

PASS-1 MACROPROCESSOR :

MAIN PROGRAM:

```
import java.util.*;
import java.io.*;
class MACRO
{
    static String mnt[][]=new String[5][3]; //assuming 5 macros in 1
program
    static String ala[][]=new String[10][2]; //assuming 2 arguments in
each macro
    static String mdt[][]=new String[20][1]; //assuming 4 LOC for each
macro
    static int mntc=0,mdtc=0,alac=0;

    public static void main(String args[])
    {
        pass1();
        System.out.println("\n*****PASS-1 MACROPROCESSOR*****\n");
        System.out.println("MACRO NAME TABLE (MNT)\n");
        System.out.println("i macro loc\n");
        display(mnt,mntc,3);
        System.out.println("\n");
        System.out.println("ARGUMENT LIST ARRAY (ALA) for Pass1\n");
        display(ala,alac,2);
        System.out.println("\n");
        System.out.println("MACRO DEFINITION TABLE (MDT)\n");
        display(mdt,mdtc,1);
        System.out.println("\n");
    }

    static void pass1()
    {
        int index=0,i;
        String s,prev="",substring;
        try
        {
            BufferedReader inp = new BufferedReader(new FileReader("input.txt"));
            File op = new File("pass1_output.txt");
            if (!op.exists())
                op.createNewFile();
            BufferedWriter output = new BufferedWriter(new
FileWriter(op.getAbsolutePath()));
            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 formation
mnt[mntc][1]=str[0];
mnt[mntc++][2]=(++mdtc)+" ";
st=new StringTokenizer(str[1],","); //tokenizing the arguments
String string[]=new String[st.countTokens()];
for(i=0;i<string.length;i++)
{
string[i]=st.nextToken();
ala[alac][0]=alac+" "; //ala table formation
index=string[i].indexOf("=");
if(index!=-1)
ala[alac++][1]=string[i].substring(0,index);
else
ala[alac++][1]=string[i];
}
}
else //automatically eliminates tagging of arguments in definition
{ //mdt formation
index=s.indexOf("&");
substring=s.substring(index);
for(i=0;i<alac;i++)
if(ala[i][1].equals(substring))
s=s.replaceAll(substring,"#+ala[i][0]);
}
mdt[mdtc-1][0]=s;
}
mdt[mdtc-1][0]=s;
}
else
{
output.write(s);
output.newLine();
}
}
output.close();
}
catch(FileNotFoundException ex)
{
System.out.println("UNABLE TO END FILE ");
}
catch(IOException e)
{
e.printStackTrace();
}
}
static void display(String a[][],int n,int m)

```

```

{
int i,j;
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
System.out.print(a[i][j]+" ");
System.out.println();
}
}
}

```

/* INPUT

```

START
MACRO
INCR &ARG3 &ARG2
ADD AREG &ARG1
MOVER BREG &ARG1
MEND

```

```

MACRO
PVG &ARG2 &ARG1
SUB AREG &ARG2
MOVER CREG & ARG1
MEND

```

```

INCR
DECR
DATA2
END
*/

```

/* OUTPUT

```

pvgcoen-3@pvgcoen3-ThinkCentre-M700:~/AA$ javac MACRO.java
pvgcoen-3@pvgcoen3-ThinkCentre-M700:~/AA$ java MACRO

```

```

*****PASS-1 MACROPROCESSOR*****

```

```

MACRO NAME TABLE (MNT)

```

```

i  macro  loc

```

```

1 INCR 1
2 PVG 5

```

```

ARGUMENT LIST ARRAY(ALA) for Pass1

```

```

0 &ARG3
1 &ARG2

```

MACRO DEFINITION TABLE (MDT)

```
INCR &ARG3 &ARG2
ADD AREG &ARG1
MOVER BREG &ARG1
MEND
PVG &ARG2 &ARG1
SUB AREG #1
MOVER CREG & ARG1
MEND
```

* /