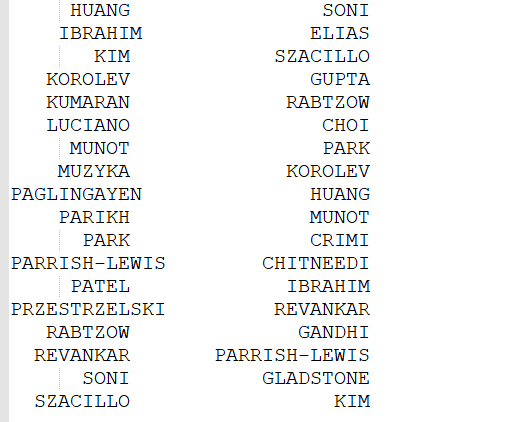
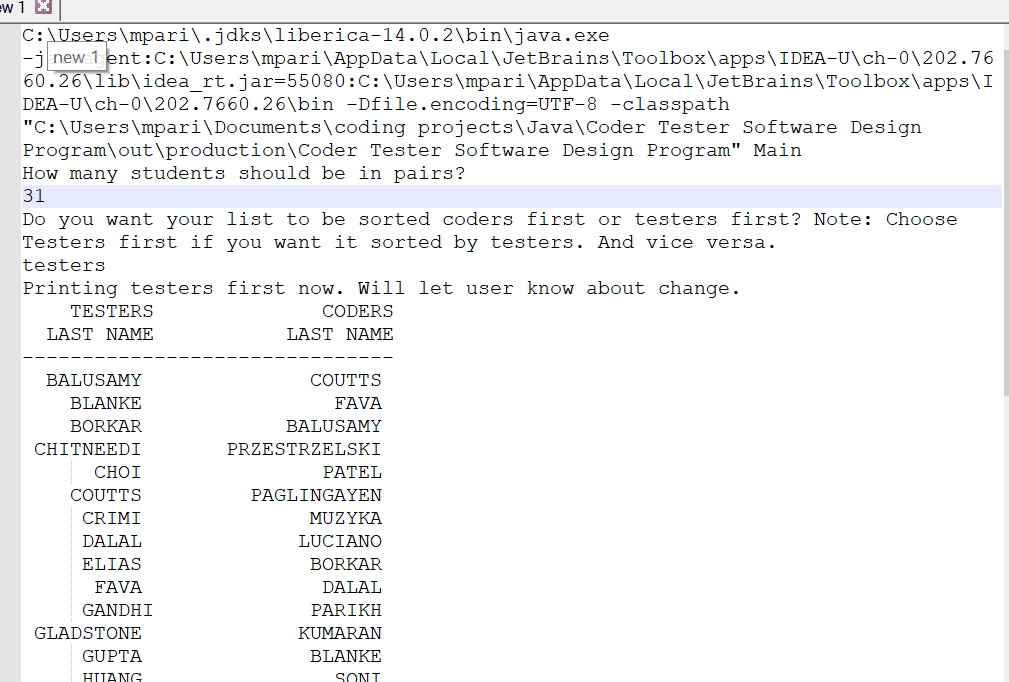
Main.java

*/\*  
Rohan Parikh  
Coder Tester lab  
29 September 2020 -  
\*/  
  
  
//Status update for 7 October 2020  
/\*  
Coder sorting is complete, just got to do testing and then let user pick choice  
 \*/  
  
import* java.io.BufferedReader;  
*import* java.io.File;  
*import* java.io.FileReader;  
*import* java.io.IOException;  
*import* java.util.ArrayList;  
*import* java.util.*Comparator*;  
*import* java.util.*List*;  
*import* java.util.Scanner;  
*import* java.util.concurrent.TimeUnit;  
  
*public class* Main {  
 *// Arrays to see if the randomindex has already been used  
 static boolean*[] *usedCoder* = *new boolean*[33];  
 *static boolean*[] *usedTester* = *new boolean*[33];  
  
 *public static void* main(String[] args) {  
 *//printing out to make sure the values are starting out as false  
 // initialize variables and creating array list* File file;  
 *//String TempeoryStorage is for coders first* String tempeoryStorage = *null*;  
 *//String testersTemperory is for testers first* String testersTempeory = *null*;  
  
 *List* <String> studentPairsCodersFirst = *new* ArrayList<>();  
 *List* <String> studentPairsTestersFirst = *new* ArrayList<>();  
 Scanner in = *new* Scanner(System.*in*);  
 *List*<String> allStudents = *new* ArrayList<>();  
  
  
  
  
 BufferedReader br;  
 *int* numOfStudents;  
  
