

16th
annual
Cy Woods Contest

General Information

If you are an In-Person competitor, you will be using PC^2 to submit. Your solutions should use file input.

If you are an online HackerRank competitor, you will be required to submit with standard input.

HackerRank contestants, please ignore the ‘Input File.’ information at the top of each problem. That is for
the in person contestants and use standard input instead.

All questions must run within 5 seconds.

We will only be supporting Java and Python.

Table of Contents

Problem:	Completed?
Welcome to the Polar Express! CHOO CHOO!!!	
Album	
Your Mom	
Among Us	
Counting Presents	
Drifting Trains	
Child Protective Services	
Santa's Laziness	
Grades	
Curling	
A Place For Everything	
Shift and Encrypt	
Sammy's Ten	
The Puppet Room	
Christmas Table Tennis	
Bitdumb	
Coins	
Rudolph's Broken Legs	
The Know-It-All	
Goodbye World	

01. WELCOME TO THE POLAR EXPRESS! CHOO CHOO!!!

Program Name: MachoTrain.java **Input File:** N/A

Mr. Armstrong wants to end the monopoly that the Polar Express has on train rides to the North Pole. He's planning to create a new competitor company, Macho Choo Choo Trains. Mr. Armstrong needs assistance designing the new trains for his company, and he wants you to help! Macho Choo Choo Trains is clearly the superior company, and therefore it needs a superior aesthetic. Print out the cool and epic design for the brand new Macho Choo Choo Trains company!

Input

N/A

Output

Print out the ASCII art under the 'Example Output to Screen' section.

Example Input File

None

Example Output to Screen

```
          |
         -+-  
          A  
         /=\\  
        i/  o \i  
       /=====\  
      /   i   \  
     i/  o * o \i  
    /==========\  
   /   *   *   \  
  i/  o   i   o \i  
 /=====\  
 /   o   i   o \  
i/  *   o   o   * \i  
/=====\  
          |   |  
o o o   |   |  
o   |   M |   |   A |   |   C |   |   H |   |   O |  
TS__[O]   |   |   |   |   |   |   |   |  
{=====|   |"----"|   |"----"|   |"----"|   |"----"|   |"----"|  
. /o--000'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'
```

02. ALBUM

Program Name: Album.java

Input File: N/A

Redemption is inevitable. John was walking down the street one dark and lonely day, when he stumbled across a broken microphone on the floor. He could not for the life of him figure out where it came from, though it looked like it came from some kind of battle. John was absolutely dumbfounded, and so he picked up the microphone and decided to bring it to his cousin, who was an expert archeologist and microphone specialist. His cousin Samantha took the microphone and looked at it. He was very disturbed by what he saw.

“Where did you get this?” he said to John. John responded, “I found it in an alley and I figured I would bring it to an expert in order to understand what happened.” Samantha spends the next two hundred and twelve hours without sleep, examining the mic day and night. She begins to tell a story.

“Once upon a time, there was a great civilization called the Swiftie Nation. The capital of this nation was Taylor Swift High School, and life in this country was very good. Everyone got free albums to listen to. Anyway, in order to record these albums, they used the sacred microphone. Unfortunately, in 2012, Katy Perry High School, a rival nation, stole the microphone in a midnight raid. For the next nine years, the microphone was never seen or heard of again.”

“WHAT HAPPENED TO THE MICROPHONE?!” John wildly exclaimed in interruption.

“I’m gettin’ there,” Samantha said impatiently. “Maybe the microphone you brought me -just maybe- might be a distant cousin’s nephew’s brother’s second dog’s owner’s pet’s relative’s uncle of the sacred microphone. If that is the case, you are going to be a rich man, John.”

John went on to record 43 and a half studio albums using the microphone, none of which actually caught on. Fortunately, Samantha’s microphone archeology business did take off and the two cousins shared the profits and both lived happily ever after as extremely rich individuals.

Input

No input.

Output

Output should consist of a single picture made up of text. This picture should be of a specific subject very important to John and Samantha. To be more specific, this should be a picture of the original microphone that Taylor Swift High School used to record their albums, NOT of the one that John found on that fateful day. Be sure to print the right picture!

