Partition into K equal sum Subsets

```
def KequalSumPartition(arr, k):
    sumArray = [0] * k
    res = [[] for i in range(k)]
    ans = []
    KequalSumPartitionUtil(k, sumArray, res, 0, 0, ans, arr)
    return ans


def KequalSumPartitionUtil(k, sumArray, res, i, ssf, ans, arr):
    if i == len(arr):
        if k == ssf:
            for l in range(1, k):
                if sumArray[l - 1] != sumArray[l]:
                    return
            # ans.append(res)
            print(res)
        return
    for j in range(len(res)):
        if len(res[j]) > 0:
            res[j].append(arr[i])
            sumArray[j] = sumArray[j] + arr[i]
            KequalSumPartitionUtil(k, sumArray, res, i + 1, ssf, ans, arr)
            res[j].pop()
            sumArray[j] = sumArray[j] - arr[i]
        else:
            res[j].append(arr[i])
            sumArray[j] = sumArray[j] + arr[i]
            KequalSumPartitionUtil(k, sumArray, res, i + 1, ssf + 1, ans,
arr)
            res[j].pop()
            sumArray[j] = sumArray[j] - arr[i]
            break


KequalSumPartition([1,2,3,4,5], 3)
```

[[1, 4], [2, 3], [5]]