ASSIGNMENT NO.3

(MACRO PASS I Code)

Input Files

1)macroInput.txt

```
MACRO
INCR    &X,&Y=,&REG=AREG
MOVER   &REG,&X
ADD     &REG,&Y
MOVEM   &REG,&X
MEND
MACRO
DECR    &A,&B,&REG=BREG
MOVER   &REG,&A
SUB     &REG,&B
MOVEM   &REG,&A
MEND
START   100
READ    N1
READ    N2
INCR    N1,Y=BREG,REG=CREG
DECR    N1,N2
STOP
N1      DS 1
N2      DS 1
END
```

2)Macro_Pass_I.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 Macro_Pass_I {

	//------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;

	//-----------------------------Prepare MDT-----------------------------------------------------------
-------------------------------
		private static void PreapareMDT() {
			FileWriter writer = null;
			try {
				writer = new
FileWriter("/home/student/Desktop/snehal/MDTable.txt");
			} catch (IOException e1) {
				e1.printStackTrace();
			}
			for(Integer strKey : MDT.keySet() ){

			   String temp="";
			   temp+=Integer.toString(strKey)+" ";

			   for(int i=0;i<MDT.get(strKey).size();i++) {
				temp+=MDT.get(strKey).get(i)+" ";
			   }
			   temp+="\n";
			   try {
					writer.write(temp);
				} catch (IOException e) {
```

```
                                        e.printStackTrace();
                                }
                        }
                        try {
                                if (writer != null) {
                                        writer.flush();
                                        writer.close();
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
                //-----------------------------Prepare MNT-------------------------------------------------

                private static void PrepareMNT() {
                        FileWriter writer = null;
                        try {
                                writer = new
FileWriter("/home/student/Desktop/snehal/MNTtable.txt");
                        } catch (IOException e1) {
                                e1.printStackTrace();
                        }
                        for(String strKey : MNT.keySet() ){
                          String temp="";
                           temp+=strKey+" ";
                           temp+=Integer.toString(MNT.get(strKey))+" ";
                           temp+="\n";
                           try {
                                        writer.write(temp);
                                } catch (IOException e) {
                                        e.printStackTrace();
                                }
                        }
                        try {
                                if (writer != null) {
                                        writer.flush();
                                        writer.close();
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
```

```
                }
        }
        //-----------------------------Prepare ALA-----------------------------------------------
        private static void Ala_Table() throws IOException {
                FileWriter writer = null;
                try {
                        writer = new
FileWriter("/home/student/Desktop/snehal/ALAtable.txt");
                } catch (IOException e1) {
                        e1.printStackTrace();
                }
                for(String strKey : AlaTable.keySet() ){

                   String temp="";
                   temp+=strKey+"\n";
                   Ala argument =new Ala();
                   argument = AlaTable.get(strKey);
                   for(String strKey1 : argument.Arguments.keySet() ){
                        temp+=strKey1+" ";
                        temp+=argument.Arguments.get(strKey1)+"\n";
                   }
                   temp+="END\n";

                   try {
                                writer.write(temp);
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
                try {
                        if (writer != null) {
                                writer.flush();
                                writer.close();
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }

    //----------------Main Method-----------------------------------------------------
```

```
	public static void main(String[] args) throws FileNotFoundException {
		//creating file pointers
		FileReader fr_input=new
FileReader("/home/student/Desktop/snehal/macroInput.txt");
		FileWriter fr_output = null;
		try {
			fr_output = new
FileWriter("/home/student/Desktop/snehal/macroOutput.txt");
		} catch (IOException e1) {

			e1.printStackTrace();
		}
		try {
			//creates a buffering character input stream
			BufferedReader br_input=new BufferedReader(fr_input);

			String s = null;
			int lineno=0;

			while(((s=br_input.readLine())!=null)) {
				if(s.charAt(0)=='S')
				{
					break;
				}

				lineno++;
				ArrayList<String> arrayList= new ArrayList<>();
				StringTokenizer tokens = new StringTokenizer(s," ");

				while(tokens.hasMoreTokens()){
					arrayList.add(tokens.nextToken());
				}

				//if we get the blank line
				if(arrayList.size()==0)
				{
					continue;
				}

				if(arrayList.get(0).equals("MACRO"))
```

```
                                        {
                                                //taking name of the macro with arguments if any
                                                s=br_input.readLine();
                                                ArrayList<String> arrayList1= new ArrayList<>();
                                                StringTokenizer tokens1 = new StringTokenizer(s,"
");
                                                while(tokens1.hasMoreTokens()){
                                                        arrayList1.add(tokens1.nextToken());
                                                }
                                                // making entry in the MNT
                                                MNT.put(arrayList1.get(0), MDTC);
                                                String curr_macroname = arrayList1.get(0);