Example Output to Screen

```
/....\  
|.....|  
\____/_  
||  
||  
||  
||  
||
```

03. YOUR MOM

Program Name: YourMom.java

Input File: yourmom.dat

The Polar Crew (the crew of the Polar Express) is hosting a contest to find the best Your Mom joke. The passengers are required to say your mom, but after that, they can say anything. The contest works in groups of 4 jokes then those are judged to find the best one. Use the ascii values of the words after your mom to find the best your mom joke, but "is" after your mom does not count towards the total ascii value. The best your mom joke has the greatest ascii sum.

Input

The first line of input will contain a single integer N , which indicates the number of test cases to follow. Each test case will consist of 4 Your Mom jokes. Each test case is separated by an empty line.

Output

Return which the joke with the greatest ascii sum with "The best Your Mom Joke is: ". There is guaranteed to be a unique greatest sum.

Example Input File

```
4
your mom is bad
Your mom is so dumb
Your mom is a villain
Your mom is a snail

Your mom doesnt work
Your mom is the microsoft store
Your mom went to goodwill
your mom is not a good insult

your mom is small
your mom is a dwarf
your mom is a zzzzz
Your mom is a target for a hitman

your mom is a hitwoman
your mom looks bad
your mom is a canadian aboriginal
your mom is trying to put notepad back on your computer
```

Example Output to Screen

```
The best Your Mom Joke is: Your mom is a villain
The best Your Mom Joke is: Your mom doesnt work
The best Your Mom Joke is: Your mom is a target for a hitman
The best Your Mom Joke is: your mom is trying to put notepad back on your computer
```

04. AMONG US

Program Name: AmongUs.java

Input File: amogus.dat

The ride on the Polar Express is a very long ride, and long rides produce bored children. The bored kids on the Polar Express decide to host a tournament to pass time. An Among Us tournament. The tournament consists of multiple rounds, with the crewmates and imposters revealed to the kids that aren't playing before the round starts. The nerdiest kid on the train, being an avid Among Us fan, knows the average imposter kills of every single player in the tournament. (He isn't playing because he is part of the PC Master Race, you mobile plebs.) He feels confident that he can predict whether the imposters will win or lose before the round starts.

He believes that if the imposters average kills combined equal or exceed the number of crewmates, imposters will win.

Notes: Among Us is a game where there are two teams, imposters and crewmates. Imposters win if they kill all the crewmates before the crewmates complete all their tasks or they vote off all the imposters.

Input

Input starts with an integer P ($1 \leq P \leq 100$) that tells the amount of players in the tournament. The following P lines contain N, a string of the player's name and an integer K ($0 \leq K \leq P$), the average kills of the player. After P lines, a new line contains an integer R ($1 \leq R \leq 10$), the number of rounds. The next R lines contain the names of players that are imposters in that round.

Output

For each round, output "Imposters Win!" if the imposter(s) win the round. If the imposter(s) lose, print out "LOL THESE IMPOSTERS SUCK!".

Example Input File

```
10
Jimbo 3
Timmy 4
Bob 8
XxEpicGamerxX 1
Patrick 3
sus 6
L 2
Armstrong 3
KingGeorgeThe3rd 7
Lauren 4
4
sus Jimbo
XxEpicGamerxX L
Jimbo Timmy Patrick
Bob
```

Example Output to Screen

```
Imposters Win!
LOL THESE IMPOSTERS SUCK!
Imposters Win!
LOL THESE IMPOSTERS SUCK!
```

05. COUNTING PRESENTS

Program Name: Counting.java

Input File: counting.dat

Big boi Santa is a very secretive man and doesn't like children finding out what Christmas presents they're getting, so he puts in more safety measures just in case someone happens to sneak into his hidden kingdom. Santa hides the actual gifts into boxes and labels them with a number so that every number corresponds to a gift. Roger, the class nerd, somehow sneaks into Santa's factory with you after the Polar Express arrives and you two end up at the gift sorting station, where gifts are sorted by numbers. You two sit there for a while counting and keeping track of what presents pass you on the conveyor belt. After some time, Roger, again not wanting to end up on his asian parents' naughty list, decides to practice sorting as his hourly math skill. You suggest that Roger sorts the frequency of the gifts from highest frequency to lowest frequency, and he immediately starts on your given task.

