ASSIGNMENT NO.2

(PASS II Assembler Code)

Input Files-

1)input_ic.txt

```
100  (01,AD)   (C,100)
100  (04,IS)  1   (S,1)
101  (01,IS)  2    (L,1)
102  (05,IS)  1   (S,2)
103  (02,IS)  3    (L,2)
 (05,AD)
104  6
105  1
106  (01,IS)  4    (L,3)
107  (01,DL)  (C,10)
 (05,AD)
117  5
118  (02,IS)  1    (L,4)
119  (02,DL)   (C,1)
 (02,AD)
120  1
```

2)littable.txt

```
1 ='6' 104
2 ='1' 105
3 ='5' 117
4 ='1' 120
```

3)symtable.txt

```
1 B 119
2 A 107
```

4)mot.txt

```
START   AD  01  0
END     AD  02  0
LTORG   AD  05  0
ADD     IS  01  1
SUB     IS  02  1
MULT    IS  03  1
MOVER   IS  04  1
MOVEM   IS  05  1
DS      DL  01  0
DC      DL  02  1
```

5)pass2.java

```
/*package pass2_assembler;*/

import java.io.*;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.StringTokenizer;

public class pass2 {
        static HashMap<String, ArrayList<String>> mnemonic = new
HashMap<String, ArrayList<String>>();
        static HashMap<Integer, ArrayList<String>> symboltab = new
HashMap<Integer, ArrayList<String>>();
        static HashMap<Integer, ArrayList<String>> littable = new
HashMap<Integer, ArrayList<String>>();
        static HashMap<String, Integer>registers = new HashMap<String,
Integer>();
        static int lc=0;

//Mnemonic Table
        public static void CreateMnemonicTable() throws
FileNotFoundException
        {
                FileReader fr=new
FileReader("/home/student/Downloads/snehal1/mot.txt");
```

```
            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());
                        }
                        String val=arrayList.get(0);
                        arrayList.remove(0);
                        mnemonic.put(val, arrayList);
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }

//Registers
            public static void initRegisters() {
                registers.put("AREG",1);
                registers.put("BREG",2);
                registers.put("CREG",3);
                registers.put("DREG",4);
            }

//Symbol Table
            private static void createSymbolTable() {

                FileReader fr = null;
                try {
                    fr = new
FileReader("/home/student/Downloads/snehal1/symtable.txt");
                } catch (FileNotFoundException e1) {
                    // TODO Auto-generated catch block
                    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());
                                }
                                Integer val=Integer.parseInt(arrayList.get(0));
                                ArrayList<String> temp= new ArrayList<>();

                                temp.add(arrayList.get(1));
                                temp.add(arrayList.get(2));
                                symboltab.put(val, temp);
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }

//Literal Table
        private static void createLiteralTable() {

                FileReader fr = null;
                try {
                        fr = new
FileReader("/home/student/Downloads/snehal1/littable.txt");
                } catch (FileNotFoundException e1) {
                        // TODO Auto-generated catch block
                        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());
                                }
                                int val=Integer.parseInt(arrayList.get(0));
                                ArrayList<String> temp= new ArrayList<>();

                                temp.add(arrayList.get(1));
                                temp.add(arrayList.get(2));
                                littable.put(val, temp);
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }

//Main
        public static void main(String[] args) throws FileNotFoundException {
                CreateMnemonicTable();
                initRegisters();
                createLiteralTable();
                createSymbolTable();

                //creating file pointers
                FileReader fr_input=new
FileReader("/home/student/Downloads/snehal1/input_ic.txt");
                FileWriter output = null;
                try {
                        output = new
FileWriter("/home/student/Downloads/snehal1/final_output.txt");
                } catch (IOException e1) {

                        e1.printStackTrace();
                }

                try {
                        BufferedReader br_input=new BufferedReader(fr_input);

                        String s = null;
```

```
                int lineno=0;

                while((s=br_input.readLine())!=null){

                        if(lineno==0)
                        {
                                lineno++;
                                continue;
                        }
                        ArrayList<String> arrayList= new ArrayList<>();
                        StringTokenizer tokens = new StringTokenizer(s,"
",false);

                        while(tokens.hasMoreTokens()){
                                arrayList.add(tokens.nextToken());
                        }
                        int tokenCount=arrayList.size();
                        String temp="";
                        System.out.println(s);
                        System.out.print("arraylist: ");
                        for(int i=0;i<arrayList.size();i++)
                        {
                                System.out.print(arrayList.get(i)+" ");
                        }

                        System.out.println(" ");
                        System.out.print(tokenCount+"=>");
                        if(tokenCount==1)
        //Assembler directive instructions
                        {
                                continue;
                        }
                        else if(tokenCount==2)
                        {

                                temp+=arrayList.get(0)+"  ";
                                temp+=arrayList.get(1)+"\n";
                                System.out.println(temp);
                        }
```

```
                        else if(tokenCount==3)
        //Declarative statements
                        {
                                temp+=arrayList.get(0)+"  ";
