Experiment No. 7

Aim: Design an implementation of pass II of two pass macro processor.

Requirement: Java(jdk-11) IDE and printout pages

Theory:

In Pass-II the macro calls are identified and the arguments are placed in the appropriate place and the macro calls are replaced by macro definitions.

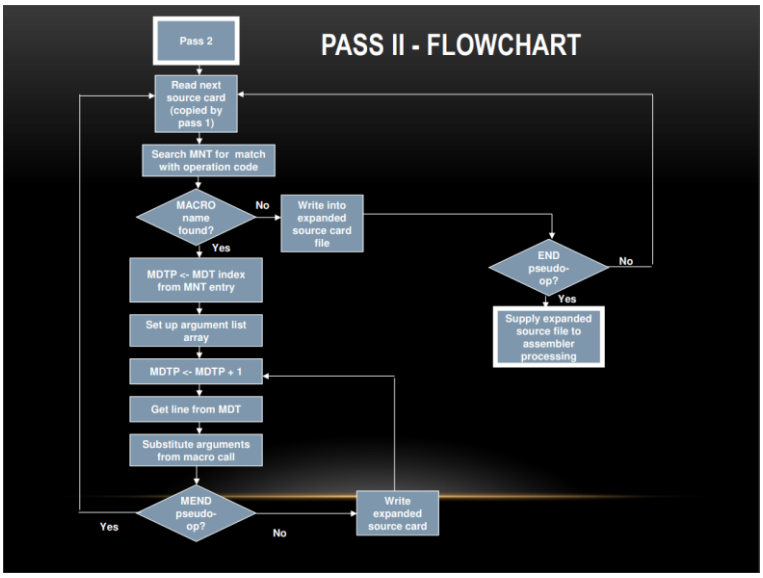
SPECIFICATION OF DATABASES

1. The input is from Pass1.
2. The output is expanded source to be given to assembler.
3. MDT and MNT are created by Pass1.
4. Macro-Definition Table Pointer (MDTP), used to indicate the next line of text to be used during macro expansion.
5. Argument List Array (ALA), used to substitute macro call arguments for the index markers in the stored macro-defns

ALGORITHM

1. This algorithm reads one line of i/p prog. at a time.
2. For each Line it checks if op-code of that line matches any of the MNT entry.
3. When match is found (i.e. when call is pointer called MDTF to corresponding macro defns stored in MDT.
4. The initial value of MDTP is obtained from MDT index field of MNT entry.
5. The macro expander prepares the ALA consisting of a table of dummy argument indices & corresponding arguments to the call.

6. Reading proceeds from the MDT, as each successive line is read, The values form the argument list one substituted for dummy arguments indices in the macro defn .
7. Reading MEND line in MDT terminates expansion of macro & scanning continues from the input file.
8. When END pseudo-op encountered, the expanded source program is given to the assembler.



Code:

```
import java.io.*;
import java.util.*;
import java.lang.*;

class pass2
{
   static int lc=0,mnti=0,mdti=0,i,j,li=0,alai=0,alac=0,alasi=0,prgi=0;
   static String[] mdt = new String[200];
   static String[] mnt = new String[100];
   static String[] ala = new String[100];
   static int[] mntin = new int[100];
   static int[] alain = new int[100];
   static int[][] alas = new int[100][3];
   static String[] prgstat = new String[200];

   public static int ifmacro(String name)
   {
      for(i=0;i<mnti;i++)
      {
         if(name.equals(mnt[i])) return i;
      }
      return -1;
```

```
    }

    public static void macroexp(int mi)
    {
        try
        {
            BufferedReader r;
            int ai=0,al=0;
            r = new BufferedReader(new FileReader("macro_definition_table.txt"));
            String line = r.readLine();
            String[] words = line.split("\\s+");
            while(true)
            {
                if(Integer.parseInt(words[0]) == mntin[mi]) break;
                line = r.readLine();
                words = line.split("\\s+");
            }
            //System.out.println(words[1]);
            for(i=0;i<alasi;i++)
            {
                if(alas[i][1]==mi)
                {
                    ai = alas[i][0];
                    al = alas[i][2];
                }
            }
            while(!words[1].equals("MEND"))
            {
                if(ifmacro(words[1])!=-1)
                {
                    line = r.readLine();
                    words = line.split("\\s+");
                    continue;
                }
                else
                {
                    for(i=ai;i<ai+al;i++)
                    {
                        //System.out.println("#"+Integer.toString(alain[i]));
                        if(words[2].contains("#"+Integer.toString(alain[i]))==true) words[2] =
words[2].replace("#"+Integer.toString(alain[i]),ala[i]);
                    }
                    String content = words[1]+" "+words[2];
                    prgstat[prgi] = content;
                    prgi++;
                }
                line = r.readLine();
                words = line.split("\\s+");
            }
        }
```

