

AssemblerP2.java

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

class pass2 {
    public static void main(String args[]) {
        String value = null;
        BufferedReader br, br1, br2;
        String input = null;
        String t = null;
        String t1 = null;
        String ss = null, ll = null;
        String pvalue, address;
        try {
            br = new BufferedReader(new FileReader("IM.txt"));
            File f = new File("Output.txt");
            PrintWriter p = new PrintWriter(f);
            while ((input = br.readLine()) != null) {
                StringTokenizer st = new StringTokenizer(input, " ");
                while (st.hasMoreTokens()) {
                    t = st.nextToken();
                    // System.out.println(t);
                    if (t.equals("AD") || t.equals("IS") || t.equals("DL")) {
                        // p.print(t+ " ");
                    } else if (t.matches("\\d*") && t.length() > 0 && st.hasMoreTokens()) {
                        p.print(t + " ");
                    } else if (t.matches("\\d*") && t.length() > 0 && !(st.hasMoreTokens())) {
                        p.println(t);
                    } else {
                        br1 = new BufferedReader(new FileReader("ST.txt"));
                        br2 = new BufferedReader(new FileReader("LT.txt"));
                        if (t.charAt(0) == 'S') {
                            char a;
                            int aa;
                            a = t.charAt(1);
                            aa = Character.getNumericValue(a);
                            while ((t1 = br1.readLine()) != null) {
                                StringTokenizer st1 = new StringTokenizer(t1, "\t");
                                ss = st1.nextToken();
                                int index = Integer.parseInt(ss);
                                if (index == aa) {
                                    pvalue = st1.nextToken();
                                    address = st1.nextToken();
                                    p.println(address);
                                }
                            }
                        } else if (t.charAt(0) == 'L') {
                            char a;
                            int aa;
                            a = t.charAt(1);
                            aa = Character.getNumericValue(a);
                            while ((t1 = br2.readLine()) != null) {
                                StringTokenizer st2 = new StringTokenizer(t1, "\t");
                                ss = st2.nextToken();
                                int index = Integer.parseInt(ss);
                                if (index == aa) {
                                    pvalue = st2.nextToken();
                                    address = st2.nextToken();
                                    p.println(address);
                                }
                            }
                        } else {
                            p.print(t + " ");
                        }
                    }
                }
            }
            p.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Assembly code:

```
START 100
MOV AREG A
MOV BREG B
MOV CREG =2
MOV DREG =3
ADD AREG BREG
SUB AREG A
A DC 05
B DS 03
END
```

Symbol Table:

0	A	106
1	B	107

Literals Table:

0	=2	110
1	=3	111

Intermediate code:

AD	1	100	
IS	4	1	S0
IS	4	2	S1
IS	4	3	L0
IS	4	4	L1
IS	1	12	
IS	2	1	S0

A DL 1
B DL 2
AD 2

Output Machine Code:

1 100
4 1 106
4 2 107
4 3 110
4 4 111
1 1 2
2 1 106
A 1
B 2
2