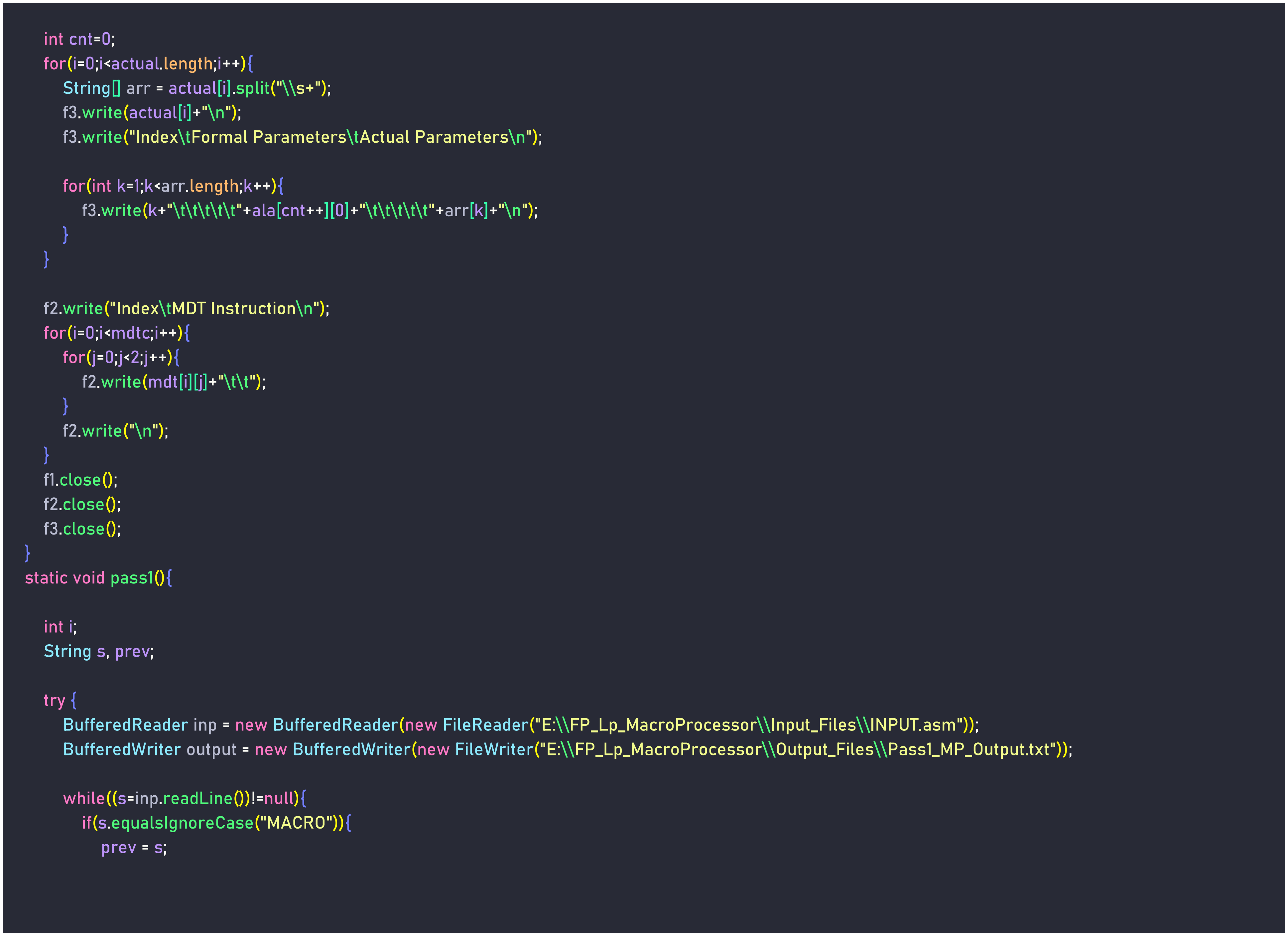
Problem Statement : Design suitable data structures and implement pass-1 of a two pass macro-processor in JAVA

Name : Anuj Mahendra Mutha **Class** : TE 4 **Batch** : M4

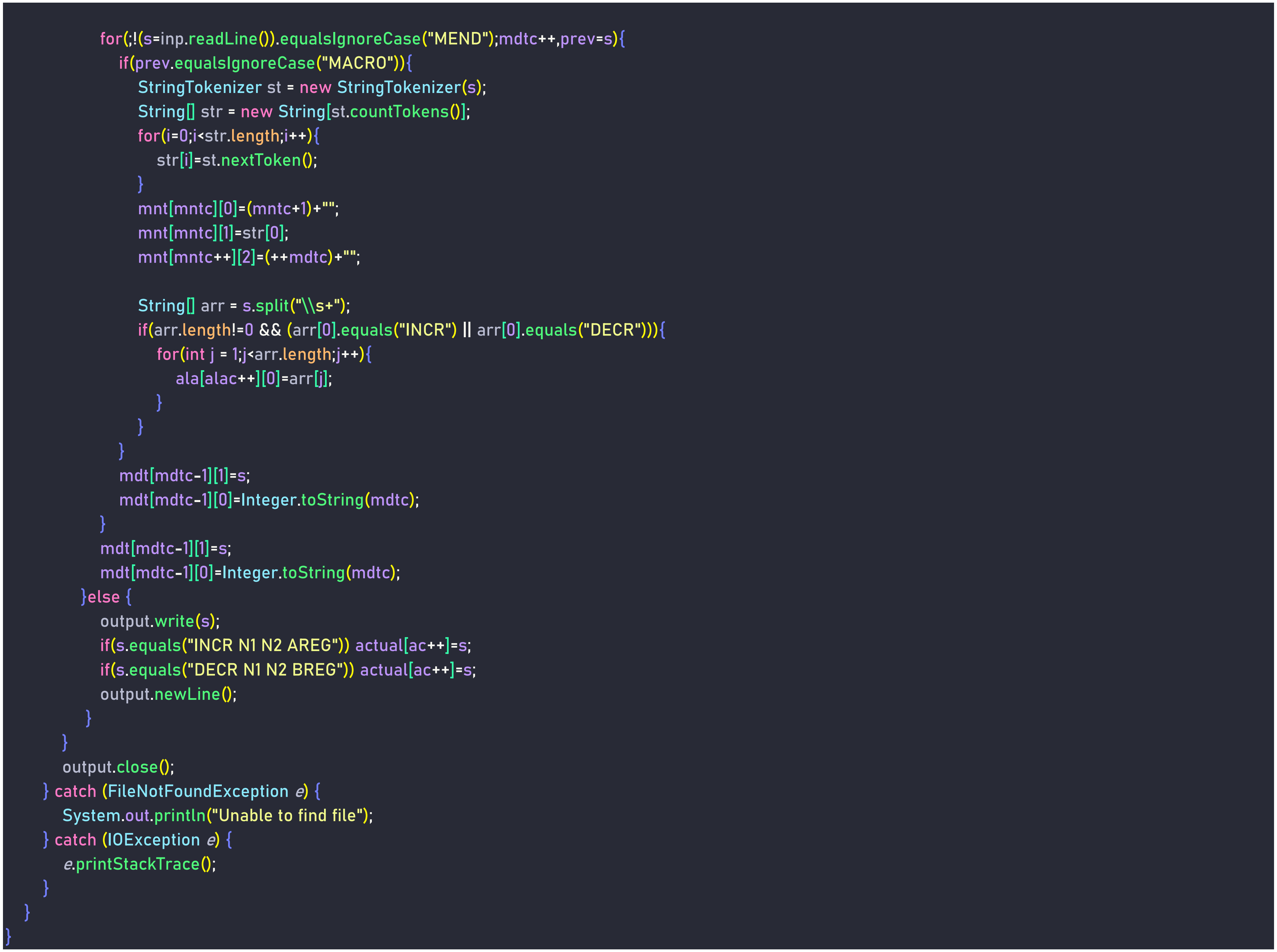
Roll Number : 31443 **Subject**: Lab Practices - 1 **DOP** : 30 December 2021

Code for Pass-1 of a Two Pass Macro-Processor :

```
package com.muthadevs;
import java.util.*;
import java.io.*;

public class Main {

    static String[][] mnt = new String[5][3];
    static String[][] ala = new String[10][2];  //DEFINING THE DATA STRUCTURES FOR STORING THE CONTENTS.
