NAME: TEJAS NALAWADE PRACTICAL NO: 3 ROLL NO: TCOD01

Title of Assignment: Design suitable data structures and implement pass-I of a two-pass macro-processor.

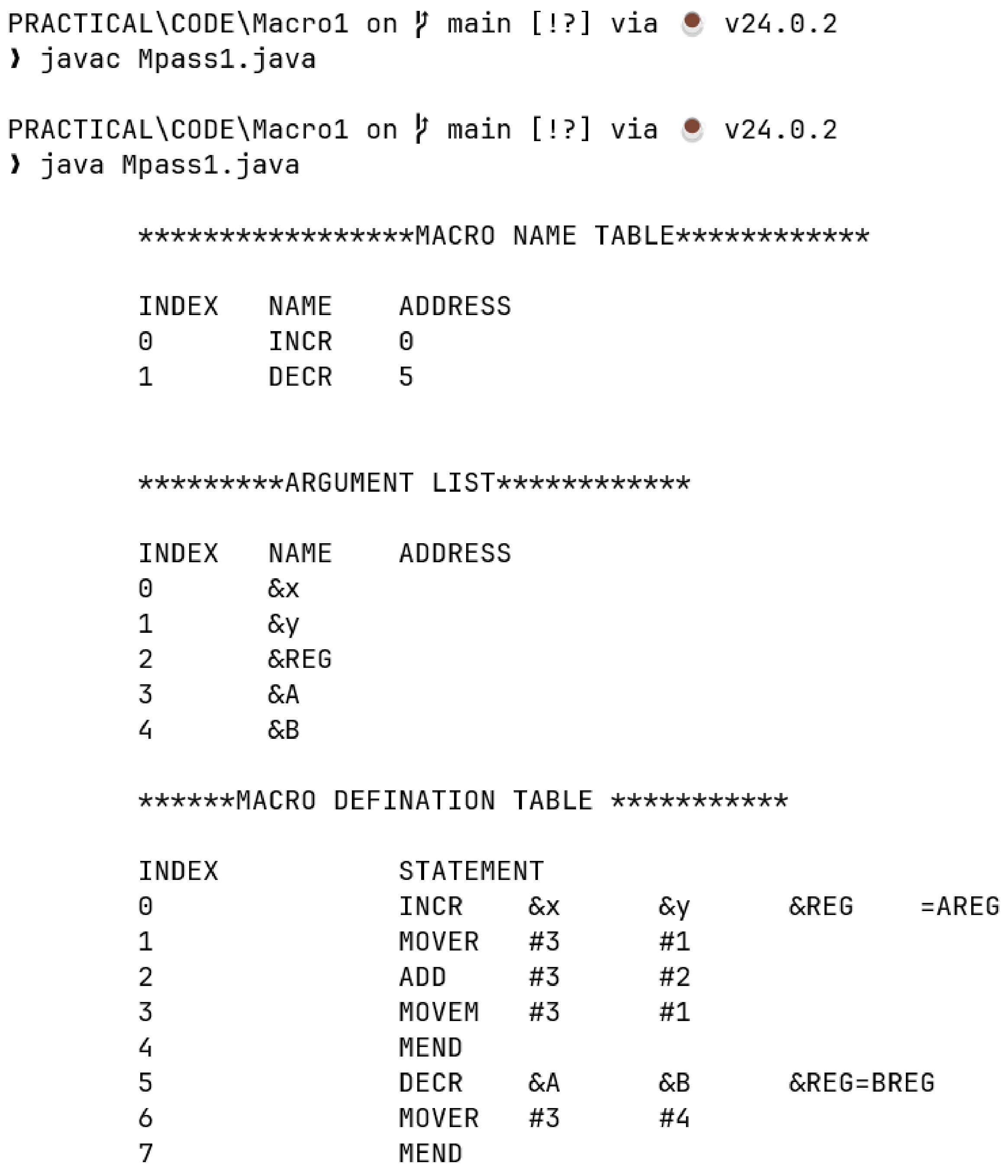
Problem Statement:
Write a program in C for a pass-II of two pass macro processor for Implementation of Macro Processor.
Following cases to be considered
 a) Macro without any parameters
 b) Macro with Positional Parameters
c) Macro with Key word parameters
d) Macro with positional and keyword parameters. (Conditional expansion, nested macro implementation not expected)

