Task 2 (60 Points, SON & Apriori):

In this task, you are asked to implement SON algorithm **"FirstName_LastName_SON.py"** in **Apache Spark** using Python. Recall that given a set of baskets, SON algorithm divides them into chunks/partitions and then proceed in two stages. First, local frequent itemsets are collected, which form candidates; next, it makes second pass through data to determine which candidates are globally frequent.

You must implement Apriori algorithm for stage one to find local frequent itemset. Make use of monotonicity concept while generating candidate itemset. You may find Python itertools package to be useful in your implementation: https://docs.python.org/3/library/itertools.html#itertools.combinations

Requirements: You must use mapPartitions() method in Spark in stage one that invokes Apriori to find frequent itemsets in each partition. You will need mapPartitions() for your second stage to find the true frequent itemsets too.

Execution Format:

bin/spark-submit FirstName_LastName_SON.py baskets.txt <support> output.txt

baskets.txt is exactly like the one shown in task 1.

<support> is a minimum support ratio in floating format like 0.1 (That is, for an itemset to be frequent, it should appear in at least 10% of the baskets). output.txt is name of the text file.

Output Format:

You should save all the frequent itemset into one file. Each line will have frequent items in sorted ascending order. Singletons followed by pairs, triples, quadruples etc. Frequent items except singletons need to be in tuple format, and are sorted in ascending order within themselves. For example: (1,2), (1,3,4) (1,4,5,8). Example output (**Not an actual output**):

output.txt

```
1
2
3
(1,2)
(1,3)
(1,4)
(2,3)
(2,4)
(3,4)
(1,2,3)
(1,2,3,4)
```