A picture containing logo

Description automatically generated

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO

**Problem Based Assignment**

**Data Structures and Algorithm**

| Submitted To: | Dr. Mohsin Ali Memon |
| --- | --- |
| Submitted by: | Syeda Anoosha Iqtidar (21SW004) |
| Dated: | 13-03-2023 |
| Section: | III |

**Step 1:**

Use [https://www.daniel-braun.com/technik/reverse-geocoding-library-for-java/](https://www.daniel-braun.com/technik/reverse-geocoding-library-for-java/%20) library to find out the city and country from the given coordinates and store them in yearly earthquake collection along with magnitude. (Collection of each year means 52 collections)

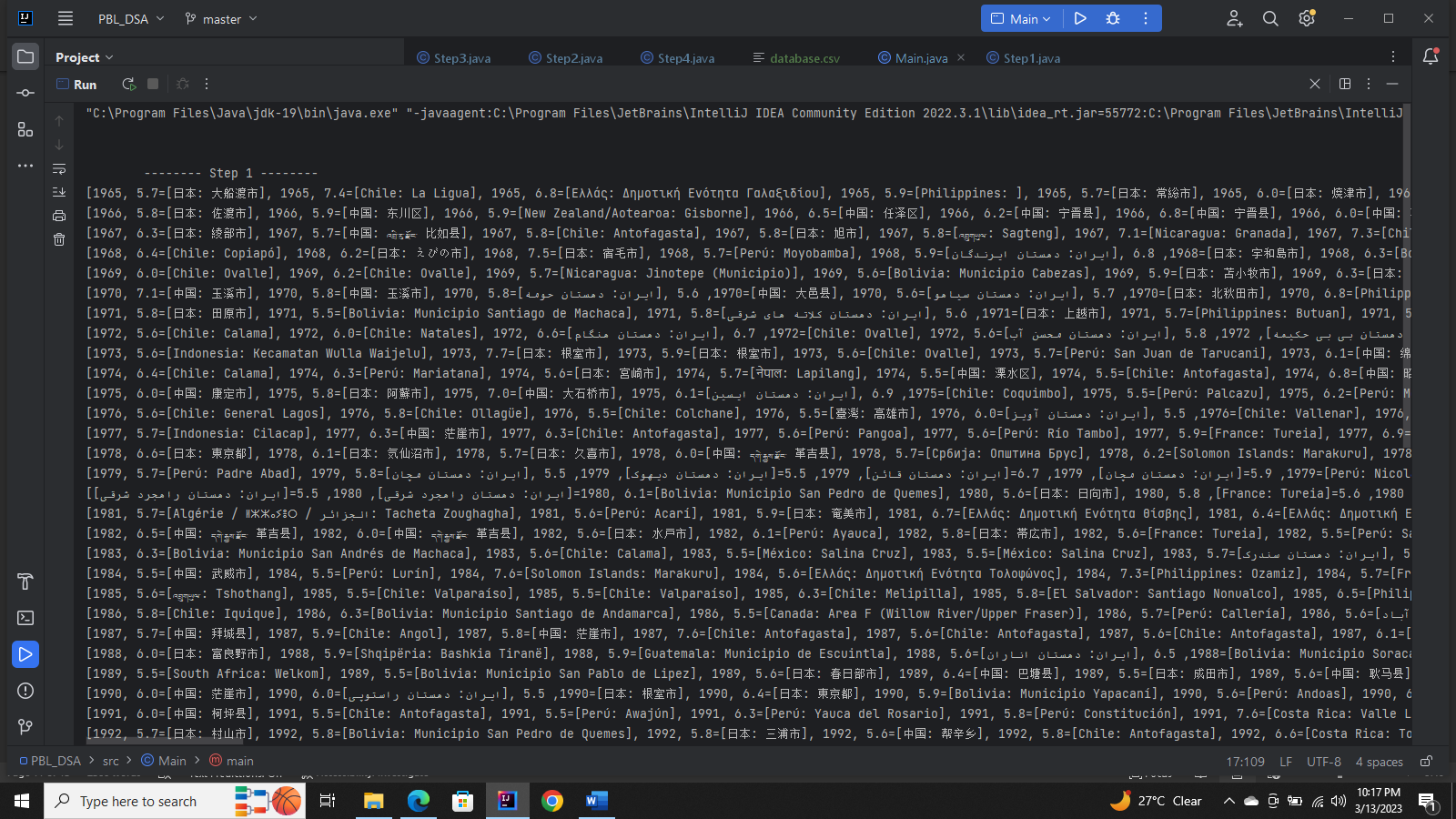
**Code:**

import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.\*;  
  
public class Step1 {  
 public ArrayList<String>[] storeEarthquakeData = new ArrayList[52];  
 Scanner sc = new Scanner(System.*in*);  
  
 public void storing() throws IOException {  
  
 for (int i = 0; i < storeEarthquakeData.length; i++) {  
 storeEarthquakeData[i] = new ArrayList<>();  
 }  
  
 File file = new File("D:\\PBL\\Valid countries and cities.txt");  
  
 FileReader fileReader = new FileReader(file);  
 BufferedReader br = new BufferedReader(fileReader);  
 String line;  
 br.readLine();  
  
 int y = 1965, L = 0, i = 0;  
 while ((line = br.readLine()) != null && L<1103) {  
  
