

A. Second Smallest Number Prime

Time limit: 2 sec, Memory Limit: 2 MB

There is an array, you have to find second smallest number from the array and check it prime or not.

Input:

The first line is number of test cases T ($1 \leq T \leq 100$) then for every test cases first line is Number of elements in the array N ($2 \leq N \leq 10$) then next line contains elements of the array using a space between two elements. The array element can be maximum ≤ 100000).

Output:

If the second smallest number is prime then just print: "The second smallest number is a prime." otherwise if not prime print: "The second smallest number is not prime." and if there is no second smallest number then print: "There is no second smallest number."

Sample Inputs:	Sample Outputs:
2 5 2 4 6 8 10 5 1 1 2 3 3	Case 1: "The second smallest number is not prime." Case 2: "The second smallest number is a prime."

Problem Setter:

Md. Abdullah Saleh

Programmer

IT Consultants Ltd

B. Count Squares

Time limit: 2 sec, Memory Limit: 2 MB

Square number is a number whose square root is an integer. As example 4 is a square number and its square root is 2. Now you will be given two numbers and you have to find how many square numbers are between them inclusively.

Input:

Input starts with an integer T (≤ 100), denoting the number of test cases.

Then each contains two integers A and B . Here ($0 < A, B < 100001$)

Output:

For each case, you have to print Case #: S , here # is the case number and S is the number of square numbers within A and B inclusive. See sample input output for better understanding.

Sample Inputs:	Sample Outputs:
3	
1 5	Case 1: 2
1 13	Case 2: 3
1 2	Case 3: 1

Problem Setter:

Md. Gulzar Hussain

C. Number of Set Bits

Time limit: 1 sec, Memory Limit: 2 MB

Do you know anything about number of set bits in an integer? If no then let me inform you that number of 1s in an integer when it converted to its binary equivalent is the number of set bits in that integer. That is let think a number 4, its binary form is 100. There is only one 1 in 100. So number of set bits in 4 is 1.

Input:

Input starts with an integer T (≤ 100), denoting the number of test cases.

Then each contains an integers N. Here ($1 < N < 2^{32}$)

Output:

For each case, you have to print Case #: S, here # is the case number and S is the number of set bits in N. See sample input output for better understanding.

Sample Inputs:	Sample Outputs:
3 4 15 12	Case 1: 1 Case 2: 4 Case 3: 2

Problem Setter:

Md. Gulzar Hussain

D. Write Vertically

Time limit: 2 sec, Memory Limit: 3 MB

There are two ways to write Japanese sentences, horizontally or vertically.

Vertical writing is the traditional Japanese writing, and it is used for Japanese language textbooks for Japanese schools in Japan, literature, newspaper, and official governmental documents. When you write Japanese vertically, it starts from top right of the paper, and books with the vertical writing open from left to right.

Your task is to make the horizontal sentences to vertical, that is, rotate all sentences 90 degree clockwise.

Input

Your input will be at most 100 English sentences, every sentences may have maximum 100 characters.

Output

Your output will be the last sentence vertically at first and sequentially first sentence will be vertically at last. See sample input output for more clarification.

Sample Inputs:	Sample Outputs:
Bangladesh. Green Be Strong.	BGB era en Seg tnl r a o d n e g s . h .

Problem Setter:

Md. Gulzar Hussain

E. ExString

Time Limit: 2 sec, Memory Limit: 3 MB

In case of two strings s_1 and s_2 , if s_1 and s_2 has different length, then smaller string should add some extra symbol to make the length equal. The resulted smaller string is said to be ExString. You have to find out ExString for two strings.

Extra Symbol adding rule:

1. If character position is even, then put # as extra symbol.
2. Else, put * as extra symbol.

For example, GUB### is the ExString for CSEGUB and GUB.

Input

Input will consist of several test cases. Each input consists of an integer T, denoting test cases. Each test case consists of two strings.

Output

Output consists of a single line with of Case #, where # is the test case number and a print the appropriate ExString.

Input	Output
2 csegub gub 123456789 123	Case 1: gub### Case 2: 123#####

Problem Setter:

Misbah Ul Hoque
Senior Lecturer
Department of Computer Science and Engineering
Green University of Bangladesh