Input

The first number of the data file will provide the number of test cases. Each test case will be one line long. The first number of each line will give the number of numbers in that test case. You can assume all the numbers in a test case are integers.

Output

Print out the 5 most frequent numbers in order from largest frequency to smallest frequency as well as their frequency separated by a space. If there are less than 5 numbers in the dataset, then print out all numbers in the dataset in order from largest frequency to smallest frequency also with their frequency separated by a space. If two or more numbers have the same frequency, print the numbers from smallest to largest with their frequencies separated by a space. Print a blank line in between the outputs of each test case. You can assume that there are more than 0 numbers in a test case and the numbers in the test cases will always be integers.

Example Input File

```
5
15 2 2 2 3 4 4 5 5 5 5 5 3 933478 933478
3 1 1 2
10 1 2 3 3 4 5 6 3 4 5
1 1
2 2 2
```

Example Output to Screen

```
5 6
2 3
3 2
4 2
933478 2

1 2
2 1

3 3
4 2
5 2
1 1
2 1

1 1

2 2
```

06. DRIFTING TRAINS

Program Name: Drift.java

Input File: drift.dat

Oh no! As the Polar Express travels across the ice, the Conductor loses control of the train and they start sliding across the ice, Tokyo Drift style. The train eventually gets caught on some rough patches and slides to a stop, magically into the shape of a perfect right triangle every single time. The Conductor grumpily states that he needs maintenance time to get the train out of the triangular shape so everyone then gets off the train and wanders around on the ice... everyone but Roger. Roger, the nerd that he is, decides to practice a different math skill every hour so he doesn't end up on his asian parents' naughty list, and just now, he decided to practice his Pythagorean Theorem. He counts how many railcars are on one side of the train triangle (leg), and how many on the slanted (hypotenuse) side, and guesses (or rather, calculates) how many railcars make up the second leg. Roger, being the friendly guy that he is, invites you to play along with him. Help him solve for the final leg.

Input

The first number of the input file gives how many times the train drifted to a stop (number of test cases). Each test case consists of two integers on one line separated by a single space. The first number of each test case, L_1 , provides the how many railcars are on one of the legs of the right triangle. The second number, H , provides how many railcars on the hypotenuse of the right triangle. You can assume that the hypotenuse is always larger than the given leg.

Output

Return the length of the other leg, L_2 , to four values after the decimal point.

Example Input File

```
4
3 4
5 13
12 20
6 10
```

Example Output to Screen

```
2.6458
12.0000
16.0000
8.0000
```

07. CHILD PROTECTIVE SERVICES

Program Name: CPS.java

Input File: cps.dat

After delivering presents to children all around the world, Santa must log the coordinates of each house he went to because the Child Protective Services (CPS) needs this information. However, his GPS broke, and he doesn't know where he is. Santa has given you his initial position and retraced his steps. Help him record children into the North Pole database to send to the CPS.

Input

The first line consists of an integer T , the number of test cases in the data set. Each test case starts with a line that has three integers: " N ", the number of steps, " X ", the initial x-coordinate, " Y ", and the initial y-coordinate. " N " lines follow, each consisting of a string " S ", the direction of movement ("up", "down", "left", or "right"), and a nonnegative integer " D ", the distance Santa moves in that step. The coordinates use the Cartesian coordinate system.

Output

For each test case, print X and Y , separated by a space, representing Santa's final coordinates

Example Input File

```
5
1 0 0
up 1
2 1 0
up 2
down 7
4 5 3
up 1
left 3
right 2
down 4
4 -10 3
left 10
down 10
right 10
up 10
7 4 3
up 2
up 5
down 6
left 30
down 10
right 100
up 8
```

Example Output to Screen

```
0 1
1 -5
4 0
-10 3
74 2
```

08. SANTA'S LAZINESS

Program Name: Lazy.java

Input File: lazy.dat

As the number of children increases every day, Santa Clause slowly becomes more and more lazy. He has decided this year that he will only give gifts to those who really want a gift. He has saved the letters he received from the children of the world for the past Y years. He will only get gifts for children who have wished for the same present for N or more years. Although he would normally have his elves do this, they are all working from home, and sending each letter to them all would be too much work for Santa Clause. Write a program to help him determine which children deserve a present, so that he does not need to leave his couch until Christmas day.

