1) Strings Repetition

You just need to take a string and a integer as an input and repeat the string upto the count given as in integer.

Sample Test Case :
Hello
2

Output :
HelloHello

2) Given a string `s`, remove duplicate letters so that every letter appears once and only once. You must make sure your result is **the smallest in lexicographical order** among all possible results

Input: `s = "bcabc"`
Output: `"abc"`

Input: `s = "cbacdcbc"`
Output: `"acdb"`

3) A magical string `s` consists of only `'1'` and `'2'` and obeys the following rules:

- The string s is magical because concatenating the number of contiguous occurrences of characters `'1'` and `'2'` generates the string `s` itself.

The first few elements of `s` is `s = "122112122112112122112……"`. If we group the consecutive `1`'s and `2`'s in `s`, it will be `"1 22 11 2 1 22 1 22 11 2 11 22 ....."` and the occurrences of `1`'s or `2`'s in each group are `"1 2 2 1 1 2 1 2 2 1 2 2 ....."`. You can see that the occurrence sequence is `s` itself.

Given an integer `n`, return the number of `1`'s in the first `n` number in the magical string `s`.

Input: `n = 6`
Output: `3`
Explanation: `The first 6 elements of magical string s is "122112" and it contains three 1's, so return 3.`

Input: `n = 1`
Output: `1`

4) Given a list of non-negative integers `nums`, arrange them such that they form the largest number and return it. Since the result may be very large, so you need to return a string instead of an integer.

Input: `nums = [10,2]`
Output: `"210"`

Input: `nums = [3,30,34,5,9]`
Output: `"9534330"`

5) Given a string `s` and an integer `k`, return *the length of the longest substring of* `s` *such that the frequency of each character in this substring is greater than or equal to* `k`.

Input: `s = "aaabb", k = 3`
Output: `3`
Explanation: `The longest substring is "aaa", as 'a' is repeated 3 times.`

Input: `s = "ababbc", k = 2`
Output: `5`
Explanation: `The longest substring is "ababb", as 'a' is repeated 2 times and 'b' is repeated 3 times.`