 String[] read= line.split("=");  
 String[] eq1 = read[0].split(", ");  
  
 int year = Integer.*parseInt*(eq1[0]);  
 if(year!=y) {  
 i++;  
 y=year;  
 storeEarthquakeData[i] = new ArrayList<>();  
 }  
 storeEarthquakeData[i].add(line);  
 L++;  
 }  
 br.close();  
  
 }  
 public void getYearlyEarthquakes(){  
  
 for (int i = 0; i < storeEarthquakeData.length; i++) {  
 Object[] arr = storeEarthquakeData[i].toArray();  
 System.*out*.println(Arrays.*toString*(arr));  
 }  
  
 }  
  
 public void getAverageCountries(){  
 System.*out*.println("\n\nEnter country name:");  
 String country = sc.nextLine();  
 int year = 1965;  
  
 for (int i = 0; i < storeEarthquakeData.length; i++) {  
 int numberOfEQ = 0;  
 String[] arr = storeEarthquakeData[i].toArray(new String[0]);  
 for (int j = 0; j < arr.length; j++) {  
 if (arr[j].contains(country)){  
 numberOfEQ++;  
 }  
 }  
 if(numberOfEQ==0){  
 System.*out*.println("No earthquake in "+year);  
 }else {  
 System.*out*.println("Average number of Earthquake in year " + year + " is " + numberOfEQ);  
 }  
 year++;  
 }  
  
 }  
}

**Algorithm:**

1. Make an array of ArrayList of String with size 52.
2. In a method, initialize these 52 objects. Each object will have separate initialization.
3. Initialize some helping variables as y=1965, L=0 and i=0.
4. Read from the file “Valid countries and cities.txt”, which contains year, city, country, and its magnitude where there is no “Unable to geocode” error and store this line at ith index of ArrayList.
5. It uses a while loop to read each line of the text file, splits the line into an array of earthquake data, extracts the year of the earthquake.
6. This process of reading the file will continue until the year in the file is equal to y.
7. When the year is not equal to y, y will hold the new value of year and i will be incremented by 1.
8. In this way, after reading the while file, we will have 52 objects containing earthquake data of all the years.

**Output:**



**Note:**

* Methods in Step1 like getYearlyEarthquakes() will retrieve all the data from all the years and getAverageCountries() will be used for Problem 1.
* For file “Valid countries and cities.txt” refer to <https://github.com/SyedaAnoosha/PBL_DSA/tree/master/src/Files>

**Step 2:**

Make a queue storing biggest (with highest magnitude) quake of each year with magnitude and country, starting from 1965 to 2016. (52 elements in the queue approx.).