Input

The first line will contain an integer T , the number of test cases that Santa needs to consider. Each test case will start with a line containing integers C , the number of children within that test case; Y , the number of letters Santa keeps from each child; and N , the number of years they need to have wished for a present for Santa to get it for them. Then follows data for C children. Each child starts with a line containing their name. Following that is Y lines, listing the names of presents requested by each child during that year, listed by space. Children may request the same present more than once in a year, but Santa Clause will only consider that as one occurrence. There is no constraint for the number of presents listed. It is guaranteed that no two children will have the same names.

Output

For each test case, output the names of all children who deserve a present, followed by a colon (:) and then the names of all presents they will get. The children should be sorted in alphabetical order, and the presents should also be sorted in alphabetical order. If a child deserves no presents, print the child's name, followed by gets dried coal for dinner. Each test case should be followed by 20 hyphens (-).

Example Input File

```
2
2 3 2
The Grinch
Candycane Eggnog Smoothie Cookies Tree Hat Grass Fur Clothes Yoga
Smoothie Cookies Tree Potatoes Computer Lamborghini Tesla
Christmas Eight Rivers Middle School Grass
Elf
Vacation Chair Sleep RedBull
Music Dogs Cats
Monster Energy Shoes Bed Home Room
1 5 5
Santa Claus
Couch Chips Movie Netflix Coke Pepsi Blanket
Pepsi Movie Blanket Sleigh Reindeer Cinema Chair Coke
Coke Potato Chips
Sleep Coke Potato Chips Bed
Computer Netflix Pepsi Coke Blanket Couch
```

Example Output to Screen

```
Elf gets dried coal for dinner
The Grinch: Cookies Grass Smoothie Tree
-----
Santa Claus: Coke
-----
```

09. GRADES

Program Name: Grades.java

Input File: grades.dat

8 Rivers Middle School is taking a field trip to the North Pole via the Polar Express! Santa knows that these students are coming, and he wants to prepare a special gift for the top student in each class. However, 8 Rivers Middle School has an incompetent gradebook system that is incapable of calculating the top students. The computer science club at 8 Rivers Middle School tried to fix this issue, but they just weren't good enough. Therefore, they have decided to outsource this task to you! Calculate the student with the highest grade in each class, so Santa can give presents to these students!

Input

The first line will contain one integer, N , the number of classes in the input. On the first line of each case, there are two integers. The first integer, S , is the number of students in the class. The second integer, G , is the number of grades that each student in the class has. The following S lines contain the name of the student, followed by the integer grades they have earned (separated by spaces). A “Z” grade in the input means that the assignment was never turned in, so therefore it is equivalent to a grade of 0. An “X” grade signals that the student is exempt from this grade, so therefore it is not computed into their average. Each student is guaranteed to have at least one turned in assignment.

Output

Print out the name of the student with the highest grade, followed by their grade rounded to 2 decimal digits.

Example Input File

```
2
2 2
Joe 75 X
Sally 92 87
4 3
Jack 25 Z 10
Nathan 100 100 100
Roger 69 70 82
Victor 90 X 94
```

Example Output to Screen

```
Sally 89.50
Nathan 100.00
```

10. CURLING

Program Name: Curling.java

Input File: curling.dat

Mr. Name is hosting a simplified game of curling where you try and put a curl down an alley a certain distance D . He has recorded the distance from the center for every shot. However, he has a lot of different shots to sort and doesn't want to sort them by hand, so he wants you to write a program so that given a distance D you can sort L shots by the absolute value of their distance from D .

Input

The input starts with a line containing one integer N (the number of following test cases) $1 < N < 20$. Each case starts with an integer L (the number of following lines) $1 < L < 10000$ and a double D (the number you need to sort to) $0 < D < 1000$. That is then followed by L lines each containing one double X (the number that must be sorted) $1 < X < 1000$. No two X s will be the same distance from D .

