Experiment No. 6

Aim: Design an implementation of pass 1 of two pass macro processor.

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

Theory:

In Pass-I the macro definitions are searched and stored in the macro definition table and the entry is made in macro name table.

SPECIFICATION OF DATABASES

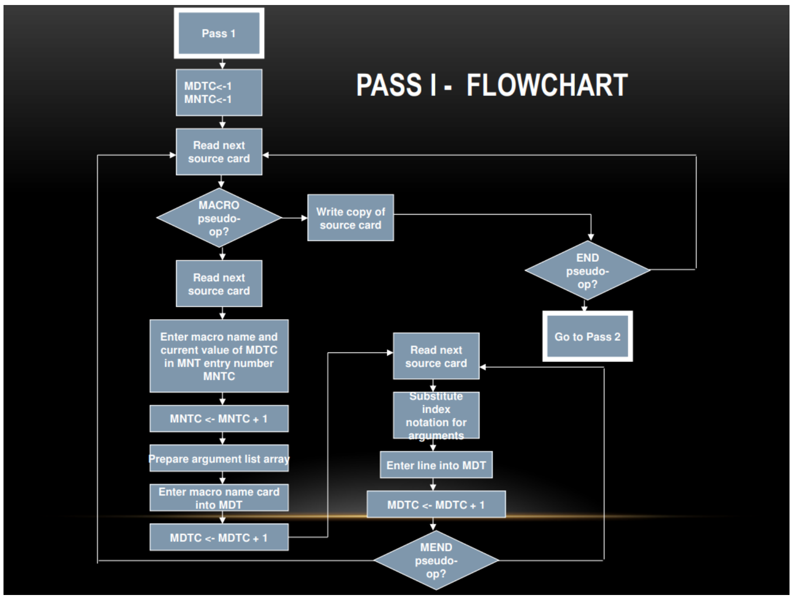
1. The input macro source program.
2. The output macro source program to be used by Pass2.
3. Macro-Definition Table (MDT), to store the body of macro defns .
4. Macro-Definition Table Counter (MDTC), to mark next available entry MDT.
5. Macro- Name Table (MNT) - store names of macros.
6. Macro Name Table counter (MNTC) - indicate the next available entry in MNT.
7. Argument List Array (ALA) - substitute index markers for dummy arguments before storing a macro-deff.

ALGORITHM

1. Pass1 of macro processor makes a line-by-line scan over its input.
2. Set MDTC = 1 as well as MNTC = 1.
3. Read next line from input program.
4. If it is a MACRO pseudo-op, the entire macro definition except this (MACRO) line is stored in MDT. The name is entered into Macro Name Table along with a pointer to the first location of MDT entry of the definition.

