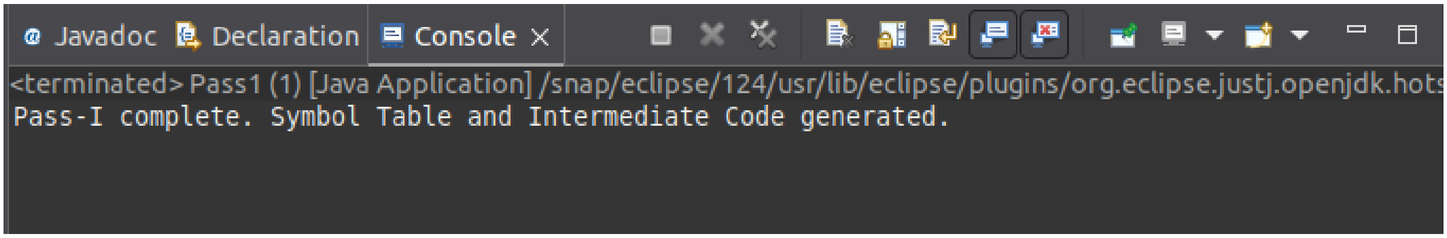
Pass1.java

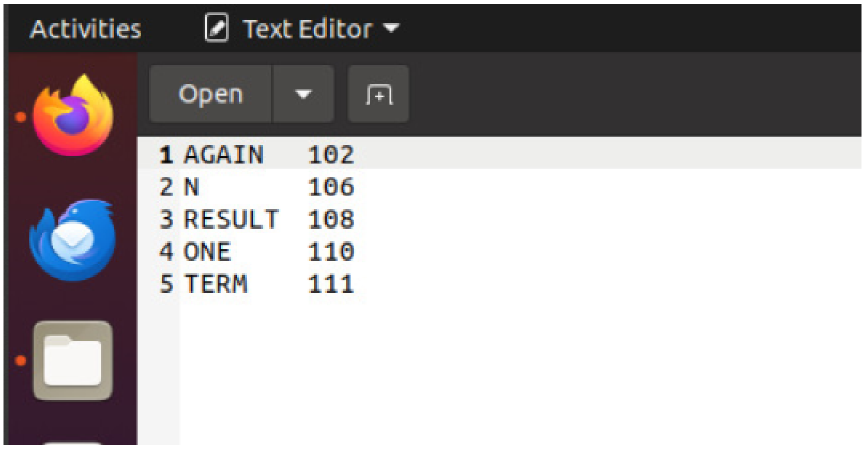
```java
package assembler;

import java.io.*;

public class Pass1 {
    public static void main(String[] args) throws IOException {
        String[][] a = {
            {"", "START", "101", ""},
            {"", "MOVER", "BREG", "ONE"},
            {"AGAIN", "MULT", "BREG", "TERM"},
            {"", "MOVER", "CREG", "TERM"},
            {"", "ADD", "CREG", "N"},
            {"", "MOVEM", "CREG", "TERM"},
            {"N", "DS", "2", ""},
            {"RESULT", "DS", "2", ""},
            {"ONE", "DC", "1", ""},
            {"TERM", "DS", "1", ""},
            {"", "END", "", ""}
        };

        int lc = Integer.parseInt(a[0][2]);

        // Symbol table
        String[][] st = new String[10][2];
        int symCnt = 0;

        BufferedWriter symtab = new BufferedWriter(new FileWriter("symtab.txt"));
        BufferedWriter inter = new BufferedWriter(new FileWriter("intermediate.txt"));

        for (int i = 1; i < a.length; i++) {
            if (!a[i][0].equals("")) {
                st[symCnt][0] = a[i][0];
                st[symCnt][1] = Integer.toString(lc);
                symtab.write(a[i][0] + "\t" + lc + "\n");
                symCnt++;
            }

            inter.write(lc + "\t" + a[i][1] + "\t" + a[i][2] + "\t" + a[i][3] + "\n");

            if (a[i][1].equals("DS")) {
                lc += Integer.parseInt(a[i][2]);
            } else {
                lc++;
            }
        }

        symtab.close();

inter.close();
          System.out.println("Pass-I                                           Code
 generated.");
                                           complete. Symbol Table and Intermediate
     }
 }
```

Output :



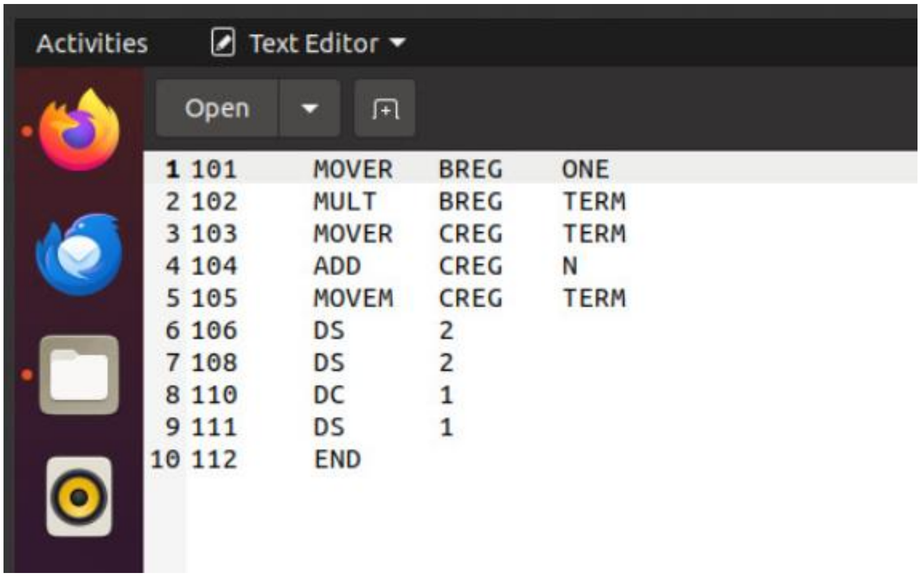
```
<terminated> Pass1 (1) [Java Application] /snap/eclipse/124/usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hots
Pass-I complete. Symbol Table and Intermediate Code generated.
```

