UMW High School Programming Contest

Overview[1]
Welcome to the UMW High School Programming Contest! This is a contest wherein teams from area high schools compete to solve a set of programming problems as fast as possible.

Problems
There are 14 problems of (roughly) increasing difficulty. The team who solves the most problems is declared the winner. In the event of a tie, the team with the best time wins. The best time also included penalties for wrong submissions. Each problem has a general description, details on the formats for input and output, and example input and output.

Input is from standard input and output is to standard output. Do not open any files inside of your program. Output of your program must match that of the correct output ***exactly***. You can solve the problems in Java, Python, or C++. Up to date versions of these languages are used to judge your entries. You must save your code using the standard extension for your language of choice (.java .py .cpp respectively). In case you need a refresher, sections at the end of your packet provide a brief overview of how to perform input and output in Java, Python, and C++.

Rules
You are allowed to have any printed material such as books or printouts. You are also allowed to access standard library documentation for your language of choice. No other use of the internet is permitted. You are also not allowed to copy and paste code from any source, be it a thumb drive, or another file that you have. Only one computer is allowed to a team.

Logging In
To log in to the system, direct your browser to **http://bit.ly/2GXI5ri (Update with new link prior to contest)**. You will then need to enter your team name and password which are provided for you. This will bring up the interface for you to submit solutions to the problems, ask questions (which will be broadcast to all participants), and see how your team is doing.

Submitting
To submit a solution, first make sure that you select the correct problem letter on the top of the site. Then click the "Choose File" button. Then choose the file that contains your source code (be sure it has the correct extension). After that it will be submitted for judging. It will receive an automatic judging response which is marked pending. Your solution will then be checked manually and a final response will be assigned.

[1] Overview page modified from Dr. Finlayson's 2018 UMW Contest

Introduction

Giraffes[2] are easily recognized - long necks, orange-reticulated skin, long legs, elongated head, a 20” tongue, big eyes with long eyelashes, and two horn-like structures called ossicones that deliver thumps during a giraffe neck fight. Giraffes can be found living in herds on the African continent and in small groups in various zoos. Recently, two giraffes - Jiffy and Jaffy - matriculated to UMW. Both are studying computer science. Jiffy is Jaffy’s older sister. She is a senior, soon to be a Summa Cum Laude graduate who is active in promoting giraffe social concerns around the world. Jaffy is a sophomore with lots of potential, but spends most of his time talking to friends, riding his bicycle, and partying. Jiffy is hoping Jaffy will learn to balance his studies and social life; otherwise, he will fail out of UMW and return to his home in South Africa. Jiffy recently returned from a Wildlife Conference in Paris where she led a panel discussion on giraffes. Those in attendance learned many giraffe facts. There are four areas of giraffe habit in sub-saharan Africa, each with its own species of giraffe - Northern Giraffe in southern Niger, Reticulated Giraffe in Kenya, Masai Giraffe in Tanzania, and Southern Giraffe in Botswana, Namibia, and South Africa. Today, these four areas have approximately 110,000 giraffes, whereas there were about 1,000,000 giraffes in the 1700. Habitat fragmentation, deforestation, war, and poaching are the primary threats to giraffes. Jiffy and Jaffy are Southern Giraffes, which have the best situation for prosperous families. Jiffy and Jaffy are from a large family, and you get to meet several of the family members in the problems. The whole time Jiffy was educating the world about giraffes, Jaffy was lollygagging around UMW. When Jiffy returned from her conference, UMW wrote an article extolling her accomplishments. Jiffy is pleased with her life. She already has a job offer from Google. Google will allow Jiffy to work in either their California office or their Johannesburg office. Jiffy is concerned about Jaffy. She wants Jaffy to become an upstanding giraffe who makes positive contributions to the giraffe society. Jaffy is just a happy undergraduate UMW student, who does the minimum to remain enrolled. The one thing that makes Jaffy sad is Janny, his highschool sweetheart who is still in South Africa. Jaffy misses Janny and worries she may get poached. Jaffy is plotting a way for Janny to study at UMW.

