The International Collegiate Programing Contest

Sponsored by FU-ICPC Community

Level 2 – Qualification contest

FCAI, Egypt

February 2024

A. Swap Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob are playing a game on an array a of n positive integers. Alice and Bob make alternating moves with Alice going first.

In his/her turn, the player makes the following move:

- If $a_1 = 0$, the player loses the game, otherwise:
- Player chooses some i with $2 \le i \le n$. Then player decreases the value of a_1 by 1 and swaps a_1 with a_i.

Determine the winner of the game if both players play optimally.

Input
The input consists of multiple test cases. The first line contains a single integer t $(1 \le t \le 2 \cdot 10^4)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n $(2 \le n \le 10^5)$ — the length of the array a.

The second line of each test case contains n integers $a_1, a_2 \ldots a_n$ $(1 \le a_i \le 10^9)$ — the elements of the array a.

It is guaranteed that sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output
For each test case, if Alice will win the game, output "`Alice`". Otherwise, output "`Bob`".

You can output each letter in any case. For example, "`alIcE`", "`Alice`", "`alice`" will all be considered identical.

Example

input	Copy
3 2 1 1 2 2 1 3 5 4 4	

output	Copy
Bob Alice Alice	

Note

In the **first testcase**, in her turn, Alice can only choose $i = 2$, making the array equal $[1, 0]$. Then Bob, in his turn, will also choose $i = 2$ and make the array equal $[0, 0]$. As $a_1 = 0$, Alice loses.

In the **second testcase**, once again, players can only choose $i = 2$. Then the array will change as follows: $[2, 1] \to [1, 1] \to [1, 0] \to [0, 0]$, and Bob loses.

In the **third testcase**, we can show that Alice has a winning strategy.

B. The answer is easy it's....

time limit per test: 1 s.
memory limit per test: 256 MB
input: standard input
output: standard output

This problem is simple and doesn't need much thinking to solve. Just follow the clear instructions and use basic logic to find the answer quickly. It's straightforward and doesn't require any complex analysis or calculations.

You were given an integer n you are required to find any integer X that is coprime with n

Input
1 line containing integer n $(1 \leq n \leq 10^9)$

Output
1 line containing integer X $(1 \leq X \leq 10^9)$

if there are multiple answers output any of them

Example

input	Copy
3	
output	**Copy**
7	

Note
two numbers are said to be **coprime** if their greatest common divisor is 1

gcd(3, 7) = 1

C. Special Offer! Super Price 999 Bourles!

Polycarpus is an amateur businessman. Recently he was surprised to find out that the market for paper scissors is completely free! Without further ado, Polycarpus decided to start producing and selling such scissors.

Polycaprus calculated that the optimal celling price for such scissors would be p bourles. However, he read somewhere that customers are attracted by prices that say something like "Special Offer! Super price 999 bourles!". So Polycarpus decided to lower the price a little if it leads to the desired effect.

Polycarpus agrees to lower the price by no more than d bourles so that the number of nines at the end of the resulting price is maximum. If there are several ways to do it, he chooses the maximum possible price.

Note, Polycarpus counts only the *trailing* nines in a price.

Input
The first line contains two integers p and d ($1 \le p \le 10^{18}$; $0 \le d < p$) — the initial price of scissors and the maximum possible price reduction.

Output
Print the required price — the maximum price that ends with the largest number of nines and that is less than p by no more than d.

The required number shouldn't have leading zeroes.

Examples

input	Copy
1029 102	

output	Copy
999	

input	Copy
27191 17	

output	Copy
27189	

D. Geometry

time limit per test: 1 second
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Geometry is a very important field in mathematics, Squares and rectangles are essential shapes in geometry, both of them have 4 right angles, but a square is a special case of a rectangle where width and height are the same.

The figure below shows a square on the left and a rectangle on the right:

If you have the width and the height of a 4 right angled shape, can you figure out if it is a square or a rectangle?

Input
The first line contains T, the number of test cases, for each test case there is a line with two integers ($1 \leq w, h \leq 1,000,000$) representing width and height, respectively.

Output
For each test case, print one line consists of 'Square' if the shape is a square, otherwise print 'Rectangle' if it is a rectangle.

