What is a High School Programming Contest

High School Programming Contest

A high school programming contest consists of teams of high school students attempting to solve the most problems in four hours.

- Team Size: A team consists of a maximum of four students.
- Teams Per High School: A high school may enter multiple teams.
- Contest Computers: Team members will be connected to the Zoom meeting using a computer. A maximum of two of the team members may have a computer with a programming environment of their choice. The team members with a programming environment use their computers to write programs that solve problems. The other team members can use paper and pencil to work on the problems. Teams decide how to allocate work.
- Programming Environment: A programming environment consists of an editor, a compiler tool chain, the ability to create programs, the ability to create test cases, and the ability to run programs and test them. For example, Netbeans is a Java programming environment and Idle is a Python programming environment.
- Solving Problems: At the start of the contest, each team is given a link to the contest judging system and a PDF containing the problems. The contest judging system also contains the problem descriptions. Teams solve problems using their brains and paper resources brought to the contest. For example, a Java textbook and notes of their favorite algorithms. Teams **CANNOT** search the Internet when solving problems.
    - Each team studies the problems in the packet, designs solutions, creates programs that solve the problem, creates test cases, and executes test cases.
    - Teams initially create programs on their computer, using their programming environment.
    - For each problem, the problem packet provides one or two test cases and expected outputs.
    - Teams use the packet test cases and expected outputs to test their programs.
    - Teams must augment the provided test cases to convince themselves their program successfully solves the problem.
    - After the team thinks their program solves the problem, they submit the program into a contest judging system.
    - For problem submissions that do not successfully solve a problem, teams re-work their solutions and re-submit.
- Contest Judging System: The contest judging system does the following.
    - Determines if the submission successfully solves the problem.
    - Tells the submitting team the success or failure of their submission.
    - Maintains a scoreboard of teams submission, indicating rankings from first, second, and so on.

- Contest Winners: At the end of 3.5 hours, the team that has solved the most problems wins the contest. The contest judging system orders the teams from first, second, and so on. When multiple teams solve the same number of problems, the teams are ordered based upon the earliest submission of the last correct submission. Time penalties for incorrect submissions factor into the ordering. We will award prizes to several teams.
- Contest Problems: Problems in the packet range from easy to difficult. The goal is for all teams to solve some problems, but in the end, a team will finish first, second, and so on.

Programming Contest Problems

Teams write programs to solve the problems and create test cases to determine if they have correctly solved the problems. The problems are solved by reading text from standard input, computing the solution, and writing text to standard output. The contest judging system has a collection of test cases with expected outputs. When a team submits their solution, the contest judging system runs the submission against test cases and compares actual output to expected output. The following are sample problems. The contest will have problems similar. We will have easy problems, medium difficult problems, and some more difficult.

Problem 1 - Pinocchio's Nose

Problem Description

After being magically given life, the puppet Pinocchio is able to walk and talk. If he is able to prove himself brave and honest, he will be turned into a real boy. While he is still a puppet, every lie he tells makes his nose grows one half inch. When getting in trouble, Pinocchio tells several lies making his nose grow quite long.

You should write a program that determines how long Pinocchio’s nose is after telling a given number of lies. Pinocchio’s nose starts as one half inch long, and grows another half inch for each lie he tells.

Input

Input consists of a single non-negative integer *L*, giving the number of lies that Pinocchio has told.

Output

Output is one line printing the length of Pinocchio’s nose.

If the length is a whole number, then do not print the number of half inches. Also, be sure to get the pluralization correct - if Pinocchio’s nose is one inch, or one half inch, then do not pluralize the word inch.

Sample Input 1

```
5
```

Sample Output 1

```
Pinocchio’s nose is 3 inches.
```

Sample Input 2

```
12
```

Sample Output 2

```
Pinocchio’s nose is 6 and 1 half inches.
```

Problem 2 - Donald and Daffy

Donald Duck plans to visit his cousin Daffy Duck. Donald studies a map and discovers various routes from his home to Daffy’s. For example, one route is 5 miles from Donald’s to Rapunzel’s, 6 miles from Rapunzel's house to Snow White’s house, and 3 miles from Snow White’s to Daffy’s for a total of 14 miles. Donald wants help adding the distances on his route.

Input

Input consists of a sequence of lines. The first line is the number of lines with distances. Subsequent lines have N integers separated by a space. The first integer on each line is the number of distances on that line.

Output

Output is a sequence of lines, where each line contains the sum of the last N-1 integers on the input line.

Sample Input

```
3
4 10 10 10 10
2 5 8
3 50 8 67
```

Sample Output

```
40
13
```

```
125
```

Problem 3 - Minnie's Letter Counter

Problem Description

Minnie Mouse is teaching Daisy Duck to read letters. Minnie has Daisy play a game where Daisy counts the number of times a specific letter appears on a line. For example, the letter a appears three times on the following line.

```
available
```

Minnie's cataracts occasionally cause her to miscount the letters. Minnie wants you to write a program that counts the number of times a specific letter occurs on a line.