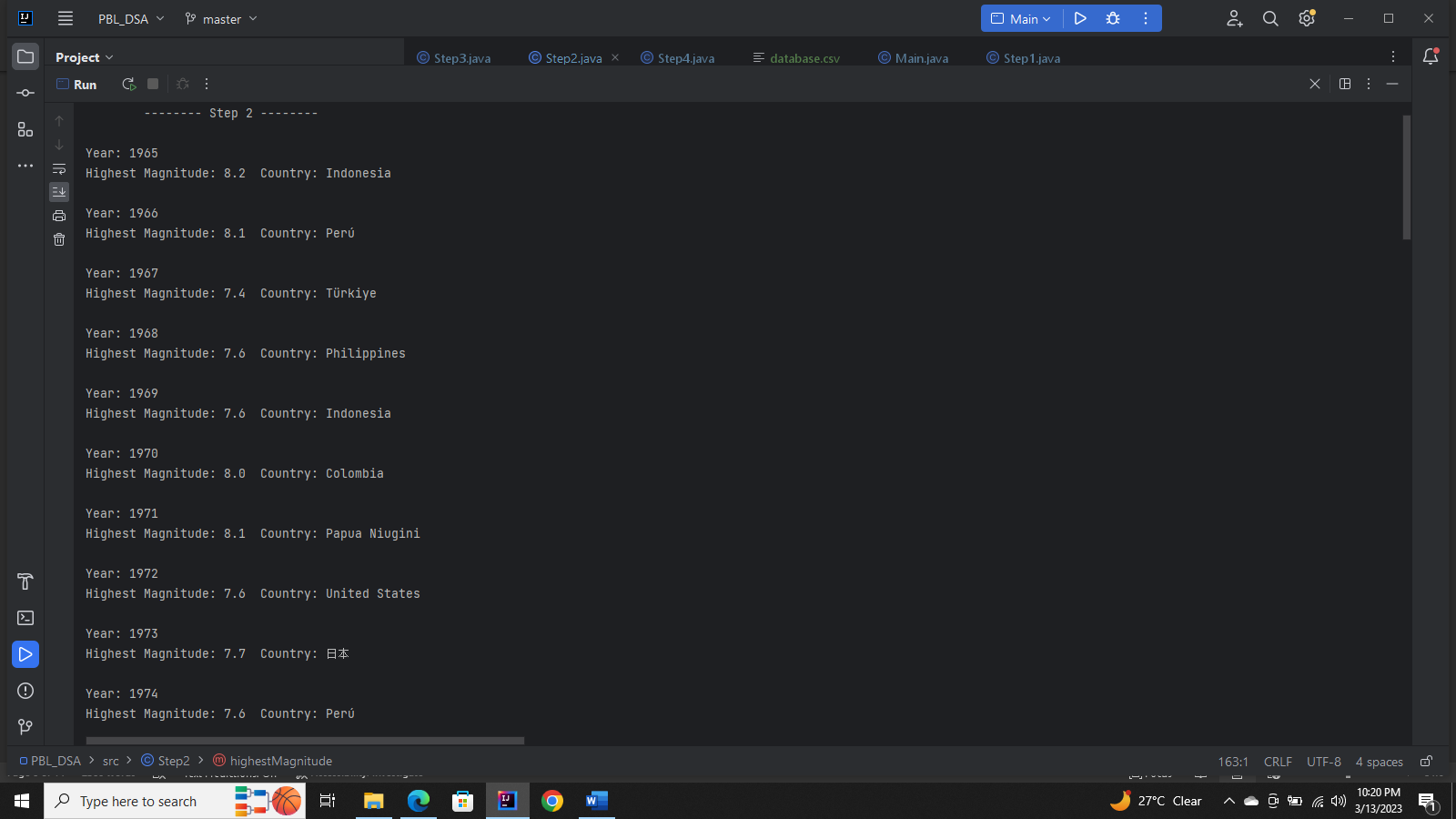
**Code:**

import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.LinkedList;  
import java.util.Queue;  
  
public class Step2 {  
  
 public Queue<Earthquake> maxEarthquake= new LinkedList<>();  
  
 public void highestMagnitude() throws IOException {  
  
 File f = new File("D:\\PBL\\Max Magnitude.txt");  
 BufferedReader br = new BufferedReader(new FileReader(f));  
 String line = "";  
  
 while((line=br.readLine()) != null){  
  
 String[] read = line.split(": ");  
 String country = read[0];  
 String[] data= read[1].split(",");  
 double mag = Double.*parseDouble*(data[0]);  
 int year = Integer.*parseInt*(data[1]);  
 maxEarthquake.add(new Earthquake(country,mag,year));  
 }  
 }  
 public void getYearlyMaxEarthquake(){  
 Earthquake[] arr = maxEarthquake.toArray(new Earthquake[0]);  
 for (int i = 0; i < arr.length; i++) {  
 System.*out*.println("\nYear: "+arr[i].getYear()+"\nHighest Magnitude: "+arr[i].getMagnitude()+" Country: "+arr[i].getCountry());  
 }  
 }  
  
 public void problem2(){  
 Earthquake[] arr = maxEarthquake.toArray(new Earthquake[0]);  
 for (int i = 0; i < arr.length; i++) {  
 if(arr[i].getYear() >= 2005 && arr[i].getYear() <= 2015 ) {  
 System.*out*.println("\nYear: "+arr[i].getYear()+"\nHighest Magnitude: "+arr[i].getMagnitude()+" Country: "+arr[i].getCountry());  
 }  
 }  
 }  
}

**Algorithm:**

1. Make a Queue of LinkedList with Earthquake as an object.
2. Read the file “Max Magnitude.txt”, which contains data about maximum earthquake of each year, line by line and extract country, year and magnitude and store it in new Earthquake object.
3. Add that Earthquake object to the queue.

**Output:**

 A screenshot of a computer

Description automatically generated

**Note:**

* Methods in Step2 like “getYearlyMaxEarthquake()” will retrieve all the max earthquakes from all the years and “problem2()” will be used for Problem 2.
* For file “Max Magnitude.txt” and class “Earthquake” refer to <https://github.com/SyedaAnoosha/PBL_DSA/tree/master/src/Files>

<https://github.com/SyedaAnoosha/PBL_DSA/blob/master/src/Earthquake.java>

**Step 3:**

Make a stack from the collections, one for each country which stores earthquake and its magnitude in the order of the event (the most recent event on top).