5. When the END pseudo-op is encountered all the macro-defns have been processed, so control is transferred to pass2

FLOWCHART



<u>**Code:**</u>

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

class pass1
{
   public static void main(String []args)
   {
      BufferedReader reader;
      int lc=0,mnti=0,mdti=0,i,j,li=0,alai=0,alac=0,alasi=0,prgi=0;
      String[] mdt = new String[200];
      String[] mnt = new String[100];
      String[] ala = new String[100];
      int[] mntin = new int[100];
      int[] alain = new int[100];
      int[][] alas = new int[100][3];
      String[] prgstat = new String[200];
      try
      {
         reader = new BufferedReader(new FileReader("prg.txt"));
         String line = reader.readLine();
         String[] words = line.split("\\s+");
         //System.out.println(sym[0]+" "+symtab[0][0]);
```

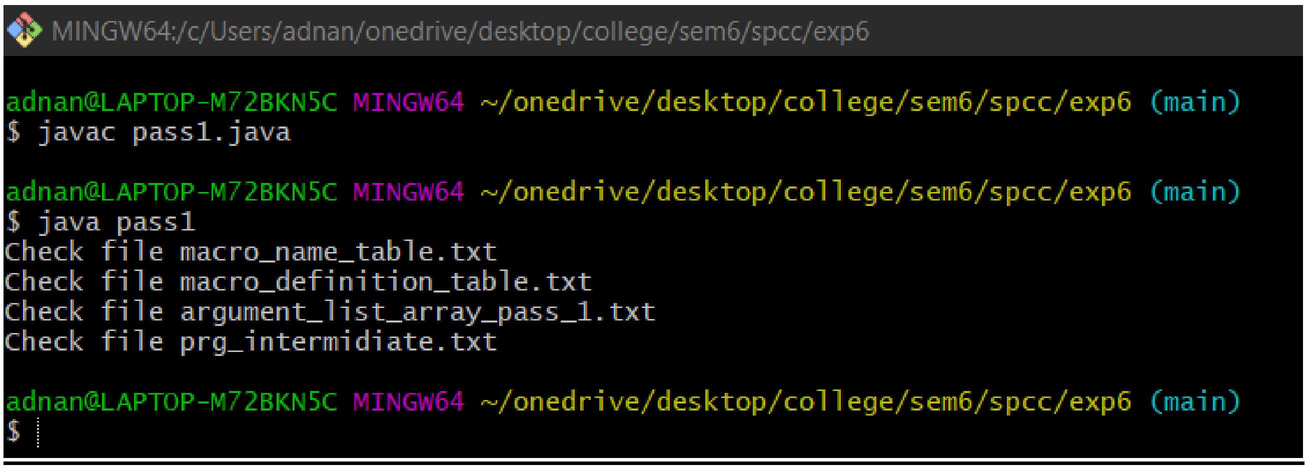
```
        while (!line.trim().equals("END"))
        {
           if(words[0].equals("MACRO"))
           {
              li=0;alac=0;
              //System.out.println("yes");
              while(!words[0].equals("MEND"))
              {
                 line = reader.readLine();
                 words = line.split("\\s+");
                 if(li==0)
                 {
                    mnt[mnti] = words[0];
                    String[] op = words[1].split(",");
                    alas[alasi][0] = alai; alas[alasi][1] = mnti;alas[alasi][2]=op.length;
                    for(i=0;i<op.length;i++)
                    {
                       ala[alai] = op[i];
                       alain[alai] = alac;
                       alac++;
                       alai++;
                    }
                    mntin[mnti] = mdti;
                    mdt[mdti] = line;
                    mnti++;
                    mdti++;
                    alasi++;
                    li++;
                 }
                 else
                 {
                    for(i=alas[alasi-1][0];i<alai;i++)
                    {
                       if(line.contains(ala[i])==true) line =
line.replace(ala[i],"#"+Integer.toString(alain[i]));
                    }
                    mdt[mdti] = line;
                    mdti++;
                 }
              }
              //System.out.println(line);
           }
           else
           {
              prgstat[prgi] = line;
              prgi++;
           }
           line = reader.readLine();
           words = line.split("\\s+");
        }
        reader.close();
        prgstat[prgi] = line;
        prgi++;
        //for(i=0;i<mnti;i++) System.out.println(mnt[i]+" "+mntin[i]);
        //for(i=0;i<mdti;i++) System.out.println(i+" "+mdt[i]);
        //for(i=0;i<alai;i++) System.out.println(alain[i]+" "+ala[i]);

        try(OutputStream fw = new FileOutputStream("macro_name_table.txt"))
        {
           for(i=0;i<mnti;i++)
           {
              // SR NO       macro name      MDTindex
              String content = i+" "+mnt[i]+"
"+mntin[i]+System.getProperty("line.separator");
              fw.write(content.getBytes(),0,content.length());
           }
        }
        catch (IOException e) { e.printStackTrace(); }
        System.out.println("Check file macro_name_table.txt");
        try(OutputStream fw = new FileOutputStream("macro_definition_table.txt"))
        {
           for(i=0;i<mdti;i++)
           {
              // SR NO       macro definition
              String content = i+" "+mdt[i]+System.getProperty("line.separator");
              fw.write(content.getBytes(),0,content.length());
           }
        }
        catch (IOException e) { e.printStackTrace(); }
        System.out.println("Check file macro_definition_table.txt");
        try(OutputStream fw = new FileOutputStream("argument_list_array_pass_1.txt"))
        {
           for(i=0;i<alai;i++)
           {
              // SR NO       argument index in mdt      argument name
              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_1.txt");
        try(OutputStream fw = new FileOutputStream("prg_intermidiate.txt"))
        {
           for(i=0;i<prgi;i++)
           {
              // program line
              String content = prgstat[i]+System.getProperty("line.separator");
              fw.write(content.getBytes(),0,content.length());
           }
        }
```

