



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

Course - System Programming and Compiler Construction (SPCC)

UID	2021300101
Name	Adwait Purao
Class and Batch	TE Computer Engineering - Batch B
Date	01/05/24
Lab #	9
Aim	Write a program to design a two pass Macro Processor
Objective	Implement a two-pass Macro Processor to facilitate macro definition, invocation, and expansion, enhancing code development efficiency in system programming.
Theory	<p>Specification of the Problem</p> <ul style="list-style-type: none">• In Pass-I the macro definitions are searched and stored in the macro definition table and the entry is made in macro name table [1]• 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. <p>Specification of Databases</p> <p>Pass 1:</p> <ul style="list-style-type: none">• The input macro source program.• The output macro source program to be used by Pass 2.• Macro-Definition Table (MDT): To store the body of macro definitions.• Macro-Definition Table Counter (MDTC): To mark the next available entry in MDT.• Macro-Name Table (MNT): To store names of macros.[1]• Macro Name Table Counter (MNTC): To indicate the next available entry in MNT.• Argument List Array (ALA): To substitute index markers for dummy arguments before storing a macro definition.[1] <p>Pass 2:</p> <ul style="list-style-type: none">• The input is from Pass 1.• The output is expanded source to be given to the assembler.• MDT and MNT are created by Pass 1.• Macro-Definition Table Pointer (MDTP): Used to indicate the next line of text to be used during macro-expansion.[1]



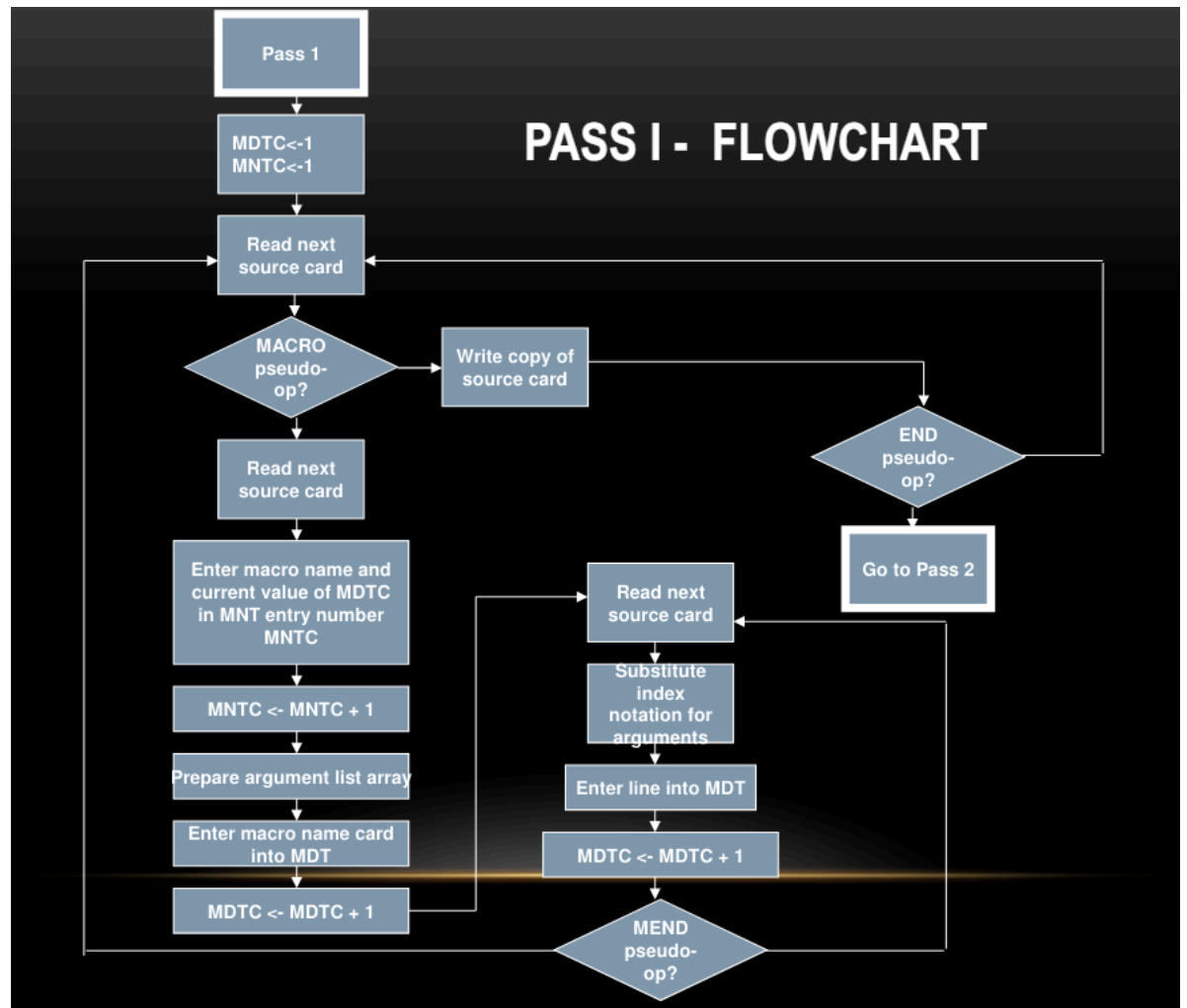
BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