Symtab.txt



```
AGAIN   102
N       106
RESULT  108
ONE     110
TERM    111
```

Intermediate.txt

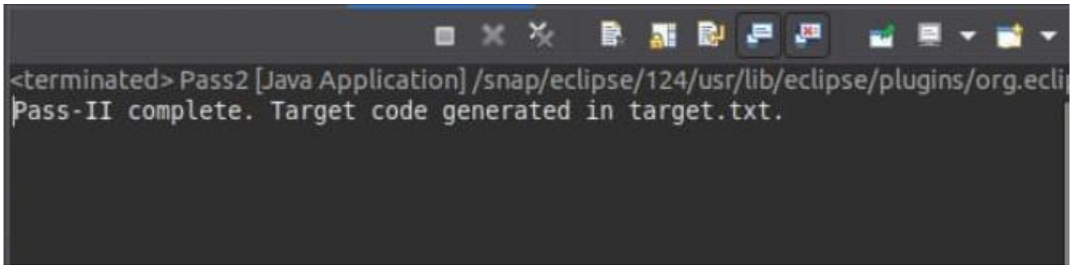


```
101     MOVER   BREG    ONE
102     MULT    BREG    TERM
103     MOVER   CREG    TERM
104     ADD     CREG    N
105     MOVEM   CREG    TERM
106     DS      2
108     DS      2
110     DC      1
111     DS      1
112     END
```

Pass2.java

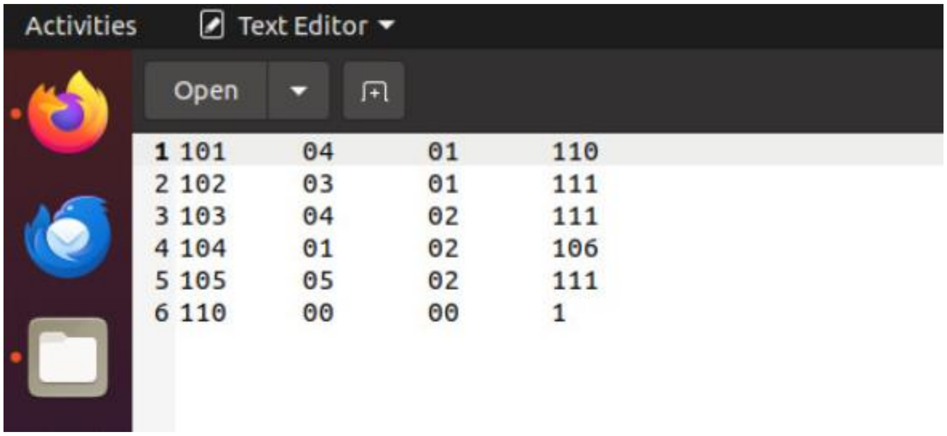
```java
package assembler;

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

public class Pass2 {
    public static void main(String[] args) throws IOException {
        BufferedReader inter = new BufferedReader(new FileReader("intermediate.txt"));
        BufferedReader symtab = new BufferedReader(new FileReader("symtab.txt"));
        BufferedWriter target = new BufferedWriter(new FileWriter("target.txt"));

        // Load symbol table into hashmap
        Map<String, String> symTable = new HashMap<>();
        String line;
        while ((line = symtab.readLine()) != null) {
            String[] parts = line.trim().split("\t");
            if (parts.length == 2)
                symTable.put(parts[0], parts[1]);
        }

        // Instruction codes and registers
        List<String> inst = Arrays.asList("STOP", "ADD", "SUB", "MULT", "MOVER", "MOVEM", "COMP", "BC", "DIV", "READ", "PRINT");
        List<String> reg = Arrays.asList("AREG", "BREG", "CREG", "DREG");

        while ((line = inter.readLine()) != null) {
            String[] parts = line.trim().split("\t");
            String lc = parts[0];
            String opcode = parts[1];
            String operand1 = parts.length > 2 ? parts[2] : "";
            String operand2 = parts.length > 3 ? parts[3] : "";

            if (opcode.equals("START") || opcode.equals("END")) continue;
            if (opcode.equals("DS")) continue;
            if (opcode.equals("DC")) {
                target.write(lc + "\t" + "00\t00\t" + operand1 + "\n");
                continue;
            }

            int opCodeVal = inst.indexOf(opcode);
            int regCodeVal = reg.indexOf(operand1);

            String address = symTable.getOrDefault(operand2, "000");

            target.write(lc + "\t" + String.format("%02d", opCodeVal) + "\t" +
                         String.format("%02d", regCodeVal) + "\t" + address + "\n");
        }


inter.close();
symtab.close();
target.close();


 }        System.out.println("Pass-II complete. Target code generated in target.txt.");
      }
```

Output



```
<terminated> Pass2 [Java Application] /snap/eclipse/124/usr/lib/eclipse/plugins/org.ecli
Pass-II complete. Target code generated in target.txt.
```

target.txt



```
101     04      01      110
102     03      01      111
103     04      02      111
104     01      02      106
105     05      02      111
110     00      00      1
```