```
            catch (IOException e) { e.printStackTrace(); }
            System.out.println("Check file prg_intermidiate.txt");
            // This file is for background processing. This stores the start index of arguments of a
macro in the ala as the ala is the same for all macros.
            try(OutputStream fw = new FileOutputStream("alas.txt"))
            {
                for(i=0;i<alasi;i++)
                {
                    // alastartindex        mntindex        number of arguments in that macro
                    String content = alas[i][0]+" "+alas[i][1]+"
"+alas[i][2]+System.getProperty("line.separator");
                    fw.write(content.getBytes(),0,content.length());
                }
            }
            catch (IOException e) { e.printStackTrace(); }
        }
        catch (IOException e) { e.printStackTrace(); }
    }
}
```

Output:

Execution:



MINGW64:/c/Users/adnan/onedrive/desktop/college/sem6/spcc/exp6

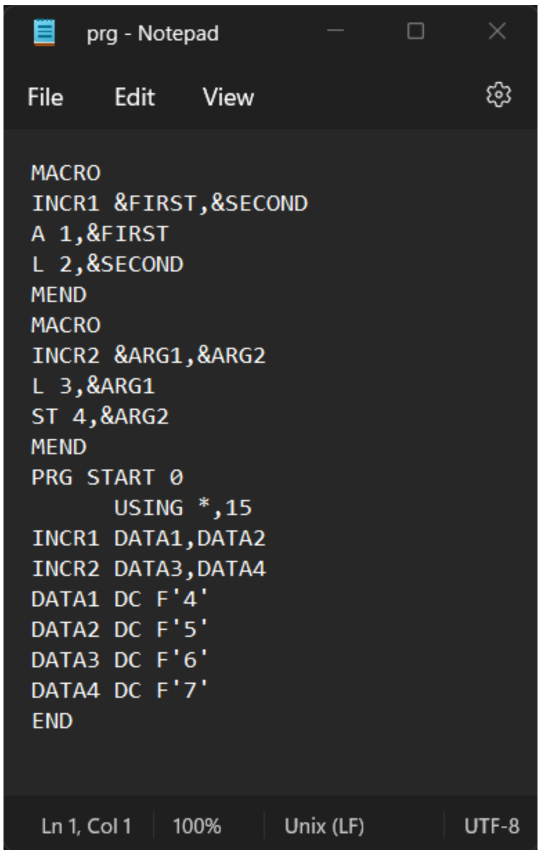
```
adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/desktop/college/sem6/spcc/exp6 (main)
$ javac pass1.java

adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/desktop/college/sem6/spcc/exp6 (main)
$ java pass1
Check file macro_name_table.txt
Check file macro_definition_table.txt
Check file argument_list_array_pass_1.txt
Check file prg_intermidiate.txt

adnan@LAPTOP-M72BKN5C MINGW64 ~/onedrive/desktop/college/sem6/spcc/exp6 (main)
$ 
```

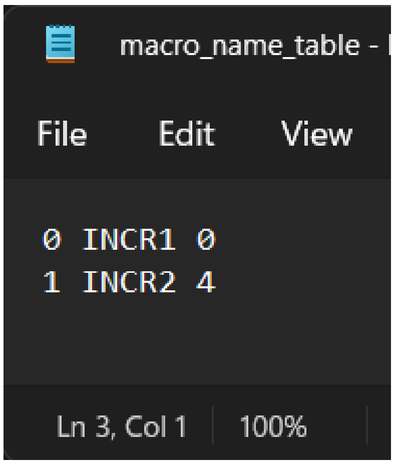
76_Adnan Shaikh

Input File:

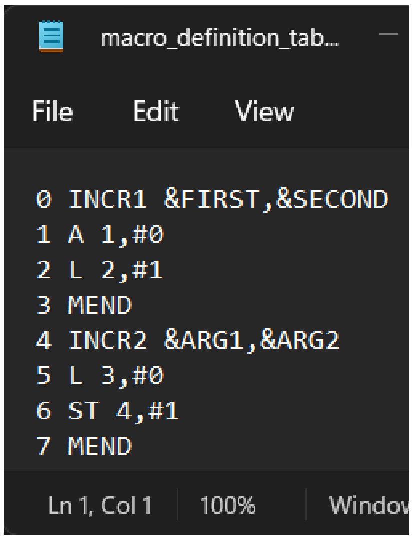