                                                //Processing all the Arguments
                                                ArrayList<String> arrayList11= new
ArrayList<>();
                                                StringTokenizer tokens11 = new
StringTokenizer(arrayList1.get(1),",");
                                                while(tokens11.hasMoreTokens()){
                                                        arrayList11.add(tokens11.nextToken());
                                                }
                                                Ala argument =new Ala();
                                                for(int i=0;i<arrayList11.size();i++)
                                                {
                                                        String curr=arrayList11.get(i),
curr_keyword;// curr keyword = formal argument
                                                        String default_val="";
                                                        //keyword type parameters
                                                        if(curr.contains("="))
                                                        {
                                                                int positionEqu=curr.indexOf('=');
                                                                //keyword parameter
                                                                if(positionEqu == curr.length()-1)
                                                                {
                                                curr_keyword=curr.substring(0, positionEqu);
                                                                        default_val = "@";
                                                                }

                                                                //default parameter
                                                                else {
```

```
        curr_keyword=curr.substring(0, positionEqu);
                                                        String defaultValue=
curr.substring(positionEqu+1, curr.length());
                                                        default_val = defaultValue;
                                                }

                                        }
                                        //Positional parameters
                                        else
                                        {
                                                curr_keyword = curr;
                                                default_val = "#" +
Integer.toString(i);
                                        }
                                        argument.Arguments.put(curr_keyword,
default_val);
                                }
                                AlaTable.put(arrayList1.get(0), argument);
                                // making entry in mdt
                                MDT.put(MDTC++, arrayList1);
                                //BODY of the macro
                                while(!(s=br_input.readLine()).equals("MEND"))
                                {
                                        ArrayList<String> arrayList2= new
ArrayList<>();
                                        ArrayList<String> arrayList22= new
ArrayList<>();
                                        StringTokenizer tokens2 = new
StringTokenizer(s," ");
                                        while(tokens2.hasMoreTokens()){

        arrayList2.add(tokens2.nextToken());
                                        }
                                        String argus = arrayList2.get(1);
                                        arrayList2.remove(1);
                                        StringTokenizer tokens3 = new
StringTokenizer(argus,",");
                                        while(tokens3.hasMoreTokens()){
```

```
        arrayList22.add(tokens3.nextToken());
                                                }
                                                for(int i=0;i<arrayList22.size();i++)
                                                {
//                                              System.out.println(arrayList22.get(i));
                                                        arrayList2.add(arrayList22.get(i));
                                                }
                                                MDT.put(MDTC++, arrayList2);
                                        }
                                        arrayList.clear();
                                        ArrayList<String> temp_mend= new
ArrayList<>();
                                        temp_mend.add("MEND");
                                        MDT.put(MDTC++, temp_mend);
                                        //ending of the macro
                                }
                                else
                                {
                                        continue;
                                }

                        }
                        do {
                                fr_output.write(s+"\n");
                                System.out.println(s);
                                s=br_input.readLine();
                                //tokenizing
                                ArrayList<String> arrayList1= new ArrayList<>();
                                StringTokenizer tokens1 = new StringTokenizer(s," ");
                                while(tokens1.hasMoreTokens()){
                                        arrayList1.add(tokens1.nextToken());
                                }
                                //checking if 1st keyword is macro name
                                if(MNT.containsKey(arrayList1.get(0)))
                                {
                                        String macroName = arrayList1.get(0);
                                        String actualArgs = arrayList1.get(1);
                                        ArrayList<String> arrayList22= new
ArrayList<>();
```

```
                                        StringTokenizer tokens3 = new
StringTokenizer(actualArgs,",");
                                        while(tokens3.hasMoreTokens()){
                                                arrayList22.add(tokens3.nextToken());
                                        }
                                        for(int i=0;i<arrayList22.size();i++)
                                        {
                                                String curr = arrayList22.get(i);
                                                if(curr.contains("="))
                                                {
                                                        int positionEqu=curr.indexOf('=');
                                                        String param =
"&"+curr.substring(0, positionEqu);
                                                        String val =
curr.substring(positionEqu+1, curr.length());

        AlaTable.get(macroName).Arguments.put(param, val);
                                                }
                                                else
                                                {
                                                        for(String strKey :
AlaTable.get(macroName).Arguments.keySet() ){

        if(AlaTable.get(macroName).Arguments.get(strKey).equals("#"+i))
                                                        {

        AlaTable.get(macroName).Arguments.put(strKey, curr);
                                                        }
                                                   }
                                                }
                                        }
                                    }
                                }while(!s.equals("END"));
                                fr_output.write("END\n");

                                PreapareMDT();
                                PrepareMNT();
                                Ala_Table();
                                br_input.close();
```

```
                }
                catch(Exception e){
                        System.out.println(e);
                }
                finally {
                        try {
                                if (fr_output != null) {
                                        fr_output.flush();
                                        fr_output.close();
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }
}
```

Output_Files:-

1)macroOutput.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
```