[redacted]
// 11.24.2020
// CSE 142 Section AH
// TA: Emilia Borisova
// Personality Test
// This program creates a Keirsey personality type Test. First, it reads given
// input file and gather information on A(ESTJ) or B(INFP). After that, it
// calculates the B percentage of responses for each dimension. Final, it
// prints out to a output file with names, answer percentages, and personality types.
```
import java.util.*;
import java.io.*;

public class Personality {

	// sets the class constant to represent the number of dimensions
	public static final int dimensions = 4;

	// Builds the structure of the program.
	public static void main(String[] args)throws FileNotFoundException {
		Scanner console = new Scanner(System.in);

		// prints the introduction and asks for name of input file and output file
		introduction();
		Scanner input = inputFile(console);
		PrintStream output = outputFile(console);

		// Makes the output file for every person's percent and personality type
		boolean file = true;
		while(input.hasNextLine()) {
			String name = input.nextLine();
			int[] percent = countArray(name);
			String type = distributeType(percent);
			if(file) {
				output.print(name + ": ");
				file = false;
			} else {
				output.print(Arrays.toString(percent) + " = " + type);
				file = true;
				output.println();
		    }
		}
	}

   // Prints out the short introduction of the program.
   public static void introduction() {
		System.out.println("This program processes a file of answers to the");
		System.out.println("Keirsey Temperament Sorter. It converts the");
		System.out.println("various A and B answers for each person into");
		System.out.println("a sequence of B-percentages and then into a");
		System.out.println("four-letter personality type.");
		System.out.println();
   }

   // Asks users the name of the input file and looks for it.
```

```
        // Returns the input file that was typing by the user.
        // Parameters:
        //    Scanner console - the Scanner to use for input
    public static Scanner inputFile(Scanner console) throws FileNotFoundException {
            System.out.print("input file name? ");
            Scanner input = new Scanner(new File(console.nextLine()));
            return input;
    }

        // Asks users the name of the output file and creates it.
        // Returns the output file that was typing by the user.
        // Parameters:
        //    Scanner console - the Scanner to use for input
    public static PrintStream outputFile(Scanner console) throws FileNotFoundException {
            System.out.print("output file name? ");
            PrintStream output = new PrintStream(console.nextLine());
            return output;
    }

        // Counts the number of A and B from each person's response of test. After that,
        // Calculates the B percentage of responses they gave for each dimension.
        // Returns an array of B percentages from each dimension.
    // Parameters:
        //    String line - read the text from input file
    public static int[] countArray(String line) {
            int[] numberA = new int[dimensions];
            int[] numberB = new int[dimensions];

            for(int i = 0; i < line.length(); i++) {
                    if(line.charAt(i) == 'A' || line.charAt(i) == 'a' ) {
                            numberA[(i % 7 + 1) / 2]++;
                    } else if (line.charAt(i) == 'B' || line.charAt(i) == 'b') {
                            numberB[(i % 7 + 1) / 2]++;
                    }
            }
            int[] percent = calculatePercent(numberA, numberB);
                  return percent;
    }

    // Calculates a person's percentage of B answer from the total answer of A and B.
        // Return the total percentage of B answers.
    // Parameters:
        //    int[] numberA - the array of answer A
        //    int[] numberB - the array of answer B
    public static int[] calculatePercent(int[] numberA, int[] numberB) {
            int[] percent = new int[dimensions];

            for(int i = 0; i < dimensions; i++) {
                    int answer = (int)Math.round(((double)numberB[i] / (numberA[i] + numberB[i])) *
100);
                    percent[i] = answer;
            }
            return percent;
    }
```

```
    // Distributes the personality type based on the precent of B answer.
        // less than 50 percent is assigned with "ESTJ"
        // greater than 50 percent is assigned with "INFP"
        // Returns the result of personality type of the person
        // Parameters:
        //    int[] percent - the total B percentage Calculated from array
    public static String distributeType(int[] percent) {
            String typeResult = "";
            String typeA = "ESTJ";
            String typeB = "INFP";

            for(int i = 0; i < dimensions; i++) {
                if(percent[i] < 50) {
                        typeResult = typeResult + typeA.charAt(i);
                } else if(percent[i] > 50) {
                    typeResult = typeResult + typeB.charAt(i);
                } else {  //percent[i] = 50
                        typeResult = typeResult + "X";
                }
            }
            return typeResult;
    }
}
```