  
 *//Reading file into an array using bufferreader  
  
 try* {  
 file = *new* File("C:\\Users\\mpari\\Documents\\coding projects\\Java\\Coder Tester Software Design Program\\src\\SD\_ClassList.txt");  
 br = *new* BufferedReader(*new* FileReader(String.*valueOf*(file)));  
 *while* (br.ready()) {  
 allStudents.add(br.readLine());  
 }  
 } *catch* (IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
  
  
 *//Asking user for how many students do they want* System.*out*.println("How many students should be in pairs?");  
 numOfStudents = in.nextInt();  
 *if* (numOfStudents > allStudents.size() || numOfStudents <= 0) {  
 System.*out*.println("Sorry, the amount of students you inputted is larger than the students" +  
 " in the file. Input a different number.");  
 numOfStudents = in.nextInt();  
 }  
  
 System.*out*.println("Do you want your list to be sorted coders first or testers first? Note: Choose" +  
 " Testers first if you want it sorted by testers. And vice versa.");  
  
 String codersOrTesters = in.next();  
  
 *//This if statement is coders first  
 if* (codersOrTesters.toLowerCase().equals("coders")) {  
  
 *int* i = 0;  
 *while* (i != numOfStudents) {  
 i++;  
  
 *// method for coders first* tempeoryStorage = ((*studentsAllCoders*(allStudents, numOfStudents)));  
 studentPairsCodersFirst.add(tempeoryStorage);  
 }  
  
  
 studentPairsCodersFirst.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output* String firstLine = String.*format*("%11S %19S ", " Coders", "Testers");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%11S %19S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-----------------------------------------------");  
 *//for loop to run for amount of coders and then also to split the string and output  
 for* (*int* P = 0; P < studentPairsCodersFirst.size(); P++) {  
 String value = studentPairsCodersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%11S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 *try* {  
 TimeUnit.*SECONDS*.sleep(10);  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 *else* {  
 *int* i = 0;  
 *while* (i != numOfStudents) {  
 i++;  
  
 *// method for testers first* testersTempeory = ((*studentsAllTesters*(allStudents, numOfStudents)));  
 studentPairsTestersFirst.add(testersTempeory);  
 }  
  
  
 studentPairsTestersFirst.sort(*Comparator*.*comparing*(String::toString));  
 *//Formatting output but this time testers first* System.*out*.println("Printing testers first now. Will let user know about change.");  
 String firstLine = String.*format*("%11S %19S ", " Testers", "Coders");  
 System.*out*.println(firstLine);  
 String secondLine = String.*format*("%11S %19S ", " Last Name", " Last Name");  
 System.*out*.println(secondLine);  
 System.*out*.println("-------------------------------");  
 *//for loop to run for amount of coders and then also to split the string and output  
 for* (*int* P = 0; P < studentPairsTestersFirst.size(); P++) {  
 String value = studentPairsTestersFirst.get(P);  
 String[] split = value.split(",");  
 String names = String.*format*("%11S %19S ", split[0], split[1]);  
 System.*out*.println(names);  
 }  
 *try* {  
 TimeUnit.*SECONDS*.sleep(10);  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 *private static* String studentsAllTesters(*List*<String> allStudents, *int* numOfStudents) {  
 *while* (*true*) {  
 *//common variable for randomindex  
 int* studentsAllLength = allStudents.size();  
 *//random index and inputting arraylist value into a string  
 int* randomIndex = (*int*) (Math.*random*() \* studentsAllLength);  
 *int* randomIndex2 = (*int*) (Math.*random*() \* studentsAllLength);  
 *if* ((!*usedTester*[randomIndex] && !*usedCoder*[randomIndex2]) && randomIndex != randomIndex2) {  
  
 String last = allStudents.get(randomIndex) + "," + allStudents.get(randomIndex2);  
 *usedCoder*[randomIndex2] = *true*;  
 *usedTester*[randomIndex] = *true*;  
 *return* last;  
 }  
 }  
 }  
  
 *public static* String studentsAllCoders(*List*<String> allStudents, *int* numOfStudents) {  
 *while* (*true*) {  
 *//common variable for randomindex  
 int* studentsAllLength = allStudents.size();  
 *//random index and inputting arraylist value into a string  
 int* randomIndex = (*int*) (Math.*random*() \* studentsAllLength);  
 *int* randomIndex2 = (*int*) (Math.*random*() \* studentsAllLength);  
 *if* ((!*usedTester*[randomIndex] && !*usedCoder*[randomIndex2]) && randomIndex != randomIndex2)  
 {  
 String last = allStudents.get(randomIndex2) + "," + allStudents.get(randomIndex);  
 *usedCoder*[randomIndex2] = *true*;  
 *usedTester*[randomIndex] = *true*;  
 *return* last;  
 }  
 }  
 }  
}

Output #1



Output #2