Examples

input	Copy
3 10 10 13 200 300 300	

output	Copy
Square Rectangle Square	

E. Fake News

time limit per test: 1 s.
memory limit per test: 256 MB
input: standard input
output: standard output

As it's the first of April, Heidi is suspecting that the news she reads today are fake, and she does not want to look silly in front of all the contestants. She knows that a newspiece is fake if it contains `heidi` as a subsequence. Help Heidi assess whether the given piece is true, but please be discreet about it...

Input

The first and only line of input contains a single nonempty string s of length at most 1000 composed of lowercase letters (`a`-`z`).

Output

Output `YES` if the string s contains `heidi` as a subsequence and `NO` otherwise.

Examples

input	Copy
abcheaibcdi	

output	Copy
YES	

input	Copy
hiedi	

output	Copy
NO	

Note

A string s contains another string p as a subsequence if it is possible to delete some characters from s and obtain p.

F. NIT Destroys the Universe

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

For a collection of integers S, define $\operatorname{mex}(S)$ as the smallest non-negative integer that does not appear in S.

NIT, the cleaver, decides to destroy the universe. He is not so powerful as Thanos, so he can only destroy the universe by snapping his fingers several times.

The universe can be represented as a 1-indexed array a of length n. When NIT snaps his fingers, he does the following operation on the array:

- He selects positive integers l and r such that $1 \leq l \leq r \leq n$. Let $w = \operatorname{mex}(\{a_l, a_{l+1} \ldots, a_r\})$. Then, for all $l \leq i \leq r$, set a_i to w.

We say the universe is destroyed if and only if for all $1 \leq i \leq n$, $a_i = 0$ holds.

Find the minimum number of times NIT needs to snap his fingers to destroy the universe. That is, find the minimum number of operations NIT needs to perform to make all elements in the array equal to 0.

Input
Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). Description of the test cases follows.

The first line of each test case contains one integer n ($1 \leq n \leq 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output
For each test case, print one integer — the answer to the problem.

Example

input Copy

```
4
4
0 0 0 0
5
0 1 2 3 4
7
0 2 3 0 1 2 0
1
1000000000
```

output Copy

```
0
1
2
1
```

Note

In the first test case, we do 0 operations and all elements in the array are already equal to 0.

In the second test case, one optimal way is doing the operation with $l = 2$, $r = 5$.

In the third test case, one optimal way is doing the operation twice, respectively with $l = 4$, $r = 4$ and $l = 2$, $r = 6$.

In the fourth test case, one optimal way is doing the operation with $l = 1$, $r = 1$.

G. Divisor Difference

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a positive integer n. Over all positive integers x, y satisfying

$$xy = n$$

find the maximum possible value of $|x - y|$ (where $|a|$ denotes the absolute value - see notes for more clarification).

Input

The first line of input contains t ($1 \le t \le 2 \cdot 10^5$) the number of test cases. The description of each test case follows.

Each test case contains one line of one integer n ($1 \le n \le 10^9$), the integer described in the problem.

Output

For each test case output one line containing one integer, the maximum value of $|x - y|$ as described in the problem.

Example

input	Copy
2 4 5	

output	Copy
3 4	

Note

The absolute value of an integer a - denoted $|a|$ - is either a if a is non-negative, or $-a$ if a is negative.

H. SwapSort

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem your goal is to sort an array consisting of n integers in at most n swaps. For the given array find the sequence of swaps that makes the array sorted in the non-descending order. Swaps are performed consecutively, one after another.

Note that in this problem you do not have to minimize the number of swaps — your task is to find any sequence that is no longer than n.

Input
The first line of the input contains integer n ($1 \le n \le 3000$) — the number of array elements. The second line contains elements of array: $a_0, a_1, ..., a_{n-1}$ ($-10^9 \le a_i \le 10^9$), where a_i is the i-th element of the array. The elements are numerated from 0 to $n-1$ from left to right. Some integers may appear in the array more than once.

Output
In the first line print k ($0 \le k \le n$) — the number of swaps. Next k lines must contain the descriptions of the k swaps, one per line. Each swap should be printed as a pair of integers i, j ($0 \le i, j \le n-1$), representing the swap of elements a_i and a_j. You can print indices in the pairs in any order. The swaps are performed in the order they appear in the output, from the first to the last. It is allowed to print $i = j$ and swap the same pair of elements multiple times.