    static String[][] mdt = new String[20][2];
    static String[] actual = new String[2];

    static int mntc = 0, mdtc = 0, alac = 0, ac = 0;

    public static void main(String[] args)throws Exception {

        pass1();

        BufferedWriter f1 = new BufferedWriter(new FileWriter("E:\\FP_Lp_MacroProcessor\\Output_Files\\MNT.txt"));
        BufferedWriter f2 = new BufferedWriter(new FileWriter("E:\\FP_Lp_MacroProcessor\\Output_Files\\MDT.txt"));
        BufferedWriter f3 = new BufferedWriter(new FileWriter("E:\\FP_Lp_MacroProcessor\\Output_Files\\ALA.txt"));

        int i,j;
        f1.write("Index\tMacro name\tMDT Index\n");
        for(i=0;i<mntc;i++){
            for(j=0;j<3;j++){
                f1.write(mnt[i][j]+"\t\t");
            }
            f1.write("\n");
        }
```



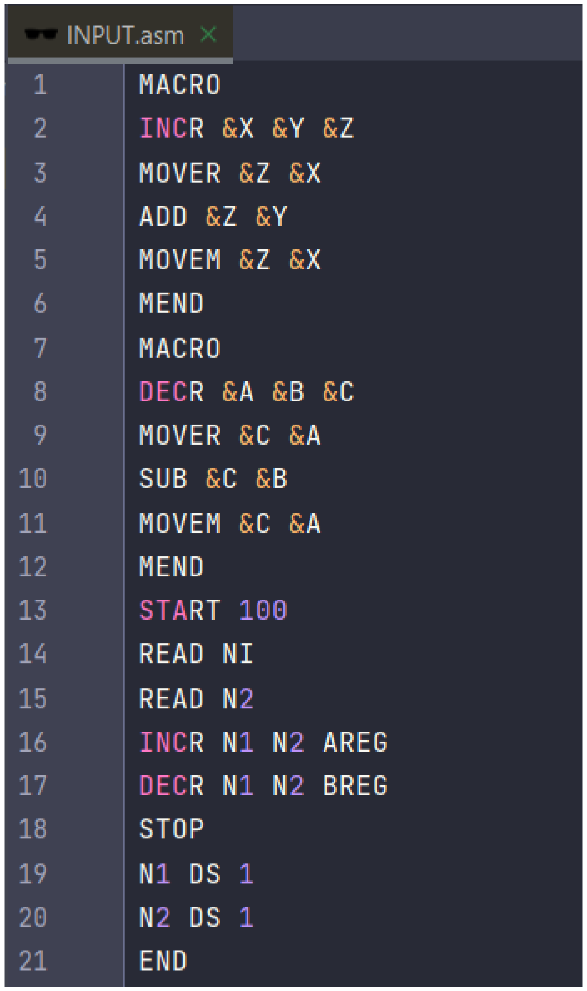
```java
        int cnt=0;
        for(i=0;i<actual.length;i++){
            String[] arr = actual[i].split("\\s+");
            f3.write(actual[i]+"\n");
            f3.write("Index\tFormal Parameters\tActual Parameters\n");

            for(int k=1;k<arr.length;k++){
                f3.write(k+"\t\t\t\t\t"+ala[cnt++][0]+"\t\t\t\t\t"+arr[k]+"\n");
            }
        }

        f2.write("Index\tMDT Instruction\n");
        for(i=0;i<mdtc;i++){
            for(j=0;j<2;j++){
                f2.write(mdt[i][j]+"\t\t");
            }
            f2.write("\n");
        }
        f1.close();
        f2.close();
        f3.close();
    }
    static void pass1(){

        int i;
        String s, prev;

        try {
            BufferedReader inp = new BufferedReader(new FileReader("E:\\FP_Lp_MacroProcessor\\Input_Files\\INPUT.asm"));
            BufferedWriter output = new BufferedWriter(new FileWriter("E:\\FP_Lp_MacroProcessor\\Output_Files\\Pass1_MP_Output.txt"));

            while((s=inp.readLine())!=null){
                if(s.equalsIgnoreCase("MACRO")){
                    prev = s;
```



```
                    for(;!(s=inp.readLine()).equalsIgnoreCase("MEND");mdtc++,prev=s){
                        if(prev.equalsIgnoreCase("MACRO")){
                            StringTokenizer st = new StringTokenizer(s);
                            String[] str = new String[st.countTokens()];
                            for(i=0;i<str.length;i++){
                                str[i]=st.nextToken();
                            }
                            mnt[mntc][0]=(mntc+1)+"";
                            mnt[mntc][1]=str[0];
                            mnt[mntc++][2]=(++mdtc)+"";

                            String[] arr = s.split("\\s+");
                            if(arr.length!=0 && (arr[0].equals("INCR") || arr[0].equals("DECR"))){
                                for(int j = 1;j<arr.length;j++){
                                    ala[alac++][0]=arr[j];
                                }
                            }
                        }
                        mdt[mdtc-1][1]=s;
                        mdt[mdtc-1][0]=Integer.toString(mdtc);
                    }
                    mdt[mdtc-1][1]=s;
                    mdt[mdtc-1][0]=Integer.toString(mdtc);
                }else {
                    output.write(s);
                    if(s.equals("INCR N1 N2 AREG")) actual[ac++]=s;