CODE:

```
import java.io.*;

public class MPass1{
   public static void main(String[] args) throws IOException{
      BufferedReader br1= new BufferedReader(new FileReader("input.txt"));
      String line;
      mdt[] MDT = new mdt[20];
      mnt[] MNT = new mnt[4];
      arglist[] ARGLIST = new arglist[10];
      boolean macro_start = false, macro_end = false, fill_arglist = false;
      int mdt_cnt= 0, mnt_cnt =0, arglist_cnt = 0;
      while((line = br1.readLine()) !=null){

         line = line.replaceAll(","," ");
         String[] tokens = line.split("\\s+");
         MDT[mdt_cnt] = new mdt();
         String stmnt = "";
         for(int i=0; i<tokens.length;i++){
            if(tokens[i].equalsIgnoreCase("mend")){
               MDT[mdt_cnt++].stmnt = "\t"+tokens[i];
               macro_end = true;
            }
            if(tokens[i].equalsIgnoreCase("macro")){
               macro_start= true;
               macro_end = false;
               break;
            }
            else if(!macro_end){
               if(macro_start){
                  MNT[mnt_cnt++] = new mnt(tokens[i],mdt_cnt);
                  macro_start = false;
                  fill_arglist = true;

               }
               if(fill_arglist){
                  while(i<tokens.length){
                     MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt+"\t"+tokens[i];
```

```
                    stmnt = stmnt+"\t"+tokens[i];
                    if(tokens[i].matches("&[a-zA-Z]+") || tokens[i].matches("&[a-zA-Z]+[0-9]+"))
                       ARGLIST[arglist_cnt++] = new arglist(tokens[i]);
                    i++;
                 }
                 fill_arglist = false;
              }
              else{
                 if(tokens[i].matches("[a-zA-Z]+")|| tokens[i].matches("[a-zA-Z]+[0-
9]+")||tokens[i].matches("[0-9]")){
                    MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt + "\t"+ tokens[i];
                    stmnt = stmnt + "\t" + tokens[i];
                 }
                 if(tokens[i].matches("&[a-zA-Z]+") || tokens[i].matches("&[a-zA-Z]+[0-9]+")){
                    for(int j=0; j<arglist_cnt; j++){
                       if(tokens[i].equals(ARGLIST[j].argname)){
                          MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt + "\t#" + (j+1);
                          stmnt = stmnt +"\t#"+(j+1);
                       }
                    }
                 }

              }
           }

        }
        if(stmnt!="" && !macro_end){
              mdt_cnt++;
        }
     }
     br1.close();

     BufferedWriter bw1 = new BufferedWriter(new FileWriter("MNT.txt"));
     System.out.println("\n\t****************MACRO NAME TABLE************");
     System.out.println("\n\tINDEX\tNAME\tADDRESS");
     for(int i=0; i<mnt_cnt;i++){
        System.out.println("\t"+i+"\t"+MNT[i].name+"\t"+MNT[i].addr);
        bw1.write(MNT[i].name+"\t"+MNT[i].addr+"\n");
     }
     bw1.close();

     bw1 = new BufferedWriter(new FileWriter("ARGLIST.txt"));
     System.out.println("\n\n\t*********ARGUMENT LIST************");
     System.out.println("\n\tINDEX\tNAME\tADDRESS");
     for(int i=0; i<arglist_cnt;i++){
        System.out.println("\t"+i+"\t"+ARGLIST[i].argname);
        bw1.write(ARGLIST[i].argname+"\n");
     }
     bw1.close();

     System.out.println("\n\t******MACRO DEFINATION TABLE ***********");
     System.out.println("\n\tINDEX\t\tSTATEMENT");

     bw1 = new BufferedWriter(new FileWriter("MDT.txt"));
     for(int i=0; i<mdt_cnt;i++){
            System.out.println("\t"+i+"\t"+MDT[i].stmnt);
            bw1.write(MDT[i].stmnt+"\n");
        }
        bw1.close();
    }
}
```

OUTPUT



```
PRACTICAL\CODE\Macro1 on  main [!?] via ☕ v24.0.2
❯ javac Mpass1.java

PRACTICAL\CODE\Macro1 on  main [!?] via ☕ v24.0.2
❯ java Mpass1.java

        *****************MACRO NAME TABLE************

        INDEX   NAME    ADDRESS
        0       INCR    0
        1       DECR    5


        *********ARGUMENT LIST************

        INDEX   NAME    ADDRESS
        0       &x
        1       &y
        2       &REG
        3       &A
        4       &B

        ******MACRO DEFINATION TABLE ***********

        INDEX           STATEMENT
        0               INCR    &x      &y      &REG    =AREG
        1               MOVER   #3      #1
        2               ADD     #3      #2
        3               MOVEM   #3      #1
        4               MEND
        5               DECR    &A      &B      &REG=BREG
        6               MOVER   #3      #4
        7               MEND
```