For Loops Challenge

Bjarne is learning about programming. Yesterday’s lesson was about *for* loops. To put his skills into practice, he had to write a number of *for* loops that printed consecutive positive integers. He was so proud of his creation that he stored the output of his program into a file. For example, the contents of the file could have looked like this:

```
5 6 7 8 9 10 11 12 13 14
56 57 58 59 60 61 62 63 64 65
100 101 102 103 104 105 106
```

Today, he opened his file and realized it is now inconsistent: the numbers weren’t sorted in ascending order anymore! His wife told him she was bored so she swapped some numbers around. He’s so frustrated he can’t rewrite the program.

Your task is to help him out. Read out the numbers and print all the C++ for loops that recreate the original file.

Input

There are many lines in the input. The i-th line contains a sequence of space-separated positive integers, where each integer is between 0 and 1000000000. It is guaranteed there are no repeated integers and that there will be at minimum one line with one integer, no line will have more than 1000 integers.

Output

Output all the for loops that generate Bjarne’s original file, one per line. Print the *for* loops in order. That is, if the numbers of the i-th loop are less than the numbers of the j-th loop, the i-th for loop must be printed first.

Notes:

- Name ‘i’ the variable of each for loop.
- Don’t use brackets.
- The *for* loop condition must be inclusive, that is, use ‘<=’.

- The increment section of the for loop must be "i++".
- The C++ code you print need not include a line ending command.
- Beware of spaces. All your *for* loops must contain the same number of spaces as this sample:

```
for (int i = a; i <= b; i++) cout << i << “ “;
```

Example

Input

```
9 6 100 1 3 105
2 4 101 102 103 104 5 7 8
```

Output

```
for (int i = 1; i <= 9; i++) cout << i << “ “;
for (int i = 100; i <= 105; i++) cout << i << “ “;
```