Jaffy on his bicycle

[2] Detailed giraffe facts from National Geographic issue 10.2019, *Giraffes* by Joshua Foer, photographs by Ami Vitale

Practice Problem A - How Old is Jaqarue

Jiffy and Jaffy have an uncle named Jaqarue. His given name is Jaque, but in college he did extensive field work in Australia studying kangaroos, where he received the nickname Jaqarue. Jaqarue's speciality is Red Giant Kangaroos. Jaffy is 19 years old and a sophomore at UMW. Uncle Jaqarue is 49 years old. Write a program that given Jaffy's age, the program prints Uncle Jaqarue's age.

Input

Input is a single, non-negative integer that is Jaffy's age.

Output

Output is one line that is the age of Jaqarue.

Sample Input 1

```
10
```

Sample Output 1

```
Uncle Jaqarue is 40 years old.
```

DO NOT TURN OVER THIS SHEET UNTIL THE CONTEST STARTS

Problem B - How Many Spots

Jiffy and Jaffy's Aunt Gingerbread is amazing at counting spots on giraffes. Giraffe spots are mainly for camouflage, but they also contain a patchwork of blood vessels that cool a giraffe. Aunt Gingerbread does not count spots one-by-one. She looks at a giraffe and immediately knows how many spots are on it. Many have asked Aunt Gingerbread to teach them her spot counting trick, but she doesn't know how she does it. Aunt Gingerbread uses her spot counting to teach addition. She counts the spots on two giraffes and teaches others how to add the spots. Write a program that adds the spots of two giraffes.

Input

The first line of input contains a non-negative integer N giving the number of pairs of giraffes that Aunt Gingerbread has counted.

Following that are 2*N lines. Each pair of lines contain the spots on a pair of giraffes.

Output

Output is N lines, where each line has the sum of the spots on a pair of giraffes.

Sample Input 1

```
4
10
12
20
25
30
38
40
47
```

Sample Output 1

```
22
45
68
87
```

Problem C - How Tall is Jessie

Jiffy and Jaffy have a young cousin named Jessie. On each Facetime call between Aunt Jessabell, Jiffy, and Jaffy; Aunt Jessabell points her camera at her little daughter Jessie. Jiffy is amazed how much Jessie has grown each time they talk. Giraffes can grow up to 19 feet tall and weigh 2500 pounds. Maybe cousin Jessie will grow this big or maybe Jessie will grow even larger. The Facetime calls are once per month, and little Jessie grows 6 inches on each call. Little Jessie is 3 feet tall on the first Facetime call. Write a program that computes how tall Jessie is after N Facetime calls.

Input

Input is a single, non-negative integer that is the number of Facetime calls.

Output

Output is one line that is the height of Jessie. The height is printed in feet and half feet.
If the height is a whole number, then do not print the number of half feet. The output line is terminated with either a . (period) or an ! (exclamation point). If Jessie's height is greater than 19 feet, terminate the output line with an exclamation point; otherwise, terminate the output line with a period.

Sample Input 1

```
2
```

Sample Output 1

```
Jessie's height is 4 feet.
```

Sample Input 2

```
5
```

Sample Output 2

```
Jessie's height is 5 and 1 half feet.
```

Sample Input 3

```
34
```

Sample Output 3

```
Jessie’s height is 20 feet!
```

Problem D - Jaffy’s Bored in a CS Class

One day in computer science class, Jaffy got bored with the lecture. The teacher did not allow smart phones in class to keep his mind occupied, so Jaffy started estimating the weight of backpacks of students sitting in a row of desks. Maybe Jaffy can estimate weight like Aunt Gingerbread can count spots. After creating the estimates, Jaffy mentally computed the total number of backpacks in the row, the average weight, the smallest weight, and the largest weight. Write a program that performs Jaffy’s mental computations.

Input

The first line of input contains a non-negative integer N giving the number of backpacks Jaffy estimated.

Following that are N lines, each containing an estimated backpack weight.

Output

Output is three lines with (1) the heaviest backpack, (2) the lightest backpack, and (3) the average weight of all backpacks. Output values must be rounded to the nearest tenth. See section Rounding to Nearest 100th and 10th at the end of packet for code to accomplish this.

