2026 Generating permutations

Consider a set $D = \{1, 2, 3, \ldots, n\}$.

A **permutation** of the elements of D is an ordered array $a = a_1 a_2 a_3 \ldots a_n$ of all the distinct elements of D. The number of permutations of D is $n!$.

Sample #1:

Let $D = \{1, 2, 3\}$, then $\#Permutations = 3! = 6$ and Permutations = {123, 132, 213, 231, 312, 321}

Sample #2:

Let $D = \{1, 2, 3, 4\}$, then $\#Permutations = 4! = 24$ and Permutations = {1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321}

Lexicographical order

Let $a = a_1 a_2 a_3 \ldots a_n$ and $b = b_1 b_2 b_3 \ldots b_n$ permutations of D, then $a < b$ if only if for some m with $m < n$ and $a_1 = b_1$, $a_2 = b_2$, $a_3 = b_3$, $\ldots$, $a_{m-1} = b_{m-1}$ and $a_m < b_m$

Check for the samples that the permutations are in lexicographical order !!!.

Generating Permutations

If the permutation b is the lexicographical successor of the permutation a, then we can get b from a using the next procedure:

1. Look for the greater m that $a_m < a_m + 1$
2. Make $b_1 = a_1$, $b_2 = a_2$, $\ldots$, $b_{m-1} = a_{m-1}$
3. Make b_m equal to the minor value of a_{m+1}, a_{m+2}, $\ldots$, a_n, that is greater that a_m.
4. Make $b_{m+1} < b_{m+2} < \cdots < b_n$

Make a program that accept in an input file the number of elements n $(0 < n < 10)$ of D and generate in an output file all the permutations of D lexicographically ordered.

Input

Input consists on several test cases.

For each case, the input is the number of elements n $(0 < n < 10)$ of the set D. There might be several runs in the input file. your program should stop when $n = 0$.

Output

For each case, the output will be the name of the run, the value of n, follow by the list of all the permutations of D lexicographically ordered. There should be a blank line between runs.

Sample Input

```
1
2
4
0
```

Sample Output

```
Run 1  n=1
1

Run 2  n=2
12
21

Run 3  n=4
1 2 3 4
1 2 4 3
1 3 2 4
1 3 4 2
1 4 2 3
1 4 3 2
2 1 3 4
2 1 4 3
2 3 1 4
2 3 4 1
2 4 1 3
2 4 3 1
3 1 2 4
3 1 4 2
3 2 1 4
3 2 4 1
3 4 1 2
3 4 2 1
4 1 2 3
4 1 3 2
4 2 1 3
4 2 3 1
4 3 1 2
4 3 2 1
```