CSE221 Assignment 01 Summer 2025

A. Odd or Even?

time limit per test: 1 second
memory limit per test: 256 megabytes

Do you know how to tell if a number is **Odd** or **Even**? You are given T numbers, and for each of those numbers, you have to tell whether the number is odd or even.

Input

The first line will contain a single integer T $(1 \le T \le 100)$. Each of the next T lines will contain a number N $(-10^5 \le N \le 10^5)$.

Output

For each N, you have to print whether the number is odd or even. Please see the sample input-output format to know what exactly you have to print.

Example

input

```
5
10
19
7
3
100
```

output

```
10 is an Even number.
19 is an Odd number.
7 is an Odd number.
3 is an Odd number.
100 is an Even number.
```

B. Can you solve Arithmetic Expressions?

time limit per test: 1 second
memory limit per test: 256 megabytes

Can you solve arithmetic expressions with your programming knowledge? Let's find it out. You will be given some arithmetic expressions, and you have to solve them.

Input

The first line will contain a number $T(1 \le T \le 1000)$ representing the number of test cases. Then for each test case, you will be given an arithmetic expression. Please see the sample input below. It is guaranteed that the numbers inside the arithmetic expression will be between 1 and 1000.

Output

For each test case, you have to print the result. Look at the sample output for reference.

Important Note: Your answer might contain floating point numbers, and in that case, your answer doesn't have to be exactly equal to the actual answer. For example, if your answer is 20.250000001 and the judge's solution is 20.25, your answer will still be considered correct. As long as it is really close to the correct solution, your solution will be considered correct. Formally speaking, if your solution is x, and the judge's solution is y, then as long as $|x - y| \le 10^{-6}$, your solution will be correct. In the above example, your solution was 20.250000001 and the judge's solution was 20.25. If you take the difference of these two numbers, they are smaller than 10^{-6}. Similarly, if the judge's solution is 19.0000000000 and your solution is 19, it is still correct, as the difference is 0, which is less than 10^{-6}.

Example

input

```
15
calculate 67 + 41
calculate 85 / 5
calculate 13 - 56
calculate 99 - 95
calculate 3 / 10
calculate 12 * 19
calculate 14 - 6
calculate 3 * 88
calculate 45 * 68
calculate 81 - 0
calculate 77 + 40
calculate 8 * 84
calculate 73 - 22
calculate 85 - 86
calculate 28 * 58
```

output

```
108.000000
17.000000
-43.000000
4.000000
0.300000
228.000000
8.000000
264.000000
3060.000000
81.000000
117.000000
672.000000
51.000000
-1.000000
1624.000000
```

C. Fast Sum

time limit per test: 1 second
memory limit per test: 256 megabytes

Your friend is trying to solve the following problem. You are given $\mathbf{T}$ test cases. For each test case, you are given an integer $\mathbf{N}$. You have to find out the summation of 1 to N. More formally, your friend has to calcuate

$$\sum_{x=1}^{x=N} x$$

Your friend wrote the following code in Python to solve it:

```
T = int(input())

for _ in range(T):
    N = int(input())
    sum = 0
    for i in range(1, N + 1):
        sum += i
    print(sum)
```

Same code in Java:

```
import java.util.Scanner;

public class Solution {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        for (int t = 0; t < T; t++) {
            int N = sc.nextInt();
            int sum = 0;
            for (int i = 1; i <= N; i++)
                sum += i;
            System.out.println(sum);
        }
    }
}
```

However, the code is not passing the online judge due to some unknown errors for large values of N.

Since you are currently studying CSE221 and have learned about time complexity, help your friend come up with a more efficient solution.

Input

The first line contains a single integer T $(1 \le T \le 10^4)$ — the number of test cases.

The next T lines each contain a single integer N $(1 \le N \le 10^6)$.

Output

For each test case, print a single integer — the summation from 1 to N.

Example

input

```
5
2
5
10
12
100
```

output

```
3
15
55
78
5050
```

D. Is Sorted?

time limit per test: 1 second
memory limit per test: 256 megabytes

You are given an array of N integers. Determine whether the given array is in non-decreasing order.

An array is said to be in non-decreasing order if, for every valid index i such that $1 \le i < N$ (1-based indexing), the condition $A[i] \le A[i+1]$ holds true.

Example: $[1, 2, 4, 5]$, $[1, 2, 2, 4, 4, 5]$ are in the non-decreasing order because every element is less than or equal to the one after it.

Input

The first line contains a single integer T $(1 \le T \le 100)$ — the number of test cases. Each testcase contains two lines.

In each test case, the first line contains a single integer N $(1 \le N \le 10^4)$ — the number of elements in the array. The second line contains N integers separated by spaces $a_1, a_2, a_3 \dots a_n$ $(1 \le a_i \le 10^6)$ — the elements of the array.

Output

For each test case, print YES if the array is in non-decreasing order. Otherwise, print NO.

Example

input

```
3
4
1 2 3 3
4
1 5 2 6
1
5
```

output

```
YES
NO
YES
```

E. Reverse Sorting

time limit per test: 1 second
memory limit per test: 256 megabytes

You are given an array of $\mathbf{N}$ integers. Your task is to sort the array in non-decreasing order using only a specific type of operation:

- In one operation, you may select any subarray of length exactly 3 and reverse it.

You can apply this operation as many times as you like (or not at all). Your goal is to determine whether it is possible to sort the array using only this operation.

Input

The first line contains a single integer N $(1 \le N \le 1000)$ — the number of elements in the array.

The second line contains N integers separated by spaces $a_1, a_2, a_3 \dots a_n$ $(1 \le a_i \le 10^6)$ — the elements of the array.

