```java

import java.io.*;
import java.util.*;

class Operator
{
  String name;
  String cls;
  int opcode;
  Operator(String a, String c, int op)
  {
    this.name = a;
    this.cls = c;
    this.opcode = op;
  }
}

class Register
{
  String name;
  int no;
  Register(String a, int op)
  {
    this.name = a;
    this.no = op;
  }
}


class Condition
{
  String name;
  int no;
  Condition(String a, int op)
  {
    this.name = a;
    this.no = op;
  }
}


class Symbol
{
  String name;
  int addr;
  int length;
  Symbol(String a, int op, int len)
  {
    this.name = a;
    this.addr = op;
    this.length = len;
  }
}



public class assembler_pass2
{
  public static void main(String[] args) throws IOException
  {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    BufferedReader objReader = new BufferedReader(new FileReader("pass1.txt"));
    String strCurrentLine;

    //Making the symbol tables
    ArrayList<Symbol> symboltable = new ArrayList<Symbol>(25);

    //Accepting the symbol table
    int cont = 1;
    while(cont == 1)
    {
    System.out.println("Enter name: ");
    String s = br.readLine();
    System.out.println("Enter address: ");
    int ad = Integer.parseInt(br.readLine());
    System.out.println("Enter size: ");
    int size = Integer.parseInt(br.readLine());

    //Adding the symbol

    symboltable.add(new Symbol(s,ad,size));

    System.out.println("Add another? ");
    cont = Integer.parseInt(br.readLine());

    }

    //Displaying the symbol table
    Iterator<Symbol> display = symboltable.iterator();
    Symbol x;
    while(display.hasNext())
    {
      x = display.next();
      System.out.println(x.name + " " + x.addr + " " + x.length);
    }

    //Creating the output files
    FileWriter fw = new FileWriter("pass2.txt");
    BufferedWriter write = new BufferedWriter(fw);
    int first = 1;
    while ((strCurrentLine = objReader.readLine()) != null)
    {
      if(first == 1)
      {
        first = 0;
      }
      else
      {
        String[] splited = strCurrentLine.split("\\s+");
        int len = splited[1].length();
        String[] inst = splited[1].substring(1,len-1).split(",");
        if(inst[0].equals("AD") && (inst[1].equals("1") || inst[1].equals("2")))
        {
          //skip
        }
        else
        {
```

```
     String pass2line = "";
     pass2line = pass2line+splited[0] + " + ";
     if(inst[0].equals("IS"))
     {
      if(inst[1].equals(0))
      {
       pass2line = pass2line +"00 0 000" + " ";
      }
      else
      {
       pass2line = pass2line + inst[1] + " ";
       int len2 = splited[2].length();
       String[] arg = splited[2].substring(1,len2-1).split(",");
       if(arg[0].equals("S"))
       {
        //Fetching from symbol table
        Symbol s = symboltable.get(Integer.parseInt(arg[1]));
        pass2line = pass2line + s.addr + " ";
       }
       else
       {
        // Register or condition
        pass2line = pass2line+ arg[0] + " ";
       }

       len2 = splited[3].length();
       arg = splited[3].substring(1,len2-1).split(",");
       if(arg[0].equals("S"))
       {
        //Fetching from symbol table
        int index = Integer.parseInt(arg[1])-1;
        Symbol s = symboltable.get(index);
        pass2line = pass2line + s.addr + " ";
       }
       else
       {
        // Register or condition
        pass2line = pass2line+ arg[0] + " ";
       }
      }


     }

     else if(inst[0].equals("DL") && inst[1].equals("1"))
     {
      pass2line = pass2line + "00 0 ";
      int len3 = splited[2].length();
      String[] arg = splited[2].substring(1,len3-1).split(",");
      pass2line = pass2line + arg[1] + " ";
     }
     write.write(pass2line);
     write.newLine();
    }
   }

  }

  write.close();



  }

}

/*
//-----------------------------------------------------------------------------------

 OUTPUT -

100) +
103) + 4 1 108
104) + 7 2 103
105) + 5 1 106
106) + 00 0 2
107) + 1 2 100
108) + 00 0 1
109) + 00 0 3

//-----------------------------------------------------------------------------------

INPUT ( PASS1 CODE)

    (AD,1) (C,100)
100) (DL,2) (C,3)
103) (IS,4) (1) (S,3)
104) (IS,7) (2) (S,2)
105) (IS,5) (1) (S,4)
106) (DL,1) (C,2)
107) (IS,1) (2) (S,1)
108) (DL,1) (C,1)
109) (DL,1) (C,3)
110) (AD,2)

//-----------------------------------------------------------------------------------
*/


```