Sorting related problems

Problem 01: **http://codeforces.com/problemset/problem/339/A**

Problem Description:

Xenia the beginner mathematician is a third year student at elementary school. She is now learning the addition operation.

The teacher has written down the sum of multiple numbers. Pupils should calculate the sum. To make the calculation easier, the sum only contains numbers 1, 2 and 3. Still, that isn't enough for Xenia. She is only beginning to count, so she can calculate a sum only if the summands follow in non-decreasing order. For example, she can't calculate sum 1+3+2+1 but she can calculate sums 1+1+2 and 3+3.

You've got the sum that was written on the board. Rearrange the summans and print the sum in such a way that Xenia can calculate the sum.

Input

The first line contains a non-empty string s — the sum Xenia needs to count. String s contains no spaces. It only contains digits and characters "+". Besides, string s is a correct sum of numbers 1, 2 and 3. String s is at most 100 characters long.

Output

Print the new sum that Xenia can count.

Sample Input:

Case 1: 3+2+1
Case 2: 1+1+3+1+3
Case 3: 2+1+2+2+2+3+1+3+1+2
Case 4:
2+3+3+1+2+2+2+1+1+2+1+3+2+2+3+3+2+2+3+3+3+1+1+1+3+3+3+2+1+3+2+3+2+1+1+3+3+3+1+2+2+1+2+2+1+2+1+3+1+1

Sample Output:

Case 1: 1+2+3
Case 2: 1+1+1+3+3
Case 3: 1+1+1+2+2+2+2+2+3+3
Case 4:
1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+2+2+2+2+2+2+2+2+2+2+2+2+2+2+2+2+2+3+3+3+3+3+3+3+3+3+3+3+3+3+3+3+3+3+3

Code:

```
#include<stdio.h>
#include<string.h>

int main(void)
{
    char arr[100];
    int i,n,j;
```

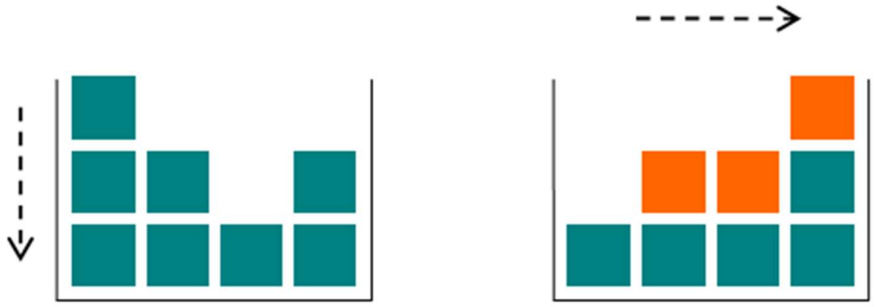
```
    int temp;

    scanf("%s",arr);

    n=strlen(arr);

    for(i=0;i<n;i+=2)
    {
        for(j=0;j<n;j+=2)
            if(arr[i]<arr[j])
            {
                temp=arr[j];
                arr[j]=arr[i];
                arr[i]=temp;
            }
    }

    printf("%s",arr);

    return 0;
}
```

Problem 02: http://codeforces.com/problemset/problem/405/A

Problem Description:

Little Chris is bored during his physics lessons (too easy), so he has built a toy box to keep himself occupied. The box is special, since it has the ability to change gravity.

There are n columns of toy cubes in the box arranged in a line. The i-th column contains a_i cubes. At first, the gravity in the box is pulling the cubes downwards. When Chris switches the gravity, it begins to pull all the cubes to the right side of the box. The figure shows the initial and final configurations of the cubes in the box: the cubes that have changed their position are highlighted with orange.



Given the initial configuration of the toy cubes in the box, find the amounts of cubes in each of the n columns after the gravity switch!

Input

The first line of input contains an integer n ($1 \le n \le 100$), the number of the columns in the box. The next line contains n space-separated integer numbers. The i-th number a_i ($1 \le a_i \le 100$) denotes the number of cubes in the i-th column.

Output

Output *n* integer numbers separated by spaces, where the *i*-th number is the amount of cubes in the *i*-th column after the gravity switch.

Sample Input:

Case 1: 4
3 2 1 2
Case 2: 5
2 1 2 1 2
Case 3: 6
100 40 60 20 1 80
Case 4: 10
1 2 3 4 5 6 7 8 9 10
Sample Output:

Case 1: 1 2 2 3
Case 2: 1 1 2 2 2
Case 3: 1 20 40 60 80 100
Case 4: 1 2 3 4 5 6 7 8 9 10

Code:

```
#include<iostream>
using namespace std;

int main(void)
{
    int n,a[100];
    int i,j,temp;

    cin>>n;
    for(i=0;i<n;i++)
        cin>>a[i];

    for(i=0;i<n;i++)
        for(j=i;j<n;j++)
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }

    for(i=0;i<n;i++)
        cout<<a[i]<<" ";

    return 0;
}
```