```java
import java.io.*;
import java.util.*;

public class PersonalityTestsShell
{
    //class constants
    // The dimensions are the 4 categories
    public static final int DIMENSIONS = 4;

    public static final String set1 = "1 8 15 22 29 36 43 50 57 64";                               // Introvert/Extrovert
    public static final String set2 = "2 3 9 10 16 17 23 24 30 31 37 38 44 45 51 52 58 59 65 66";  // Sensing/Intuition
    public static final String set3 = "4 5 11 12 18 19 25 26 32 33 39 40 46 47 53 54 60 61 67 68"; // Thinking/Feeling
    public static final String set4 = "6 7 13 14 20 21 27 28 34 35 41 42 48 49 55 56 62 63 69 70"; // Judging/Perceiving

    public static void main(String[] args) throws FileNotFoundException {
        Scanner console = new Scanner(System.in);
        Scanner input = getInput(console);
        PrintStream out = getOutput(console);
        process(input, out);
    }

    /*This method prompts the user for a file name that holds the information for this program
    your program must keep asking for a file name that exists in the current directory console
    Scanner until the user gives a legal file name. this method returns the file object*/
    // Type in a file name to read in the A's and B's. In this case we're using personality.txt
    public static Scanner getInput(Scanner console) throws FileNotFoundException {
        System.out.println("Enter the output file name: ");
        String outputFile = console.nextLine();
        return new Scanner(new PrintStream(outputFile));
    }

    /* This method prompts the user for an output file name so
     the result of this program can be written to it.
     This method will keep prompting the user until we get a valid file name and returns the file object  */

    // Ask the user for a filename to save the output to. In this case, we're using the name, result.txt
    public static PrintStream getOutput(Scanner console) throws FileNotFoundException {
        System.out.print("Output file name: ");
        String name = console.nextLine();
        return new PrintStream(new File(name));
    }

    /* This method process the file that contains  the information.
    the parameters to this method are the input file object and the output file object.
    this method reads the information for each person. the first line holds he name of
    the person and the next line holds the user's answers to the personality questions.
    after reading the answers to the questions for the person, methods count and percent and personalityType
    must be called in this method*/
    public static void process(Scanner input, PrintStream out)
    {

    }

    // Fills the aCount and bCount arrays with counts of how many
    // As and Bs this person answered in each dimension.
    // the methods isInTheSet and output needs to be called in this method
    public static void count(PrintStream out, String answers, int[] aCount, int[] bCount)
    {
        // countA[] and countB[] are arrays containing the counts for each dimension.
        // countA[]  -> [0, 0, 0, 0]
        // DIMENSION:    1  2  3  4
        // countB[]  -> [0, 0, 0, 0]

        // We need to figure out which dimension to increase the count for
        // We have a string of answers.
        // We need to get each letter, or answer, one at a time

        // Step 1: Get each letter from the answer string
        //         Use a for loop to go through the answer string
        // Step 2: For each question, figure out which set, or dimension, it comes from
        //         This is where we use isInTheSet(). We use that method to help us figure that out
        //         if the question is in first  set, isInTheSet(1), then we increment set 1
        //         if the question is in second set, isInTheSet(2), then we increment set 2
        //         if the question is in third  set, isInTheSet(3), then we increment set 3
        //         if the question is in fourth set, isInTheSet(4), then we increment set 4
        // Step 3: Check if the answer is an 'A' or a 'B'
        //         if the answer is an 'A', increment countA
        //         if the answer is a  'B', increment countB
        //         if the answer is a  '-', don't do anything
        // Step 4: Call output(out, aCount, bCount);
        //         This will save the data from aCount and bCount to the output file, result.txt
    }

    /* This method outputs the content of the array aCount
    and bCount to the output file. refer to the provided output file*/
    public static void output(PrintStream out, int[] aCount, int[] bCount)
    {

    }

    // This method finds if the question number
    // being processed is in the given set or not
    public static boolean inTheSet(String set, int num)
    {

        return false;
    }

    // uses the given A and B count arrays to compute and return
    // a new array of the percentage of B answers out of the (A+B) total
    // for each dimension
    public static int[] percents(PrintStream out, int[] aCount, int[] bCount) {
        return null;
    }

    // returns the personality type that corresponds to the given
    // percentages, such as "INTJ"
    public static void personalityType(PrintStream out, int[] percents)
    {
    }
}
```