- Argument List Array (ALA): Used to substitute macro call arguments for the index markers in the stored macro definitions.

Algorithm for Pass 1

- Pass 1 of the macro processor makes a line-by-line scan over its input.
- Set MDTC = 1 as well as MNTC = 1.[1]
- Read the next line from the input program.
- If it is a MACRO pseudo-op, the entire macro definition except this (MACRO) line is stored in MDT.
- The name is entered into the Macro Name Table along with a pointer to the first location of MDT entry of the definition.
- When the END pseudo-op is encountered, all the macro definitions have been processed, so control is transferred to Pass 2.





BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

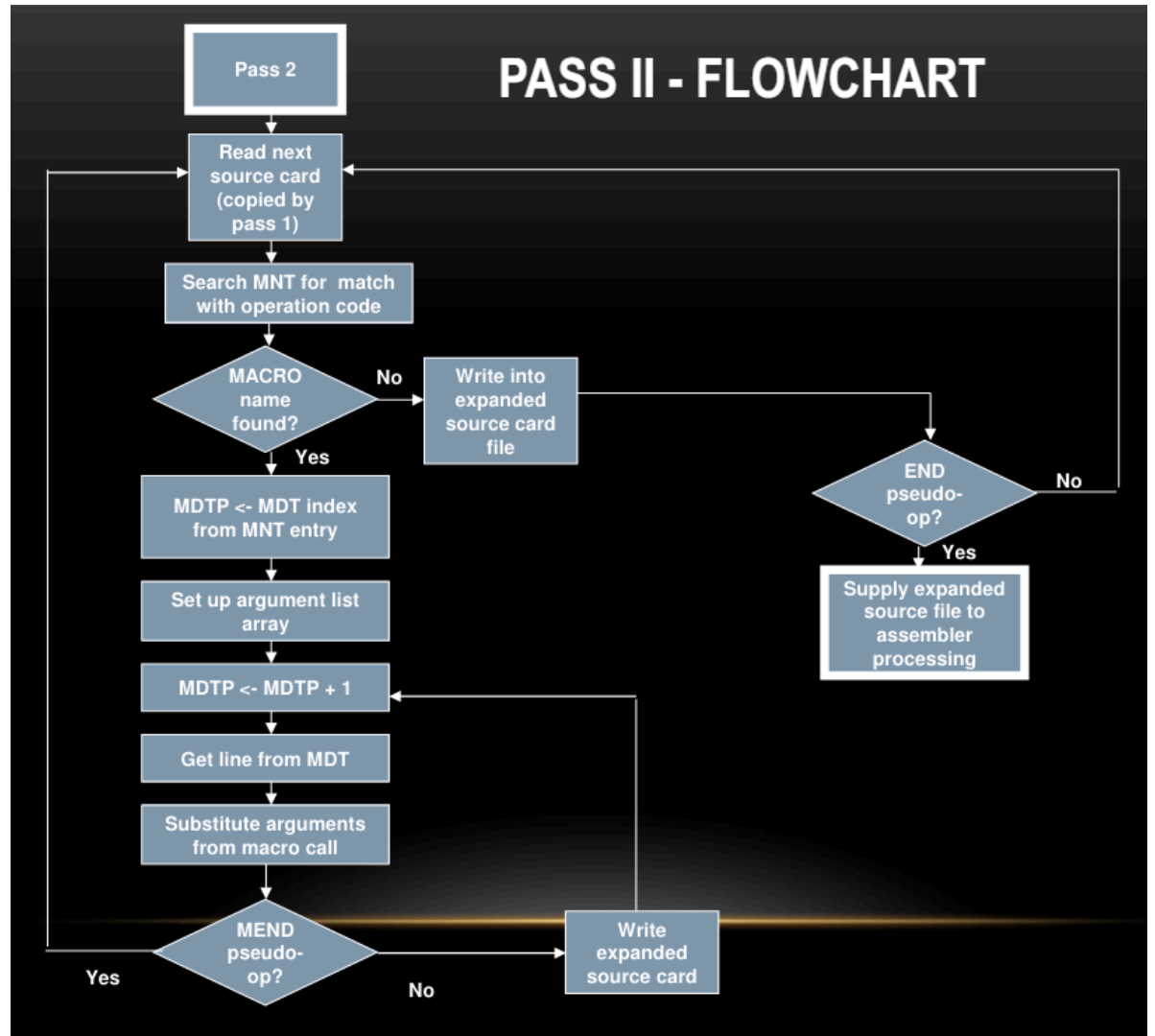
Algorithm for Pass 2

1. This algorithm reads one line of input program at a time.
2. For each line, it checks if the opcode of that line matches any of the entries in the Macro Name Table (MNT).[1]
3. When a match is found (i.e., when a call is pointed), MDTP points to the corresponding macro definition stored in the Macro Definition Table (MDT).
4. The initial value of MDTP is obtained from the MDT index field of the MNT entry.
5. The macro expander prepares the Argument List Array (ALA) consisting of a table of dummy argument indices and corresponding arguments to the call.
6. Reading proceeds from the MDT, as each successive line is read. The values from the argument list are substituted for dummy argument indices in the macro definition.[1]
7. Reading the MEND line in MDT terminates the expansion of the macro, and scanning continues from the input file.
8. When the END pseudo-op is encountered, the expanded source program is given to the assembler.



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering



Applications

Language Expansion: Macro processors are often used for language expansion. This involves defining new language constructs that can be expressed in terms of existing language components.[2]

Systematic Text Replacements: Macro processors can be used for systematic text replacements that require decision making. This can be particularly useful in large codebases where certain patterns need to be updated or refactored.[2]

Text Reformatting: Macro processors can also be used for text reformatting, such as conditional extraction of material from an HTML file. This can be useful in a variety of contexts, from web development to data analysis.[2]



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

Macro Assembler: It performs expansion of each macro call in a program into a sequence of assembly language statements and also assembles the resultant assembly language program.

Macro Pre-processor: It only processes the macro call. Other statements are processed with the help of an assembler. A macro pre-processor merely performs expansion of macros in a program.

**Implementation /
Code**

MacroProcessor.java

```
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;

public class MacroProcessor {

