

Page: 116
Date: 11/1

Assignment A4

Title: Pass II of a two pass macro processor

Problem Statement:

Write a Java Program for Pass II of a two pass macro processor.

The output of Assignment A3 should be the input for this assignment

Date of Completion:

Objectives:

- i. To implement Pass II of a two pass macro processor
- ii. To implement Pass II using concepts of OOP

Outcomes:

- i. Student should be able to implement Pass II of a two pass macro processor
- ii. Implement Pass II using the concepts of OOP
- iii. Understand the need to implement a Pass II macro processor over Pass I

Software & Hardware Requirements:

Fedora 64 bit OS

Eclipse Java IDE

500 GB HDD

4GB RAM

i5 Processor.

Theory :

A Macro Preprocessor :

A Macro Preprocessor takes source program containing macro definitions & macros calls & translates it into assembly language without macro definitions or calls.

It has to perform the following basic tasks

- 1) Recognize Macro Definition
- 2) Save Macro Definition
- 3) Recognize Macro Calls
- 4) Expand Macro Calls
- 5) Substitute Arguments

Tasks in Pass I

1. Scan all Macro Definitions
2. Enter name in MNT & in library
3. Store Macro definition in MDT
4. Add info to MNT indicating location of name in MDT
5. Prepare Argument List Array

Tasks in Pass II

1. Examine all statements in assembly source program to detect Macro calls
2. Locate Macro name in MNT
3. Establish correspondence b/w formal & actual param
4. Obtain info from MNT regarding position of Macro Definition in MDT
5. Expand Macro call by picking up model statements p MDT.

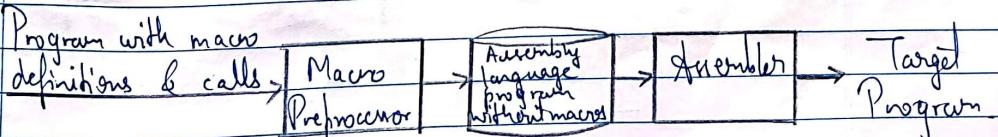
Page:
Date:

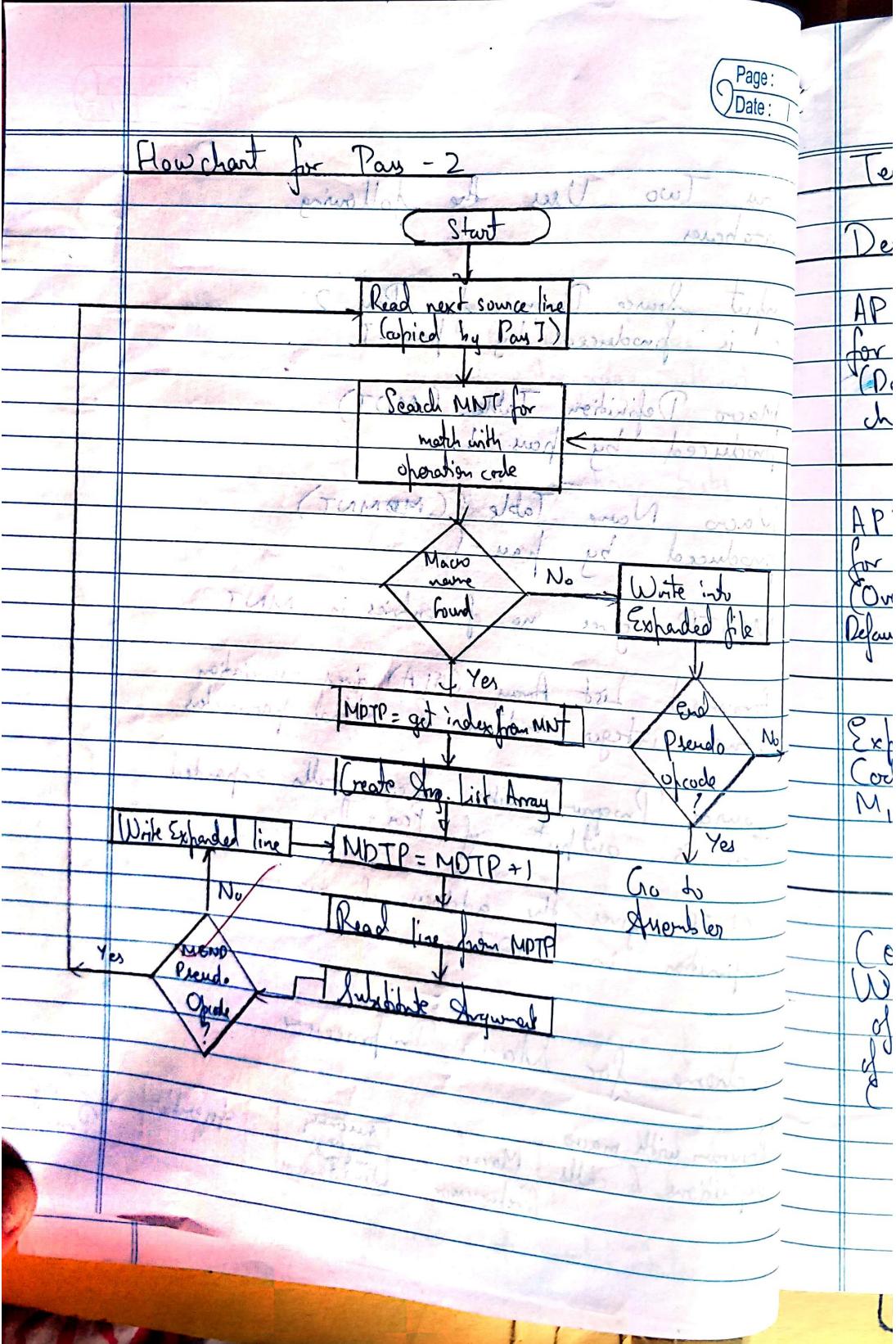
Page: 6
Date: 11/9

Pass Two Uses the following databases

- i) Input Source Program for Pass 2
It is produced by pass - 1
- ii) Macro Definition Table (MDT)
produced by pass - 1
- iii) Macro Name Table (MNT)
produced by pass - 1
- iv) MNT gives no. of entries in MNT
- v) Argument List Array (ALA) giving association between integer indices & actual parameters
- vi) Source Program with macro calls expanded
This is output of Pass - 1
- vii) MDT gives the address of macro definition in MDT

Scheme for Macro Preprocessor.





Test Cases

Description	Input	Output	Result
AP TAB for M1 (Default check)	M1 10, 20, & B = CREG	1 10 2 20 3 AREG 4 CREG	Success
AP TAB for M2 (Overriding Default Check)	M2 100, 200, & U = AREG, & V = BREG	1 100 2 200 3 BREG 4 AREG	Success
Expanded Code for M1	M1 10, 20, & B = CREG	MOVER AREG, 10 ADD AREG, = '1' MOVER AREG, 20 ADD AREG, = '1'	Success

Conclusion:
We have successfully implemented part II
of a two part Assembly using concepts
of OOP & taking output assignment A/3
(Part I) as input.

Success
11/02/20