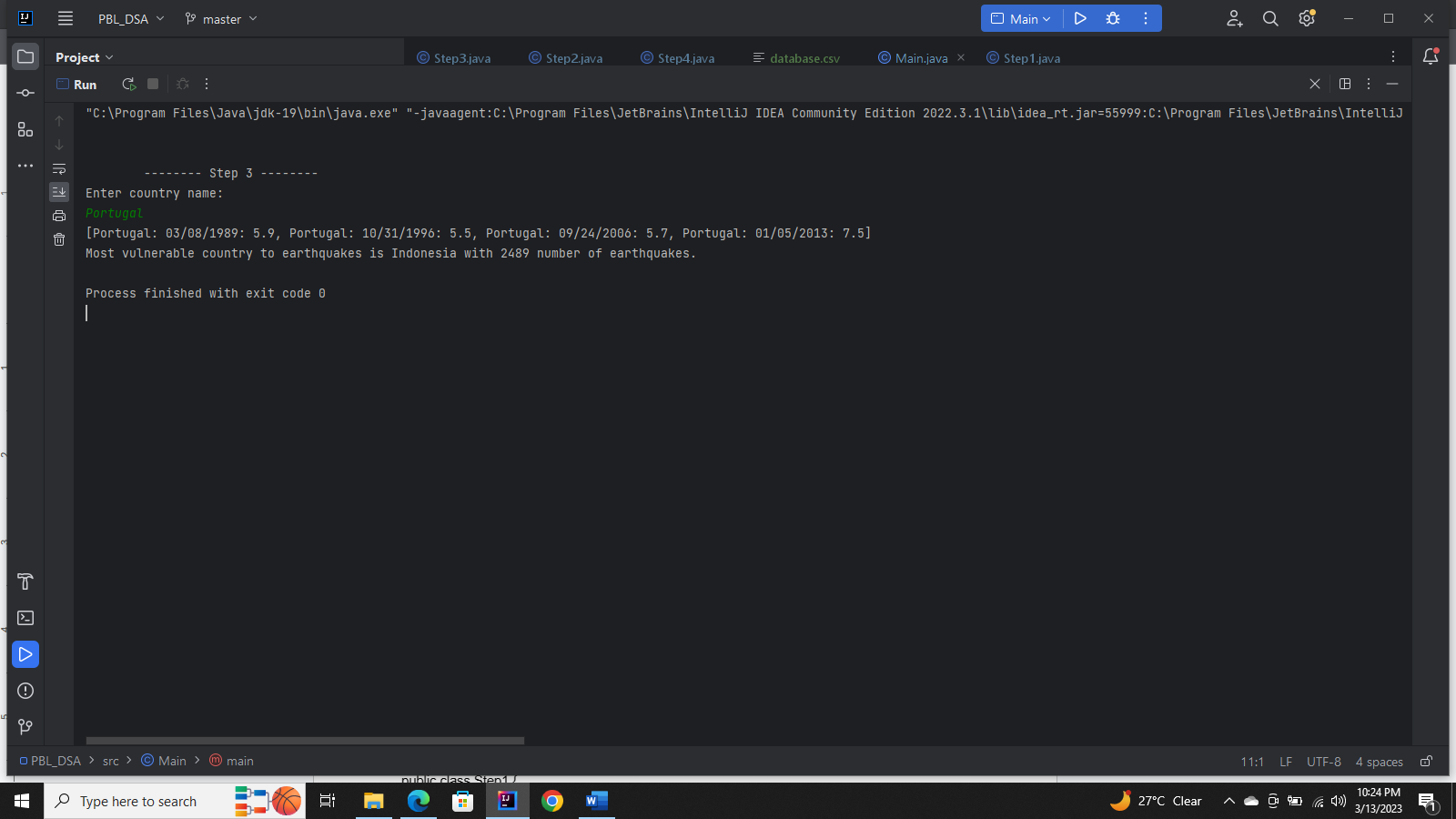
**Code:**

import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.Arrays;  
import java.util.Scanner;  
import java.util.Stack;  
public class Step3 {  
 public static Stack<Object>[] *recentEQ* = new Stack[110];  
 Scanner sc = new Scanner(System.*in*);  
  
 public static String[] *countries* = {"United States", "中国", "Indonesia", "ایران", "Philippines", "Chile", "Vanuatu", "臺灣", "Papua Niugini",  
 "Ελλάς", "Argentina", "नेपाल", "日本", "México", "República Dominicana", "Solomon Islands", "South Georgia and the South Sandwich Islands",  
 "Colombia", "Perú", "Antigua and Barbuda", "Zambia", "Guatemala", "မြန်မာ", "République démocratique du Congo", "New Zealand/Aotearoa",  
 "Venezuela", "Монгол улс ᠮᠤᠩᠭᠤᠯ ᠤᠯᠤᠰ", "Bolivia", "Россия", "Türkiye", "Тоҷикистон", "France", "Brasil", "India", "Ecuador", "Tonga",  
 "Viti", "Australia", "Panamá", "Belau", "Canada", "Ísland", "افغانستان", "پاکستان", "འབྲུགཡུལ་", "Costa Rica", "Nicaragua", "British Indian Ocean Territory",  
 "Shqipëria", "ኢትዮጵያ", "مصر", "Italia", "South Africa", "El Salvador", "Bosna i Hercegovina / Босна и Херцеговина", "Кыргызстан",  
 "Oʻzbekiston", "Djibouti جيبوتي", "Қазақстан", "România", "ประเทศไทย", "Madagasikara / Madagascar", "Gabon", "Norge", "Cuba", "España",  
 "Tanzania", "België / Belgique / Belgien", "Malaysia", "Србија", "Crna Gora / Црна Гора", "ኤርትራ Eritrea إرتريا", "Algérie / ⵍⵣⵣⴰⵢⴻⵔ / الجزائر",  
 "Việt Nam", "ປະເທດລາວ", "Polska", "Honduras", "العراق", "Türkmenistan", "Saint Kitts and Nevis", "Guinée", "اليمن", "საქართველო", "বাংলাদেশ", "Kalaallit Nunaat",  
 "Malawi", "Portugal", "Kenya", "South Sudan", "Micronesia", "България", "Uganda", "Hrvatska", "Maroc / ⵍⵎⵖⵔⵉⴱ / المغرب", "Jamaica", "السودان", "Trinidad and Tobago",  
 "Հայաստան", "Северна Македонија", "Paraguay / Paraguái", "Κύπρος - Kıbrıs", "السعودية", "Dominica", "Azərbaycan", "سوريا", "Soomaaliya الصومال",  
 "Slovenija", "Kosova / Kosovo", "Moçambique", "Ayiti"};  
  
  
 public int countryIndex(String country){  
 for (int i = 0; i < *countries*.length; i++) {  
 if(country.equalsIgnoreCase(*countries*[i])){  
 return i;  
 }  
 }  
 return -1;  
 }  
  
 public Stack<Object>[] getStack() throws IOException {  
  
 for (int i = 0; i < *recentEQ*.length; i++) {  
 *recentEQ*[i] = new Stack<>();  
 }  
  
