**NOTE: THIS ASSIGNMENT IS SEPARATED INTO TWO PARTS. SORTING ARRAYS & SORTING AND SEARCHING HASHMAP. THE CODE WILL NOT RUN PROPERLY IF THE ENTIRE WORD DOCUMENT IS COPY AND PASTED, THEY MUST BE RUN SEPARATELY.**

**PART ONE: SORTING ARRAYS**

import java.util.\*;  
  
*/\*PART ONE: Develop a program that asks the user to enter a capital for a U.S. state. Upon receiving the user input,  
the program reports whether the user input is correct. For this application, the 50 states and their capitals are  
stored in a two-dimensional array in order by state name. Display the current contents of the array  
then use a bubble sort to sort the content by capital. Next, prompt the user to enter answers for all the  
state capitals and then display the total correct count. The user's answer is not case-sensitive.\*/*

public class main {  
 public static void main (String[] args) {  
 *//For this application, the 50 states and their capitals are stored in a two-dimensional array in order by state name.* String[][] statesCapitals = {  
 {"Alabama", "Alaska", "Arizona", "Arkansas",  
 "California", "Colorado", "Connecticut", "Delaware",  
 "Florida", "Georgia", "Hawaii", "Idaho",  
 "Illinois", "Indiana", "Iowa", "Kansas",  
 "Kentucky", "Louisiana", "Maine", "Maryland",  
 "Massachusetts", "Michigan", "Minnesota", "Mississippi",  
 "Missouri", "Montana", "Nebraska", "Nevada", "New Hampshire",  
 "New Jersey", "New Mexico", "New York", "North Carolina",  
 "North Dakota", "Ohio", "Oklahoma", "Oregon",  
 "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota",  
 "Tennessee", "Texas", "Utah", "Vermont", "Virginia",  
 "Washington", "West Virginia", "Wisconsin", "Wyoming"},  
  
 {"Montgomery", "Juneau", "Phoenix", "Little Rock",  
 "Sacramento", "Denver", "Hartford", "Dover",  
 "Tallahassee", "Atlanta", "Honolulu", "Boise",  
 "Springfield", "Indianapolis", "Des Moines", "Topeka",  
 "Frankfort", "Baton Rouge", "Augusta", "Annapolis",  
 "Boston", "Lansing", "St. Paul", "Jackson",  
 "Jefferson City", "Helena", "Lincoln", "Carson City",  
 "Concord", "Trenton", "Santa Fe", "Albany",  
 "Raleigh", "Bismarck", "Columbus", "Oklahoma City",  
 "Salem", "Harrisburg", "Providence", "Columbia",  
 "Pierre", "Nashville", "Austin", "Salt Lake City",  
 "Montpelier", "Richmond", "Olympia", "Charleston",  
 "Madison", "Cheyenne"}  
 };  
  
 *//Develop a program that asks the user to enter a capital for a U.S. state.* System.*out*.println("Hello! Welcome to my Comp Sci 201 final assignment. Happy to have you here");  
 System.*out*.println("Enter any U.S. State Capital: ");  
  
 *//Adding scanner to accept user input.* Scanner find = new Scanner(System.*in*);  
 *//The nextLine method reads a STRING value from the user -- (in this case, any US capital)* String userInput = find.nextLine();  
  
 */\*Iterating through the capitals to see whether the user input a US capital;  
 \* Setting the boolean value so if our user is incorrect, we can print the else statement once  
 \* and not iterate through the entire array and print the incorrect statement 50 times\*/* boolean correct = true;  
 for (int i = 0; i < statesCapitals[1].length; i++) {  
 if (statesCapitals[1][i].equalsIgnoreCase(userInput)) {  
 System.*out*.println("Yes, " + userInput + " is a capital. Great job!");  
 correct = true;  
 break;  
 *//Need to break the loop if the user is incorrect, so we reset the boolean to exit the loop.* } else  
 correct = false;  
 }  
 *//If user is incorrect, we print this statement and still show them the capitals because we are nice like that.* if (!correct) {  
 System.*out*.println("Whoops, that is not a capital city. But I will show you the answers...");  
 }  
  