                                temp+=arrayList.get(1).substring(1, 3)+"  ";
                                String Slast_token=arrayList.get(2);
                                int index=0;
                                String si="";
                                for(int i=3; (Slast_token.charAt(i)!=')' &&
i<Slast_token.length()); i++)
                                {
                                        si+=Slast_token.charAt(i);
                                }
                                temp+=si+"\n";
                                System.out.println(temp);
                        }
                        else if(tokenCount==4)
        //Imperative statements with register
                        {
                                System.out.println("4th
token"+arrayList.get(3));
                                temp+=arrayList.get(0)+"  ";// address
                                temp+=arrayList.get(1).substring(1, 3)+"  ";
        //opcode

                                String Slast_token=arrayList.get(2);
        //operand 1

        if(Slast_token.charAt(0)=='1'||Slast_token.charAt(0)=='2'||Slast_token.cha
rAt(0)=='3'||Slast_token.charAt(0)=='4')
                                {
                                        temp+=Slast_token+" ";
                                }
                                else if(Slast_token.charAt(1)=='L')
        //literal
                                {
                                        int index=0;
                                        String si="";
                                        for(int i=3; (Slast_token.charAt(i)!=')'
&& i<Slast_token.length()); i++)
```

```
                                        {
                                                si+=Slast_token.charAt(i);
                                        }
                                        index=Integer.parseInt(si);
                                        temp+=littable.get(index).get(1)+"  ";

                                }
                                else if(Slast_token.charAt(1)=='S')
                                {
                                        int index=0;
                                        String si="";
                                        for(int i=3; (Slast_token.charAt(i)!=')'
&& i<Slast_token.length()); i++)
                                        {
                                                si+=Slast_token.charAt(i);
                                        }
                                        index=Integer.parseInt(si);
                                        temp+=symboltab.get(index).get(1)+"  ";
                                }
                                else if(Slast_token.charAt(1)=='C')
                                        {
                                                int index=0;
                                                String si="";
                                                for(int i=3;
(Slast_token.charAt(i)!=')' && i<Slast_token.length()); i++)
                                                {
                                                        si+=Slast_token.charAt(i);
                                                }
                                                index=Integer.parseInt(si);
                                                temp+=si;
                                        }

                                Slast_token=arrayList.get(3);
                                if(Slast_token.charAt(1)=='L')   //literal
                                {
                                        int index=0;
                                        String si="";
                                        for(int i=3; (Slast_token.charAt(i)!=')'
&& i<Slast_token.length()); i++)
                                        {
```

```
                                        si+=Slast_token.charAt(i);
                                    }
                                    index=Integer.parseInt(si);
                                    temp+=littable.get(index).get(1)+"  ";

                                }
                                else if(Slast_token.charAt(1)=='S')
                                {
                                    int index=0;
                                    String si="";
                                    for(int i=3; (Slast_token.charAt(i)!=')'
&& i<Slast_token.length()); i++)
                                    {
                                        si+=Slast_token.charAt(i);
                                    }
                                    index=Integer.parseInt(si);
                                    temp+=symboltab.get(index).get(1)+"  ";
                                }
                                else if(Slast_token.charAt(1)=='C')
                                    {
                                        int index=0;
                                        String si="";
                                        for(int i=3;
(Slast_token.charAt(i)!=')' && i<Slast_token.length()); i++)
                                        {
                                            si+=Slast_token.charAt(i);
                                        }
                                        index=Integer.parseInt(si);
                                        temp+=si;
                                    }
                                temp+="\n";
                                System.out.println(temp);
                            }
                        output.write(temp);
                    }
                }
                catch(Exception e){
                    System.out.println(e);
                }
                finally {
```

```
                        try {
                                if (output != null) {
                                        output.flush();
                                        output.close();
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }
}
```

Output Files:-

final_output.txt

```
100  04  1 119
101  01  2 104
102  05  1 107
103  02  3 105
104  6
105  1
106  01  4 117
107  01  10
117  5
118  02  1 120
119  02  1
120  1
```