	Peitam Mogal TE (B)-17
	Assignment No. 4.
/	Pass - 2 Macroprocessor
	TESTINATO PROJECT OF THE STATE OF THE PARTY.
	MINI DESIGN OF OUTSITE OF PASS -0
0.11	IN THE PROPERTY OF THE PARTY OF
	Problem Starement - North a Tava margan to
11/1	pass I of a two-pass marro-processor The
_	butter of assignment -a (MNT MIDT and
	fre without any macro defination) should be
_	input for this assignment.
11.18	E. Enter the mater defination into Maria to
	Theory: - Deling of Delinging
1 7 7	1: Macro processor (Defination)
7.1	A macro processor is a program that reads
77.9	a file (or files) and scans them for certain
Ta al	I. If mend preudo opende is encountated
	next source of input dayor Is read.
	2: Basic tasks performed by macro processor:
	a) Recognize macrodefination.
1	b) Save the defination.
	c) Recognize call. mont mani viel
	d) Expanded calls and substitute arguments.
	B If end precide code is dound then or to hu
	- Pass 1 Macro Defination
	- Pass 2 Macro call and Expansion.
	loss s-Macoo Galls and Evension
	Pass 1 Marzo Detination:
	Pass 1 algorithm examines each line of the input data
	for maon pseudo apcode Following are steps that are
	pertamed dunna Pasc 1 algorithm:
	1. Tuitalize MOTE and MNICIOITA Value one,
	so that previous value of MATE & MNTC 18 Set of
	1

· > 011 (090100)35/ (111)
2. Read the first input data
3. If the data contains MACRO pseudo opcode then A. Read the next data input
A. Read the next data input
13. they the name of the made and edistill idua
OF MOTO IN MNTO
c. Increase the counter value of MNF by value
bus Talonemm) s- talonapera to Trapo
D. Prepare the argument list array respective
to the macro found were all the
E. Enter the macro defination into MPT. Increase The counter of MPT By value one:
The counter of MPT By value one
F. Kead next line of the input data.
a substitute the index notations for dummy
arguments passed in MACRO.
H. Increase the counter of the MIDT by value one.
I. If mend pseudo opcode is encountered then next source of input data Is read.
next source of input data Is read.
. I re expand data input. 20 108:
a) Recognize macrodefination.
4. If macro pseudo opcode is not encountered in
aufu silpur inen
A. A copy of input data is created.
B. If end pseudo code is found then go to Pasiz C- Otherwise read next source of input data.
c. otherwise read next source of input data.
Pace and bas the copie copie.
Pass 2 Macro Calls and Expansion
Format Contract Contr
2. Examine each operation code for finding respective entry in the MNT.
respective entry in the MNT. Code for finding
3. It pame of the masso is
A. A pointer is set to the encountered then
A. A pointer is set to the MNT entry where
7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

-I-IDALE I-I-I
namic of
name of macro is found.
This pointer is called Macro Defination Table Pointer (MPTP).
B. Prepare argument list away containing a table of
dummy argument.
C. Increase The value of MOTO by value and
The Diff of the Control of the Contr
E. Substitute the values from the arguments list of
The macrofor Dummy arguments. F. If mend pseudo apcode is found then next
Source of input data is Read.
a. Else expands datainput.
4. When macro name is not tound than create arounded
data file
5. If end pseudo rocode is encountered than trad the
expanded source fileto Ascembler foromico coina.
6. Else read next source of elajoinput.
TAMER 2 SAT
Draw Plowchart Nx. t algorithm:
Input: 7 A9 3 91000 52 4 . 12917 8 370M
INPUT
MACRO
FNCR 1 & FIRST, &SECOND = DAMA9
A 10 & FIRST
L 2, & SECOND
MEND MACRO
INCR 2 4 ARAI & ARAI, & ARAZ = DATA 5
L 3, & ARGI
ST 49 4 ARUZ
MEND
PRUZ START
USING #, BASE

