

Practice Problems

PRACTICE [BIZZ] BizzFuzz

Description

It's time to enter the real world! After 4 rewarding years in college, you're ready for your first big job interview. The company loves you, but they need to make sure you know how to program! They've given you one final test before you can land the job. They call it: BizzFuzz.

Input Specification

<No input>

Output Specification

Print out the all of the numbers from 1 to 100 (inclusive: interval [1, 100]) -- but there's a twist.

If the number is a multiple of 4, print out Bizz.

If the number is a multiple of 7, print out Fuzz.

If the number is a multiple of 4 AND 7, print out BizzFuzz.

If the number is NOT a multiple of 4 OR 7, print out the number.

Sample Input/Output

Input	Output
	1 2 3 Bizz 5 6 Fuzz Bizz 9 10 11 Bizz 13 Fuzz 15 Bizz 17 18 19 Bizz Fuzz 22 23 Bizz 25 26 27 BizzFuzz 29 30 (etc)

PRACTICE [ECHO] Echo

Description

Your little brother is being **so** annoying today! You know what really gets on his nerves, though? Repeating everything he says!

Input Specification

The first line of input will be an integer $0 < N < 100$ representing the number of lines to follow.

The following N lines will each contain a string S with $1 \leq \text{length}(S) \leq 50$.

Output Specification

For each line of input, S , print out S to stdout.

Sample Input/Output

Input	Output
5 You're being mean! Hey, stop doing that. Stop copying me! Mom, do you hear this? *angry glare*	You're being mean! Hey, stop doing that. Stop copying me! Mom, do you hear this? *angry glare*

Competition Problems

[FIBS] FibsJubsMagoo (Easy - Type: Ad Hoc)

Description

You thought BizzFuzz was easy? Well now that you're on the job, it's time to put those skills to the test once again! Your boss has assigned you as lead developer on the company's newest product: FibsJubsMagoo. People loved BizzFuzz, but it's a fast-paced market and people are already asking when you'll release the next app to revolutionize the app store. Show them your talent!

Input Specification

<No input>

Output Specification

Print out the numbers from 1 to 200 (inclusive: interval [1, 200]) according to these special rules:

- If the number is a multiple of 2, 5, AND 9, print out FibsJubsMagoo.
- If the number is a multiple of both 2 AND 5, print out FibsJubs.
- If the number is a multiple of both 2 AND 9, print out FibsMagoo.
- If the number is a multiple of both 5 AND 9, print out JubsMagoo.
- If the number is only a multiple of 2, print out Fibs.
- If the number is only a multiple of 5, print out Jubs.
- If the number is only a multiple of 9, print out Magoo.

Sample Input/Output

Input	Output
	1 Fibs 3 Fibs Jubs Fibs 7 Fibs Magoo FibsJubs 11 Fibs 13 Fibs Jubs Fibs 17 FibsMagoo (etc)

[SOCKS1] Socks I (Easy - Type: Strings)

Description

The ACM President absolutely HATES laundry day! Worst of all, he hates trying to match all of his socks. Being the good friend that you are, you decide to visit him to help match all of his socks. The President needs your help in determining how many pairs of socks he has.

Input Specification

The first line of input will be an integer $0 < N < 1,000$ representing the number of lines to follow. The following N lines will each contain a single word: red, green, blue, white, or black, followed by a newline character.

Output Specification

On the first line, output the number of pairs of red socks the President can make.

On the second line, output the number of pairs of green socks the President can make.

On the third line, output the number of pairs of blue socks the President can make.

On the fourth line, output the number of pairs of white socks the President can make.

Finally, on the fifth line, output the number of pairs of black socks the President can make.

Sample Input/Output

Input	Output
7 red blue green green blue black white	0 1 1 0 0

[SOCKS2] Socks II (Easy, Type: Strings, Arrays)

Description

Oh boy... laundry day again! This time the ACM President doesn't even care if his socks are mismatched. Can you help him find out how many pairs of socks he can make? (That way he can count the days until he has to do laundry again)

Input Specification

The first line of input will be an integer $0 < N < 1,000$ representing the number of lines to follow. The following N lines will each contain a single word: red, green, blue, white, or black, followed by a newline character.