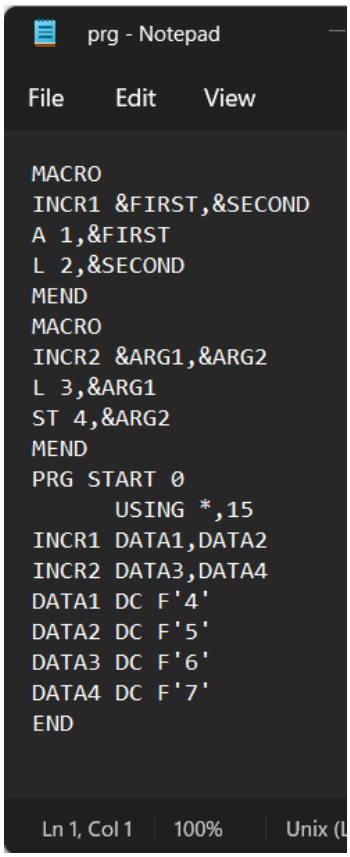
```
        catch (IOException e) { e.printStackTrace(); }
    }
    public static void main(String []args)
    {
        BufferedReader reader;
        try
        {
            reader = new BufferedReader(new FileReader("macro_name_table.txt"));
            String line = reader.readLine();
            while(line!=null)
            {
                String[] words = line.split("\\s+");
                mnt[mnti] = words[1];
                mntin[mnti]= Integer.parseInt(words[2]);
                mnti++;
                line = reader.readLine();
            }
            //for(i=0;i<mnti;i++) System.out.println(mnt[i]+" "+mntin[i]);
            reader = new BufferedReader(new FileReader("macro_definition_table.txt"));
            line = reader.readLine();
            while(line!=null)
            {
                String[] words = line.split("\\s+");
                mdt[mdti] = words[1];
                mdti++;
                line = reader.readLine();
            }
            //for(i=0;i<mdti;i++) System.out.println(i+" "+mdt[i]);
            reader = new BufferedReader(new FileReader("argument_list_array_pass_1.txt"));
            line = reader.readLine();
            while(line!=null)
            {
                String[] words = line.split("\\s+");
                alain[alai] = Integer.parseInt(words[1]);
                ala[alai] = words[2];
                alai++;
                line = reader.readLine();
            }

            reader = new BufferedReader(new FileReader("alas.txt"));
            line = reader.readLine();
            while(line!=null)
            {
                String[] words = line.split("\\s+");
                alas[alasi][0] = Integer.parseInt(words[0]);
                alas[alasi][1] = Integer.parseInt(words[1]);
                alas[alasi][2] = Integer.parseInt(words[2]);
                alasi++;
                line = reader.readLine();
            }
```

```
        //for(i=0;i<alai;i++) System.out.println(alain[i]+" "+ala[i]);
    }
    catch (IOException e) { e.printStackTrace(); }
    try
    {
        reader = new BufferedReader(new FileReader("prg_intermidiate.txt"));
        String line = reader.readLine();
        String[] words = line.split("\\s+");
        while (!line.trim().equals("END"))
        {
            int ai=0;
            int macval = ifmacro(words[0]);
            //System.out.println(words[0]+" "+macval);
            if(macval!=-1)
            {
                //System.out.println(macval);
                String[] op = words[1].split(",");
                for(i=0;i<alasi;i++)
                {
                    if(alas[i][1]==macval)
                    {
                        ai = alas[i][0];
                    }
                }
                for(i=ai;i<ai+op.length;i++)
                {
                    ala[i]=op[i-ai];
                    //System.out.println(ala[i]);
                }
                macroexp(macval);
            }
            else
            {
                prgstat[prgi] = line;
                prgi++;
            }
            line = reader.readLine();
            words = line.split("\\s+");
        }
        //for(i=0;i<prgi;i++) System.out.println(i+" "+prgstat[i]);
        reader.close();
    }
    catch (IOException e) { e.printStackTrace(); }
    try(OutputStream fw = new FileOutputStream("prg_expanded.txt"))
    {
        for(i=0;i<prgi;i++)
        {
            // program statement
            String content =prgstat[i]+System.getProperty("line.separator");
            fw.write(content.getBytes(),0,content.length());
```

```
            }
        }
        catch (IOException e) { e.printStackTrace(); }
        try(OutputStream fw = new FileOutputStream("argument_list_array_pass_2.txt"))
        {
            for(i=0;i<alai;i++)
            {
                // SR NO        argument index in mdt        argument name(Replaced with actual
arguments)
                String content = i+" "+alain[i]+" "+ala[i]+System.getProperty("line.separator");
                fw.write(content.getBytes(),0,content.length());
            }
        }
        catch (IOException e) { e.printStackTrace(); }
        System.out.println("Check file argument_list_array_pass_2.txt");
        System.out.println("Check file prg_expanded.txt");
    }
}
```

Output:

Input File:



Execution:

```
MINGW64:/c/Users/adnan/onedrive/Desktop/college/sem6/spcc/exp7

adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/Desktop/college/sem6/spcc/exp7 (main)
$ javac pass2.java

adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/Desktop/college/sem6/spcc/exp7 (main)
$ java pass2
Check file argument_list_array_pass_2.txt
Check file prg_expanded.txt

adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/Desktop/college/sem6/spcc/exp7 (main)
$
```

Argument list array pass2 and expanded program:

argument_list_array_p...

File Edit View

```
0 0 DATA1
1 1 DATA2
2 0 DATA3
3 1 DATA4
```

Ln 1, Col 1 100% Wi

prg_expanded - Note...

File Edit View

```
PRG START 0
      USING *,15
A 1,DATA1
L 2,DATA2
L 3,DATA3
ST 4,DATA4
DATA1 DC F'4'
DATA2 DC F'5'
DATA3 DC F'6'
DATA4 DC F'7'
```

Ln 1, Col 1 100% Wir

<u>Conclusion:</u> Thus we have Implemented program for pass 2 of two pass macro processor.