Sample Input

```
7
29
20.1
2
34
12.7
24.6
23.2
```

Sample Output

```
Heaviest backpack: 34.0
Lightest backpack: 2.0
```

```
Average weight of all backpacks: 20.8
```

Problem E - Jaffy's Gobbling Engine

One day in English class, Jaffy is daydreaming instead of paying attention to the teacher. Jaffy envisions a stream of numbers and operators being fed into a Gobbling Engine. Jaffy dreams of his Gobbling Engine becoming famous just like Charles Babbage's Difference Engine - a mechanical computer created in 1820 and programmed by Ada Lovelace. At supper, Jaffy tells Jiffy about his daydream. Jiffy says she can easily write a program that is Jaffy's Gobbling Engine. Jiffy opens her laptop and proceeds to program a Gobbling Engine. You, too, can write a Gobbling Engine.

Input

The first line of input contains a non-negative odd integer N giving the number of lines input to the Gobbling Engine.

Following that are N lines that alternate between a number and an operator. An operator is one of four characters: `+ - * /`. Each character performs the arithmetic operation traditionally associated with it.

Output

Output is a number that is the result of the add, multiply, subtract, or divide pattern. Output values must be rounded to the nearest tenth. See section Rounding to Nearest 100th and 10th at the end of packet for code to accomplish this.

Sample Input 1

```
3
1
+
1
```

Sample Output 1

```
2
```

Sample Input 2

```
17
45
```

```
+
2
/
23
-
6
*
67
+
13
/
3
-
46
*
5
```

Sample Output 2

```
-650.1
```

Problem F - Jaffy Drives a Convertible

Jaffy has always loved riding bicycles. Last semester Jaffy failed his freshman English for the second time. This semester Jaffy is taking freshman English at the Germanna campus in Locust Grove, which is too far from UMW for a bicycle ride. Jaffy learned to drive a car. He found a convertible, which with its top down, accommodates his long neck. Jaffy wants to track his gas usage and expenses for his car. Jaffy's convertible is flexible - like his neck. The gas tank size and the miles per gallon fluctuate from day to day. Given his convertible's miles per gallon, a full tank of gas, a distance to drive, and the cost per gallon of gas; Jaffy wants to know how much gas his car consumes driving a certain distance and how much it costs to refill the gas tank after driving that distance.

Input

Input consists of four lines of input. The first line is the number of miles Jaffy drives or attempts to drive. The second line is the miles per gallon that Jaffy's convertible gets. The third line is the number of gallons of gas Jaffy's tank holds. The fourth line is the current price of a gallon of gas. For this problem, Jaffy's gas tank is full.

Output

The output is four lines. The first line states the distance Jaffy drove or the distance he drove before he ran out of gas. The second line is the amount of gas consumed on the drive. The third line is how many additional miles Jaffy can drive (0 if the tank is empty). The fourth line is how much it costs to refill his gas tank. Output values must be rounded to the nearest hundredth. See section Rounding to Nearest 100th and 10th at the end of the packet for code to accomplish this.

Sample Input

```
113.4
16
300
2.17
```

Sample Output

```
Jaffy drove 113.40 miles!
Jaffy used 7.09 gallons of gas!
Jaffy can drive for 4686.60 more miles!
It will cost Jaffy $15.38 to refill his tank!
```

Problem G - Jaffy fixes Janny's Photos

Jaffy is feeling sad because he has not seen his beloved Janny in over a year. His only comforts are some digital photos of Janny. Unfortunately, the photos are oriented incorrectly. Jaffy must rotate the photo counterclockwise so he can gaze upon Janny's beauty. Each photo has a specific degree for rotation - for example 90° and 180°. Janny has an inexpensive camera with a resolution of 10 x 10. Each photo is a 10 x 10 grid of symbols with a space in between each symbol. Even though the photos do not look like much to you, they bring great joy to Jaffy.

Input

The first line contains a non-negative integer N giving the number of degrees of counterclockwise rotation. The value on this line will be evenly divisible by 90. For example, 0, 90, 180, 270, and 360.