Output

The numbers X sorted by their proximity to D each on a new line. End each case with "-----"(10 hyphens).

Example Input File

```
3
1 10.0
10.0
4 4.0
5.0
0.0
6.0
1.0
5 212.7
212.6
100.0
212.7
212.9
212.0
```

Example Output to Screen

```
10.0
-----
5.0
6.0
1.0
0.0
-----
212.7
212.6
212.9
212.0
100.0
-----
```

11. A PLACE FOR EVERYTHING

Program Name: APlace.java

Input File: aplace.dat

Wuh oh! It looks like Santa and his board of execs are at it again! This time, we need to make a program to distribute a number of tickets for the Polar Express to every neighborhood on a grid. Get on it quick, before you get fired!

Input

A number " n ", which will be the number of following input sets, and " x ", where $1 \leq x \leq 10$, the allowed width of your grid, followed by " a ", where $0 \leq a \leq$

Output

A grid of width "x" containing every digit in "a", with one space between each row element. Put blank lines between each test case.

Example Input File

```
3  
3 123456789  
3 123456789101112  
4 1234
```

Example Output to Screen

1
2
3
4
5
6
7
8
9

1	2	3
4	5	6
7	8	9
1	0	1
1	1	2

1 2 3 4

12. SHIFT AND ENCRYPT

Program Name: Shift.java

Input File: shift.dat

You are a newly-hired elf working in Santa's factory in the North Pole working in the sorting department. This job requires you to travel around the factory dropping off toys and gifts in the correct places, and many of these places are blocked off by another one of Santa's safety measures: encrypted passwords. Santa gives you a word and a number and you must correctly encrypt the word or else you will not be able to access the gift vaults. Since this is a very niche skill set, Santa gives you practice problems so you can quickly encrypt the word when you are working. To encrypt the word, you must shift the string to the right by the given integer. After the shifting is done, you must reverse the string to finalize the password.

Input

The first line of the input is an integer, T, the number of test cases in the dataset. Each test case consists of two lines, one with an integer N, that represents the number of characters you should shift the string by; the second line of each test case contains the string that you want to shift and reverse. The input string may be more than one word. The input number will always be less than the length of the given string.

Output

Print the resulting string after it has been shifted and reversed.

Example Input File

```
2
1
Armstrong
2
Computer science rocks
```

Example Output to Screen

```
nortsmrAg
cor ecneics retupmoCsk
```

13. SAMMY'S TEN

Program Name: Sammy10.java

Input File: sammy10.dat

Meanwhile, as the Polar Express is traveling to the North Pole, Sammy Klaws is planning a heist on Amazon! He has reunited his old friends that he served in the Christmas Wars of '18 with. He has James, the tech specialist, adept at scrambling Amazon's security and comms; Tristan, able to quietly and efficiently infiltrate; Zeki and Steven, the distractions, able to keep a crowd's attention through even a major catastrophe; Old Man Austin (or was it Jackson?), ignored by society, able to walk out with the gifts; Sid, able to make people vomit by showing them his special hand tricks(DON'T ASK); The Bens, a dynamic duo of destruction to hinder further operations from Amazon; and Adah, the getaway sleigh driver. The plan for the heist is simple. Old Man Jackson (Austin?) will pose as a homeless man living outside Amazon HQ. James will get a job as a temporary security contractor with Amazon. Tristan will live in the vents. On the day of the heist, Zeki and Steven will pull the infamous 'Help' distraction (a la Thor Ragnarok). James will disable the security while Tristan gets the Old Man and the Bens into the facility. Adah will land the sleigh on the roof of the facility. Sammy will join the Old Man in getting presents to the sleigh while the Bens disable the facility. After the sleigh is loaded, Adah will take to the skies while the rest of the team covertly exfiltrate. If things go poorly, there is a designated extraction zone for Adah to pick up the team if things go south. Through the operation, the team will be using radios. The radio chatter will be random to throw off Amazon, but if the notorious warlord Stroud is mentioned, then the team needs an extraction.

