LP Practical Exam

Name: Rutuja Banginwar

Roll No: 31303

Aim: Pass One of Two Pass Macro Processor

Input:

```
MACRO
M1      &X, &Y, &A=AREG, &B=
MOVER &A, &X
ADD     &A, ='1'
MOVER &B, &Y
ADD     &A, ='5'
MEND
MACRO
M2      &P, &Q, &U=CREG, &V=DREG
MOVER &U, &P
MOVER &V, &Q
ADD     &U, ='15'
ADD     &V, ='10'
MEND
START
M1      10, 20, &B=CREG
M2      100, 200, &V=AREG, &U=BREG
M2      100, 200, &V=AREG, &U=BREG
END
```

Code:

```
import java.io.*;
import java.util.*;

public class LPA3{
    public void generateMacroIC() throws Exception{
        //Initialization
        BufferedReader  br = new BufferedReader(new FileReader("src\\input.txt"));

        BufferedWriter MNTbw = new BufferedWriter(new FileWriter("MNT.txt"));
        BufferedWriter PNTABbw = new BufferedWriter(new FileWriter("PNTAB.txt"));
        BufferedWriter KPDTABbw = new BufferedWriter(new FileWriter("KPDTAB.txt"));
        BufferedWriter MDTbw = new BufferedWriter(new FileWriter("MDT.txt"));
        BufferedWriter ICbw = new BufferedWriter(new FileWriter("NoMacroCode.txt"));

        String line;
        LinkedHashMap<String,Integer> PNTAB = new LinkedHashMap<String,Integer>();
        ArrayList<String> allMacroNames = new ArrayList<String>();
        String macroName = null;
        int pp=0,kp=0;
        int ppindex=0,kpindex=0;
        int mdtp=1,kpdtp=0;
        int insideMacroFlag = 0;

        while((line = br.readLine()) != null){
            String split[] = line.split("\\s+");
            if(split[1].equalsIgnoreCase("MACRO")){      //Checking Macro has been started
or not
                insideMacroFlag = 1;                        //Inside the macro
                line=br.readLine();                          //Reading Macro Prototype
                split = line.split("\\s+");

                pp=0;
                kp=0;ppindex=0;
                macroName = split[1];
                allMacroNames.add(split[1]);

                for(int i=2;i<split.length;i++){
                    split[i] = split[i].replaceAll("[&,]","");
                    if(split[i].contains("=")){         //'=' this means keyword parameter
                        kp++;
                        String keywordParam[] = split[i].split("=");
                        PNTAB.put(keywordParam[0],++ppindex);
                        if(keywordParam.length == 2){
                            KPDTABbw.write(++kpindex + "\t" + keywordParam[0] + "\t" +
keywordParam[1] + "\n"); //For default value
                        }
                        else{
```

```
                            KPDTABbw.write(++kpindex + "\t" + keywordParam[0] + "\t" + "-"
+ "\n");
                        }
                    }
                    else{                               //For Positional Parameter
                        pp++;
                        PNTAB.put(split[i],++ppindex);
                    }
                }
                //writing to PNTAB file
                if(!PNTAB.isEmpty()){
                    PNTABbw.write(macroName + "\n");
                    for(Map.Entry<String,Integer> lhm: PNTAB.entrySet()){
                        PNTABbw.write(lhm.getValue() + " " + lhm.getKey() + "\n");
                    }
                    PNTABbw.write("\n");
                }
                //writing entry in MNTtab File
                MNTbw.write(split[1] + "\t" + pp + "\t" + kp + "\t" + mdtp + "\t" +
((kp==0)?("-"):(kpdtp+1)) + "\n");
                kpdtp=kpindex;
            }
            else if(split[1].equalsIgnoreCase("MEND")){
                MDTbw.write(mdtp + "\t" + split[1]);
                MDTbw.write("\n");
                mdtp++;

                PNTAB.clear();  //One macro is finished
                insideMacroFlag = 0; //out of macro
            }
            else if(insideMacroFlag == 1){
                MDTbw.write(mdtp + "\t");
                //Writing to MDT file
                for(int i=1;i<split.length;i++){
                    if(split[i].contains("&")){   //If it is a operand
                       // split[i] = split[i].replaceAll("[&,]","");
                        //MDTbw.write("(P, "+PNTAB.get(split[i]) + ")\t");
                        String op = split[i].replaceAll("[&,]","");

                        if(PNTAB.containsKey(op)){
                            if(split[i].contains(",")){
                                MDTbw.write("(P, " + PNTAB.get(op) + "),\t");
                            }
                            else{
                                MDTbw.write("(P, " + PNTAB.get(op) + ")\t");
                            }
                            //System.out.println(op);
                        }
```

```
                        else{
                            if(allMacroNames.contains(split[1]) &&
(split[i].contains("="))){
                                MDTbw.write(split[i] + "\t");
                            }
                            else{
                                System.out.println("Invalid Parameter: "+op);
                            }
                        }
                    }
                    else{    //If it is a opcode
                        MDTbw.write(split[i] + "\t");
                    }
                }
                MDTbw.write("\n");
                mdtp++;
            }
            else{
                ICbw.write(line + "\n");  //Other than Macro Code
            }
        }
        //Closing all files
        MNTbw.close();
        PNTABbw.close();
        KPDTABbw.close();
        MDTbw.close();
        ICbw.close();
        br.close();
        System.out.println("Macro Processing Finished");
    }

    public static void main(String args[]) throws Exception{
        LPA3 passOne = new LPA3();
        passOne.generateMacroIC();

    }
}
```

Output:

PNTAB.txt:

M1

1 X

2 Y

3 A

4 B

M2

1 P

2 Q

3 U

4 V

KPDTAB.txt

1	A	AREG
2	B	-
3	U	CREG
4	V	DREG

MNT.txt

M1	2	2	1	1
M2	2	2	6	3

MDT.txt

```
1       MOVER (P, 3),   (P, 1)
2       ADD     (P, 3),   =’1’
3       MOVER (P, 4),   (P, 2)
4       ADD     (P, 3),   =’5’
5       MEND
6       MOVER (P, 3),   (P, 1)
7       MOVER (P, 4),   (P, 2)
8       ADD     (P, 3),   =’15’
9       ADD     (P, 4),   =’10’
10      MEND
```

NOMacroCode.txt

```
        START
        M1      10, 20, &B=CREG
        M2      100, 200, &V=AREG, &U=BREG
        M2      100, 200, &V=AREG, &U=BREG
        END
```