Input

The input is defined as follows.

- The first line is the letter that is to be counted.
- The second line is the number of subsequent lines of input.
- There will be an additional X lines of input where X is the number on line 2.

The number of subsequent lines of input defined by input line 2 will be greater than or equal to 1. You shall ignore case when counting.

Output

The output will be one line for each of the X input lines where each output line contains the count of characters of the corresponding input line.

Sample Input - Example 1

```
a
3
hat
this
available
```

Sample Output - Example 1

```
1
0
3
```

Sample Input - Example 2

```
s
2
MissiSSippi Rivers
This is my time to Shine.
```

Sample Output - Example 2

```
5
3
```

Problem 5 - Closest Lion

Problem Description

Mufasa has just purchased cell phones for Simba and Nala. Conveniently, the cell phones are small and fit nicely in an unobtrusive collar made of hemp. The cell phones periodically transmit the locations Simba and Nala. Mufasa has programmed the cell phones to transmit position as a pair of coordinates - x and y. Sometimes Simba and Nala walk together and sometimes they walk separately. Mufasa wants a program that tells him who is closer - Simba or Nala.

- Simba and Nala are in a grid that can be viewed as a plane.
- The x-coordinates have a range of -20 to 20.
- The y-coordinates have a range of -20 to 20.
- Mufasa are located at position (0,0).
- Simba and Nala are somewhere in the grid.
- If Simba and Nala are the same distance from Mufasa, your answer shall be Nala is closer

Input

The input is defined as follows.

- One line with four doubles with a space between each value.
- The first pair of numbers are the x and y coordinates of Nala.
- The second pair of numbers are the x and y coordinates of Simba.

Output

The output will be one line that is the lion that is closest to Mufasa. One of the following lines must be output.

```
Nala is closer
Simba is closer
```

Sample Input - Example 1

```
3 -2 -5 -3
```

Sample Output - Example 1

```
Nala is closer
```

Sample Input - Example 2

```
3 3 -2 2
```

Sample Output - Example 2

```
Simba is closer
```

Problem 6 - Ariel's Palindromes

Problem Description

Ariel was awake early one morning. She knocked on Sebastian's door hoping Sebastian would join her for some adventures. Sebastian was half asleep when he mumbled, "A nut for a jar of tuna." Ariel wrote this silly statement in her journal. Later that night when studying her journal, Ariel realized Sebastian's silly statement was the same when read in either direction. Ariel searched the Internet and discovered the concept of a palindrome. Write a program for Ariel that checks for palindromes and reverses strings.

Input

The input is defined as follows.

- The first line is the number of subsequent lines of input. This number will be greater than or equal to 1.
- The subsequent lines contain text – up to 50 characters.

Output

The output will be one line for each of the input lines with text. The output line will be Palindrome if the input line is a palindrome; otherwise, the output line will be a reverse of the input line. When checking for palindrome, you will ignore case and you will skip punctuation and spaces.

Sample Input - Example 1

```
3
Hello there!
Madam I'm Adam
Rise to vote sir.
```

Sample Output - Example 1

```
!ereht olleH
Palindrome
Palindrome
```

Sample Input - Example 2

```
2
A nut for a jar of tuna.
This is my time to Shine.
```

Sample Output - Example 2

```
Palindrome
.enihS ot emit ym si sihT
```

Problem 7 - Beast's Birthday Shopping

Problem Description

The Beast wants to buy Belle two birthday presents. The Beast is well loved throughout the village and has great credit scores so all of the local stores have given him a line of credit. The Beast decides he wants to purchase a pair of presents from each store where the total cost consumes his entire line of credit. Write a program that helps the Beast with his shopping. The Beast receives a credit C at a local store and where he can buy two items. The Beast first walks through the store and creates a list L of all available items. From this list he buys two items that add up to the entire value of the credit. Your program's output consists of the two integers indicating the positions of the items in your list (smaller number first).

Input

The first line of input gives the number of cases, N. N test cases follow. For each test case there is:

- One line containing the value C, the amount of credit you have at the store.

- One line containing the value I, the number of items in the store.
- One line containing a space separated list of I integers. Each integer P indicates the price of an item in the store.
- Each test case will have exactly one solution.

Output

For each test case, output one line containing "Case #x: " followed by the indices of the two items whose price adds up to the store credit. For this problem, the index number is the position of the item – e.g., the second item is index 2. The lower index must be output first.
Limits

```
N ≤ 10, 5 ≤ C ≤ 1000, 1 ≤ P ≤ 1000, 3 ≤ I ≤ 100
```

Sample Input

```
3
100
3
5 75 25
200
7
150 24 79 50 88 345 3
8
8
2 1 9 4 4 56 90 3
```

Sample Output - Example 1

```
Case #1 2 3
Case #2 1 4
Case #3 4 5
```