 */\*These, and following try catch statements, make the program pause for a moment so not everything is printed  
 \* at once... couldn't figure out how to hide them... \*/* try {  
 Thread.*sleep*(3200);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 System.*out*.println("The following are all of the US states and their capitals: ");  
  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 *//Display the contents of the array. For loop created to print out each capital and state.* for (int i = 0; i < statesCapitals[1].length; i++) {  
 System.*out*.println(statesCapitals[1][i] + " is the capital of " + statesCapitals[0][i]);  
 }  
  
  
 try {  
 Thread.*sleep*(3000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 *//Use bubble sort to sort content by capital* System.*out*.println("Now we will sort the states and capitals ordered alphabetically by capital.");  
 System.*out*.println("The output will be displayed horizontally.");  
  
  
 int n = statesCapitals[1].length;  
 */\*Nested for loop for bubble sort. We are comparing two adjacent values.  
 \* This takes the first value (outer loop) and compares it to every capital (inner loop). Then it takes  
 \* the next value (outer loop) and compares it to every capital again.\*/* for (int i = 0; i < n; i++) {  
 for (int j = i + 1; j < n; j++){  
 *//Initialize these variables for switching i and j. We need these within scope.* String tempCapital;  
 String tempState;  
  
 */\*We compare lexicographically, or alphabetically (A is less than B, B is greater than A);  
 \* i is one element, j is the next. If it is > 0, that means j comes first, and we switch the positions. \*/* if (statesCapitals[1][i].compareToIgnoreCase(statesCapitals[1][j]) > 0)  
 {  
 */\*Here we are switching i and j for the capitals.  
 \* Remember we need the temp variable to be able to reassign values in the array. The temp variable  
 \* is a holding spot for either i or j to move to while we are switching the elements.\*/* tempCapital = statesCapitals[1][i];  
 statesCapitals[1][i] = statesCapitals[1][j];  
 statesCapitals[1][j] = tempCapital;  
  
 *//We need to keep the capitals with the states, so we reassign that order as well.* tempState = statesCapitals[0][i];  
 statesCapitals[0][i] = statesCapitals[0][j];  
 statesCapitals[0][j] = tempState;  
 }  
 }  
 }  
 *//Print out the contents of the sorted array.* System.*out*.println("Sorted Array: " + Arrays.*deepToString*(statesCapitals));  
  
 try {  
 Thread.*sleep*(6000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
  
 */\*Next, prompt the user to enter answers for all the state capitals and then  
 \* display the total correct count. The user's answer is not case-sensitive.\*/* System.*out*.println("Now we will see how many capitals you can name.");  
 System.*out*.println("When you are done, type stop.");  
  
 */\* Setting up our scanner to accept user input. Creating our arraylist to store user entries. Creating  
 \* another array of only capitals, so we can iterate. Initializing counter variable. Initializing  
 \* boolean variable for flow control of while loop\*/* Scanner keyboard = new Scanner(System.*in*);  
 ArrayList<String> gameList = new ArrayList<>();  
 String [] capitals = statesCapitals[1];  
 int counter = 0;  
 boolean gameStop = false;  
  
 *//While loop to accept user input and determines whether that input equals an element in the sub-array of capitals.* while(!gameStop) {  
 String userEntry = keyboard.nextLine();  
  
 if (userEntry.equalsIgnoreCase("stop")) {  
 gameStop = true;  
 System.*out*.println("Great job with your entries... drum roll please...");  
  
 } else {  
 gameList.add(userEntry);  
 }  
 }  
 */\*Nested for loop compares user's guess to the sub-array. If one matches, we add one to the counter variable.  
 \* This counter variable will be what we use to print the number of correct guesses. \*/* for(String userGuess : gameList){  
 for (String capital : capitals){  
 if(userGuess.equalsIgnoreCase(capital))  
 counter +=1;  
 }  
 }  
 *//Displaying the total correct count.* System.*out*.println("Your final score: " + counter);  
 }  
}

**PART TWO: SORTING AND SEARCHING HASHMAP**

import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
import java.util.TreeMap;  
  
