Assignment 3

Thread and Collection

1. Soham has n sweets, where the i^{th} sweet is of type “Sandesh” (say) while (i+1)th type is “Pithe” (say). So, the types need to be encoded by numerals for the assignment. He noticed that he started to gain weight.

His classmate Arjeesh advised him to only eat n / 2 of the sweets and distribute the rest among friends. Soham still wishes to taste the maximum number of different types of sweets while not ignoring his classmate’s suggestion.

So, if there are n sweets of k different varieties then Soham can have a maximum of n/2 sweets of x (<=k) different types of y different combinations.

Write an appropriate Java code using collections framework such that the system takes n (even or odd) and k as input and provides the value of x along with at least some of the y combinations if not all.

2. Extract the words and their frequencies from a text file. Then store them as key value pairs in a TreeMap. Sort them in descending order using Comparator. Now, display the greatest and the least key value pairs from the collection.

3. Given preorder and inorder traversal outcomes of a family tree, construct and return the corresponding *binary tree*. If two nodes in the tree has the same depth then they are cousins. So, print the names who are cousins in generations of the family history. Try to use java collections classes and functions as much as you can.

4. You have the number of goals scored by the football teams of the world cup for a year. Store them in a collection. Find the product of all the scores except itself.

The product of any prefix or suffix is expected to fit in Java’s int range. Otherwise, remove the very high scorers.

Write program to display the products for the corresponding scores as input. The program should run in less than O(n^2) time and should not use ‘/’.

5. One thread takes an input from user and increments an integer variable by that amount. Another thread reduces the variable by a fixed amount. Execute two versions of each thread simultaneously and all working on same variable. Once all threads are over display the value of the variable. Repeat the threads unless different results emerge for repeated executions as a consequence of parallel programming. Don’t use lambda functions.

Modify the problem stated in ensuring mutual exclusion on shared variable.

6. You have the four functions:

- printFizz that prints the word "fizz" to the console,
- printBuzz that prints the word "buzz" to the console,
- printFizzBuzz that prints the word "fizzbuzz" to the console, and
- printNumber that prints a given integer to the console.

You are given an instance of the class FizzBuzz that has four functions: fizz, buzz, fizzbuzz and number. The same instance of FizzBuzz will be passed to four different threads:

- **Thread A:** calls fizz() that should output the word "fizz".
- **Thread B:** calls buzz() that should output the word "buzz".
- **Thread C:** calls fizzbuzz() that should output the word "fizzbuzz".
- **Thread D:** calls number() that should only output the integers.

Repeat the threads unless different results emerge for repeated executions as a consequence of parallel programming. You may use lambda function for only creating Runnable objects.

7. Modify the given class so that the i^{th} token (**1-indexed**) of the series is:

- "fizzbuzz" if i is divisible by 5 and 7,
- "fizz" if i is divisible by 5 and not 9,
- "buzz" if i is divisible by 7 and not 3, or
- i if i is not divisible by 7 or 5.

If n=10 then the output will be [1,2,3,4,fizz,6,buzz,8,9,fizz]. You may use lambda function for only creating Runnable objects.

8. You are asked to create a notice board. Anybody can read the notice board parallelly but when any one is writing on it, all others should wait for the modification to finish. So, the waiting threads would display a suitable message to indicate that it is waiting for the update to finish.