 File file = new File("D:\\PBL\\Stack.txt");  
 BufferedReader bf = new BufferedReader(new FileReader(file));  
 int j = 0;  
 String line = "";  
 while ((line = bf.readLine()) != null) {  
 String[] read = line.split(", ");  
 for (int i = 0; i < read.length; i++) {  
 *recentEQ*[j].push(read[i]);  
 }  
 j++;  
 }  
 return *recentEQ*;  
 }  
  
 public void getAllRecentEarthquake() throws IOException {  
  
 System.*out*.println("Enter country name:");  
 String country = sc.nextLine();  
 int index = countryIndex(country);  
 while(index == -1){  
 System.*out*.println("Enter valid country name!");  
 country = sc.nextLine();  
 index = countryIndex(country);  
 }  
  
 Stack<Object>[] recentStack = getStack();  
 Object[] arr = recentStack[index].toArray();  
 System.*out*.println(Arrays.*toString*(arr));  
  
 }  
 public void problem3() throws IOException {  
  
 System.*out*.println("Enter country name:");  
 String country = sc.nextLine();  
 int index = countryIndex(country);  
  
 while(index == -1){  
 System.*out*.println("Enter valid country name!");  
 country = sc.nextLine();  
 index = countryIndex(country);  
 }  
  
 System.*out*.println("Number of recent Earthquakes you want to get: ");  
 int recentNumber = sc.nextInt();  
  
 Stack<Object>[] recentStack = getStack();  
 Object[] arr = recentStack[index].toArray();  
  
 int length = arr.length - recentNumber;  
 for (int i = length; i < arr.length ; i++) {  
 System.*out*.println(arr[i]);  
 }  
 }  
   
 public void countryWithMostNumberOfEQ(){  
 int num = 0;  
 String country = "";  
 for (int i = 0; i < *recentEQ*.length ; i++) {  
 int size = *recentEQ*[i].size();  
 if(size > num){  
 country = String.*valueOf*(*recentEQ*[i].peek());  
 country = country.substring(0,country.indexOf(":"));  
 num = size;  
 }  
 }  
 System.*out*.println("Most vulnerable country to earthquakes is "+country+" with "+num+" number of earthquakes.");  
 }  
  
}

**Algorithm**

1. Create an array of stack of Object type with 110 sizes as there are 110 countries in total.
2. Create a String array of countries including all 110 countries, which are extracted from original database “database.csv”.
3. Method “countryIndex” will return the index of the country at ith index, only if the country passed in the argument becomes equal to the one in countries array.
4. In the method “getStack”, read from the file “Stack.txt”.
5. Read the file line by line and split it accordingly and store the data in an array.
6. Loop though this array and store the contents at the jth index of stack.
7. Increment j by 1.
8. It continues till we have the 110 objects of each country with their corresponding data.
9. Return this stack.

**Output:**



**Note:**

* Methods in Step3 like getAllRecentEarthquake () will retrieve all the recent earthquakes for any country, “countryWithMostNumberOfEQ()” will be used for Problem 1 and “problem3()”will be used for Problem 3.
* For files “Stack.txt” and “database.csv” refer to <https://github.com/SyedaAnoosha/PBL_DSA/tree/master/src/Files>

**Step 4:**

Make a linked list which saves the one most recent earthquake with magnitude and country name from each country (use the stack from step 3).

