her reactions. She is stands at the center of the 11x11 room at coordinate (5,5). The distance is calculated by the horizontal offset plus the vertical offset from the center of the room. Her wailing is as follows:

**If a peasent is within;**

4 spaces - She jumps on a chair and waves her arms around in an insane manner!

6 spaces - She clenches the chair she is sitting on and screams!

8 spaces - She is wary and calls for Zaki!

**Input:** An integer representing how many data sets there are, followed by the actual set of coordinates ranging from 0-10 inclusive. The coordinates are arranged in an X,Y fashion.

**Output:** Professor Hilford's reactions to the positions of the peasants.

**Sample Input:** (Data File)

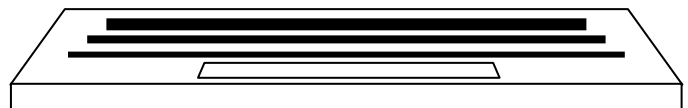
```
3
2,3
8,9
10,0
```

**Sample Output:**

```
She jumps on a chair and waves
her arms around in an insane
manner!
```

```
She clenches the chair she is
sitting on and screams!
```

```
She is wary and calls for Zaki!
```





## Five-point-seven

### Description

After all this chivalry and hard work it is finally time for Zaki to try to be a real knight. His first test is to duel the dreaded Scot! Sadly, Zaki doesn't know a thing about dueling with herrings. He pleas to Nilrem, who as always, gives in and will help. Nilrem on the fly makes up a strategy for success! He lists a group of signals and a trigger signal which signifies the action he should perform.

### Signals

Nose – low blow

Hips – shank

Hat – hamstring

Arms – parry

Chest – riposte

Ear – cross-thrust

**Assumptions:** The activator shall only be signed once.

**Input:** An integer representing how many data sets are to follow. Each data set contains a series of signs followed by the activator signal on the next line.

**Output:** A statement about what Zaki should do.

### Sample Input: (Data File)

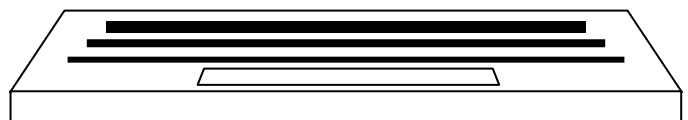
```
3
Nose Arms Hat Chest Ear Hips Hat
Ear
Nose Arms Hat Chest Ear Hips Hat
Chest
Nose Arms Hat Chest Ear Hips Hat
Arms
```

### Sample Output:

```
Zaki should shank his opponent.

Zaki should cross-thrust his
opponent.

Zaki should hamstring his
opponent.
```





## African or European?

### Description

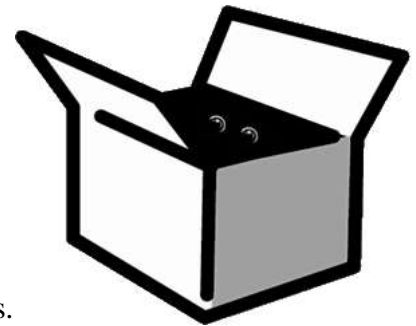
Apparently, “the average airspeed of an un-laden swallow” has nothing to do with whether it is African or European. Rather, it is dependent on the radius of the swallow, (which is spherical, of course,) and may be calculated with the equation below. Ye must help me answer these questions three, which are fortunately all about the air speed of swallows. For safety reasons all said swallows are boxed in cubical crates.

$$AS = 4 \cdot \pi \cdot r^2$$

**Assumptions:**  $\pi$  is 3.14 for all calculation purposes.

**Input:** An integer representing the number of executions the program must complete followed by a series of volumes for different sized crates holding the swallows.

