```
2 import java.io.*;
 3 import java.util.*;
 5 class Operator
 6 {
 7
     String name;
8
     String cls;
9
     int opcode;
10
     Operator(String a, String c, int op)
11
12
       this.name = a;
13
       this.cls = c;
14
       this.opcode = op;
15
16 }
17
18 class Register
19 {
20
     String name;
21
     int no;
22
     Register(String a, int op)
23
24
       this.name = a;
25
       this.no = op;
26
27 }
28
29
30 class Condition
31 {
32
     String name;
33
     int no;
34
     Condition(String a, int op)
35
36
       this.name = a;
37
       this.no = op;
38
39 }
40
41
42 class Symbol
43 {
44
     String name;
45
     int addr;
46
     int length;
47
     Symbol(String a, int op, int len)
48
49
       this.name = a;
50
       this.addr = op;
51
       this.length = len;
52
53 }
54
55
56
57 public class assembler_pass2
58 {
59
     public static void main(String[] args) throws IOException
60
     {
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
61
 62
63
        BufferedReader objReader = new BufferedReader(new
    FileReader("pass1.txt"));
64
        String strCurrentLine;
65
 66
        //Making the symbol tables
        ArrayList<Symbol> symboltable = new ArrayList<Symbol>(25);
67
68
69
        //Accepting the symbol table
 70
        int cont = 1;
 71
        while(cont == 1)
 72
 73
        System.out.println("Enter name: ");
 74
        String s = br.readLine();
75
        System.out.println("Enter address: ");
 76
        int ad = Integer.parseInt(br.readLine());
 77
        System.out.println("Enter size: ");
 78
        int size = Integer.parseInt(br.readLine());
 79
 80
        //Adding the symbol
 81
82
        symboltable.add(new Symbol(s,ad,size));
83
84
        System.out.println("Add another? ");
85
        cont = Integer.parseInt(br.readLine());
 86
87
        }
88
89
        //Displaying the symbol table
90
        Iterator<Symbol> display = symboltable.iterator();
91
        Symbol x;
92
        while(display.hasNext())
93
94
          x = display.next();
          System.out.println(x.name + " " + x.addr + " " + x.length);
95
96
97
98
        //Creating the output files
99
        FileWriter fw = new FileWriter("pass2.txt");
100
        BufferedWriter write = new BufferedWriter(fw);
101
        int first = 1;
102
        while ((strCurrentLine = objReader.readLine()) != null)
103
        {
104
          if(first == 1)
105
          {
106
            first = 0;
107
          }
          else
108
109
            String[] splited = strCurrentLine.split("\\s+");
110
111
            int len = splited[1].length();
            String[] inst = splited[1].substring(1,len-1).split(",");
112
            if(inst[0].equals("AD") && (inst[1].equals("1") ||
113
    inst[1].equals("2")))
114
            {
115
              //skip
            }
116
117
            else
118
            {
```

```
119
              String pass2line = "";
120
              pass2line = pass2line+splited[0] + " + ";
              if(inst[0].equals("IS"))
121
122
123
                if(inst[1].equals(0))
124
                  pass2line = pass2line +"00 0 000" + " ";
125
126
                }
127
                else
128
129
                  pass2line = pass2line + inst[1] + " ";
130
                  int len2 = splited[2].length();
131
                  String[] arg = splited[2].substring(1,len2-1).split(",");
132
                  if(arg[0].equals("S"))
133
                    //Fetching from symbol table
134
                    Symbol s = symboltable.get(Integer.parseInt(arg[1]));
135
                    pass2line = pass2line + s.addr + " ";
136
137
                  }
138
                  else
139
                  {
140
                    // Register or condition
141
                    pass2line = pass2line+ arg[0] + " ";
142
143
144
                  len2 = splited[3].length();
                  arg = splited[3].substring(1,len2-1).split(",");
145
146
                  if(arg[0].equals("S"))
147
148
                    //Fetching from symbol table
149
                    int index = Integer.parseInt(arg[1])-1;
150
                    Symbol s = symboltable.get(index);
151
                    pass2line = pass2line + s.addr + " ";
152
                  }
153
                  else
154
155
                    // Register or condition
156
                    pass2line = pass2line+ arg[0] + " ";
157
                  }
                }
158
159
160
161
162
163
              else if(inst[0].equals("DL") && inst[1].equals("1"))
164
165
                pass2line = pass2line + "00 0 ";
166
                int len3 = splited[2].length();
                String[] arg = splited[2].substring(1,len3-1).split(",");
167
                pass2line = pass2line + arg[1] + " ";
168
169
170
              write.write(pass2line);
171
              write.newLine();
172
            }
          }
173
174
175
        }
176
177
       write.close();
178
```

```
179
180
181
   }
182
183 }
184
185 /*
186 //-----
187
188 OUTPUT -
189
190 100) +
191 103) + 4 1 108
192 104) + 7 2 103
193 105) + 5 1 106
194 106) + 00 0 2
195 107) + 1 2 100
196 108) + 00 0 1
197 109) + 00 0 3
198
199 //----
200
201 INPUT ( PASS1 CODE)
202
   (AD,1) (C,100)
203
204 100) (DL,2) (C,3)
205 103) (IS,4) (1) (S,3)
206 104) (IS,7) (2) (S,2)
207 105) (IS,5) (1) (S,4)
208 106) (DL,1) (C,2)
209 107) (IS,1) (2) (S,1)
210 108) (DL,1) (C,1)
211 109) (DL,1) (C,3)
212 110) (AD, 2)
213
214 //-----
215 */
216
217
```