ASSIGNMENT NO.4

(MACRO PASS II Code)

Input Files

1)Pass_II_Input.txt

```
START   100
READ    N1
READ    N2
INCR    N1,Y=BREG,REG=CREG
DECR    N1,N2
STOP
N1      DS 1
N2      DS 1
END
```

2)MDTable.txt

```
1 INCR &X,&Y=,&REG=AREG
2 MOVER &REG &X
3 ADD &REG &Y
4 MOVEM &REG &X
5 MEND
6 DECR &A,&B,&REG=BREG
7 MOVER &REG &A
8 SUB &REG &B
9 MOVEM &REG &A
10 MEND
```

3)MNTtable.txt

```
INCR 1
DECR 6
```

4)ALAtable.txt

```
INCR
```

```
    &X N1
    &Y BREG
    &REG CREG
    END
    DECR
    &A N1
    &B N2
    &REG BREG
    END
```

5)Pass2MacroProcessor.java

```
import java.io.*;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.StringTokenizer;
@SuppressWarnings("unused")
class Ala{
        // name_argument -> index, actual Argument
        HashMap<String, String>Arguments = new HashMap<String, String>();
}

public class pass2MacroProcessor {
        //------data structures needed-------------------------------------------------------------------
----------------------------------------

                        // name of macro -> index in MDT
                        static HashMap<String, Integer>MNT = new HashMap<String,
Integer>();
                        // index -> mnemonic , arguments (2/3)
                        static HashMap<Integer, ArrayList<String>> MDT = new
HashMap<Integer, ArrayList<String>>();
                        // name_of_macro -> all variable in class
                        static HashMap<String, Ala>AlaTable = new HashMap<String,
Ala>();
                        // MDT table counter
                        static int MDTC=1;
                        //MNT table counter
                        static int MNTC=1;

        //------create MNTtable ----------------------------------------------------------------------------
-------------------------
                        private static void createMNTtable() {
```

```
                FileReader fr = null;
                try {
                    fr = new
FileReader("/home/student/Desktop/snehal/MNTtable.txt");
                } catch (FileNotFoundException e1) {
                    e1.printStackTrace();
                }
          BufferedReader br=new BufferedReader(fr);
                String s=null;
                try {
                    while((s=br.readLine())!=null){
                        StringTokenizer tokens = new
StringTokenizer(s," ",false);
                        ArrayList<String> arrayList= new
ArrayList<>();
                        while(tokens.hasMoreTokens()){

    arrayList.add(tokens.nextToken());
                        }
                        MNT.put(arrayList.get(0),
Integer.parseInt(arrayList.get(1)));
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
    //------create MDTtable-----------------------------------------------------------------------------
--------------------------

            private static void createMDTtable() {

                FileReader fr = null;
                try {
                    fr = new
FileReader("/home/student/Desktop/snehal/MDTable.txt");
                } catch (FileNotFoundException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
          BufferedReader br=new BufferedReader(fr);
                String s=null;
                try {
                    int i = 1;
                    while((s=br.readLine())!=null){
                        StringTokenizer tokens = new
StringTokenizer(s," ",false);
                        ArrayList<String> arrayList= new
ArrayList<>();
                        while(tokens.hasMoreTokens()){

    arrayList.add(tokens.nextToken());
                        }
                        MDT.put(i, arrayList);
                        i++;
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
    //------create ALAtable------------------------------------------------------------------------------
-------------------------
            private static void createALAtable() {

                FileReader fr = null;
                try {
                    fr = new
FileReader("/home/student/Desktop/snehal/ALAtable.txt");
                } catch (FileNotFoundException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
          BufferedReader br=new BufferedReader(fr);
                String s=null;
                try {
                    int lineno=1;
                        while((s=br.readLine())!=null )
                        {
                            //System.out.println(s);
                            Ala argument =new Ala();
                            String curr=s;

    while(!(s=br.readLine()).equals("END")){
                                StringTokenizer
tokens = new StringTokenizer(s," ",false);
                                ArrayList<String>
arrayList= new ArrayList<>();

    while(tokens.hasMoreTokens()){

    arrayList.add(tokens.nextToken());
```

```
                                }

    //System.out.println(s);

    argument.Arguments.put(arrayList.get(0), arrayList.get(1));
                            }
                            AlaTable.put(curr, argument);
                        }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }

 //------Handling Macro-------------------------------------------------------------------------------------
------------------
        private static String handleMacro(ArrayList<String> arrayList,int index) {

            return null;
        }

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

            createMNTtable();
            createMDTtable();
            createALAtable();

            //creating file pointers
            FileReader fr_input=new
FileReader("/home/student/Desktop/snehal/Pass_II_Input.txt");
            FileWriter fr_output = null;
            try {
                fr_output = new FileWriter("/home/student/Desktop/snehal
/PassII_OUTPUT.txt ");

            } catch (IOException e1) {

                e1.printStackTrace();
            }

            try {
                //creates a buffering character input stream
                BufferedReader br_input = new BufferedReader(fr_input);
                String s = null;

                while(((s=br_input.readLine())!=null)) {
                    ArrayList<String> arrayList= new
ArrayList<>();
                    StringTokenizer tokens = new
StringTokenizer(s," ");

                    while(tokens.hasMoreTokens()){
                        arrayList.add(tokens.nextToken());
                    }
                    String curr = arrayList.get(0);

                    //if current line in macro call
                    if(MNT.containsKey(curr)) // first check in mnt
table
                    {
                        String temp="";
                        //taking start position from mnt
                        int startPos =
MNT.get(arrayList.get(0));

                        for(int i = startPos+1;
!MDT.get(i).get(1).equals("MEND") ; i++ )
                        {
                            //for each parameter fetching
value form ala
                            ArrayList<String>
arrayList1= new ArrayList<>();
                            arrayList1 = MDT.get(i);
                            temp+=arrayList1.get(1)+" ";

                            for(int
j=2;j<arrayList1.size();j++)
                            {

                                if(j ==
arrayList1.size()-1)
temp+=AlaTable.get(curr).Arguments.get(arrayList1.get(j));
                                else

temp+=AlaTable.get(curr).Arguments.get(arrayList1.get(j))+", ";
                            }
                            temp+="\n";
                        }

                        fr_output.write(temp);
                    }
```

```
                                        // other than macro call statement
                                        else
                                        {
                                                fr_output.write(s+"\n");
                                        }
                                }
                        }
                        catch(Exception e){
                                System.out.println(e);
                        }
                        finally {
                                try {
                                        if (fr_output != null) {
                                                fr_output.flush();
                                                fr_output.close();
                                        }
                                } catch (IOException e) {
                                        e.printStackTrace();
                                }
                        }
                        System.out.println("successfully Executed :) \n please check output
in finaloutput file ");
                }

}
```

Output Files:-

1)PassII_OUTPUT.txt

```
START   100
READ    N1
READ    N2
MOVER CREG, N1
ADD CREG, BREG
MOVEM CREG, N1
MOVER BREG, N1
SUB BREG, N2
MOVEM BREG, N1
STOP
N1      DS 1
N2      DS 1
```

END