```java
package edu.wmich.cs3310.a6.setup;
/*
* CS3310 A5
* Asgn 6 (CS3310 F15)
* @author Jonah Groendal
* Last changed: 12/9/2015
*
* This class creates MapGraph.bin and CityNameList.csv
*/
import java.io.*;

public class Setup {

    public static void main(String[] args) throws IOException {
        Scanner emd = new Scanner(new File("EuropeMapData.csv"));
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter
("CityNameList.csv", false)));
        DataOutputStream binOut= new DataOutputStream(new FileOutputStream
("MapGraph.bin"));
        String temp[] = new String[3];
        Short tempS[] = new Short[3];

        //Seek to beginning of map names
        for (int i=0; i< 11; i++) {
            emd.nextLine();
        }
        //Print city names to CityNameList.csv
        for (int i=0; i< 21; i++) {
            temp = emd.next().split(",");
            out.print(String.format("%-10s,%s,", temp[0], temp[1]));
        }
        out.close();

        // Seek to beginning of record data
        for (int i=0; i<5; i++) {
            emd.nextLine();
        }
        // Loop through record data and build MapGraph.bin
        int currentBinIndex = 0;
        binOut.writeShort(21);
        while (emd.hasNextLine()) {
            temp = emd.nextLine().split(",");
            tempS[0] = Short.parseShort(temp[0]);
            tempS[1] = Short.parseShort(temp[1]);
            tempS[2] = Short.parseShort(temp[2]);
            if (tempS[1] < tempS[0]) {
                Short tempTemp = tempS[0];
                tempS[0] = tempS[1];
```

```
                tempS[1] = tempTemp;
            }
            // Calculate desired index and "seek" to that position while filling
in empty records with max short value
            int desiredIndex = 209 - ((20-tempS[0])*(21-tempS[0])/2) + (tempS
[1]-tempS[0]);
            while (desiredIndex > currentBinIndex) {
                binOut.writeShort(32767);
                currentBinIndex++;
            }
            binOut.writeShort(tempS[2]);
            currentBinIndex++;
        }
        while (currentBinIndex < 210) {
            binOut.writeShort(32767);
            currentBinIndex++;
        }
        emd.close();
        binOut.close();
    }

}
```