The next 10 lines are the photo. Each line contains 10 characters separated by a space.

Output

10 lines each containing 10 characters separated by a space that is a counterclockwise rotation of the input. The degrees of rotation are on the first line of the input.

Sample Input

```
90
# # # # # $ # # # #
# # # # $ # $ # # #
# # # $ # $ # $ # #
# # # # $ # $ # # #
# # # # # $ # # # #
# # # # # $ # # # #
# # # # $ # $ # # #
# # # # $ # $ # # #
# # $ $ $ $ $ $ $ #
# # # # # # # # # #
```

Sample Output

```
# # # # # # # # # #
# # # # # # # # $ #
# # $ # # # # # $ #
# $ # $ # # $ $ $ #
$ # $ # $ $ # # $ #
# $ # $ # # $ $ $ #
# # $ # # # # # $ #
# # # # # # # # $ #
# # # # # # # # # #
# # # # # # # # # #
```

Problem H - Janny’s Beautiful Ring

After fixing Janny’s photos and gazing at them, Jaffy decided to get Janny a beautiful present. Jaffy knows that Janny loves jewelry in general and rings in particular. Jaffy does not have much money, which is typical for college students. Jaffy decides he can create a ring for Janny and print it using UMW’s free 3D printer. The programming contest must settle for 2D printing. Print Janny’s beautiful ring.

Input:

None.

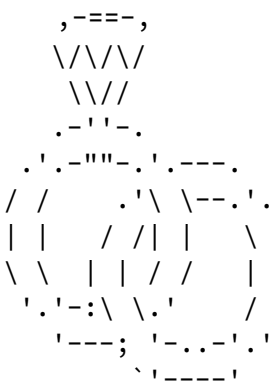
Output

Shown in the sample

Sample Input

None.

Sample Output



Problem I - Helping Aunt Gingersnap

Jiffy and Jaffy's Aunt Gingerbread has a sister named Gingersnap. Aunt Gingersnap has just finished a long journey from South Africa to south Niger, the home of Reticulated Giraffes. Aunt Gingerbread is visiting her mother. Unfortunately, Niger has various ongoing hostilities. One such hostility left several mines between Aunt Gingersnap's current location and her mother's home. In front of Aunt Gingersnap is a deadly minefield where one wrong step can cause great bodily harm. Aunt Gingersnap has found encoded instructions that the warring party uses to safely traverse the minefield. The instructions are encoded using a Caesar cipher. A cipher is an algorithm that translates plain text to encrypted text and encrypted text to plain text. The Caesar cipher shifts letters to the right when encrypting and to the left when decrypting. For example, C is two characters to the right of A and M is four characters to the left of Q. The key on a Caesar cipher is an integer, which shifts letters. When encoding a string, the key shifts to the right. When decoding a string, the key shifts letter to the left. The plain text (unencoded) string DAD when encrypted with a key of 3 shifts each letter 3 positions to the right, resulting in the string GDG. Decrypting GDG with a key of 3 shifts each letter 3 position to the left, resulting in the plain text string DAD. The plain text string ZAC when encrypted with a key of 3 shifts each letter 3 positions to the right resulting in the encrypted string CDF. Decrypting CDF with a key of 3 shifts each letter 3 positions to the left, resulting in the string ZAC.

The only person Aunt Gingersnap trusts to properly decrypt the instructions is her niece Jiffy, who is an outstanding computer science major at UMW. Aunt Gingersnap texts the encrypted instructions to Jiffy who runs a program to decrypt the instructions and texts the plain text back to Aunt Gingersnap.

You are to write Jiffy's program to decrypt the instructions.

Input

The first line of input contains a non-negative integer N giving the number of pairs of lines.

Following that are 2*N lines. Each pair of lines contain a key and an encrypted message. The first line in the pair is positive non-negative integer, which is a key. The second line in the pair is an encrypted message. The encrypted message can contain upper case letters and other characters. The upper case and lower case letters are encrypted and must be decrypted. The other characters are not changed.

Output

Output is N+1 lines. The first N lines are the decrypted message. Line N+1 is `Traversed the Minefield!`, including the exclamation point.

