itertools.permutations() ★

30 more points to get your next star!

Rank: **1346588** | Points: **40/70**

You have successfully solved itertools.permutations() Share

You are now 30 points away from the 2nd star for your python badge.

Try the next challenge | **Try a Random Challenge**

Problem Submissions Leaderboard Editorial

RATE THIS CHALLENGE

★ ★ ★ ★ ★

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

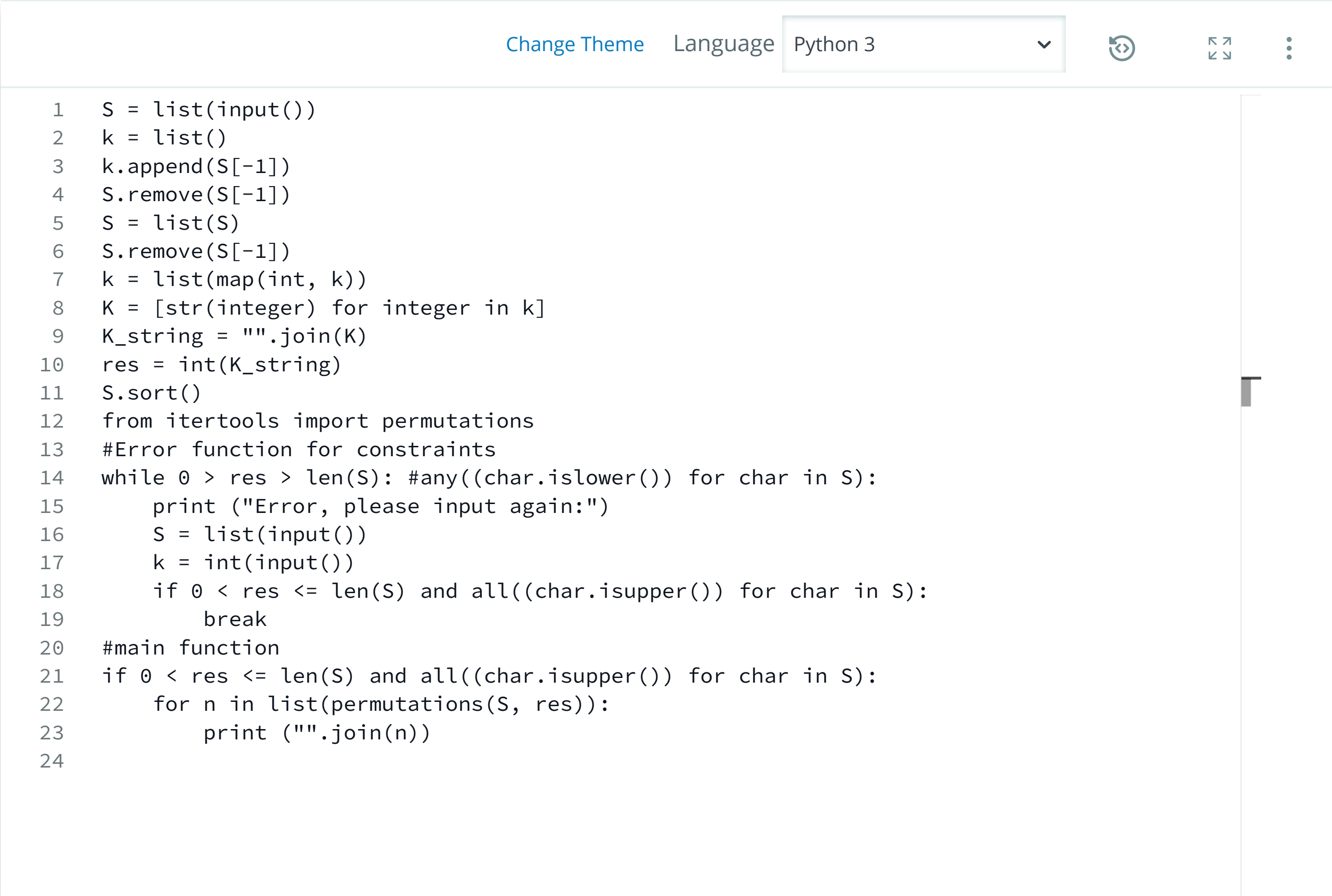
```
HACK 2
```

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.



```
S = list(input())
k = list()
k.append(S[-1])
S.remove(S[-1])
S = list(S)
S.remove(S[-1])
k = list(map(int, k))
K = [str(integer) for integer in k]
K_string = "".join(K)
res = int(K_string)
S.sort()
from itertools import permutations
#Error function for constraints
while 0 > res > len(S): #any((char.islower()) for char in S):
    print ("Error, please input again:")
    S = list(input())
    k = int(input())
    if 0 < res <= len(S) and all((char.isupper()) for char in S):
        break
#main function
if 0 < res <= len(S) and all((char.isupper()) for char in S):
    for n in list(permutations(S, res)):
        print ("".join(n))

```

Line: 11 Col: 9

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

✓ **Test case 0**

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

Success

Input (stdin) Download

```
HACK 2
```

Expected Output Download

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

Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy