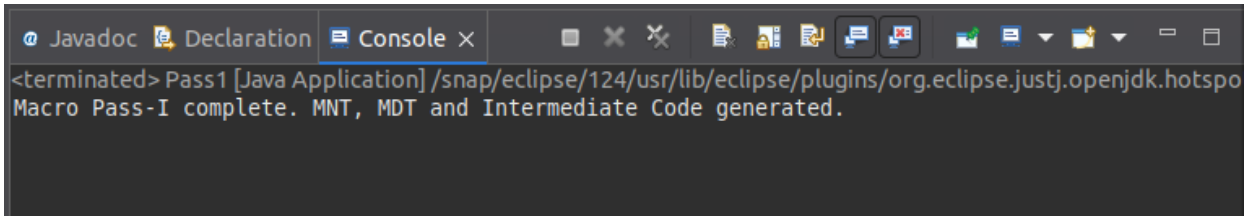
**Pass1.java**

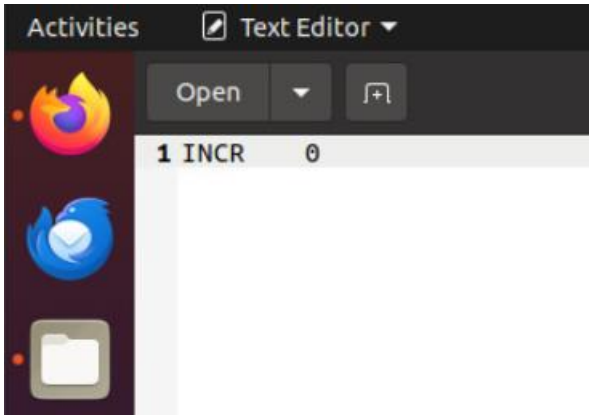
```
package Macroprocessor;

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

public class Pass1 {
    public static void main(String[] args) throws IOException {
        // Sample macro code (can be replaced with file input)
        String[] code = {
            "MACRO",
            "INCR &ARG1",
            "A 1,&ARG1",
            "MEND",
            "START 101",
            "INCR TERM",
            "MOVEM AREG,TERM",
            "END"
        };

        List<String[]> MNT = new ArrayList<>();
        List<String> MDT = new ArrayList<>();
        BufferedWriter inter = new BufferedWriter(new
FileWriter("intermediate.txt"));
        BufferedWriter mntFile = new BufferedWriter(new FileWriter("mnt.txt"));
        BufferedWriter mdtFile = new BufferedWriter(new FileWriter("mdt.txt"));

        boolean isMacro = false;
        int mdtIndex = 0;

        for (int i = 0; i < code.length; i++) {
            String line = code[i];
            String[] parts = line.trim().split("\\s+", 2);

            if (parts[0].equals("MACRO")) {
                isMacro = true;
                String[] defParts = code[i + 1].trim().split("\\s+", 2);
                String macroName = defParts[0];
                MNT.add(new String[]{macroName, String.valueOf(mdtIndex)});
                i++; // skip macro name line
                continue;
            }

            if (parts[0].equals("MEND")) {
                MDT.add("MEND");
                mdtIndex++;
                isMacro = false;
                continue;
            }

```

```
49            if (isMacro) {
50                MDT.add(line);
51                mdtIndex++;
52            } else {
53                inter.write(line + "\n");
54            }
55        }
56
57        // Writing MNT
58        for (String[] entry : MNT) {
59            mntFile.write(entry[0] + "\t" + entry[1] + "\n");
60        }
61
62        // Writing MDT
63        for (int i = 0; i < MDT.size(); i++) {
64            mdtFile.write(i + "\t" + MDT.get(i) + "\n");
65        }
66
67        inter.close();
68        mntFile.close();
69        mdtFile.close();
70
71        System.out.println("Macro Pass-I complete. MNT, MDT and Intermediate Code
    generated.");
72    }
73 }
```

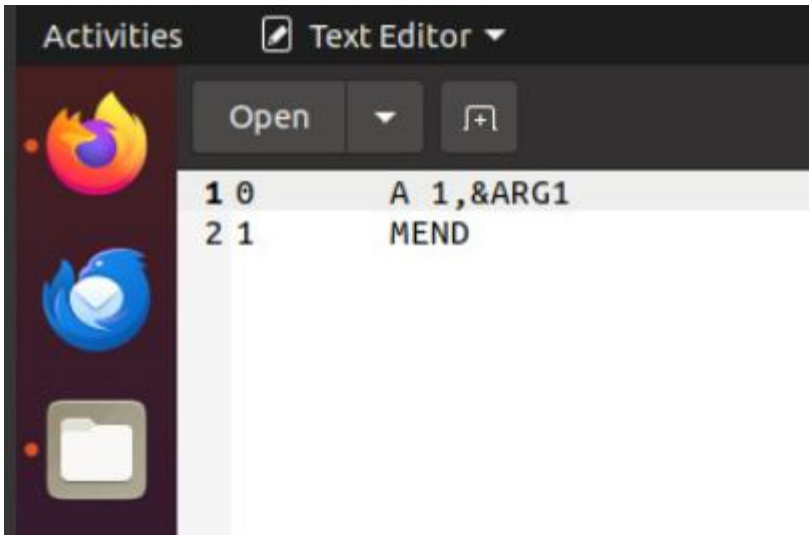
**Output :**



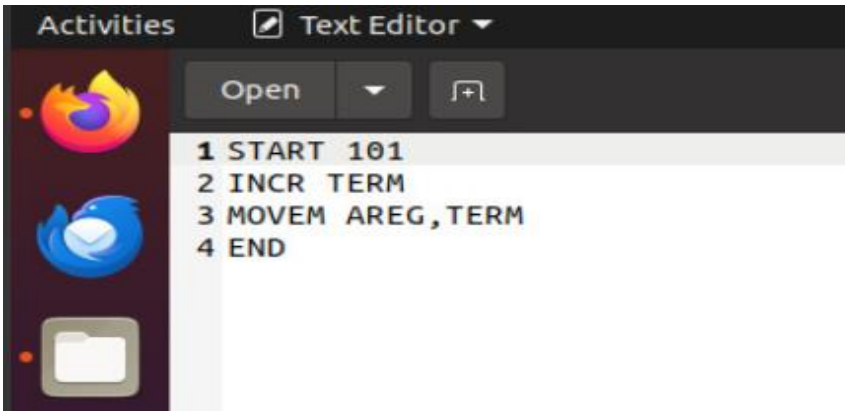

**MNT.txt**



**MDT.txt**




**Intermediate.txt**

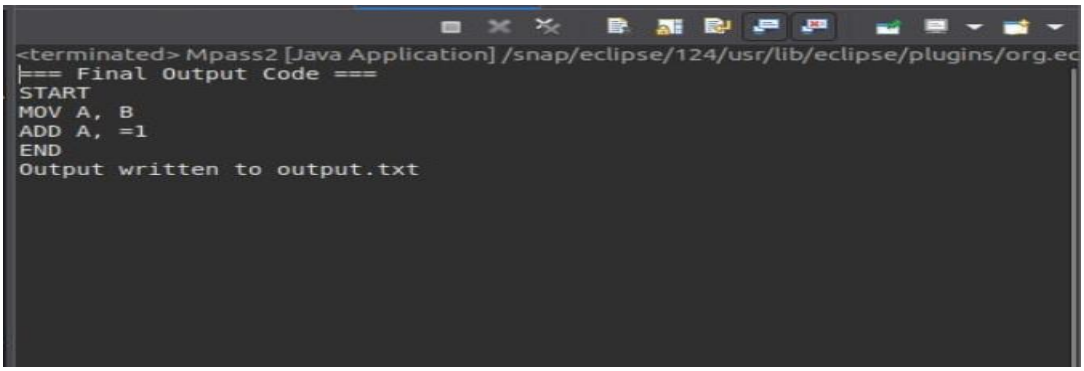


**Pass2.java**

```java
package Macroprocessor;

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

public class Pass2 {
    private final Map<String, Integer> MNT = new HashMap<>();
    private final List<String> MDT = new ArrayList<>();
    private final List<String> intermediateCode = new ArrayList<>();
    private final List<String> outputCode = new ArrayList<>();

    public Pass2() {
        // Simulating already available MNT, MDT, and intermediate code from Pass-I

        // Macro Name Table: Macro name → MDT index
        MNT.put("INCR", 0);

        // Macro Definition Table
        MDT.add("INCR &ARG");
        MDT.add("ADD &ARG, =1");
        MDT.add("MEND");

        // Intermediate Code (no macro definitions)
        intermediateCode.add("START");
        intermediateCode.add("MOV A, B");
        intermediateCode.add("INCR A");
        intermediateCode.add("END");
    }

    public void expandMacros() {
        for (String line : intermediateCode) {
            String[] tokens = line.trim().split("\\s+");

            if (tokens.length == 0) continue;

            String macroName = tokens[0];

            if (MNT.containsKey(macroName)) {
                int index = MNT.get(macroName);
                String actualArg = tokens.length > 1 ? tokens[1] : "";

                // Skip the macro header in MDT
                index++;

                while (!MDT.get(index).equalsIgnoreCase("MEND")) {
                    String defLine = MDT.get(index);

                    // Replace formal argument with actual argument
                    String expandedLine = defLine.replace("&ARG", actualArg);
```

```
            outputCode.add(expandedLine);
            index++;
        }
             } else {
                 outputCode.add(line);
             }
         }
     }

     public void displayOutput() {
         System.out.println("=== Final Output Code ===");

         try (BufferedWriter writer = new BufferedWriter(new
FileWriter("output.txt"))) {
             for (String line : outputCode) {
                 System.out.println(line);           // Console output
                 writer.write(line + "\n");          // Write to output.txt
             }
             System.out.println("Output written to output.txt");
         } catch (IOException e) {
             System.out.println("Error writing to file: " + e.getMessage());
         }
     }

     public static void main(String[] args) {
         Pass2 pass2 = new Pass2();
         pass2.expandMacros();
         pass2.displayOutput();
     }
}
```

**Output**




**Output.txt**