Sample Input

```
2
3
Jr Ohiw 10 ihhw.
4
Ks Pijx 100 jiix.
```

Sample Output

```
Go Left 10 feet.
Go Left 100 feet.
Traversed the Minefield!
```

Problem J - Aunt Gingersnap's Mother's Present

Aunt Gingersnap's mother, Gingerale, was extremely grateful for Jiffy's help in decrypting the instructions. Gingerale decided to give Jiffy a grand present. She retreated to an old tree and returned with a dusty map, which had directions to a buried treasure. Her great-great-great grandfather Gingerbreadman, who lived in the 1700's when there were 1,000,000 giraffes in Africa (today there are about 110,000), owned a solid gold giraffe sculpture. Gingerbreadman buried the sculpture and wrote directions on a piece of parchment. Gingerbreadman was one of the few literate giraffes of his time. The map has remained in the Ginger family ever since. Mother Gingerale gave the map to Aunt Gingersnap. You must help Aunt Gingersnap find the treasure.

Input

The first line contains a non-negative integer N giving the number of lines in directions.

Following are N lines, each giving one direction. Each of these directions say either “Walk P paces”, where P is a positive integer; or “Turn DIR” where DIR is one of north, south, east, or west.

Output

Output consists of one line giving the final location of the treasure relative to the starting position. It must have the form “Treasure is Y paces to the [north/south] and X paces to the [east/west].”

If either X or Y are zero, you must not print that part of the message (as in Sample Output 2). If both X and Y are 0, you should simply output “Treasure is right here.”

Sample Input 1

```
7
Walk 4 paces
Turn east
Walk 10 paces
Turn south
Walk 6 paces
Turn east
Walk 2 paces
```

Sample Output 1

```
Treasure is 2 paces to the south and 12 paces to the east.
```

Sample Input 2

```
8
Turn east
Walk 4 paces
Turn west
Walk 7 paces
Turn north
Walk 5 paces
Turn east
Walk 3 paces
```

Sample Output 2

```
Treasure is 5 paces to the north.
```

Problem K - Jaffy Rides His Bicycle

Jaffy loves to bicycle all of the trails close to UMW. He spends so much time bicycling that he failed freshman English twice. He perfectly paces himself on these bike trails, so that no matter the terrain or weather, he pedals at an exact, constant speed. Jaffy is wondering how long it will take him to traverse a bike trail given his speed and the length of the trail.

Input

The first line of input contains a non-negative integer N giving the number of trails Jaffy pedals.

Following that are 2*N lines grouped in pairs. The first line of a pair is speed in miles per hour that Jaffy pedals. The second line of a pair is the length of the trail in miles.

Output

Output is N lines, where each line has the time it takes Jaffy to pedal the trail. Time is printed in hours, minutes, and seconds.

Sample Input

```
3
10
1
20
5
3
6
```

Sample Output

```
Jaffy can bike the trail in 0 hours, 6 minutes, 0 seconds.
Jaffy can bike the trail in 0 hours, 15 minutes, 0 seconds.
Jaffy can bike the trail in 2 hours, 0 minutes, 0 seconds.
```

Problem L - Jaffy the Fireman

Jaffy is beginning to realize he needs to balance his social, study, and service time. Jaffy decides to become a volunteer fireman for the local Fredericksburg Fire Department. The Fire Department is excited to have Jaffy because he will be helpful reaching taller structures. After fire fighting training, Jaffy is ready to serve as a volunteer. On his first night as a volunteer, he has to put out fires in various rooms of a tall structure. Help Jaffy put out the fires.

Input

The first line of input contains a non-negative integer W that is the number of gallons of water in a tanker Jaffy has to extinguish the various fires.

The second line of input contains a non-negative integer N giving the number of floors in the building.

The third line of input contains a non-negative integer M giving the number of rooms on each floor.

Following that are N lines. Each line has M non-negative integers separated by a space. Each line represents a floor in the building. The floor number is 1 to N, where floor N is the first line read. Each non-negative integer is the number of gallons of water required to extinguish the fire in that room. A zero indicates there is no fire in that room. The room number is 1 to M, where room M is the last number on a floor line. Each room is identified by two numbers - floor and room.