Output Specification

On a single line, output how many mismatched pairs of socks can be made, given that the President has matched all other socks that he can.

NOTE: You ONLY output the number of MISMATCHED pairs of socks.

Sample Input/Output

Input	Output
5 red red red blue black	1
(Explanation: a mismatched pair of red + blue or red + black socks can be made. The one remaining sock that CAN'T be matched is ignored.)	

[BIZ] Starting a Business (Easy, Type: Float, Arrays)

Description

Your friends have decided to start a business where you sell lemons! Each person has to “buy in” by supplying lemons for the business to sell, but each person has a different number of lemons to contribute. You decide to have each person’s ownership in the company equal to the number of lemons they supply.

Input Specification

The first line of input will be an integer $0 < N < 10,000$ representing the number of lines to follow. The following N lines will each consist of a single number $0 < M < 100$ representing the number of lemons someone can provide.

Output Specification

Your program should output N lines showing the rounded percentage of the total lemons from the group that the corresponding line from the input has.

Note: View the sample input/output for clarification.

Note 2: *Rounded* not truncated. For example, if the percentage is 77.6%, you should output 78%.

Sample Input/Output

Input	Output
6	10%
2	10%
2	19%
4	19%
4	24%
5	19%
4	

(Explanation: There are 21 lemons in total. The 10% from the output corresponds to the “2” from the input, because $2 / 21 = 10\%$ when rounded. Similarly, the 19% corresponds to the 4 from the input, and the 24% corresponds to the 5 from the input.)

[AIRUP] Airline Upgrades (Easy-Medium, Type: Sorting)

Description

It is a commonly known fact that airline upgrades are biased towards travelers with certain names. A new airline company, Bob Airlines (CEO: Bob), has hired you to create their airline upgrade system.

Input Specification

The first line will contain two integers N, M with $0 < M < N < 10000$. N is the number of passengers and M is the number of available upgrades. The following N lines will be the names of passengers in the format "FirstName LastName" followed by a newline character. Names are less than 30 characters in length. All last names will be unique.

Output Specification

Your program should output the M passengers chosen for upgrades. Upgrades are given in the following order:

1. Passengers with a first name of Bob
 - a. If there are multiple Bobs traveling, sort alphabetically by last name.
2. Everyone else in alphabetical order (by last name)

Sample Input/Output

Input	Output
7 3 Bob Anderson Adam Robinson Lynn Todd Bob Smith Christine Jenkins Arthur Cain Carolyn Peterson	Bob Anderson Bob Smith Arthur Cain

[ZOMBIE] Zombie Invasion (Medium-Hard, Type: Graph)

Description

Oh no! Zombies have invaded, and you take refuge in a safe house. It doesn't have many supplies, though, and there's one not far away with much better supplies. You have a map of where all the zombies are, and you try to figure out if there exists a safe path from you to the new safe house.

Input Specification

The first line of input will contain 2 integers $0 < N, M < 10$. The following N lines will each contain M characters followed by a newline. Each character will either be "O" representing a safe space to be, or "X", representing a space with infected zombies that you do not want to be. There will also be one character that is "S", representing your starting point, and one character that is "E", representing your ending point.

Output Specification

Output "True" if there exists a path from S to E without traveling through an "X" character, "False" otherwise. Note that you can only travel vertically and horizontally.

Sample Input/Output

(Example 1) Input	Output
5 5 S0000 000X0 X000X 0X000 0000E	true
(Example 2) Input	Output
5 6 S000X0 0000X0 00XX00 0X000E 0X0000	false

[CLEAN] Cleaning Crew (Medium-Hard, Type: Graph)

Description

Your parents have asked you to help clean the kitchen. The floor is a mess, and they've asked you to help mop it. The family is going to split up the work, though, so you won't have to mop the whole thing! I realize this story is starting to sound a little weird and make no sense, but hear me out! They will tell you to mop a certain section of dirty floor and then you can be done with your chores for today. Stop laughing, this story totally makes sense.