```
MACRO
INCR1 &FIRST,&SECOND
A 1,&FIRST
L 2,&SECOND
MEND
MACRO
INCR2 &ARG1,&ARG2
L 3,&ARG1
ST 4,&ARG2
MEND
PRG START 0
      USING *,15
INCR1 DATA1,DATA2
INCR2 DATA3,DATA4
DATA1 DC F'4'
DATA2 DC F'5'
DATA3 DC F'6'
DATA4 DC F'7'
END
```

Macro name table and definition table:

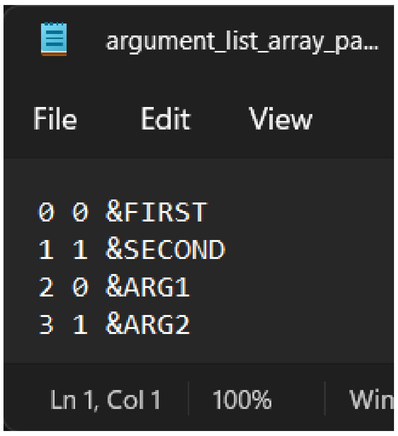


```
0 INCR1 0
1 INCR2 4
```

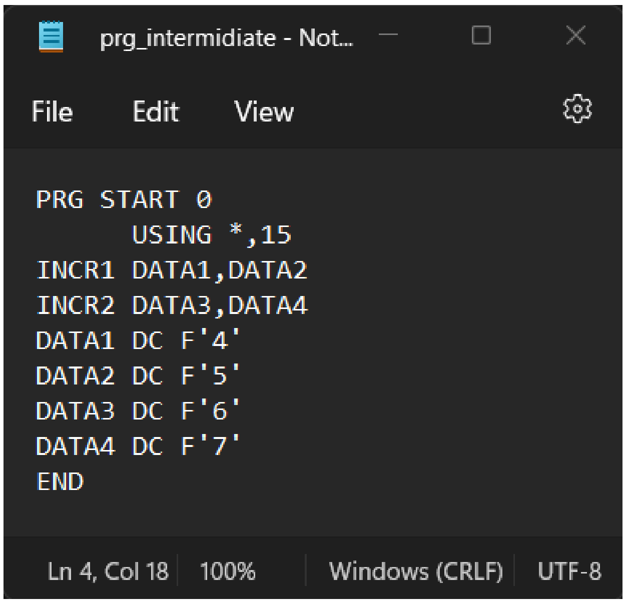


```
0 INCR1 &FIRST,&SECOND
1 A 1,#0
2 L 2,#1
3 MEND
4 INCR2 &ARG1,&ARG2
5 L 3,#0
6 ST 4,#1
7 MEND
```

Argument list array pass1 and program intermediate:



```
0 0 &FIRST
1 1 &SECOND
2 0 &ARG1
3 1 &ARG2
```


prg_intermidiate - Not...
File Edit View
Ln 4, Col 18 100% Windows (CRLF) UTF-8

```
PRG START 0
      USING *,15
INCR1 DATA1,DATA2
INCR2 DATA3,DATA4
DATA1 DC F'4'
DATA2 DC F'5'
DATA3 DC F'6'
DATA4 DC F'7'
END
```

Conclusion: Thus we have Implemented program for pass 1 of two pass Macro Processor.