max planck institut
informatik

Competitive Programming

Saarland University — Summer Semester 2020

Julian Baldus, Markus Bläser, Karl Bringmann, Marian Dietz, Simon Schwarz,
Christoph Weidenbach, Dominic Zimmer

Assignments Week 1

Deadline: **May 12, 2020 at 16:00 sharp**

Please submit solutions to the problems in our judge system, available at
`https://compro.mpi-inf.mpg.de/`.
You can find your credentials on your personal status page in our CMS.

Problem	`helloworld`	`lineup`	`outoforder`	`theanswer`
Points	3	3	3	3
Difficulty				

Please note:

- Your solution will be judged immediately after submitting. This may take some time, depending on the current server load.
- You can submit as many times as you want. However, don't abuse the server or try to extract the secret test cases.
- If your solution is **accepted**, you will receive the points specified in the table above.
- If you get **another verdict**, you will receive 0 points.

Hello World!

Problem ID: helloworld

This problem is intended to familiarize you with the contest system. Solve it before attempting any other problem.
Wandering around and greeting people is hard in quarantine times. Therefore, please write a program to say `Hello` to everybody.
You will read their names from stdin, and write the output to stdout. Note that the output is case-sensitive, i.e. do not say `hello`. Before submitting, you should check that your output for the sample input exactly matches the sample output.

Input

The first line of the input contains an integer t. The following t lines each contain a name s.

Output

For each test case, print a line containing "Hello s!". Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- The names s consist of 1 to 100 lower or upper case english letters.

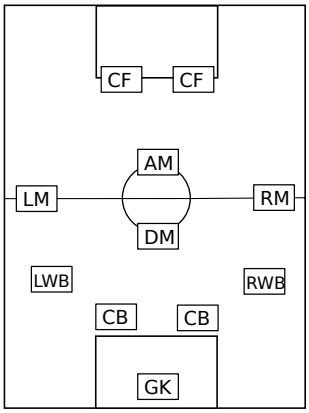
Sample Input 1	Sample Output 1
2 Dieter Lea	Hello Dieter! Hello Lea!

Lineup

Problem ID: lineup

After first discussions of a resumption of the Bundesliga, you, as the first coach need to decide on a lineup for your first game. Unfortunately, some of your players have been infected, and therefore you can not use the usual lineup you used before. Luckily for you, exactly 11 players of your team seem not infected. Now you want to form the strongest team with the players that are left.

You have already decided on the tactical formation you wish to use, so now you need to select the players who should fill each of the 11 positions in the team. Since there are only 11 players left, it should be clear who is playing, but this still leaves the question of where to put which player.



Most players have a favoured position on the field where they are strongest, but some players are proficient in different positions. Your assistant has rated the playing strength of each of your 11 players in each of the 11 available positions in your formation, where a score of 100 means that this is an ideal position for the player and a score of 0 means that the player is not suitable for that position at all. Find the lineup which maximises the sum of the playing strengths of your players for the positions you assigned them. Each position must be occupied by exactly one player.

Input

The input consists of 11 lines, one for each player, where the i-th line contains 11 integer numbers s_{ij} between 0 and 100. s_{ij} describes the i-th player's strength on the j-th position.

Output

Print x, the maximum of the sum of player strengths over all possible lineups.

Constraints

- $0 \leq s_{ij} \leq 100$ for all $1 \leq i, j \leq 11$.

Sample Input 1	Sample Output 1
100 0 0 0 0 0 0 0 0 0 0 0 80 70 70 60 0 0 0 0 0 0 0 40 90 90 40 0 0 0 0 0 0 0 40 85 85 33 0 0 0 0 0 0 0 70 60 60 85 0 0 0 0 0 0 0 0 0 0 0 95 70 60 60 0 0 0 45 0 0 0 80 90 50 70 0 0 0 0 0 0 0 40 90 90 40 70 0 0 0 0 0 0 0 50 70 85 50 0 0 0 0 0 0 0 66 60 0 80 80 0 0 0 0 0 0 50 50 0 90 88	970

Out of order

Problem ID: outoforder

Dieter likes to collect stamps. He has recently sorted his collection by increasing value. To keep the stamps properly ordered, he decided to glue them into his collection book. However, after he was done, he noticed that some of his stamps were out of order. It is possible but very difficult to remove stamps from the book and glue them into a different position, so Dieter only wants to do this if he can sort the collection with very few operations. Can you tell him if it is possible to sort the collection by swapping only two stamps with each other?

Input

The first line of the input contains an integer n. The second line contains n integers x_i, $1 \leq i \leq n$, describing the values of a stamp in Dieter's stamp collection. The collection is not sorted in increasing order, i.e. there is at least one position i with $x_i > x_{i+1}$.

Output

If there are two stamps such that swapping them yields a sorted collection, print their two positions in any order. Otherwise, print `impossible`.

Constraints

- $1 \leq n \leq 100000$
- $1 \leq x_i \leq 1000$ for all $1 \leq i \leq n$

Sample Input 1

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

Sample Output 1

```
3 5
```

Sample Input 2

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

Sample Output 2

```
impossible
```

The Answer to everything

Problem ID: theanswer

"The Answer to the Ultimate Question of Life, the Universe, and Everything is 42."

After heavy computational effort about the question to that answer, Deep Thought 2.0 finally came up with a question: If you have n integer numbers, are there any three of them that sum up to 42?

Input

The first line of input contains an integer n such that $1 \leq n \leq 5000$.

The next line contains n integers between -10^{15} and 10^{15}, separated by spaces. The integers are pairwise distinct.

Output

If, among the n given numbers, there are three that sum to 42, print them in any order. The three numbers must be distinct. If there are multiple solutions, print any of them. If there are no such three integers, print `impossible`.

Sample Input 1	Sample Output 1
4 21 15 10 11	21 10 11

Sample Input 2	Sample Output 2
5 1 2 3 4 5	impossible

Sample Input 3	Sample Output 3
5 47 -1 -2 -3 -4	47 -1 -4