A. Odd or Even?

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 ($-105 \le N \le 105$).

Output

For each N, you have to print whether the number is odd or even.

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?

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. 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. Your answer might contain floating point numbers; it doesn’t have to be exactly equal to the actual answer.

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

You are given **T** test cases. For each test case, you are given an integer **N**. You have to calculate the sum from 1 to N.

Input
The first line contains a single integer T ($1 \le T \le 104$) — the number of test cases.
The next T lines each contain a single integer N ($1 \le N \le 106$).

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

Input
The first line contains a single integer T ($1 \le T \le 100$). Each test case has two lines:

- First line: integer N ($1 \le N \le 104$) — the number of elements.
- Second line: N integers $a1, a2, \ldots, an$ ($1 \le ai \le 106$).

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

Example
Input:

3

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

Output:

YES
NO
YES

E. Reverse Sorting

You are given an array of N integers. You may select any subarray of length exactly 3 and reverse it. Determine whether it is possible to sort the array using only this operation.

Input

The first line: integer N ($1 \le N \le 1000$)
Second line: N integers $a1, a2, \ldots, an$ ($1 \le ai \le 106$)

Output

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

Example

Input:

4
2 3 1 1

Output:

YES

F. An Ancient Sorting Algorithm

Sort an array in non-decreasing order by swapping only adjacent elements with the same parity.

Input
The first line: integer N ($1 \le N \le 1000$)
Second line: N integers $a1, a2, \ldots, an$ ($1 \le ai \le 106$)

Output
Print the final array after sorting as much as possible.

Example
Input:

7
4 2 4 7 1 6 1

Output:

2 4 4 1 7 6 1

G. Sorting Again??
Rank students by marks; in case of ties, lower ID gets priority. Minimize number of swaps.

Input
The first line: integer N ($1 \le N \le 1000$)
Second line: N integers, Student IDs
Third line: N integers, corresponding marks

Output
First line: Minimum swaps. Then lines of ID and mark sorted.

Example
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

H. Trains?

Sort train schedules lexicographically by train name. If names tie, later departure prioritized; if still tied, earlier input prioritized.

Input

First line: integer N ($1 \leq N \leq 100$)
Next N lines: each train schedule in the format:

```
TrainName will departure for Destination at HH:MM
```

Output

Print the sorted train schedules.

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
