

```

/**
 * CT255 - Assignment 4
 * Skeleton code for Steganography assignment.
 *
 * @author Michael Schukat
 * @version 1.0
 */
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Steganol {
    /**
     * Constructor for objects of class Steganol
     */
    public Steganol() {
    }

    /**
     * Oisin:
     * 01001111 01101001 01110011 01101001 01101110
     * Mc Laughlin:
     *
     * 010011010110001101001100011000010111010101100111011010000110110001101001011
     * 01110
     *
     * Removed spaces for problem 2 because it wouldn't work, spent 2 hours
     * trying to fix it but it wouldn't work :(
     */

    /**
     * 00 = 1 Space
     * 01 = 2 Spaces
     * 10 = 3 Spaces
     * 11 = 4 Spaces
     */

    public static void main(String[] args) {
        String arg1, arg2, arg3, arg4;
        boolean err = false;
        if (args != null && args.length > 1) { // Check for minimum number
of arguments
            arg1 = args[0];
            arg2 = args[1];
            if (arg2.isEmpty()) {
                err = true;
            } else if ((arg1.equals("A")) && (args.length > 3)) {
                // Get other arguments
                arg3 = args[2];
                arg4 = args[3];
                if ((arg3.isEmpty()) || (arg4.isEmpty())) {
                    err = true;
                } else {
                    // Hide bitstring
                    hide(arg2, arg3, arg4);
                }
            } else if (arg1.equals("E")) {
                // Extract bitstring from text
                retrieve(arg2);
            }
        }
    }
}

```

```

        } else {
            err = true;
        }
    } else {
        err = true;
    }
    if (err) {
        System.out.println();
        System.out.println("Use: Stegano1 <A:E><Input
File><OutputFile><Bitstring>");
        System.out.println("Example: Stegano1 A inp.txt out.txt
0010101");
        System.out.println("Example: Stegano1 E inp.txt");
    }
}

static void hide(String inpFile, String outFile, String binString) {
    BufferedReader reader;
    BufferedWriter writer;
    try {
        reader = new BufferedReader(new FileReader(inpFile));
        writer = new BufferedWriter(new FileWriter(outFile));
        String line = reader.readLine();

        if (binString.length() % 2 != 0) {
            binString += "0";
        }

        int i = 0;
        while (line != null) {

            // Your code starts here
            // Store amended line in output file
            if (i < binString.length()) {
                /*
                 //Problem 1
                 //Get current bit as character as '0' or '1'
                 char bit = binString.charAt(i);

                 if (bit == '0') {
                     line += " "; // Append one space for '0'
                 } else if (bit == '1') {
                     line += "  "; // Append two spaces for '1'
                 }
                 */

                //Problem 2
                // Get two bits at a time
                String bits = binString.substring(i, i + 2);

                String encodedSpaces = "";

                //Debugging output
                System.out.println("Encoding bits: " + bits + " on
line: " + line);
                System.out.println("Adding encoded spaces: " +
encodedSpaces + "");

                if (bits.equals("00")) {

```

```

        line += " "; // 1 space for "00"
    } else if (bits.equals("01")) {
        line += "  "; // 2 spaces for "01"
    } else if (bits.equals("10")) {
        line += "   "; // 3 spaces for "10"
    } else if (bits.equals("11")) {
        line += "    "; // 4 spaces for "11"
    }

    }

//          if (i < binString.length()) {
//              System.out.println("Not all bits were encoded. Ran
out of lines in the input file.");
//          }

        writer.write(line);
        writer.newLine();
        // read next line
        line = reader.readLine();

        //Problem 1
        i++;

        //Problem 2
        i += 2;
    }
    reader.close();
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

static void retrieve(String inpFile) {
    BufferedReader reader;
    try {
        reader = new BufferedReader(new FileReader(inpFile));
        String line = reader.readLine();

        String result = "";
        while (line != null) {

            // Your code starts here
            // System.out.println(line);
            // read next line

            /*
            //Problem 1
            // If line ends with two spaces, append '1' to result
            if (line.endsWith("  ")) {
                result += "1";
            }
            // If line ends with one space, append '0' to result
            else if (line.endsWith(" ")) {
                result += "0";
            }
            // If line does not end with space, append nothing
            else {

```

```

        result += " ";
    }
    */

    //Problem 2
    //Trim line to remove white spaces.
    line = line.trim();

    if (line.endsWith("    ")) {
        result += "11"; //4 Spaces, add 11 to result
    } else if (line.endsWith("   ")) {
        result += "10"; //3 Spaces, add 10 to result
    } else if (line.endsWith("  ")) {
        result += "01"; //2 Spaces, add 01 to result
    } else if (line.endsWith(" ")) {
        result += "00"; //1 Spaces, add 00 to result
    }
    //
    //
    //
    else {
        result += " ";
    }

    line = reader.readLine();
}
//Problem 1
//System.out.println("Expected Output: \n01001111 01101001
01110011 01101001 01101110\n");

//Problem 2
System.out.println("Expected Output:
\n0100110101100011010011000110000101110101011001110110100001101100011010010
1101110\n");

System.out.println("Actual Output: \n" + result);

    reader.close();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```