acmASCIS

Self-Review Problems

Arrays

acmASCIS Student Chapter
Spreading Knowledge
acm

Lumberjack Sequencing

Another tale of lumberjacks?. Let see.

The lumberjacks are rude, bearded workers, while foremen tend to be bossy and simpleminded. The foremen like to harass the lumberjacks by making them line up in groups of ten, ordered by the length of their beards. The lumberjacks, being of different physical heights, vary their arrangements to confuse the foremen. Therefore, the foremen must actually measure the beards in centimeters to see if everyone is lined up in order.

Your task is to write a program to assist the foremen in determining whether or not the lumberjacks are lined up properly, either from shortest to longest beard or from longest to shortest.

Input

The input starts with line containing a single integer **N**, **0** < **N** < **20**, which is the number of groups to process. Following this are **N** lines, each containing ten distinct positive integers less than **100**.

Output

There is a title line, then one line per set of beard lengths. See the sample output for capitalization and punctuation.

Sample Input:

```
3
13 25 39 40 55 62 68 77 88 95
88 62 77 20 40 10 99 56 45 36
91 78 61 59 54 49 43 33 26 18
```

Sample Output:

```
Lumberjacks:
Ordered
Unordered
Ordered
```

WERTYU

A common typing error is to place the hands on the keyboard one row to the right of the correct position. So "**Q**" is typed as "**W**" and "**J**" is typed as "**K**" and so on. You are to decode a message typed in this manner.

Input

Input consists of several lines of text. Each line may contain digits, spaces, upper case letters (except **Q**, **A**, **Z**), or punctuation shown above [except back-quote (`)]. Keys labelled with words [*Tab, BackSp, Control,* etc.] are not represented in the input.

Output

You are to replace each letter or punction symbol by the one immediately to its left on the QWERTY keyboard shown above. Spaces in the input should be echoed in the output.

Sample Input:

```
O S, GOMR YPFSU/
```

Sample Output:

```
I AM FINE TODAY.
```

Alaska

The Alaska Highway runs **1422** miles from Dawson Creek, British Columbia to Delta Junction, Alaska. Brenda would like to be the first person to drive her new electric car the length of the highway. Her car can travel up to **200** miles once charged at a special charging station. There is a charging station in Dawson Creek, where she begins her journey, and also several charging stations along the way. Can Brenda drive her car from Dawson City to Delta Juntion and back?

Input

The input contains several scenario. Each scenario begins with a line containing **n**, a positive number indicating the number of charging stations. **n** lines follow, each giving the location of a filling station on the highway, including the one in Dawson City. The location is an integer between **0** and **1422**, inclusive, indicating the distance in miles from Dawson Creek. No two filling stations are at the same location. A line containing **0** follows the last scenario.

Output

For each scenario, output a line containing **POSSIBLE** if Brenda can make the trip. Otherwise, output a line containing the word **IMPOSSIBLE.**

Sample Input:

```
2
0
900
8
1400
1200
1000
800
600
400
200
0
0
```

Sample Output:

```
IMPOSSIBLE
POSSIBLE
```

Exact Sum

Peter received money from his parents this week and wants to spend it all buying books. But he does not read a book so fast, because he likes to enjoy every single word while he is reading. In this way, it takes him a week to finish a book.

As Peter receives money every two weeks, he decided to buy two books, then he can read them until receive more money. As he wishes to spend all the money, he should choose two books whose prices summed up are equal to the money that he has. It is a little bit difficult to find these books, so Peter asks your help to find them.

Input

Each test case starts with **2** ≤ **N** ≤ **10000**, the number of available books. Next line will have **N** integers, representing the price of each book, a book costs less than **1000001**. Then there is another line with an integer **M**, representing how much money Peter has. There is a blank line after each test case. The input is terminated by end of file (EOF).

Output

For each test case you must print the message: **Peter should buy books whose prices are i and j.**, where **i** and **j** are the prices of the books whose sum is equal to **M** and **i** ≤ **j**. You can consider that is always possible to find a solution, if there are multiple solutions print the solution that minimizes the difference between the prices **i** and **j**. After each test case you must print a blank line.

Sample Input:

```
2
40  40
80

5
10  2  6  8  4
10
```

Sample Output:

Peter should buy books whose prices are 40 and 40.

Peter should buy books whose prices are 4 and 6.

B2-Sequence

A B2-Sequence is a sequence of positive integers $\mathbf{1 \le b_1 < b_2 < b_3 \ldots}$ such that all pairwise sums $\mathbf{b_i + b_j}$, where $\mathbf{i \le j}$,, are different.

Your task is to determine if a given sequence is a B2-Sequence or not.

Input

Each test case starts with $\mathbf{2 \le N \le 100}$, the number of elements in a sequence. Next line will have **N** integers, representing the value of each element in the sequence. Each element $\mathbf{b_i}$ is an integer such that $\mathbf{b_i \le 10000}$. There is a blank line after each test case. The input is terminated by end of file (EOF).

Output

For each test case you must print the number of the test case, starting from 1, and a message indicating if the corresponding sequence it is a B2-Sequence or not. See the sample output below. After each test case you must print a blank line.

Sample Input:

```
4
1  2  4  8

4
3  7  10  14
```

Sample Output:

```
Case #1: It is a B2-Sequence.

Case #2: It is not a B2-Sequence.
```

Challenging Problem

Age Sort

You are given the ages (in years) of all people of a country with at least 1 year of age. You know that no individual in that country lives for 100 or more years. Now, you are given a very simple task of sorting all the ages in ascending order.

Input

There are multiple test cases in the input file. Each case starts with an integer **n** (**0<n<=2000000**), the total number of people. In the next line, there are **n** integers indicating the ages. Input is terminated with a case where **n** = **0**. This case should not be processed.

Output

For each case, print a line with **n** space separated integers. These integers are the ages of that country sorted in ascending order.

Sample Input:

```
5
3 4 2 1 5
5
2 3 2 3 1
0
```

Sample Output:

```
1 2 3 4 5
1 2 2 3 3
```