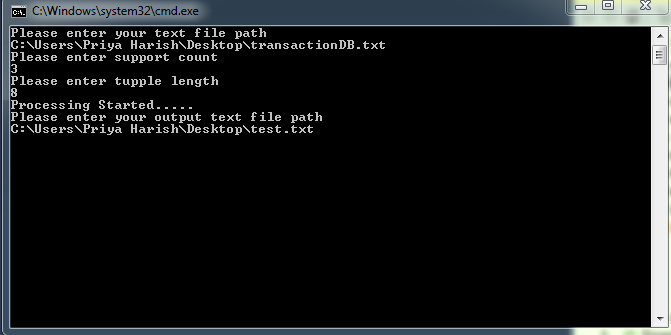
**Application Execution**

Application asks user to input

1. Input file
2. Support Count
3. Tuple Count
4. Output file path



**Application Design**

|  |  |  |
| --- | --- | --- |
| File or Class Name | Function Name | Tasks performed |
| Program.cs | Main | 1. Contains logic to accept input file and write a test output file 2. Contains logic to loop till candidate set returned is empty 3. Contains logic to consolidate frequent item set for k+ |
| TransactionSetGenerator | GenerateHashMap | 1. Collects all UNIQUE items from transaction and its count 2. It assigns integer value to all strings. These integer equivalent are used for further processing |
| GenerateHashTransactionSet | Converts transaction set to integer equivalent using hash map created in function GenerateHashMap |
| GenerateFrequentSet | InitialFrequentItemSetK1 | Returns items whose count is greater than min support count |
| GenerateFrequentItemSet | Generates frequent item set for k=2. Special function for K=2 is implemented because it at this step subset creation is not required. |
| GenerateFrequentItemSet | Generates frequent item set for all k = 3+ |
| GetSupportCount | Used to get support count by traversing through whole transaction |
| GenerateCandidateSet | GenerateCandidate | Generates candidate item set for k=2. Logic for K=2 is different as it doesn’t need subset to be checked |
| GenerateCandidate | Generates candidate item set for k= 3+ |
| CreateSubset | Pruning step. Checks subset of candidate is present in frequent item set or not |