```java
import java.util.HashMap;
import java.util.Map;

public class Code
{
  /** Returns the binary representation of the parsed dest field (string) (3 bits)*/
  public static String dest(String mnemonic)
  {
    //ex: dest("DM") returns "011"
    return HashMap_DEST.get(mnemonic);  }

  /** Returns the binary representation of the parsed comp field (string) (7bits)*/
  public static String comp(String mnemonic)
  {
    //ex: comp("A+1") returns "0110111" or comp("D&M") returns "1000000"
    return HashMap_COMP.get(mnemonic);  }

  /** Returns the binary representation of the parsed jump field (string) (3 bits)*/
  public static String jump(String mnemonic)
  {
    //ex: jump("JNE") returns "101"
    return HashMap_JUMP.get(mnemonic);
  }

  private static final Map<String, String> HashMap_DEST = new HashMap<>()
  {
    // According to the language specification:
    {
      put(null, "000");
      put("M", "001");
      put("D", "010");
      put("MD", "011");
      put("A", "100");
      put("AM", "101");
      put("AD", "110");
      put("AMD", "111");
    }
  };
  private static final Map<String, String> HashMap_COMP = new HashMap<>()
  {
    // According to the language specification:
    {
      //a=0
      put("0", "0101010");
      put("1", "0111111");
      put("-1", "0111010");
      put("D", "0001100");
      put("A", "0110000");
      put("!D", "0001101");
      put("!A", "0110001");
      put("-D", "0001111");
      put("-A", "0110011");
      put("D+1", "0011111");
      put("A+1", "0110111");
      put("D-1", "0001110");
      put("A-1", "0110010");
      put("D+A", "0000010");
      put("D-A", "0010011");
      put("A-D", "0000111");
```

```
      put("D&A", "0000000");
      put("D|A", "0010101");
      //a=1
      put("M", "1110000");
      put("!M", "1110001");
      put("-M", "1110011");
      put("M+1", "1110111");
      put("M-1", "1110010");
      put("D+M", "1000010");
      put("D-M", "1010011");
      put("M-D", "1000111");
      put("D&M", "1000000");
      put("D|M", "1010101");
    }
  };
  private static final Map<String, String> HashMap_JUMP = new HashMap<>()
  {
    // According to the language specification:
    {
      put(null, "000");
      put("JGT", "001");
      put("JEQ", "010");
      put("JGE", "011");
      put("JLT", "100");
      put("JNE", "101");
      put("JLE", "110");
      put("JMP", "111");
    }
  };

}

```