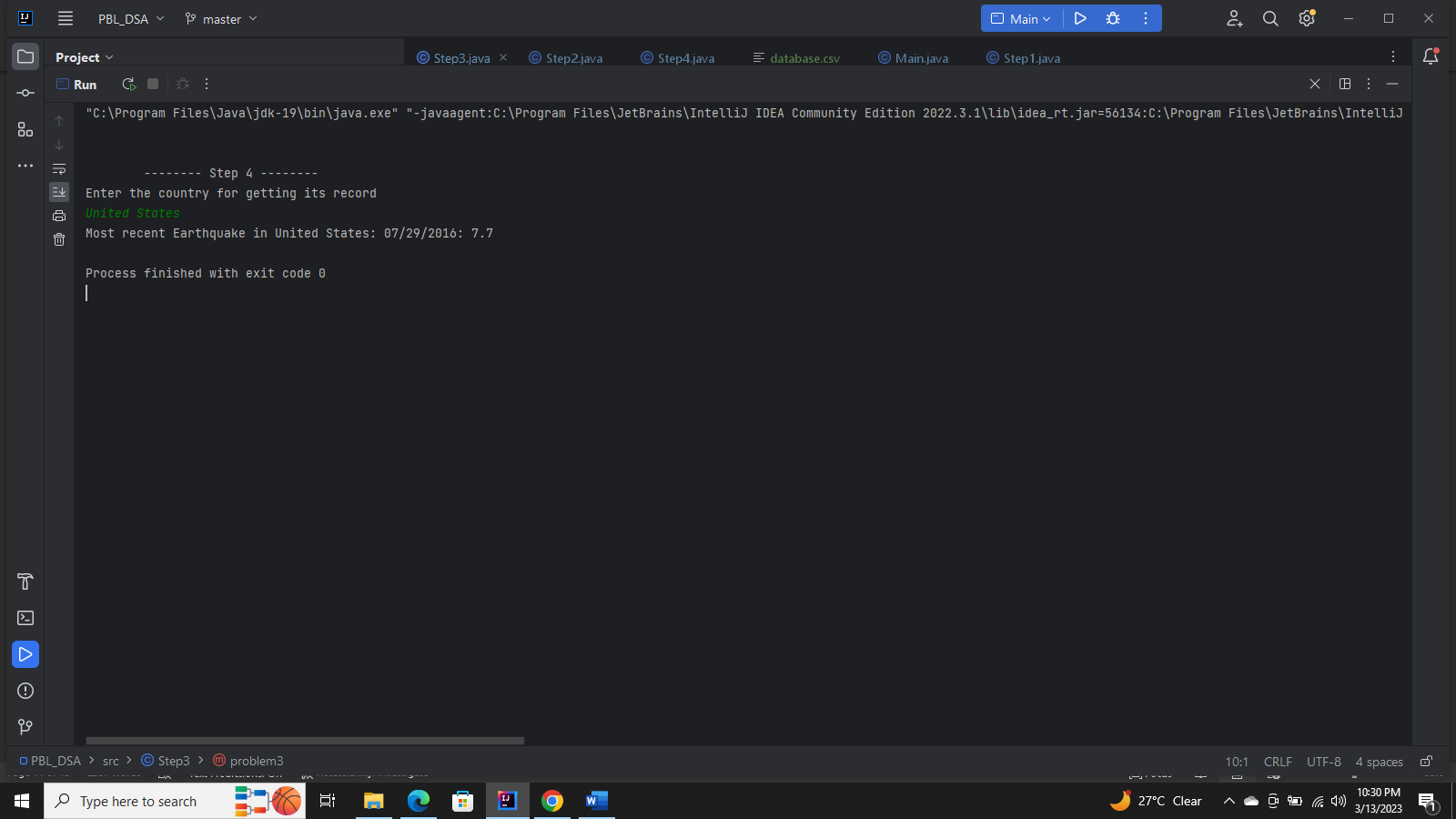
**Code:**

import java.io.IOException;  
import java.util.LinkedList;  
import java.util.Scanner;  
import java.util.Stack;  
  
public class Step4 {  
 Step3 step3 = new Step3();  
 Stack<Object>[] newStack = step3.getStack();  
  
 Scanner sc = new Scanner(System.*in*);  
 LinkedList<Object> mostRecentEQ = new LinkedList<>();  
  
 public Step4() throws IOException {}  
 public void storeMostRecentEarthquake(){  
 for (int i = 0; i < newStack.length; i++) {  
 if(newStack[i] != null) {  
 mostRecentEQ.add(newStack[i].peek());  
 }  
 }  
 }  
 public void getMostRecentEarthquake() throws IOException {  
  
 System.*out*.println("Enter the country for getting its record");  
 String country = sc.nextLine();  
 int index = step3.countryIndex(country);  
  
 while(index == -1){  
 System.*out*.println("Enter the valid country for getting its record");  
 country = sc.nextLine();  
 index = step3.countryIndex(country);  
 }  
  
 System.*out*.println("Most recent Earthquake in "+ mostRecentEQ.get(index));  
 }  
 public void problem4() {  
  
 System.*out*.println("All the Most Recent Earthquakes in all the countries above 6 Magnitude: \n");  
 double maxMag = 6.0;  
 for (int i = 0; i < mostRecentEQ.size(); i++) {  
 String data = mostRecentEQ.get(i).toString();  
 String[] read = data.split(": ");  
 double mag = Double.*parseDouble*(read[2]);  
 if(mag > maxMag){  
 System.*out*.println(data);  
 }  
 }  
 }  
}

**Algorithm:**

1. Declare and initialize a Step3 class object.
2. Create an array of stack of object type and store the step3 stack in it through “getStack” method.
3. Create a LinkedList of Object type.
4. In a method storeMostRecentEarthquake (), using a for loop peek the ith object from the stack and store it in LinkedList.
5. It will continue till the stack’s length and at the end we will have a LinkedList containing the most recent earthquake of all the countries.

**Output:**



**Note:**

* Methods in Step4 like getMostRecentEarthquake () will retrieve all the most recent earthquakes from any country, and “problem4()” will be used for Problem 4.

**Problem 1:**

How to find the average number of earthquakes per year for each country and which country is most vulnerable to earthquakes (which country has the most number of earth quakes)?

**Code:**

Click [here](#30j0zll) to get the average number of earthquakes per year for each country.

Click [here](#2et92p0) to get the country which is most vulnerable to earthquakes.

