```
package assembler;

import java.util.*;

public class BinaryTable {

    // C-instruction   ->   [destination] = [computation] ; [jump]
    private final HashMap<String,String> compTable = new HashMap<>();  // Computation 
table for a = 0.
    private final HashMap<String,String> compATable = new 
HashMap<>(); //                    for a = 1.
    private final HashMap<String,String> destTable = new HashMap<>();  // Destination 
table
    private final HashMap<String,String> jumpTable = new HashMap<>();  // Jump table

    public BinaryTable() {
        compTable.put("0", "101010");
        compTable.put("1", "111111");
        compTable.put("-1", "111010");
        compTable.put("D", "001100");
        compTable.put("A", "110000");
        compTable.put("!D", "001101");
        compTable.put("!A", "110001");
        compTable.put("-D", "001111");
        compTable.put("-A", "110011");
        compTable.put("D+1", "011111");
        compTable.put("A+1", "110111");
        compTable.put("D-1", "001110");
        compTable.put("A-1", "110010");
        compTable.put("D+A", "000010");
        compTable.put("D-A", "010011");
        compTable.put("A-D", "000111");
        compTable.put("D&A", "000000");
        compTable.put("D|A", "010101");

        compATable.put("M", "110000");
        compATable.put("!M", "110001");
        compATable.put("-M", "110011");
        compATable.put("M+1", "110111");
        compATable.put("M-1", "110010");
        compATable.put("D+M", "000010");
        compATable.put("D-M", "010011");
        compATable.put("M-D", "000111");
        compATable.put("D&M", "000000");
        compATable.put("D|M", "010101");

        destTable.put("NULL", "000");
        destTable.put("M", "001");
        destTable.put("D", "010");
        destTable.put("MD", "011");
        destTable.put("A", "100");
        destTable.put("AM", "101");
        destTable.put("AD", "110");
        destTable.put("AMD", "111");

        jumpTable.put("NULL", "000");
        jumpTable.put("JGT", "001");
        jumpTable.put("JEQ", "010");
        jumpTable.put("JGE", "011");
        jumpTable.put("JLT", "100");
        jumpTable.put("JNE", "101");
        jumpTable.put("JLE", "110");
        jumpTable.put("JMP", "111");
    }

    public String comp(String comp) { // Returns the binary code of a computation comp
(acccccc)
        if (comp.contains("M")) return "1" + compATable.get(comp);
        return "0" + compTable.get(comp);
    }

    public String dest(String dest) {
        return destTable.get(dest);
    }

    public String jump(String jump) {
        return jumpTable.get(jump);
    }

}
```