Output

Output is one line for each room fire extinguished or partially extinguished, along with a summary statement. Extinguishing a fire consumes water from Jaffy's tanker. Room fires must be extinguished in the following order

- Floor 1, room 1; floor 1, room 2, …, floor 1 room M
- Floor 2, room 1; floor 2, room 2, …, floor 2 room M
- …
- Floor N, room 1; floor N, room 2, …, floor N, room M

Each room extinguish line has the following format.

```
Room X, Floor Y
```

The summary statement is one of the following.

```
All room fires extinguished.

The building is still burning!
```

Sample Input 1

```
10
3
3
0 0 3
2 0 2
0 1 1
```

Sample Output 1

```
Room 2, Floor 1
Room 3, Floor 1
Room 1, Floor 2
Room 3, Floor 2
Room 3, Floor 3
All room fires extinguished.
```

Sample Input 2

```
100
2
5
0 0 0 0 2
9 0 50 0 40
```

Sample Output 1

```
Room 1, Floor 1
Room 3, Floor 1
Room 5, Floor 1
The building is still burning!
```

Problem M - Jaffy and Janny Together

During Jaffy's volunteer fire fighting, he saved a prominent citizen of Fredericksburg. In doing so, Jaffy sustained several injuries. The room of the prominent citizen was so high that Jaffy had

to use his 20” long tongue to open the door and extinguish the fire. Giraffe’s tongues are tough, but the extremely hot door handle severely burned Jaffy’s tongue. Also Jaffy’s large nasal cavities were scarred when hot embers landed in them. Jaffy was recuperating in Mary Washington Hospital, when the prominent citizen visited with news that they were placing $20,000 into Jaffy’s account. When the $20,000 is added to Jaffy’s account, he has a total of $20,200 dollars. At this same time Janny texted Jaffy that she had been accepted to UMW, but she needed $18,000 to book a special flight on an airplane that could accommodate the heights of giraffes. Jaffy has healed and he is visiting an ATM to transfer money from his account to Janny’s so they can reunite at UMW. He has to enter a correct password to enact the transfer. Help Jaffy perform this bank transaction. There are three password algorithms at the ATM.

1. Passwords can be any length and can be any combination of characters
2. Passwords must be a minimum of 12 characters, and they must contain uppercase letters, lowercase letters, numbers, and special characters from the set `{!@#$%^&*()_-+}`
3. Passwords must be a minimum of 16 characters and the characters must alternate between letter (either uppercase or lowercase) and special characters from the set `{!@#$%^&*()_-+}`

Input

The first line of input contains a non-negative integer N that is the password algorithm defined above. N will be 1, 2, or 3.

The second line is a password that must adhere to the password algorithm in order to allow the transaction to occur.

The third line is the amount to be withdrawn. This is a non-negative integer.

The fourth line is the withdrawal fee. This is a non-negative integer that is the dollar amount fee the bank charges for the withdrawal. The account must contain enough money to cover the amount to be withdrawn plus the withdrawal fee.

Output

Output is one of the following.

```
Invalid password.
Not enough funds in the account.
Successful withdrawal. Account balance is $x.xx.
```

Sample Input 1

```
1
```

```
Ab#c4d
$20,000
1
```

Sample Output 1

```
Successful withdrawal. Account balance is $199.00.
```

Sample Input 2

```
2
Ab#c45d
$10,000
10
```

Sample Output 2

```
Invalid password.
```

Sample Input 3

```
3
A#c4d5d$Z^Q{M&R%W@
$40,000
10
```

Sample Output 3

```
Not enough funds in the account.
```

Problem N - Janny and Jaffy's Wedding

Not only is Janny enrolling at UMW, but Janny and Jaffy are getting married. When Janny received the ring you printed in Problem H, she and Jaffy decided to get married. Giraffes tend to marry early. Jeography is Jaffy's best man, and Jeography has flown to Fredericksburg to help Jaffy with wedding details. Jeography asks Jaffy if they will have sparkling grape juice at the wedding as it is a favourite beverage of giraffes. Giraffes are a bit quirky when it comes to drinking sparkling grape juice, and giraffes have unique taste buds when it comes to drinking sparkling grape juice. Each giraffe only likes to drink sparkling grape juice when it's at a certain temperature as any other temperature doesn't taste quite right. Each wedding guest has their own drinkable sparkling grape juice temperature.