**Algorithm:**

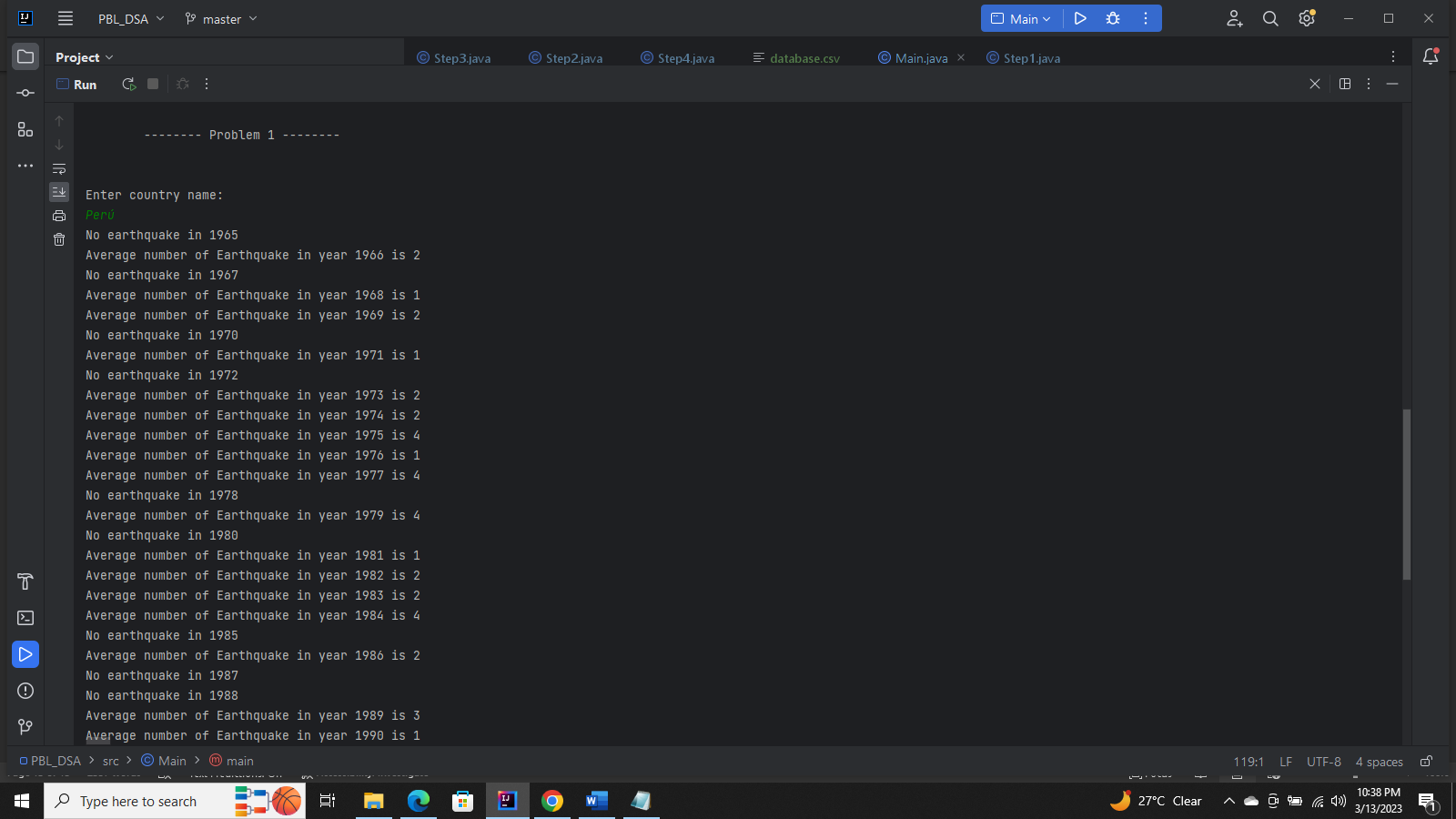
For average number of earthquakes per year for each country.

1. In the method [getAverageCountries](#30j0zll), user is asked to input a country.
2. Initialize a year variable with value equal to 1965.
3. In a for loop, which will continue till ArrayList’s array length, initialize a variable “numEQ” to count the number of earthquakes to 0.
4. Create a String array to store the data of ArrayList at ith index.
5. In a nested for loop, which will loop till the string array’s length, if the data at the jth index contains country, then “numEQ” will increment by 1.
6. In the outer for loop, the number of earthquakes will be printed for a particular year.
7. The year variable will increment.

For the country which is most vulnerable to earthquakes.

1. In the [countryWithMostNumberOfEQ](#2et92p0) method, initialize variables int num to 0 and a String country to “”.
2. The for loop will continue till stack’s length. Inside a for loop declare a size variable and give it value to the size of stack at ith index.
3. If size > num, then num will assign the value of size ad in the country at ith index will be stored in country.
4. Outside the for loop the country with number of earthquakes will be printed.

**Output:**

A screenshot of a computer

Description automatically generated

**Problem 2:**

Which are the biggest earthquakes from 2005 to 2015 and occurred and in which country (use step 2)?

**Code:**

Click [here](#1fob9te) to get code.

**Algorithm:**

1. The whole queue will be stored in an Earthquake type array.
2. The for loop will iterate till this array’s length.
3. Inside the for loop there is an of condition to check whether the earthquakes getting are between 2005 and 2015 years.
4. If condition is true, the corresponding data will be printed.

**Output:**

A screenshot of a computer

Description automatically generated

**Problem 3:**

How to determine the recent 5 earthquakes from each country?