    private static HashMap<String,Integer> mnt = new
HashMap<String,Integer>();
    private static ArrayList<String> mdt = new ArrayList<String>(),ala
= new ArrayList<String>();
    private static String FileContent;
    private static Pass1 pass1 = new Pass1();
    private static Pass2 pass2 = new Pass2();

    public static void main(String[] args) throws IOException {

        System.out.println("***** Two Pass Macro-Processor
*****\n\n");
        /*System.out.println("Enter Name of File containing assembler
program\n");
        String FileName = scan.nextLine();*/
        FileContent =
getContentFrom("C:\\Users\\asapur\\OneDrive\\SPCC\\EXPERIMENTS\\exp9\\J
ava\\asscode.txt");
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
//System.out.println(FileContent +"Contents of MNT are : \n\n");

pass1.start();

System.out.println("Contents of MNT are : \n\n");
System.out.println(mnt);
System.out.println("\n\nContents of MDT are : \n\n");
System.out.println(mdt);
System.out.println("\n\nContents of ALA are : \n");
System.out.println(ala);

pass2.start();

}

private static String getContentFrom(String fileName) throws
IOException {

    String content;
    File file = new File(fileName);
    FileInputStream fis = new FileInputStream(file);
    DataInputStream dis = new DataInputStream(fis);

    int f_count = fis.available();

    byte[] b = new byte[f_count];
    dis.readFully(b);

    content = new String(b);

    dis.close();
    return content;
}

public static String getFileContent() {
    return MacroProcessor.FileContent;
}
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
}

public static Integer getIndexFromMnt(String name) {

    if(mnt.containsKey(name))
        return mnt.get(name);
    return -1;
}

public static Boolean mntContains(String name) {

    if(mnt.containsKey(name))
        return true;

    return false;
}

public static void updateMnt(String name, int mdtp) {
    mnt.put(name, mdtp);
}

public static Boolean inMnt(String name) {

    if(mnt.get(name) == null) {
        return false;
    }

    return true;
}

public static String getLineFromMdt(int index) {
    return mdt.get(index);
}

public static void updateMdt(String line) {
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
mdt.add(line);  
}  
  
public static String getLineFromAla(int index) {  
    return ala.get(index);  
}  
  
public static void updateAla(String arg) {  
    ala.add(arg);  
}  
  
public static String getArgumentAtIndex(int index) {  
    return ala.get(index);  
}  
  
public static int getAlaSize() {  
    return ala.size();  
}  
  
public static void initAla() {  
    ala = new ArrayList<String>();  
}  
  
public void prepareALA(String[] lineContent2) {  
  
    if(lineContent2[1].matches(".,.+")) {  
        String[] commaSepr = lineContent2[1].split(",");  
        for(int i=0; i<commaSepr.length; i++) {  
            updateAla(commaSepr[i]);  
        }  
        return;  
    }  
    for(int i=1; i<lineContent2.length; i++) {  
        updateAla(lineContent2[i]);  
    }  
}
```




BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
}  
  
}
```

Pass1.java

```
public class Pass1 extends MacroProcessor{  
  
    private String[] fileContent,lineContent;  
    private String currentLine;  
    private int mdtp = 0;  
  
    public void start() {  
  
        int i=1;  
        fileContent = getFileContent().split("\\r?\\n");  
        currentLine = fileContent[0];  
  
        while(!currentLine.equals("END")) {  
  
            lineContent = currentLine.split("\\s+");  
  
            if(lineContent[0].equals("MACRO")) {  
  
                currentLine = fileContent[i++];  
                lineContent = currentLine.split("\\s+");  
                if(inMnt(lineContent[0])) {  
                    System.out.println(lineContent[0] + " already  
exists !!!!!, Skipping this MACRO");  
                    continue;  
                }  
                updateMnt(lineContent[0],mdtp);  
                prepareALA(lineContent);  
                updateMdt(this.currentLine);  
                mdtp++;  
                currentLine = fileContent[i++];  
            }  
        }  
    }  
}
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
        while(!currentLine.equals("MEND")) {

            String substitutedLine =
substitute_index_notation(currentLine);
            updateMdt(substitutedLine);
            mdtp++;
            currentLine = fileContent[i++];

        }

        updateMdt("MEND");
        mdtp++;
        initAla();
        continue;
    }
    currentLine = fileContent[i++];

}

}

private String substitute_index_notation(String currentLine2) {

    lineContent = currentLine.split("\\s+");
    String subsLine = "";
    Boolean isFound = false;
    int alaSize = getAlaSize();
    for(String currentWord : lineContent) {

        if(currentWord.matches("."+",".+")) {

            String[] commaSepr = currentWord.split(",");
            for(String comma : commaSepr) {
                for(int i=0; i<alaSize; i++) {
                    if(comma.equals(getArgumentAtIndex(i))) {
                        subsLine = subsLine + i + " ";
                    }
                }
            }
        }
    }
}
```



**BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY**
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
                isFound = true;
                break;
            }
        }
    }

    }else {
        for(int i=0; i<alaSize; i++) {
            if(currentWord.equals(getArgumentAtIndex(i))) {
                subsLine = subsLine + i + " ";
                isFound = true;
                break;
            }
        }
        if(!isFound)
            subsLine = subsLine + currentWord + " ";
    }

}

return subsLine;
}

}
```

Pass2.java

```
public class Pass2 extends MacroProcessor {

    private String newFileContent = "";
    private String finalFileContent = "";
    private int mdtp;

    public void start() {

        updateFileContent();
    }
}
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
String[] lineContent = newFileContent.split("\\r?\\n");

for(String currentLine : lineContent) {

    String[] line = currentLine.split("\\s+");

    if(macroNameFoundAt(line)) {

        //System.out.println("macro name found " + line[0]);

        mdtp = getIndexFromMnt(line[0]);
        mdtp++;

        prepareALA(line);

        String mdtLine = getLineFromMdt(mdtp);
        while(!mdtLine.equals("MEND")) {

            String[] mdtSpace = mdtLine.split("\\s+");

            finalFileContent = finalFileContent + mdtSpace[0]
+ " ";

            for(int i=1; i<mdtSpace.length; i++) {
                //System.out.println(mdtSpace[i]);
                mdtSpace[i] =
getLineFromAla(Integer.parseInt(mdtSpace[i]));
                finalFileContent = finalFileContent +
mdtSpace[i] + " ";
            }
            //mdtLine = mdtSpace.toString();
            finalFileContent = finalFileContent + "\\n";
            mdtLine = getLineFromMdt(++mdtp);
        }

        initAla();
    }
}
```



**BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY**
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

```
        continue;

    }

    finalFileContent = finalFileContent + currentLine + "\n";

}

System.out.println("\n\nFinal content of file after processing
: \n\n"+finalFileContent);

}

private boolean macroNameFoundAt(String[] line) {

    String macroName = "";
    macroName = line[0];

    if(mntContains(macroName))
        return true;

    return false;
}

private void updateFileContent() {

    String[] temp = getFileContent().split("\\r?\\n");
    int i = 0;
    while(!temp[i].contains("START"))
        i++;

    for(;i<temp.length;i++) {

        newFileContent = newFileContent + temp[i] + "\n";

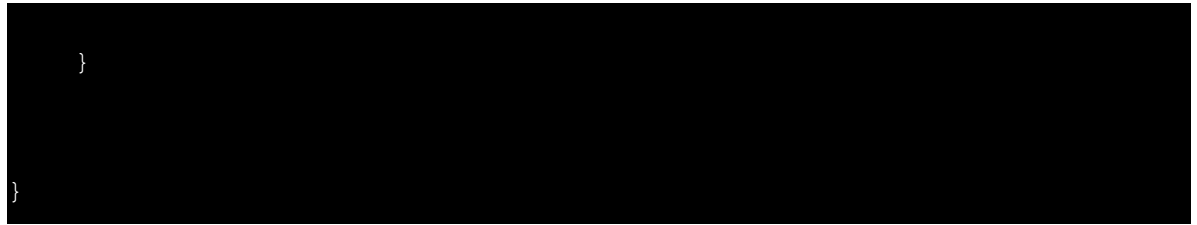
    }

}
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering



asscode.txt (Input file)

**MACRO
INCR &X,&Y
MOVER &X,&Y
MEND**

**MACRO
DECR &X,&Y
MOVEM &Y,&X
MEND**

**MACRO
PRN &X,&Y,&Z
MOVER &X,&Y
MOVEM &Y,&Z
PRINT &X,&Y
PRINT &Z
MEND**

**START 100
READ N1
INCR N1,N2
DECR N1,N2
READ N2
INCR N1,N2
INCR N3,N4
DECR N3,N4
PRN A1,A2,Z9
STOP
END**



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

Output

```
aspur@LAPTOP-LG4IQEFB MINGW64 ~/OneDrive/SPCC/EXPERIMENTS/exp9/Java
$ javac MacroProcessor.java

aspur@LAPTOP-LG4IQEFB MINGW64 ~/OneDrive/SPCC/EXPERIMENTS/exp9/Java
$ java MacroProcessor
***** Two Pass Macro-Processor *****

Contents of MNT are :

{INCR=0, PRN=6, DECR=3}

Contents of MDT are :

[INCR &X,&Y, MOVER 0 1 , MEND, DECR &X,&Y, MOVEM 1 0 , MEND, PRN &X,&Y,&Z, MOVER 0 1 ,
MOVEM 1 2 , PRINT 0 1 , PRINT 2 , MEND]

Contents of ALA are :

[]

Final content of file after processing :
```



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]

Department of Computer Engineering

Final content of file after processing :

```
START 100
READ N1
MOVER N1 N2
MOVEM N2 N1
READ N2
MOVER N1 N2
MOVER N3 N4
MOVEM N4 N3
MOVER A1 A2
MOVEM A2 Z9
PRINT A1 A2
PRINT Z9
STOP
END
```

```
aspur@LAPTOP-LG4IQEFB MINGW64 ~/OneDrive/SPCC/EXPERIMENTS/exp9/Java
$ █
```

Conclusion

In this experiment, we concentrated on crafting a two-pass Macro Processor in Python, aiming to streamline code development processes by enabling the definition and utilization of macros. Utilizing a two-pass methodology, the processor adeptly recognizes macro definitions, expands macro calls, and seamlessly integrates the expanded code into the primary source code. We delved into core macro processing principles, encompassing definition, invocation, expansion, and related functionalities. Through the effective execution of the furnished example code, we showcased the practical application of macro processing techniques, underscoring their importance in system programming and compiler development.

References

- [1] *Two Pass Macroprocessor | PDF | Macro (Computer Science) | Algorithms.* (n.d.). Scribd. Retrieved May 1, 2024, from <https://www.scribd.com/document/286294135/Two-Pass-Macroprocessor>
- [2] *Macro Processor.* (2019, July 10). GeeksforGeeks. <https://www.geeksforgeeks.org/macro-processor/>