Since Jaffy and Janny want their wedding to be perfect, they will only serve sparkling grape juice at temperatures everyone will enjoy. To make sure everyone is satisfied, they want to provide each giraffe their own bottle at their desired temperature. The refrigerator at the wedding has an internal temperature of 40 degrees F. All of the guests grape juice bottles are in the refrigerator and are exactly 40 degrees. Jaffy knows that every minute outside the refrigerator the grape juice will increase by 0.5 degrees, but the grape juice will not get warmer than the outside temperature. Jaffy wants a list of the times he should remove his friend's grape juice bottles ordered by the amount of time prior to the wedding the bottles should be removed. The guests whose bottles must be removed earlier will appear first in the list. If multiple guest's bottles must be removed at the same time, they must be ordered alphabetically by their names.

Input

- The first N lines define wedding guests and their desired temperature of sparkling grape juice. Each line contains two values. The name of the guest is a sequence of letters without spaces. Following the name is a space. Following the space is a non-negative integer, which is the guest's desired temperature.
- The line following the guests' request contains a single period.
- The next line is an integer, which is the temperature outside on the wedding day

Output

The output is a sequence of lines, one line for each guest in the input. The output first shows guests whose temperature can be achieved. They are ordered by the amount of time prior to the wedding the bottles should be removed. Each line has the guest's name followed by a colon and a phrase that describes how much time prior to the wedding the bottle should be removed from the refrigerator. The time has two components - an integer for the hours and an integer for the minutes. When the hours are one, print the word hour. When the minutes are one, print the word minute. The guests who have unobtainable temperatures appear at the end of the list. Print their name followed by a colon and the phrase " is unable to get the right temperature." in the order they appear in the input.

Sample Input

```
Jaffy 70
Janny 40
Jeography 10
Gingersnap 80
.
70
```

Sample Output

```
Jaffy: remove bottle from fridge 1 hour and 0 minutes before the wedding.
Janny: remove bottle from fridge at the time of the wedding.
Jeography: unable to get the right temperature.
Gingersnap: unable to get the right temperature.
```

Epilogue

After getting married, Janny and Jaffy moved into a modest apartment in downtown Fredericksburg. Janny received a full academic scholarship at UMW to study computer science. The Fredericksburg Fire Department hired Jaffy to a full-time position, and he became a part time student at UMW. Jaffy's study habits improved considerably, and he plans to finish his degree in computer science. With Jaffy becoming a responsible young adult, Jiffy accepted a job with Google in California. Jiffy sold her golden giraffe statue to a museum, and donated $3,000,000 to a giraffe conservatory. Cousin Jessie, the little giraffe that grows 6 inches per month, is almost a full-grown giraffe. Aunt Jessabell is so proud of Jessie, who is studying diligently in order to matriculate to UMW. Aunt Gingerbread, the magical spot counter, appeared on Jimmy Fallon live. She is currently applying her spot counting trade in Las Vegas, helping to locate gambling cheaters. Uncle Jaqarue attended Janny and Jaffy's wedding in Fredericksburg and decided to do some field work in the US. He is helping Texas solve their wild pig problem. Due to hostilities in Niger, Aunt Gingersnap could not get a flight to Janny and Jaffy's wedding. Aunt Gingersnap has watched the YouTube video of the wedding at least a dozen times. If you recall, Janny helped Aunt Gingerbread decipher a message to find Aunt Gingersnap's golden giraffe statue. Aunt Gingersnap has been studying cybersecurity ever since. The US Cybercommand has hired Aunt Gingersnap, and she will soon fly to the US. Aunt Gingersnap's job pays so well, she is bringing Aunt Gingerbread and cousin Jessie with her. Jaffy and Janny are planning a summertime party at UMW where all of their family can visit.

