```
1 import java.util.HashMap;
 2 import java.util.Map;
 3
 4 public class Code
5 {
     /** Returns the binary representation of the parsed dest field (string) (3 bits)*/
 7
     public static String dest(String mnemonic)
 8
9
       //ex: dest("DM") returns "011"
       return HashMap_DEST.get(mnemonic); }
10
11
     /** Returns the binary representation of the parsed comp field (string) (7bits)*/
12
     public static String comp(String mnemonic)
13
14
     {
       //ex: comp("A+1") returns "0110111" or comp("D&M") returns "1000000"
15
16
       return HashMap_COMP.get(mnemonic); }
17
     /** Returns the binary representation of the parsed jump field (string) (3 bits)*/
18
19
     public static String jump(String mnemonic)
20
       //ex: jump("JNE") returns "101"
21
22
       return HashMap_JUMP.get(mnemonic);
23
24
25
     private static final Map<String, String> HashMap_DEST = new HashMap<>()
26
27
       // According to the language specification:
28
       {
          put(null, "000");
29
         put("M", "001");
put("D", "010");
30
31
         put("MD", "011");
put("A", "100");
32
33
         put("AM", "101");
34
         put("AD", "110");
35
         put("AMD", "111");
36
37
       }
38
     };
39
     private static final Map<String, String> HashMap_COMP = new HashMap<>()
40
41
       // According to the language specification:
42
43
         //a=0
         put("0", "0101010");
44
         put("1", "0111111");
45
         put("-1", "0111010");
46
         put("D", "0001100");
47
         put("A", "0110000");
48
         put("!D", "0001101");
49
         put("!A", "0110001");
put("-D", "0001111");
50
51
          put("-A", "0110011");
52
53
         put("D+1", "0011111");
         put("A+1", "0110111");
put("D-1", "0001110");
put("A-1", "0110010");
54
55
56
         put("D+A", "0000010");
put("D-A", "0010011");
57
58
59
          put("A-D", "0000111");
```

localhost:4649/?mode=clike 1/2

```
put("D&A", "0000000");
put("D|A", "0010101");
60
61
62
                  //a=1
                  put("M", "1110000");
63
                  put("!M", "1110001");
64
                 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");
65
66
67
68
69
70
71
72
              }
73
74
          };
75
          private static final Map<String, String> HashMap_JUMP = new HashMap<>()
76
77
              // According to the language specification:
78
              {
                  put(null, "000");
79
                 put(null, "000");
put("JGT", "001");
put("JEQ", "010");
put("JGE", "011");
put("JLT", "100");
put("JNE", "101");
put("JLE", "110");
put("JMP", "111");
80
81
82
83
84
85
86
87
              }
88
          };
89
90 }
91
```

localhost:4649/?mode=clike 2/2