Input

The first line is an integer, N, representing the number of test cases. There will then be N test cases, with each test case at most 20 lines long, containing random space-separated strings. Each test case ends with "=".

Output

If "Stroud" is mentioned (does not have to be by itself, can be within a word), print "Bravo seven commencing extraction". Otherwise, print "Bravo seven remaining dark".

Example Input File

```
3
All the infections that the sun sucks up
From bogs, fens, flats, on Prosper fall and make him
By inch-meal a disease! His spirits hear me
And yet I needs must curse. But theyll nor pinch,
Fright me with urchinshows, pitch me i the mire,
Nor lead me, like a firebrand, in the dark
Out of my way, unless he bid em; but
For every trifle are they set upon me;
Sometime like apes that mow and chatter at me
And after bite me, then like hedgehogs which
Lie tumbling in my barefoot way and mount
Their pricks at my footfall; sometime am I
All wound with adders who with cloven tongues
Do hiss me into madness.
=
stroud
=
Hey, I could go on and on and on
I could explain every natural phenomenon
The sun, the stars, the clouds
That was just Maui messing around
I killed an eel, I buried its guts
It sprouted a tree, now you got coconuts
Whats the lesson what is the takeaway
```

13. SAMMY'S TEN

Page 2

Dont mess with Stroud when hes on a breakaway
=

Example Output to Screen

Bravo seven remaining dark
Bravo seven remaining dark
Bravo seven commencing extraction

14. THE PUPPET ROOM

Program Name: Puppet.java

Input File: puppet.dat

Remember the REALLY creepy puppet room on the Polar Express. Well, this year's non-believers have found their way in. And, on top of that, the ghosts of Christmas pasts are REALLY mad at these kids dissing Christmas. So, the spirits have become malicious and are jumpscaring the kids a la FNAF. Nathan, immune to jumpscares, is trying to find the last kid stuck in the puppet car. However, a puppet is imitating the cries for help. Nathan will go to the one closest to him. If the puppet and the child are the same distance away, the puppet's cries will drown out the trapped child's screams, and Nathan will go to the puppet.

Input

The first line will contain an integer N, the number of data sets. Each data set is a 10x10 character matrix representing a map. Each data set will be separated by an "_" (underscore).

"#" represents a traversable area.

"H" represents an impassable scrap heap.

"S" represents where Nathan is starting.

"J" represents the puppets ready to jumpscare.

"F" represents the tangible fear of the child.

Output

For each data set, print "Nathan saved the child." if the child is saved and "The spirits feast on blood." if Nathan walks into the ambush.

Example Input File

```
3
SHHHHHHHHJ
#####
#HHHHHHHH#
####F#####
HHH#HHHHH#
HHH####
HHHHHHHHH#
#####
##H#H#HHH
HH##H#H#H#H
```

```
—
JHHHHHHHHH
#HHHHHHHHH
#HHHHHHHHH
#HHHHHHHHH
###H###S
#HHHHHHHHH
#HHHHHHHHH
#HHHHHHHHH
#HHHHHHHHH
FHHHHHHHHH
```

```
—
HHHHHHHHHH
HHHHHHHHHH
HHHHJHHHH
HHHH#HHHH
HHHH#HHHH
HHHHSHHHH
```

14. THE PUPPET ROOM

Page 2

```
HHHHH#HHHH  
HHHHH#HHHH  
HHHHHFHHHH  
HHHHHHHHHH
```

Example Output to Screen

```
Nathan saved the child.  
The spirits feast on blood.  
The spirits feast on blood.
```

15. CHRISTMAS TABLE TENNIS

Program Name: TableTennis.java

Input File: tabletennis.dat

After all of the traveling they have done, Santa's reindeers, specifically Prancer, Dasher, Comet, and Vixen, have gotten bored and decided to play a game of table tennis. However, this isn't just your average table tennis game—it has 4 players. Additionally, they roll a 100000-sided die to decide how many points they need to win. The game ends when one of the reindeer becomes the first player to achieve the winning point value.

Input

The input consists of an unknown number of lines, each containing a test case. There will be no more than 2500 lines in the input. The first and only line of each test case consists of W , the number of points required to win, followed by a string G , the game representation, containing only P, D, C, and V. The letters P, D, C, and V each represent one point earned by Prancer, Dasher, Comet, and Vixen, respectively.

