Partition into K subset which are non-empty

```
def partitionInKSubsets(n, k):
    ans = [[] for i in range(k)]
    partitionInKSubsetsUtil(n, k, ans, 0, 1)
    return


def partitionInKSubsetsUtil(n, k, ans, sets, i):
    if i > n:
        if sets == k:
            print(ans)
        return

    for j in range(len(ans)):
        if len(ans[j]) > 0:
            ans[j].append(i)
            partitionInKSubsetsUtil(n, k, ans, sets, i + 1)
            ans[j].pop()
        else:
            ans[j].append(i)
            partitionInKSubsetsUtil(n, k, ans, sets + 1, i + 1)
            ans[j].pop()
            break


partitionInKSubsets(3, 2)
```

```
[[1, 2], [3]]

[[1, 3], [2]]

[[1], [2, 3]]
```

```
def KpartitionSet(n, k):
    ans = [None] * k
    helper(n, k, ans, 0, 1)
    return


def helper(n, k, ans, setCount, i):
    if i > n:
        if setCount == k:
            print(ans)
        return
    for j in range(k):
        if ans[j] != None:
            ans[j] = ans[j] + str(i)
            helper(n, k, ans, setCount, i + 1)
            ans[j] = ans[j][:-1]
        else:
            ans[j] = str(i)
            helper(n, k, ans, setCount + 1, i + 1)
            ans[j] = None
            break


KpartitionSet(4, 3)
```