Output

Print YES if it is possible to sort the array using only the allowed operations. Otherwise, print NO.

Examples

input

```
4
2 3 1 1
```

output

```
YES
```

input

```
6
2 5 5 1 5 5
```

output

```
NO
```

input

```
1
2
```

output

```
YES
```

input

```
2
6 6
```

output

```
YES
```

Note

For the Sample Input 1, one way to sort the array $[2, 3, 1, 1]$ using only the allowed operation:

1. Reverse the subarray $[3, 1, 1]$ which becomes $[1, 1, 3]$. So the resulting array: $[2, 1, 1, 3]$
2. Reverse subarray $[2, 1, 1]$ which becomes $[1, 1, 2]$. So the resulting array: $[1, 1, 2, 3]$

F. An Ancient Sorting Algorithm

time limit per test: 1 second
memory limit per test: 256 megabytes

You are given an array of N integers. You have to sort the array in non-decreasing order using a custom sorting algorithm with the following constraint:

- You may only swap adjacent elements with the same parity (i.e., both even or both odd)

Sort the array in non-decreasing order until no more such swaps are possible and print the final array.

Input

The first line contains a single integer N $(1 \le N \le 1000)$ — the number of elements in the array.

The second line contains N integers separated by spaces $a_1, a_2, a_3 \dots a_n$ $(1 \le a_i \le 10^6)$ — the elements of the array.

Output

Print the final array after sorting the array in non-decreasing order until no more such swaps are possible.

Examples

input

```
7
4 2 4 7 1 6 1
```

output

```
2 4 4 1 7 6 1
```

input

```
5
3 5 9 7 1
```

output

```
1 3 5 7 9
```

input

```
14
4 8 2 9 1 5 4 6 8 1 7 13 11 8
```

output

```
2 4 8 1 5 9 4 6 8 1 7 11 13 8
```

input

```
1
221
```

output

```
221
```

G. Sorting Again??

time limit per test: 1 second
memory limit per test: 256 megabytes

Suppose you are given a task to rank the students. You have gotten the marks and ID of the students. Now your task is to rank the students based on their marks using a sorting algorithm. If two or more students get the same mark, then students with the lower ID will get prioritized. See the input and output for a better understanding.

However, **you have to keep in mind that your sorting algorithms perform the minimum number of swapping operations.**

Input

The first line of the input file will contain an integer $N(1 \le N \le 1000)$. The second line will contain N integers, representing the Student ID, $S_i(1 \le S_i \le 1000)$. The next line will contain the N integers, $S_m(1 \le S_m \le 1000)$, which denotes the obtained mark of the corresponding students.

Note: It is guaranteed that the student IDs are unique. In other words, $S_i \ne S_j$ if $i \ne j$.

Output

The first line of the output must contain a number X which denotes the number of minimum swaps. The rest of the N lines will contain the Student ID and obtained marks sorted based on the instruction above. See the sample output for a better understanding.

Important Note: Since you are asked to minimize the number of swaps, if your number of swaps doesn't match with the judge's answer, your solution will be considered incorrect.

Look at the first sample input. It can be shown that this can be sorted with only 4 swaps. It can also be shown that it is not possible to sort this in less than 4 swaps.

Examples

input

```
7
7 4 9 3 2 5 1
40 50 50 20 10 10 10
```

output

```
Minimum swaps: 4
ID: 4 Mark: 50
ID: 9 Mark: 50
ID: 7 Mark: 40
ID: 3 Mark: 20
ID: 1 Mark: 10
ID: 2 Mark: 10
ID: 5 Mark: 10
```

input

```
4
7 2 5 3
80 60 80 50
```

output

```
Minimum swaps: 2
ID: 5 Mark: 80
ID: 7 Mark: 80
ID: 2 Mark: 60
ID: 3 Mark: 50
```

H. Trains?

time limit per test: 3 seconds
memory limit per test: 256 megabytes

You have been recently recruited as the Software Engineer at Jumanji Railway Software System. You have a big task at hand. You will be given $N(1 \le N \le 100)$ schedule of the train. The next N line will contain the name of the train and the departure time. See the input format for better understanding.

Your task is to write a sorting algorithm that will group the trains in the lexicographical order based on the name of the trains. If two or more trains have the same name, then the train with the latest departure time will get prioritized. If there is still a tie, then the train which comes first in the input will come first.

Input

The first line will contain an integer $N(1 \le N \le 100)$. For the next N lines, i_{th} line will describe i_{th} train. Please see the sample input for better understanding.

Please note that the names of the trains and destinations don't contain any white spaces, and the length of the names and destinations will be at most 100. For example, look at the following description:

```
DhumketuExpress will departure for Chittagong at 02:30
```

Here, **DhumketiExpress** is the name of the train **Chittagong** is the destination, and they don't contain any whitespaces, and their length is less than 100.

Output

Print the train description in the sorted order (specified above). Please see the output format for better understanding.

Example

input

```
8
ABCD will departure for Mymensingh at 00:30
DhumketuExpress will departure for Chittagong at 02:30
ABC will departure for Dhaka at 17:30
ABCD will departure for Chittagong at 01:00
ABC will departure for Khulna at 03:00
ABC will departure for Barisal at 03:00
ABCE will departure for Sylhet at 23:05
PadmaExpress will departure for Dhaka at 19:30
```

output

```
ABC will departure for Dhaka at 17:30
ABC will departure for Khulna at 03:00
ABC will departure for Barisal at 03:00
ABCD will departure for Chittagong at 01:00
ABCD will departure for Mymensingh at 00:30
ABCE will departure for Sylhet at 23:05
DhumketuExpress will departure for Chittagong at 02:30
PadmaExpress will departure for Dhaka at 19:30
```