Contest Input/Output[3]

This guide contains the basic way of doing input and output in Java, Python, and C++ in case you need a quick refresher. In programming contests, you always do input/output from/to the terminal screen. Do not open any files for input or output. Do not output any prompts. If a problem instructs you to read a line containing a number, just read the number - it will be there. Do not print a prompt like “Enter a number”.

Java Input

Input in Java can be done with the `Scanner` class which must be imported first:
`import java.util.Scanner;`
Then a `Scanner` object must be created:
`java.util.Scanner in = new java.util.Scanner(System.in);`
1. To read in one line of input into a string, use the scanner’s `nextLine` method: `String line = in.nextLine();`
2. To read in a single word into a `String`, stopping at a space, use the `next` method: `String word = in.next();`
3. To read a numerical value use the `nextInt` method for integers, or the `nextDouble` method for real numbers:

```
int number1 = in.nextInt();
double number2 = in.nextDouble();
```

Java Output

1. To output a `String` constant or variable, use the `System.out.println` function which takes one argument, prints it to the screen, then prints a new line:

```
// print a message
System.out.println("This will be printed")
// print a value
int x;
System.out.println(x);
```

2. To output something without a new line at the end, use the `System.out.print` function which outputs its argument with no newline:

```
// print a message with no new line
System.out.print("The value of X is ")
// print a value with no newline
int x;
System.out.print(x);
```

[3] Section modified from Dr. Finlayson’s Spring 2018 UMW Contest.

Python Input

1. The input function in Python read in one line of input and returns it as a string. For example:

```
line = input() # read in a string
```

2. In order to convert from a string to a number, you can use the int function for integers, or the float function for real numbers:

```
number = int(input()) # read in string and convert to int
```

3. In order to break a line of input into multiple strings, separated by spaces, you can use the `.split()` function which returns a list of strings:

```
line = input() # read in whole line
words = line.split() # split line into strings based on spaces
```

Python Output

1. Output in Python is done with the print function which outputs all of its arguments, separated by spaces, and puts a newline at the end:

```
print("X is equal to", x) # print msg, number, and new line
```

2. In order to prevent print from putting spaces between each item, pass sep=' ' as an argument:

```
print("X is equal to", x, sep= ) # no space between msg and value
```

3. In order to prevent print from putting a new line at the end, pass end=' ' as an argument:

```
print("X is equal to", x, end= ) # no new line added
```

C++ Input

1. To skip over whitespace (spaces or new lines), and read in a single value (such as an integer or string), use `cin >> value;`. For example:

```
int number;
cin >> number; // read one integer
char str1[100];
cin >> str1; // read one character string
string str2;
cin >> str2; read one string object
```

2. To read in one entire line of input, which may contain spaces, use `cin.getline`. For example:

```
char str1[100];
cin.getline(str1, 100) // read character string upto 100 chars
string str2;
getline(cin, str2); // read in a string object
```

C++ Output

Output is done with `cout <<` in C++ which can take any built-in data type. For example:

```
int x;
cout << "X is equal to " << x << endl; // print msg, int, new line
```

Rounding to Nearest 100th and 10th

For problems that print a floating point number to the nearest 10th and 100th, use the following techniques.

Java Rounding

The following demonstrates ways to print a double to the nearest 10th and 100th.

```
double d = 1205.6358;
System.out.print("d: " + String.format("%.1f", d)); // 10th
System.out.print("d: " + String.format("%.2f", d)); // 100th
```

Python Rounding

The following demonstrates ways to print a floating point number to the nearest 10th and 100th.

```
d = 1205.6358
print("d: {:0.1f}, d: {:0.1f}\n".format(d, d)) // 10th
print("d: %.1f, d: %.1f" % (d, d))
print("d: {:0.2f}, d: {:0.2f}\n".format(d, d)) // 100th
print("d: %.2f, d: %.2f" % (d, d))
```

C++ Rounding

The following demonstrates ways to print a floating point number to the nearest 10th and 100th.

```
double d = 1205.6358;
cout<<d<<setprecision(1)<<x<<endl; // 10th
cout<<d<<setprecision(2)<<x<<endl; // 100th
```