**Output:** The average airspeed for each of the respective swallows.



**Sample Input:** (Data file)

```
2
27 ft^3
125 ft^3
```

**Sample Output:**

```
28.26 ft^2
78.50 ft^2
```





# True Chivalry

## Description

Zaki's next knightly duty is to repress the peasants. If he properly infringes their constitutional rights, he shall progress along the path to knighthood. However, he doesn't have to repress just any peasants: he must repress them in a very specific order so that he can make sure he's doing so properly.

**Assumptions:** The range of numbers given will not be less than 1 or greater than 25. The columns need to be 3 wide and right justified.

**Input:** An UNKNOWN number of data sets. Each data set contains one integer which represents the length of the side of a spiral square.

**Output:** A spiral square with the number 1 in the top left corner and proceeding from there to complete the square.

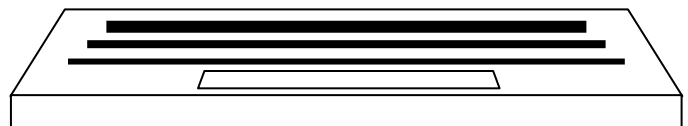
### Sample Input: (Data File)

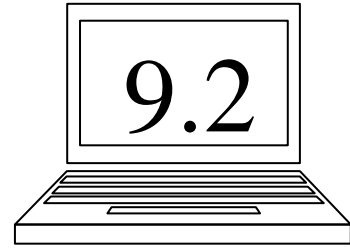
```
7
4
```

### Sample Output:

```
 1  5  9 13 17 21  2
24                    6
20                    10
16                    14
12                    18
 8                    22
 4 23 19 15 11  7  3
```

```
 1  5  9  2
12                    6
 8                    10
 4 11  7  3
```





# Chewie says: Gggaaaaarr

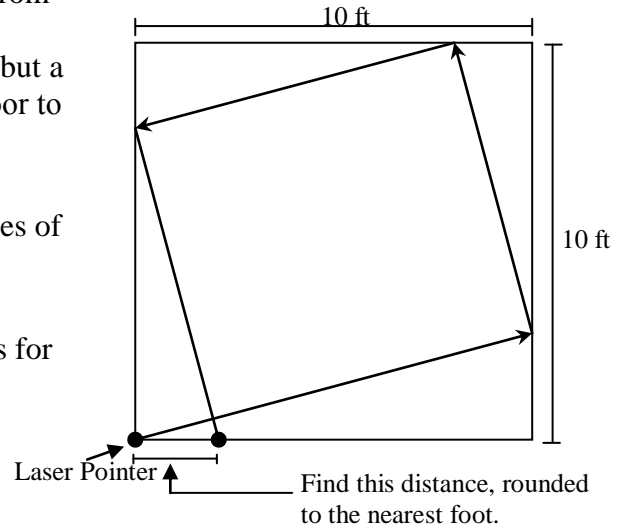
## Description

Zaki received a strange birthday present from some man named San Holo. It was some strange laser pointing device that puts a hole in anything but a mirror. An attached card says it's a key to the door to Excalibur.

**Assumption:** The laser does not follow the rules of physics and will always bounce off at  $90^\circ$ .

**Input:** An unknown number of possible degrees for the laser. The degrees will be between  $0 - 90$  inclusive.

**Output:** The distances from the wall. If the angle is greater than  $45$ , then the laser will bounce clockwise.



### Actual Input: (Data File)

35 degrees  
90 degrees  
50 degrees

### Actual Output:

3 feet  
0 feet  
2 feet



## The Holy Grail

### Description

Michael the Enchanter has informed Sir Zaki of the location of the Holy Grail, the only problem is that it is located across the world! Thankfully, Tim has given Sir Zaki a map of the world, complete with all the Bridges of Peril he must cross. Sir Zaki would like to reach the Grail with as little peril as possible, so he needs your help. Given a set of countries and the peril level of crossing to other countries, determine the least perilous route for Sir Zaki to take.

**Assumptions:** You can assume we will give your program a little more time to process the potentially large set of judge data.