Output

Print the name of the reindeer who won each game of Christmas table tennis.

Example Input File

```
11 DDDDPVCVDPPPVVCCCPDPDVCVVPPCCDCDDCP  
5 CPDDPPCCCVPVDVD  
20 CVPVVPPCVDPPVDPVVCVPDPPDVCVCDCDPPVCPVVCVPCCCDPCPDVDCVCDPPDV
```

Example Output to Screen

```
Prancer  
Dasher  
Vixen
```

16. BITDUMB

Program Name: Bitdumb.java

Input File: bitdumb.dat

Roger's dog Nathan is bitdumb. Roger wants it to be bitwise. If the dog isn't trained by the time the Polar Express comes, Nathan won't be able to go see Santa and the elves in the North Pole. Given a positive integer D , get Nathan to do a trick and then feed this D to Nathan. Help Roger train his dog.

To get Nathan to do a trick, give him the value of $1 \wedge 2 \wedge 3 \wedge \dots \wedge D$, where \wedge is the bitwise XOR operator. To feed Nathan, give him the value of $1 | 2 | 3 | \dots | D$, where $|$ is the bitwise OR operator.

Input

The first line contains an integer T , the number of test cases. T lines follow, each with D , $1 \leq D \leq 2^{1024}$.

Output

For each test case, print the trick number followed by the feeding number, with a space in between. Good luck helping Roger train his dog!

Example Input File

```
4
5129
263776
2487512
48228443
```

Example Output to Screen

```
1 8191
263776 524287
2487512 4194303
0 67108863
```

17. COINS

Program Name: Coins.java

Input File: coins.dat

Millie Moose is really excited to go to the annual Cy Mustard Meese Holiday Party, so that she can drink some mustard. Unfortunately, her car broke and she has to ride the Polar Express to the party, which requires admission to be paid in exact change. Millie has a ton of really weird coins and she doesn't know if it would be possible to come up with the exact amount. Write a program that determines whether there is any combination of coins that adds up to the full amount without going over. Not every coin must be used, and each coin can only be used once.

Input

Input begins with an integer t , the number of test cases, followed by t test cases each containing the following: p , being the full price of the ticket in cents ($0 < p < 100,000$), followed by a line containing the number, c , of coins in Millie's pocket, followed by c lines each containing the value in cents of a coin.

Output

For each case, if there is any combination of coins that add up to the full price, print "MUSTARD ACQUIRED.", as Millie will be able to get to the party and drink some mustard. If not, print "TOO BAD."

Example Input File

```
3
199
5
100
50
22
49
66
254
2
100
50
590
7
10
20
30
1232
42352
560
120
```

Example Output to Screen

```
MUSTARD ACQUIRED.
```

```
TOO BAD.
```

```
MUSTARD ACQUIRED
```

18. RUDOLPH'S BROKEN LEGS

Program Name: Rudolph.java

Input File: rudolph.dat

During the quarantine, Santa and his reindeers have been experimenting with new activities to fill up their time. Their most recent activity has been soccer: Rudolph, especially, enjoys kicking the toys of the children that Santa dislikes. However, one of these children requested a 500 lb. dumbbell, which caused Rudolph to break his legs.

Since Santa does not care about workplace compliance, he is still forcing Rudolph to deliver presents this Christmas. Full of fear for his own life, Rudolph has called you to help him find if he will survive. Your task is to help Rudolph find a set of paths between children such that Rudolph can visit all the children that he is able to access using only those paths and the sum of the length of the paths is minimal.

Rudolph may start his journey at the home of any child, and he may visit a child more than once. Some children are on the naughty list and have no paths connecting them to other children. For these, Rudolph does not need to deliver presents to them.

Input

The input will start with an integer T , denoting the number of test cases. Each test case begins with integers V and N , the number of children that Rudolph must visit and the number of paths available for Rudolph to take between children. The following N lines denote a possible undirected path between two children: an integer representing child A , an integer representing child B , and an integer W representing the length of the path. Integers A and B will be zero-indexed.

