// Kelvin Kellner

// Mrs. Cooper

// June 4th, 2019

// ICS 4UI Culminating Coding Challenges

// Day 2 - How Cold Is It?

import java.io.\*;

import java.util.Arrays;

// City Class

class City implements Comparable<City>

{

public int temperature;

public String name;

// Comparable method is used for sorting of objects automatically using "Array.sort()".

// Sort based on temperature first (lowest to highest), then alphabetically (ascending).

@Override

public int compareTo(City that) {

int tempResult = Integer.compare(this.temperature, that.temperature);

if(tempResult == 0)

return this.name.compareTo(that.name);

else

return tempResult;

}

}

// Main Class

public class Day2HowColdIsIt

{

public static void main(String[] args) throws IOException

{

String fileNameIn = "citytemp.txt";

String fileNameOut = "findcoldest.txt";

try

{

// File Input and Data Storage

int lines = countLines(fileNameIn);

City[] cities = new City[lines/2];

FileReader fileIn = new FileReader(fileNameIn);

BufferedReader read = new BufferedReader(fileIn);

for(int i=0;i<lines/2;i++)

{

// Process each city and store it in the array

cities[i] = new City();

try

{

cities[i].temperature = Integer.parseInt(read.readLine());

cities[i].name = read.readLine();

}

catch(Exception e)

{

System.out.println("Error.\nTemperature and name data could not be processed for city #" + i + ".\nEnsure that the temperature is written as an integer.\n\nMessage...\n");

e.printStackTrace();

}

}

read.close();

// Sort the cities (using comparable!)

Arrays.sort(cities);

try

{

// Output the 3 coldest

FileWriter fileOut = new FileWriter(fileNameOut);

PrintWriter write = new PrintWriter(fileOut);

write.println("COLD");

write.println(cities[2].name);

write.println("COLDER");

write.println(cities[1].name);

write.println("COLDEST");

write.println(cities[0].name);

write.close();

System.out.println("\nDONE.\n\nSee \"" + fileNameOut + "\" for results.");

}

catch(Exception e)

{

System.out.println("Error.\nCould not save to file.\nCheck that you are attempting to save in the correct location.\n\nMessage...\n");

e.printStackTrace();

}

}

catch(Exception e)

{

System.out.println("Error.\nCould not open or process the file.\nCheck that the file name matched that at the top of the program.\n\nMessage...\n");

e.printStackTrace();

}

}

// Returns the number of lines of text found within in a given text file

public static int countLines(String fileName) throws IOException

{

FileReader fileIn = new FileReader(fileName);

BufferedReader read = new BufferedReader(fileIn);

int count = 0;

while(read.readLine()!=null)

count++;

read.close();

return count;

}

}