itertools.permutations()

itertools.permutations(iterable[, r])

This tool returns successive r length permutations of elements in an iterable.

If r is not specified or is `None`, then r defaults to the length of the iterable, and all possible full length permutations are generated.

Permutations are printed in a lexicographic sorted order. So, if the input iterable is sorted, the permutation tuples will be produced in a sorted order.

Sample Code

```
>>> from itertools import permutations
>>> print permutations(['1','2','3'])
<itertools.permutations object at 0x02A45210>
>>>
>>> print list(permutations(['1','2','3']))
[('1', '2', '3'), ('1', '3', '2'), ('2', '1', '3'), ('2', '3', '1'), ('3', '1', '2'), ('3', '2', '1')]
>>>
>>> print list(permutations(['1','2','3'],2))
[('1', '2'), ('1', '3'), ('2', '1'), ('2', '3'), ('3', '1'), ('3', '2')]
>>>
>>> print list(permutations('abc',3))
[('a', 'b', 'c'), ('a', 'c', 'b'), ('b', 'a', 'c'), ('b', 'c', 'a'), ('c', 'a', 'b'), ('c', 'b', 'a')]
```

Task

You are given a string S.
Your task is to print all possible permutations of size k of the string in lexicographic sorted order.

Input Format

A single line containing the space separated string S and the integer value k.

Constraints

$0 < k \leq len(S)$
The string contains only *UPPERCASE* characters.

Output Format

Print the permutations of the string S on separate lines.

Sample Input

```
HACK 2
```

Sample Output

```
AC
AH
AK
CA
CH
CK
HA
HC
```

```
HK
KA
KC
KH
```

Explanation

All possible size 2 permutations of the string "**HACK**" are printed in lexicographic sorted order.