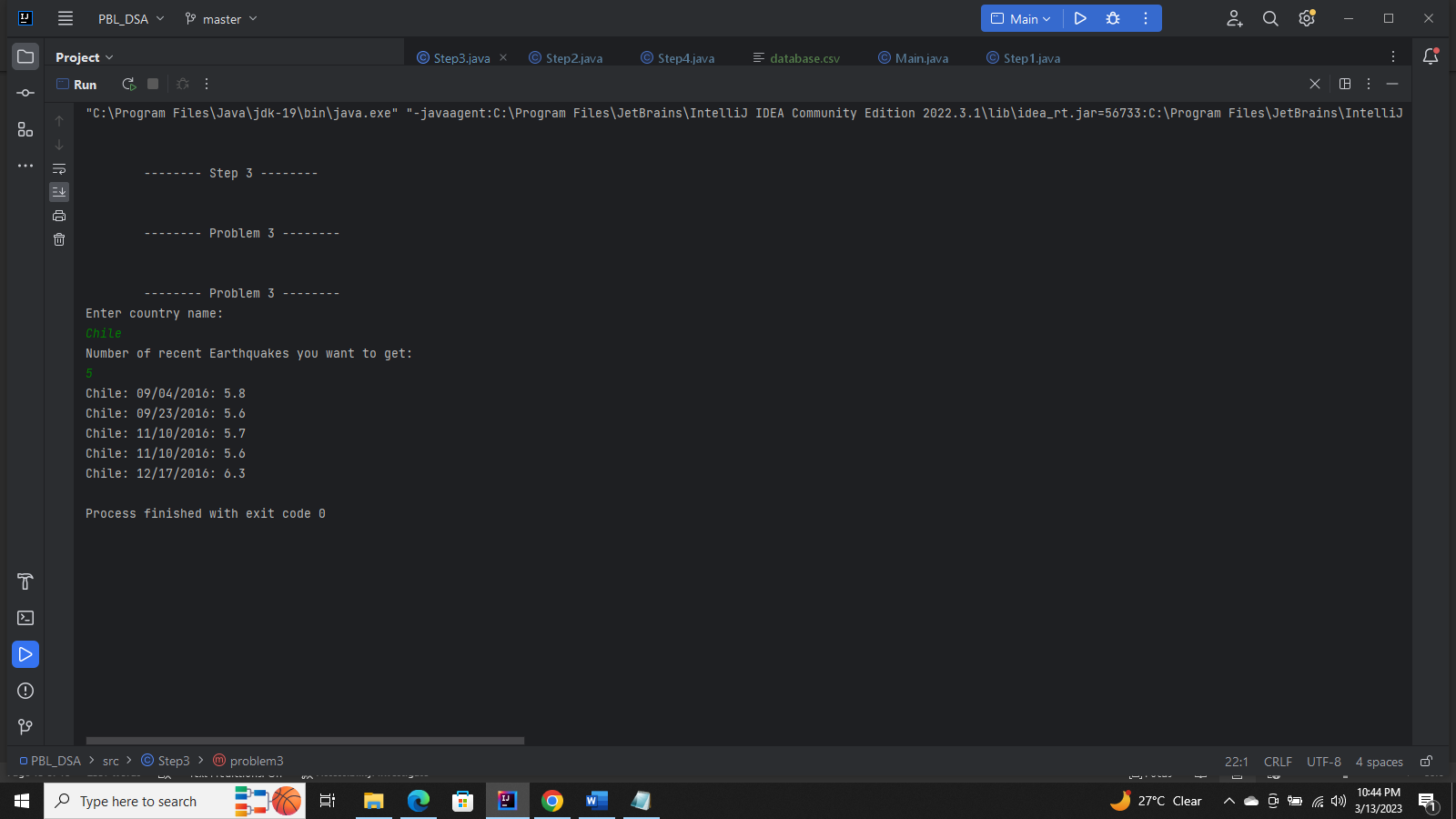
**Code:**

Click [here](#3znysh7) to get code.

**Algorithm:**

1. The user will be asked for entering a country, if that country exists than program continues else it will print out a statement to enter valid country.
2. It will then ask to know the number of recent earthquakes to be printed.
3. The stack at the same index at which the country is stored will convert to array.
4. The length variable will hold the value of number of earthquakes asked subtracted from the array’s length.
5. The for loop will iterate till this array’s length, printing the data.

**Output:**



**Problem 4:**

How to find the most recent above 6 magnitude earthquakes (use step 4).

**Code:**

Click [here](#tyjcwt) to get code.

**Algorithm:**

1. A variable stores tha magnitude 6 value.
2. The LinkedList will be converted to array at ith index inside a for loop.
3. If the magnitude at the at ith index is greater than the variable, then that earthquake will be printed.

**Output:**

A screenshot of a computer

Description automatically generated

**Main Class:**

This is the main class for all the classes.

public class Main {  
 public static void main(String[] args) throws Exception {  
  
 //-------Step1------  
 System.*out*.println("\n\n -------- Step 1 -------- ");  
 Step1 step1 = new Step1();  
 step1.storing();  
 step1.getYearlyEarthquakes();  
  
  
 //-------Step2------  
 System.*out*.println("\n\n -------- Step 2 -------- ");  
 Step2 step2 = new Step2();  
 step2.highestMagnitude();  
 step2.getYearlyMaxEarthquake();  
  
 //------Step3-------  
 System.*out*.println("\n\n -------- Step 3 -------- ");  
 Step3 step3 = new Step3();  
 step3.getAllRecentEarthquake();  
  
 //------Step4-------  
 System.*out*.println("\n\n -------- Step 4 -------- ");  
 Step4 step4 = new Step4();  
 step4.storeMostRecentEarthquake();  
 step4.getMostRecentEarthquake();  
   
 //------Problem 1-------  
 System.*out*.println("\n\n -------- Problem 1 -------- ");  
 step1.getAverageCountries();  
 step3.countryWithMostNumberOfEQ();  
  
 //------Problem 2-------  
 System.*out*.println("\n\n -------- Problem 2 -------- ");  
 step2.problem2();  
  
 //------Problem 3-------  
 System.*out*.println("\n\n -------- Problem 3 -------- ");  
 step3.problem3();  
  
 //------Problem 4-------  
 System.*out*.println("\n\n -------- Problem 4 -------- ");  
 step4.problem4();  
  
  
 }  
}