**Input:** Input files will contain the number of countries in the world, the country Sir Zaki begins in, and the country the Holy Grail is located in, followed by a list of the countries and the peril levels of crossing from the said country to another country.

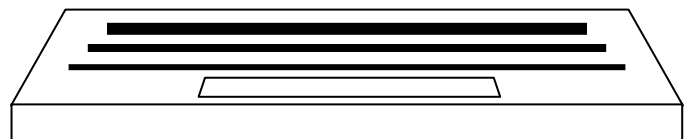
**Output:** Output will be the countries traveled to in order.

### Sample Input: (Data File)

```
5
England
Germany
France Germany 16 England 8 Spain 10 Austria 12
Germany France 20 Spain 12 Austria 8
England France 15 Spain 9
Spain England 8 France 4 Austria 9
Austria Germany 4 France 10 England 3 Spain 13
```

### Sample Output:

```
England->Spain->Austria->Germany
```





## Text 44444\*

### Description

This chronic death thing is ridiculous! The writing staff suffered the loss of, yet, another brilliant mind. This writer was working on the rest of the packet and intended to leak the packet via a test message. There seems to be a key log of the packet. Write a program to decode and send your text message given the number of times a key is hit. Ex; 44 is an H

1 @ . ? 1	2 ABC2	3 DEF3
4 GHI4	5 JKL5	6 MNO6
7 PQRS7	8 TUV8	9 WXYZ9
* Send	0 0	# Space

**Assumptions:** All text decoded should be in upper case.

**Input:** An unknown series of Integers and Symbols representing the keys for the text.

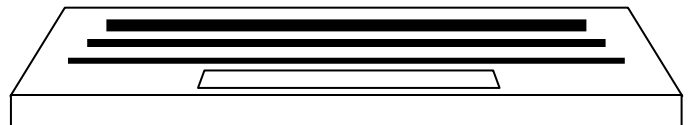
**Output:** The decoded text messages.

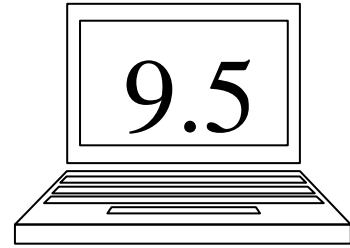
### Sample Input: (Data File)

```
22 8 8 777 999 # 55 666* 9  
6 88 7777 8 # 333 444 66 444*
```

### Sample Output:

```
BTRY LO  
MUST FINI
```





## A few times I've been around that track...

### Description

Apparently, no one cares that Squire Zaki now wields the legendary Excalibur. He must participate in time trials on his horse. Our incompetent timer needs help determining how long Zaki spends on each lap around the jousting octagon. Each time he completes a lap, his galloping speed slows down 10 percent of his previous lap.

**Assumptions:** No solutions will require rounding past the hundredths place and all input shall be in seconds.

**Input:** An unknown number of total lap times and the number of laps completed respectively.

**Output:** The split times for each lap formatted to two decimal places.

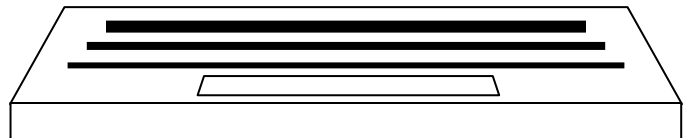
#### Sample Input: (Data File)

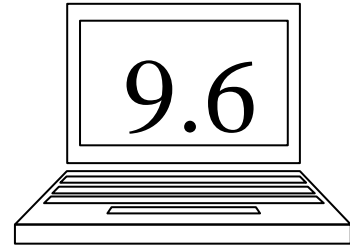
```
464.1 seconds
4 laps
231.7 seconds
3 laps
```

#### Sample Output:

```
100.00 seconds
110.00 seconds
121.00 seconds
133.10 seconds

70.00 seconds
77.00 seconds
84.70 seconds
```





## Yet another amazing coincidence

### Description

The height of a shrubbery in centimeters is an amazing number. Apparently, performing Kaprekar's process on this height determines how many times said shrubbery must be struck with a herring in order to cut it down and appease yon Knights who Sayeth 'Ni' and finally become a great Knight!