*/\* PART TWO: Now revise the code to store the pairs of each state and its capital in a Map using the HashMap function.  
Display the content of the Map, then use the TreeMap class to sort the map while using a binary search tree for storage.  
Next, your program should prompt the user to enter a state, and it should then display the capital for the state.\*/*public class main {  
 public static void main(String[] args) {  
 String[][] statesCapitals = {  
 {"Alabama", "Alaska", "Arizona", "Arkansas",  
 "California", "Colorado", "Connecticut", "Delaware",  
 "Florida", "Georgia", "Hawaii", "Idaho",  
 "Illinois", "Indiana", "Iowa", "Kansas",  
 "Kentucky", "Louisiana", "Maine", "Maryland",  
 "Massachusetts", "Michigan", "Minnesota", "Mississippi",  
 "Missouri", "Montana", "Nebraska", "Nevada", "New Hampshire",  
 "New Jersey", "New Mexico", "New York", "North Carolina",  
 "North Dakota", "Ohio", "Oklahoma", "Oregon",  
 "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota",  
 "Tennessee", "Texas", "Utah", "Vermont", "Virginia",  
 "Washington", "West Virginia", "Wisconsin", "Wyoming"},  
  
 {"Montgomery", "Juneau", "Phoenix", "Little Rock",  
 "Sacramento", "Denver", "Hartford", "Dover",  
 "Tallahassee", "Atlanta", "Honolulu", "Boise",  
 "Springfield", "Indianapolis", "Des Moines", "Topeka",  
 "Frankfort", "Baton Rouge", "Augusta", "Annapolis",  
 "Boston", "Lansing", "St. Paul", "Jackson",  
 "Jefferson City", "Helena", "Lincoln", "Carson City",  
 "Concord", "Trenton", "Santa Fe", "Albany",  
 "Raleigh", "Bismarck", "Columbus", "Oklahoma City",  
 "Salem", "Harrisburg", "Providence", "Columbia",  
 "Pierre", "Nashville", "Austin", "Salt Lake City",  
 "Montpelier", "Richmond", "Olympia", "Charleston",  
 "Madison", "Cheyenne"}  
 };  
  
  
 *// Now revise the code to store the pairs of each state and its capital in a Map using the HashMap function.* HashMap<String, String> hMap = new HashMap<String, String>(statesCapitals[1].length);  
  
 *//For loop to assign elements to HashMap.* for (int i = 0; i < statesCapitals[1].length; i++) {  
 hMap.put(statesCapitals[0][i], statesCapitals[1][i]);  
 }  
  
 *//Display the content of the HashMap. Setting the key and value of the HashMap to equal the states/capitals.* for(Map.Entry<String, String> mapElement : hMap.entrySet()) {  
 String key = mapElement.getKey();  
  
 String value = (mapElement.getValue());  
  
 System.*out*.println(key + " : " + value);  
 }  
  
 System.*out*.println("Now, you can enter a state. After your entry, the program will tell you the capital. " +  
 "Type stop to stop the program.");  
 System.*out*.println("\*\*Please enter the state with the first letter capitalized\*\* (ex. Texas, New York, Ohio):");  
  
 try {  
 Thread.*sleep*(6000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 */\*Use the TreeMap class to sort the map while using a binary search tree for storage.  
 \* NOTE: The treemap is a binary sort tree itself. \*/* TreeMap<String, String> tMap = new TreeMap<>(hMap);  
  
 *//Initialize scanner for user input and boolean for while loop flow control.* Scanner keyboard = new Scanner(System.*in*);  
 boolean treeStop = false;  
  
 */\*Checking user input to compare against the treemap using the containsKey() method.  
 If user enters an actual state, the program will tell them the capital.\*/* while (!treeStop) {  
 String userEntry = keyboard.nextLine();  
 if (tMap.containsKey(userEntry)){  
 System.*out*.println(tMap.get(userEntry) + " is the capital of " + userEntry);}  
 *//User can type stop to end the program and exit the loop.* else if (userEntry.equalsIgnoreCase("stop")){  
 treeStop = true;  
 System.*out*.println("Goodbye. Thank you for grading my assignment. Hope you have a nice day.");  
 continue;  
 }  
 }  
 }  
}