Input Specification

The first line of input will contain 4 integers $0 < N, M, I, J < 10$. The following N lines will each contain M characters followed by a newline. Each character will be either `C` to represent a clean space or `D` to represent a dirty space. The space you should begin cleaning from is row I , column J . You will clean the dirty space at $[I, J]$, all adjacent dirty spaces to $[I, J]$, and continuing to clean all adjacent dirty spaces in your dirty section of the floor.

Note: Row and column indexing starts at 0. The top left corner is $[0, 0]$ and the bottom right corner is $[N - 1, M - 1]$.

Note 2: It is GUARANTEED that the space $[I, J]$ will be marked as dirty.

Output Specification

You should output the graph after cleaning ONLY the dirty spot on the floor. You need to keep cleaning in a zone around the dirty spot until you've cleaned the whole dirty area. If you've ever used a program like Paint, it may help to imagine the problem like the paint fill tool. You continue outwards from a spot and "fill" all spots that are the same color.

Sample Input/Output

(Example 1) Input	Output
4 4 2 3 DCCC CCDD DDCD DDCC	DCCC CCCC DDCC DDCC
(Explanation: We first clean the space at $[2, 3]$, then expand to $[1, 3]$, then $[1, 2]$. Once we are surrounded by all clean spaces, we are done cleaning)	
(Example 2) Input	Output
4 5 2 1 CDCDC DCDCD CDCDC DCDCD	CDCDC DCDCD CCDCD DCDCD

[CENSUS] Census (Hard - Type: Sorting)

Description

You've been hired by the city of NumbersOnly to conduct a census! Oddly, NumbersOnly doesn't actually care about any data of its citizens other than their ages. Your task is to find out who lives in the city, sort them by age, and report back to the mayor.

Input Specification

The first line of input will be an integer $10 < N < 2,000,000,000$ representing the number of lines to follow. The following N lines will each consist of a single number $0 < M < 100$ representing the age of a citizen in NumbersOnly.

Output Specification

On line 1, print out how many citizens of NumbersOnly are 1 year old.

On line 2, print out how many citizens of NumbersOnly are 2 years old.

...

On line k , print out how many citizens of NumbersOnly are k years old.

...

On line 99, print out how many citizens of NumbersOnly are 99 years old.

Sample Input/Output

Input	Output
10 6 9 11 2 4 4 3 8 8 8	(Output is simplified for space reasons) 0 1 1 2 0 1 0 3 1 0 1 (lines 12 - 99 are 0)
(Explanation: no one from the census was age 1, there was one person age 2, one person age 3, two people age 4, no people age 5, one person age 6, no people age 7, three people age 8, one person age 9, no people age 10, one person age 11, and no one age 12 or older) NOTE: Even if an entry is zero, it MUST be printed. This output was simplified due to space limitations.	

[HWHELP] Homework Helper (Hard, Type: Strings, Math, Ad Hoc)

Description

Your favorite math teacher is extremely busy this weekend and has asked for your help grading tests. He will hand you a stack of tests and you will assign a grade to them. Luckily, these are just basic pre-algebra tests, so the grading should go by quickly!

Input Specification

The first line of input will contain two integers $0 < N, M < 10$ where N represents the number of students you will need to grade and M represents the number of problems for each test.

You will then receive N sets of $(M + 1)$ lines in the following format:

- The first line: A string S with $1 \leq \text{length}(S) \leq 20$ representing this student's name
- The next M lines: a math equation of the form $\text{LHS} = \text{RHS}$ where...
 - LHS is a single integer
 - RHS will be an equation that may use the symbols $+$, $-$, $*$, $/$, $($, and $)$.
 - There will be NO SPACES between each symbol
 - Each line L will have $1 \leq \text{length}(L) \leq 50$

Note: Looking at the sample input/output will probably make this much clearer.

Output Specification

Your program should output M lines showing the student's name followed by a colon and a space, then their score (rounded to the nearest percent).

Note 2: *Rounded* not truncated. For example, if the score is 77.6%, you should output 78%.

Sample Input/Output

Input	Output
4 3 Alice 4=2+2 5=3*5 10=1+1 Bob 1=0+1 100=5*4*3*2*1*(5/6) 0=0 Charlie 8=8-1+1 54=5+4 71=7*10+1	Alice: 33% Bob: 100% Charlie: 67%