### Directions

Kaprekar's process takes any number (548) and puts it in descending order (854) then subtracts it from its ascending order (458). You keep repeating this process until your answer repeats itself. Find out the number of times the process must be accomplished until two answers are consecutive.

**Input:** The height of an unknown number of shrubberies in centimeters.

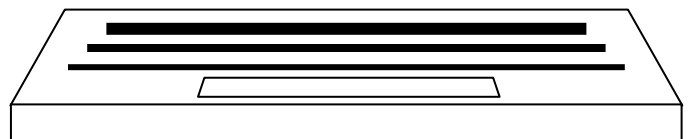
**Output:** The number of herrings that must be struck for said trees.

**Sample Input:** (Data File)

321

**Sample Output:**

6





## One last gag

### Description

After becoming a great, great, Knight, the new “Sir” Zaki is given the quest to infiltrate Castle Anthrax and sabotage a *great weapon*. On his way he comes upon a faux seller of shrubberies who calls himself a “shrubber” and fakes his wares with prime shrubberies in order to create larger trees. He has far too many orders today and needs some help.

### Directions

Every number has prime factors. Ex:  $18 = 3 * 3 * 2$ .  
Your task is that given a number you should output the prime factors for the given number.

**Input:** An unknown number of large tree sizes

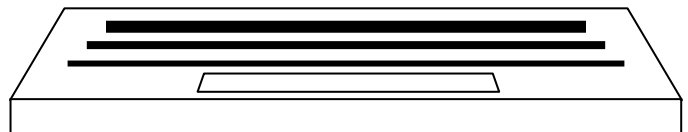
**Output:** The actual sized trunks this peddler will use; in descending order.

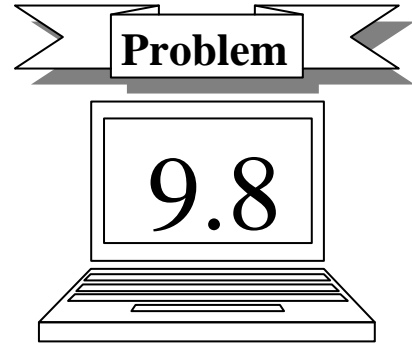
**Sample Input:** (Data File)

```
7
81
34
```

**Sample Output:**

```
7
3 * 3 * 3 * 3
17 * 2
```





# Tis just a flesh wound

## Description

After a night long infiltrating, Sir Zaki faced one last obstacle and the most random and unnecessary of sights; the Black Knight. The Black Knight said very little but just, “I move for no man.” Your mission is to help Zaki flank the Black Knight and perhaps come upon the *great weapon*.

## Directions

Given a grid of size 6x6, determine if the Black Knight can be flanked. If so, determine if Zaki can reach those positions. The Black Knight is flankable if two spaces which are beside him and opposite of each other are empty. Zaki can move either up, down, left, or right, but not diagonally. As such, the flankable positions must be to the Black Knight’s left, right, above or below him. Output whether the Black Knight is flankable and, if so, whether the positions are attainable. Note that Sir Zaki can not move onto the same space as the Black Knight.

- ‘B’ - denotes the Black Knight                      Z’ - denotes the Zaki’s location
- ‘\*’ - denotes an empty area

**Input:** An unknown number of data sets.

**Sample Input:** (data file)

```
Z * X X * *
* * * * * X
* X * * * *
X * * X * X
X X * B * *
X X * * X *

Z * X X * *
* * * * * X
* X * * * *
X * * X * X
X X * B X *
X X * * X *

Z * X X * *
* X * * * X
* X * * * *
X * * X * X
X X * B * *
X X * * X *
```

**Sample Output:**