If there are multiple answers, print any of them. It is guaranteed that at least one answer exists.

Examples

input	Copy
5 5 2 5 1 4	

output	Copy
2 0 3 4 2	

I. Sort the Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Being a programmer, you like arrays a lot. For your birthday, your friends have given you an array a consisting of n **distinct** integers.

Unfortunately, the size of a is too small. You want a bigger array! Your friends agree to give you a bigger array, but only if you are able to answer the following question correctly: is it possible to sort the array a (in increasing order) by reversing **exactly one** segment of a? See definitions of segment and reversing in the notes.

Input
The first line of the input contains an integer n ($1 \le n \le 10^5$) — the size of array a.

The second line contains n distinct space-separated integers: $a[1], a[2], ..., a[n]$ ($1 \le a[i] \le 10^9$).

Output
Print "`yes`" or "`no`" (without quotes), depending on the answer.

If your answer is "`yes`", then also print two space-separated integers denoting start and end (start must not be greater than end) indices of the segment to be reversed. If there are multiple ways of selecting these indices, print any of them.

Examples

input	Copy
3 3 2 1	
output	Copy
yes 1 3	

input Copy

```
4
2 1 3 4
```

output Copy

```
yes
1 2
```

input Copy

```
4
3 1 2 4
```

output Copy

```
no
```

input Copy

```
2
1 2
```

output Copy

```
yes
1 1
```

Note

Sample 1. You can reverse the entire array to get $[1, 2, 3]$, which is sorted.

Sample 3. No segment can be reversed such that the array will be sorted.

Definitions

A segment $[l, r]$ of array a is the sequence $a[l], a[l+1], ..., a[r]$.

If you have an array a of size n and you reverse its segment $[l, r]$, the array will become:

$a[1], a[2], ..., a[l-2], a[l-1], a[r], a[r-1], ..., a[l+1], a[l], a[r+1], a[r+2], ..., a[n-1], a[n]$.

J. Depression

time limit per test: 1 s.
memory limit per test: 256 MB
input: standard input
output: standard output

Do you remember a kind cartoon "Beauty and the Beast"? No, no, there was no firing from machine guns or radiation mutants time-travels!

There was a beauty named Belle. Once she had violated the Beast's order and visited the West Wing. After that she was banished from the castle...

Everybody was upset. The beautiful Belle was upset, so was the Beast, so was Lumiere the candlestick. But the worst thing was that Cogsworth was upset. Cogsworth is not a human, but is the mantel clock, which was often used as an alarm clock.

Due to Cogsworth's frustration all the inhabitants of the castle were in trouble: now they could not determine when it was time to drink morning tea, and when it was time for an evening stroll.

Fortunately, deep in the basement are lying digital clock showing the time in the format `HH:MM`. Now the residents of the castle face a difficult task. They should turn Cogsworth's hour and minute mustache hands in such a way, that Cogsworth began to show the correct time. Moreover they need to find turn angles in degrees for each mustache hands. The initial time showed by Cogsworth is `12:00`.

You can only rotate the hands forward, that is, as is shown in the picture:

As since there are many ways too select such angles because of full rotations, choose the smallest angles in the right (non-negative) direction.

Note that Cogsworth's hour and minute mustache hands move evenly and continuously. Hands are moving independently, so when turning one hand the other hand remains standing still.

Input

The only line of input contains current time according to the digital clock, formatted as `HH:MM` ($00 \le$ `HH` ≤ 23 , $00 \le$ `MM` ≤ 59). The mantel clock initially shows `12:00`.

Pretests contain times of the beginning of some morning TV programs of the Channel One Russia.

Output

Print two numbers x and y — the angles of turning the hour and minute hands, respectively ($0 \le x, y < 360$). The absolute or relative error in the answer should not exceed 10^{-9}.

Examples

input	Copy
12:00	
output	**Copy**
0 0	

input	Copy
04:30	
output	**Copy**
135 180	

input	Copy
08:17	
output	**Copy**
248.5 102	

Note

A note to the second example: the hour hand will be positioned exactly in the middle, between 4 and 5.