Output

For each test case, print out a single integer, denoting the total length of the set of paths that allows Rudolph to deliver presents to every child that he can access.

Example Input File

```
2
6 8
0 1 2
0 2 11
0 3 1
1 3 9
1 4 10
2 5 20
3 4 3
3 5 1
8 11
0 1 3
0 2 5
1 3 4
2 3 12
3 4 9
3 7 8
4 5 4
4 6 5
4 7 1
5 6 6
6 7 20
```

18. RUDOLPH'S BROKEN LEGS

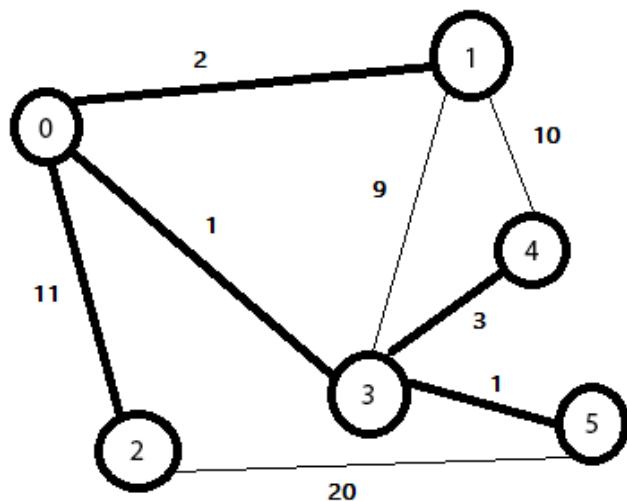
Page 2

Example Output to Screen

18
30

Explanation

The graph shown below corresponds with the first test case. It can be proven that the set of paths shown (bolded in black) has the minimum total length of all paths that visit all nodes. It has a combined weight of $2+11+1+3+1=18$. Thus, this is our output.



19. THE KNOW-IT-ALL

Program Name: Know.java

Input File: know.dat

The Know-It-All is annoying, and Chris needs something to distract him, so he can focus on saving the Polar Express and Christmas.

Chris has come up with a game: given a string X and a string Y, count how many times string X can be made up using the letters in string Y, case sensitive, in order. However, Chris does not want to check all of the Know-It-All's answers, so he wants you to write a program to verify them for him.

In other words, your goal is to find the number of unique sequences s of increasing indices into the string Y such that the concatenation of $y[s[0]] + y[s[1]] + \dots + y[s[n]]$ is equal to the string X.

Input

The first line contains an integer T, the number of datasets for you to interpret. Each dataset has two lines, the first being phrase X, and the second being phrase Y.

Output

For each test case, print the total number of times string X appears as a subsequence of string Y.

Example Input File

```
4
Christmas is here!
CChristmas is here!!
Santa has 212 SPIRIT.
Santa has 212 SPIRIT
sistersarahhasthebell
siststeer sarah hhass tthhee bell. i took it from her
TheConductorisMacho
The Conductor is the unnamMed conductor of the Polar Express. He is Macho.
```

Example Output to Screen

```
4
0
384
201
```

Explanation

For the first test case, the four unique sequences are as follows:

1. CChristmas is here!!
2. CChristmas is here!!
3. CChristmas is here!!
4. CChristmas is here!!

The highlighted characters denote the subsequence of string Y that makes up string X. Thus, string X appears as a subsequence of string Y four times, giving us our answer.

20. GOODBYE WORLD

Program Name: Bye.java

Input File: bye.dat

Santa sometimes flies too high and exits the atmosphere. Fortunately, he's protected by Christmas magic and won't die. You, however, are a reindeer pulling his sleigh and it's not healthy for reindeers to be in space!

Input

You will receive an integer T , the number of test cases. The following T lines each contain an integer X , the number of miles above Earth's surface at which you are flying.

Output

Space starts at 62 miles above Earth's surface. For each test case, if you're at or above this height, output "Goodbye World". If you're still in Earth's atmosphere, output "Flying smoothly!".

Example Input File

```
3
4
90
62
```

Example Output to Screen

```
Flying smoothly!
Goodbye World
Goodbye World
```