ASSIGNMENT A-3

TITLE: Pass I of a two pass macro processor.

PROBLEM STATEMENT:

Design suitable data structures and implement pass-I of a two-pass macro-processor using OOP features in Java.

OBJECTIVE:

- Understand the internals of language translators
- Handle tools like LEX and YACC
- Understand the operating system internals and functionalities with implementation point of view

S/W PACKAGES AND HARDWARE REQUIREMENTS:

- 64-bit open source Linux (Fedora 20)
- Eclipse IDE, JAVA
- 64-bit architecture I3 or I5 machines
- LEX and YACC

OUTCOME:

We will be able to:

- Identify and create the MDT, MNT
- Pass the parameters to the macro
- To separate the macro definitions from the source code

THEORY:

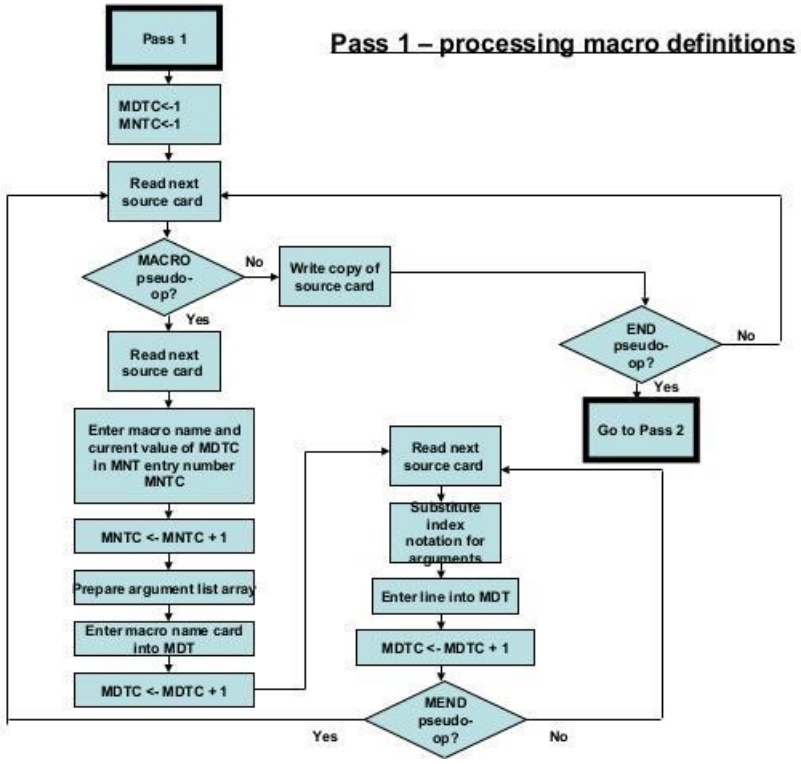
Macro processing feature allows the programmer to write shorthand version of a program (modular programming). The macro processor replaces each macro invocation with the corresponding sequence of statements i.e. macro expansion.

Tasks done by the macro processor

- Recognize macro definitions
- Save the macro definition recognize macro calls
- Expand macro calls

Tasks in pass I of a two pass macro processor

- Recognize macro definitions
- Save the macro definition(Create MDT,MNT,ALA)Perform processing of assembler directives(e.g. BYTE, RESW directives can affect address assignment)
- Create intermediate code file.



CONCLUSION:

Thus, we successfully implemented Lexical Analysis to count number of words, lines and characters.