Assignment

Pick **2 of the 3** tasks below that you want to solve. Send the solution to our inbox **within 2 hours** of receiving the assignment.
Send the solution - python files - one per task; if you create more files for a task then zip it.
Do your best in all aspect of coding, i.e. consider it as production code.

Task 1

Using Python 3 and Django - write an app/service providing following REST API for a branch ticketing system.
Required services:

- Generate a new serial number.
    - Also returning generation datetime and rank(position) in the queue
- Get actual/active waiting number.
- Delete the last active number.

For example:

There is a list of active tickets - current state:
- 1245, 2017-09-01 15:22, 0
- 1246, 2017-09-01 15:42, 1
- 1250, 2017-09-01 16:32, 2

Generate new serial number returns:
- 1251, 2017-09-01 19:20, 3

Get actual/active waiting number returns:
- 1245, 2017-09-01 15:22, 0

After deletion of the last active number the list looks as follows:
- 1246, 2017-09-01 15:42, 0
- 1250, 2017-09-01 16:32, 1
- 1251, 2017-09-01 19:20, 2

Task 2

There is a horizontal row of **n** cubes. The length of each cube is given. You need to create a new vertical pile of cubes. The new pile should follow these directions: if $cube^I$ is on top of $cube^J$ then $sideLength^J$ >= $sideLength^I$.
When stacking the cubes, you can only pick up either the leftmost or the rightmost cube each time.
Print "Yes" if it is possible to stack the cubes. Otherwise, print "No". Do not print the quotation marks.

Input Format
The first line contains a single integer **T**, the number of test cases.
For each test case, there are 2 lines.
The first line of each test case contains **n**, the number of cubes.
The second line contains **n** space separated integers, denoting the sideLengths of each cube in that order.

Constraints
1 T 5
$1 < n < 10^5$
$1 < sideLength < 2^{31}$

Output Format
For each test case, output a single line containing either "Yes" or "No" without the quotes.

Sample Input

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

Sample Output

```
  Yes
  No
```

Explanation
In the first test case, pick in this order: left - 4, right - 4, left - 3, right - 3, left - 2, right - 1.
In the second test case, no order gives an appropriate arrangement of vertical cubes. 3 will always come after either 1 or 2.

Use python3, read input from STDIN, write output to STDOUT

Task 3

You are given a string S.
S contains alphanumeric characters only.
Your task is to sort the string in the following manner:

- All sorted lowercase letters are ahead of uppercase letters.
- All sorted uppercase letters are ahead of digits.
- All sorted odd digits are ahead of sorted even digits.

Input Format
A single line of input contains the string.

Constraints
0 < len(S) < 1000

Output Format
Output the sorted string.

Sample Input
Sorting1234

Sample Output
ginortS1324

Note: !!!
a) Using *join*, *for* or *while* anywhere in your code, even as substrings, will result in a score of zero.
b) You can only use *sorted* function once in your code. A zero score will be awarded for using *sorted* more than once.

Use python3, read input from STDIN, write output to STDOUT