                    if(s.equals("DECR N1 N2 BREG")) actual[ac++]=s;
                    output.newLine();
                }
            }
            output.close();
        } catch (FileNotFoundException e) {
            System.out.println("Unable to find file");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Input File:

1. INPUT.asm

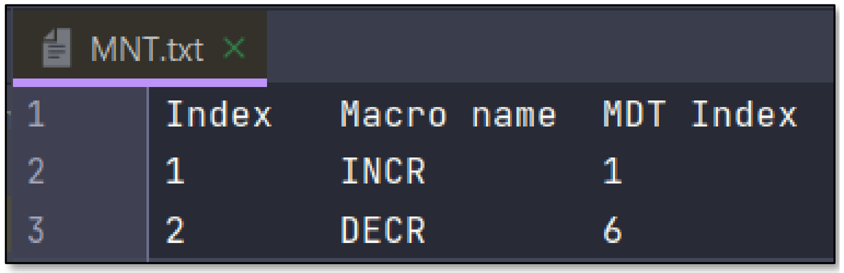


INPUT.asm

```
MACRO
INCR &X &Y &Z
MOVER &Z &X
ADD &Z &Y
MOVEM &Z &X
MEND
MACRO
DECR &A &B &C
MOVER &C &A
SUB &C &B
MOVEM &C &A
MEND
START 100
READ NI
READ N2
INCR N1 N2 AREG
DECR N1 N2 BREG
STOP
N1 DS 1
N2 DS 1
END
```

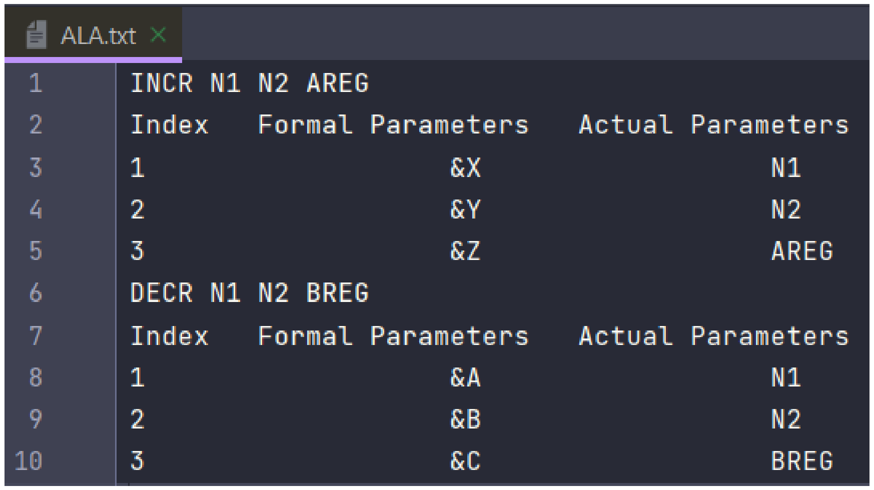
Output(s):

1) MNT.txt (Macro Name Table)



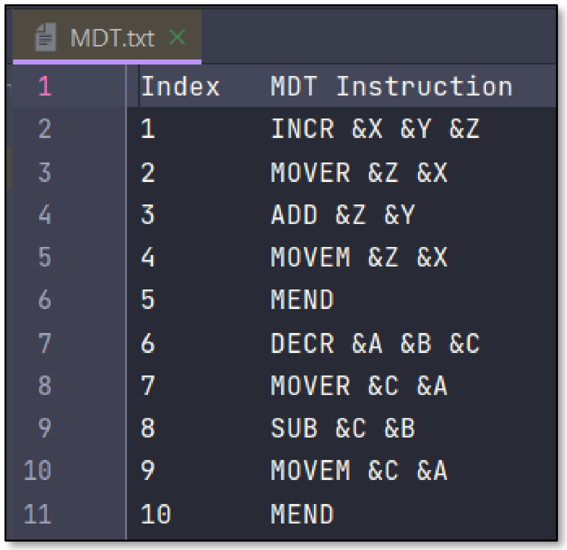
```
Index    Macro name   MDT Index
1        INCR         1
2        DECR         6
```

2) ALA.txt (Argument List Array)



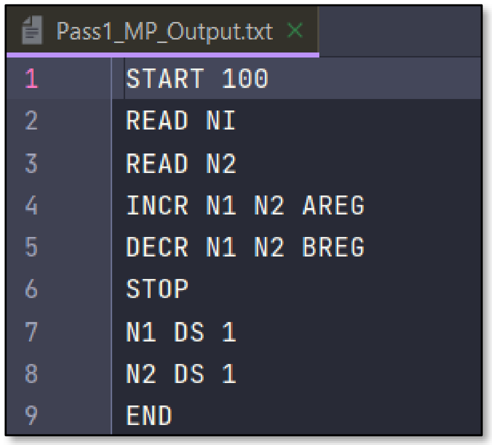
```
INCR N1 N2 AREG
Index   Formal Parameters   Actual Parameters
1                    &X                   N1
2                    &Y                   N2
3                    &Z                   AREG
DECR N1 N2 BREG
Index   Formal Parameters   Actual Parameters
1                    &A                   N1
2                    &B                   N2
3                    &C                   BREG
```

3) MDT (Macro Definition Table)



```
Index   MDT Instruction
1       INCR &X &Y &Z
2       MOVER &Z &X
3       ADD &Z &Y
4       MOVEM &Z &X
5       MEND
6       DECR &A &B &C
7       MOVER &C &A
8       SUB &C &B
9       MOVEM &C &A
10      MEND
```

4) Pass-1 Macro-Processor Output



```
START 100
READ NI
READ N2
INCR N1 N2 AREG
DECR N1 N2 BREG
STOP
N1 DS 1
N2 DS 1
END
```