	INCRI DATAI
	INCRI DATAI INCRI DATAI INCRI DATAI
9	FOUR DC F'4'
1	FOUR DC F'4'
	BASE EQU 8
	BASE EQU 8 TEMP PS: 1F
1917	END TO THE STATE OF THE STATE O
	Property Property Street Street
	Output baroles aborgs obused busin II I
	= = = PASS 1 = 1= 7-3 = 1010 Want 25 101002
	ALA: Jugajojohoh zbroncia 2219 . D
Panded	[& FIRST, & SECOND] 21 amon sour mond.
	[& ARas, & ARaz]
199 69	E. IT end pseudo apade 18 epacentered then the
	- MNT of relamped delif Dans bolance
	FINCE YOU DEXT SOLICE OF FOR FRONT
	[INCR 2 , 4] : militable force freedows !! (used)
	0. 5 - 0
	MACRO CONTRACTOR OF THE PROPERTY OF THE PROPER
	A PANIS 1, #028 1291 8 1 90 NT
	MEN 21#1 7297 9 01 A
	D YARAI, & ARAZ = PATA 5
	INCR 11- CIGA ACTORY OF TARAP COM
	1084 H 8
	ST 3, #0
	4) #)
	MEN TRANSPORT
	D STATE OF THE PARTY

100	DATE!
	====== PASS 2 ==================================
	MDT: & FIRST, & SCIOND=DAT INCR 19
1	
-	A Litte
-	1,# 0 L 2,#1
/	MEN 2,#1
/	D. C.
_	TOTAL STATE OF THE
	PRQ 2 STAR
1	TO BAS C
	USIN
	a
	A PATA 1
	Lariopso 2011 2, PATA9
	L 3, DATA3
	ST 4 > PATA 4
Why !	FOUR DC P'4'
	FIVE DC F'5'
	BASE EQUIPMENT 8
	TEMP DS 1F
	PROSENIE S
	ama & val Pramalami of the
	END
	ALA: 10. Market of the Alandaria
1	[DATA 1, DATA 9]
1	[DATA 3 , DATA 4]
+	LUMIN 33 UNIM 4
1	Dan 1 ?
1	Ponclusion:
+	Thus pass # of Macro Processor is impreme and
1	and ALA file is generated.
1	Thus pass # of Macro processor is implemented And ALA file is generated:

Assignment No. 04 [PASS-2 Macroprocessor]

Problem Satement: Write a Java program for pass-II of a two-pass macro-processor. The output of assignment-3 (MNT, MDT and file without any macro definitions) should be input for this assignment.

