

Problem 1: No More Shouting

Points: 5

Author: Brett Reynolds, Orlando, Florida, United States



Problem Background

It's common knowledge that on the internet, TYPING IN ALL UPPERCASE LETTERS ISN'T VERY POLITE. It just looks like you're shouting at people, which isn't a very good way to hold a conversation. You've been asked to design a browser extension that can (forcibly) calm everyone down by converting UPPERCASE SHOUTING into lowercase whispers. Try to stay calm as you solve this problem.

Problem Description

Your program will be given lines of text in which all letters are uppercase. You must convert these letters to lowercase without otherwise changing the content of the text.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include a single line of text consisting of uppercase letters, numbers, spaces, and/or punctuation.

2

```
THIS SENTENCE IS IN ALL CAPS  
SHOUTING ISN'T NICE.
```

Sample Output

For each test case, your program must output the provided string after replacing all uppercase letters with their lowercase equivalents. Spaces, numbers, and punctuation should not be modified.

```
this sentence is in all caps  
shouting isn't nice.
```

Problem 2: Sum It Up

Points: 5

Author: Shelly Adamie, Fort Worth, Texas, United States

Problem Background

Adding up numbers is very easy, unless you add a twist. If two numbers are the same, sum their sums!

Problem Description

Your program will be given two numbers. If they are not equal, return their sum. If they are equal, return double their sum.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include a single line consisting of two non-negative integers separated by spaces.

```
5
1 3
2 2
3 2
13 13
125 9
```

Sample Output

For each test case, your program must output the value calculated according to the rules described above.

```
4
8
5
52
134
```

Problem 3: Goofy Gorillas

Points: 5

Author: Shelly Adamie, Fort Worth, Texas, United States

Problem Background

The local zoo's most popular exhibit contains two gorillas. However, the gorillas can cause the zookeepers some issues. We need to be able to alert the zookeepers of trouble in the gorilla compound.



Problem Description

Your program will be given information about whether each of the gorillas is smiling or not. We need to alert the zookeepers if both gorillas are smiling (which might mean they're causing trouble), or if neither gorilla is smiling (which might mean they're about to fight). If only one gorilla is smiling, everything is probably ok.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include a single line containing two boolean values ("true" or "false") separated by spaces.

```
2
true false
true true
```

Sample Output

For each test case, your program must output "true" if the zookeepers should be alerted about potential trouble, or "false" if everything seems ok.

```
false
true
```

Problem 4: Caught Speeding

Points: 10

Author: Holly Norton, Fort Worth, Texas, United States

Problem Background

You are driving a little too fast, and a police officer pulls you over. He needs to determine how big your speeding ticket should be; fortunately for you, he's decided to give you a bit of a break if it happens to be your birthday.



Problem Description

Your program should compute the ticket you are going to receive based on the speed you were travelling:

- If your speed is 60 or less, you don't get a ticket.
- If your speed is between 61 and 80 inclusive, you get a small ticket.
- If your speed is 81 or higher, you get a big ticket.

If today is your birthday, all of these numbers are increased by 5 (for example, you can drive up to 65 without getting a ticket).

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will consist of a single line, including two values separated by spaces:

- A positive integer representing your speed
- The word "true", indicating today is your birthday, or the word "false", indicating it is not.

```
3
60 false
82 false
83 true
```

Sample Output

For each test case, your program must print a single line, as follows:

Problem 4: Caught Speeding

- Print “no ticket” if you do not receive a ticket
- Print “small ticket” if you receive a small ticket
- Print “big ticket” if you receive a big ticket

no ticket

big ticket

small ticket