Name - Vasu Kalariya
Roll - PE29
Sub - SSC

# Lab Assignment - 3

**Assignment Title:** Design of Pass 1 of Two Pass Macroprocessor

**Aim:** Design suitable data structure & implement pass 1 of Two Pass Macroprocessor.

**Objective:**

Design suitable data structure & implement pass 1 of Two Pass Macroprocessor. Input should consist of a one macro definition and one macro call and few assembly language instructions.

**Theory:**

→ Description about the Macroprocessor.

A macro instruction is the notational convenience for the programmer. For every occurrence of a macro the whole macro body or macro block of statement gets expanded in the main source code. Thus Macro instructions make writing code more convenient. Macro represents a group of commonly used statements in the source programming language. Macro Processor replaces each macro instruction with the expansion of macros. A macro consist of macro name, set of formal parameters and a body of code. Macro name with a set of actual parameters, is replaced by some code, generated from macro body. This is called macro expansion.

→ Data Structure required for 2 pass macroprocessor.

There are three main data structure involved in ~~our~~ macroprocessor

- MDT TAB : The macro defination themselves are stored in a defination table (MDT TAB), which contains the macro prototype.
- MNT TAB : The ~~non~~ macro names are entered into MNT TAB which serves as an index of MDT TAB
- ALA : It is an argument table which is used during the expansion of macro invocations

→ Flowchart for Pass 1
{At the end}.

Algorithm for Pass 1:

1 Initialization of counter for MDT & MNT
2 Read next instruction (and divide it into its various field as label. mnemonic)
3 if opcode = MACRO goto Step 5
else go to step 4
4 (a) Write copy of instruction to output of Pass 1
(b) Check whether opcode = END or not
(c) If OPCODE "!" END goto Step 2
(d) if OPCODE = END goto Pass 2 i.e End of this algo. for Pass 1
5 (a) Read Next instruction
(b) Enter <macro-name, MDTC> into MNT at MNTC
MDTC is
MDTC is entered in MNT at available row
(c) MNTC - MNTC + 1
(d) Prepare Argument List Array
(e) Enter macroname instruction in MDT at MDTC
(i) MDTC - MDTC + 1.
6 (a) Read next card
(b) Substitute Index notations for dummy arguments
(c) Enter this instruction into MDT.

(d) MDTC - MDTC + 1

(e) If OPCODE of this instruction is MEND then goto Step 2.
else goto Step 6.a

Input : Assembly Language Program

Output :

1] Program without Macro Defination (Pass - 1)

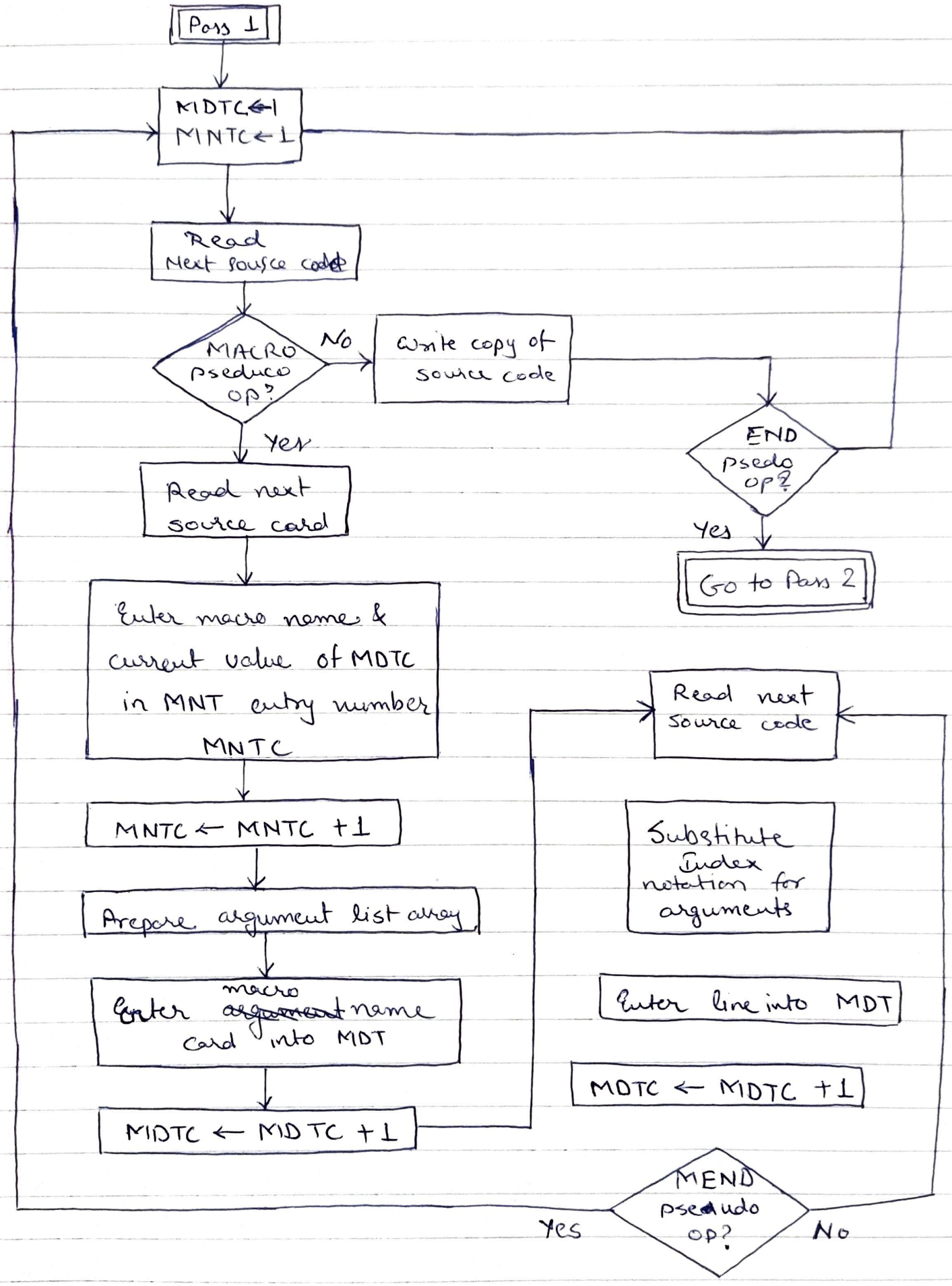
2] Macro Defination Table (MDT)
Index MDT - Instruction

3] Macro Name Table (MNT)
Index Macro Name MDT - index

4] Argument List Array (ALA)
Index Dummy Argument.

Platform : Linux (Java)

Conclusion : The function of Pass 1 in assembler is studied along errors comming in each pass.


Pass 1
MDTC ← 1
MNTC ← 1
Read Next source code
MACRO pseduo op?
No
Write copy of source code
END pseudo op?
Yes
Go to Pass 2
Yes
Read next source card
Enter macro name & current value of MDTC in MNT entry number MNTC
MNTC ← MNTC + 1
Prepare argument list array
Enter macro name card into MDT
MDTC ← MDTC + 1
Read next source code
Substitute Index notation for arguments
Enter line into MDT
MDTC ← MDTC + 1
MEND pseudo op?
Yes
No