1. Pass 2 Macro Code:

```
import java.io.*;
import java.util.HashMap;
import java.util. Vector;
public class macroPass2 {
        public static void main(String[] Args) throws IOException{
                 BufferedReader b1 = new BufferedReader(new FileReader("intermediate.txt"));
                 BufferedReader b2 = new BufferedReader(new FileReader("mnt.txt"));
                 BufferedReader b3 = new BufferedReader(new FileReader("mdt.txt"));
                 BufferedReader b4 = new BufferedReader(new FileReader("kpdt.txt"));
                 FileWriter f1 = new FileWriter("Pass2.txt");
                 HashMap<Integer,String> aptab=new HashMap<Integer,String>();
                 HashMap<String,Integer> aptabInverse=new HashMap<String,Integer>();
                 HashMap<String,Integer> mdtpHash=new HashMap<String,Integer>();
                 HashMap<String,Integer> kpdtpHash=new HashMap<String,Integer>();
                 HashMap<String,Integer> kpHash=new HashMap<String,Integer>();
                 HashMap<String,Integer> macroNameHash=new HashMap<String,Integer>();
                 Vector<String>mdt=new Vector<String>();
                 Vector<String>kpdt=new Vector<String>();
                 String s,s1;
                 int i,pp,kp,kpdtp,mdtp,paramNo;
                 while((s=b3.readLine())!=null)
                         mdt.addElement(s);
                 while((s=b4.readLine())!=null)
                         kpdt.addElement(s);
                 while((s=b2.readLine())!=null){
                         String word[]=s.split("\t");
                         s1=word[0]+word[1];
                         macroNameHash.put(word[0],1);
                         kpHash.put(s1,Integer.parseInt(word[2]));
                         mdtpHash.put(s1,Integer.parseInt(word[3]));
                         kpdtpHash.put(s1,Integer.parseInt(word[4]));
                 while((s=b1.readLine())!=null){
                         String b1Split[]=s.split("\\s");
                         if(macroNameHash.containsKey(b1Split[0])){
                                  pp= b1Split[1].split(",").length-b1Split[1].split("=").length+1;
                                  kp=kpHash.get(b1Split[0]+Integer.toString(pp));
                                  mdtp=mdtpHash.get(b1Split[0]+Integer.toString(pp));
                                  kpdtp=kpdtpHash.get(b1Split[0]+Integer.toString(pp));
                                  String actualParams[]=b1Split[1].split(",");
                                  paramNo=1;
                                  for(int j=0; j< pp; j++){
                                          aptab.put(paramNo, actualParams[paramNo-1]);
                                          aptabInverse.put(actualParams[paramNo-1],paramNo);
                                  i=kpdtp-1;
                                  for(int j=0;j< kp;j++){
```

```
String temp[]=kpdt.get(i).split("\t");
                                                                                                                          aptab.put(paramNo,temp[1]);
                                                                                                                          aptabInverse.put(temp[0],paramNo);
                                                                                                                          paramNo++;
                                                                                                 i=pp+1;
                                                                                                  while(i<=actualParams.length){
                                                                                                                          String initializedParams[]=actualParams[i-1].split("=");
                        aptab.put (aptabInverse.get (initialized Params [0].substring (1, initialized Params [0].length ())), initialized Params [1], initialized Params [2], initialized Params [3], initialized Params [4], initialized Params [6], initialized Params [6]
].substring(0,initializedParams[1].length()));
                                                                                                 i=mdtp-1;
                                                                                                  while(mdt.get(i).compareToIgnoreCase("MEND")!=0){
                                                                                                                          fl.write("+");
                                                                                                                          for(int j=0;j<mdt.get(i).length();j++){
                                                                                                                                                  if(mdt.get(i).charAt(j)=='#')
                                                                                                                                                                           f1.write(aptab.get(Integer.parseInt("" +
mdt.get(i).charAt(++j))));
                                                                                                                                                  else
                                                                                                                                                                           fl.write(mdt.get(i).charAt(j));
                                                                                                                          fl.write("\n");
                                                                                                                          i++;
                                                                                                  aptab.clear();
                                                                                                 aptabInverse.clear();
                                                                         else
                                                                                                 fl.write("+ "+s+"\n");
                                                 b1.close();
                                                 b2.close();
                                                 b3.close();
                                                 b4.close();
                                                 fl.close();
                         }
}
OUTPUT:
OUTPUT:
Pritam-spos@Pritam-HP:~/SPOSL$ javac macroPass2.java
Pritam-spos@Pritam-HP:~/SPOSL$ java macroPass2
Pritam-spos@Pritam-HP:~/SPOSL$ cat Pass2.txt
Intermediate - -
M1 10,20,&b=CREG
M2 100,200,&u=&AREG,&v=&BREG
Kpdt--
                         AREG
a
b
                        CREG
u
                        DREG
```

pass2 --

- + MOVE AREG,10
- + ADD AREG,='1'
- + MOVER AREG,20
- + ADD AREG,='5'
- + MOVER &AREG,100
- + MOVER &BREG,200
- + ADD & AREG,='15'
- + ADD &BREG,='10'

MNT --

M1 2 2 1 1 M2 2 2 6 3

MDT --

MOVE #3,#1

ADD #3,='1'

MOVER #3,#2

ADD #3,='5'

MEND

MOVER #3,#1

MOVER #4,#2

ADD #3,='15'

ADD #4,='10'

MEND