Text

Description automatically generated

**T.Y.B.Tech (CSE)**

[System Software and Compilers(SSC)](https://mitwpu.instructure.com/courses/3210)

**Lab Assignment No – 3**

**Name: Aniruddha Shende**

**Roll number: PD-05**

**Batch: D1**

**Panel: D**

A piece of paper with writing

Description automatically generated with medium confidence

A piece of paper with writing

Description automatically generated with medium confidence

A picture containing text, receipt

Description automatically generated

Text, letter

Description automatically generated

**INPUT :**

**Table

Description automatically generated**

**Java Code:**

1. **Main.java file**

package com.lab1;

public class Main {

public static void main(String[] args) {

Pass1\_MACRO.printPass1\_MACRO();

}

}

1. **MDT Table file**

package com.lab1;

import java.util.LinkedHashMap;

public class MDTtable {

private static int location\_counter = 0;

private static LinkedHashMap<String, String> MDT = new LinkedHashMap<String, String>();

public static int getLocation\_counter() {

return location\_counter;

}

public static void add(String instructions) {

location\_counter += 1;

MDT.put(Integer.toString(location\_counter), instructions);

}

public static void printMDT() {

for (String key : MDT.keySet()) {

System.out.println(key + " " + MDT.get(key));

}

}

}

1. **MNT Table file**

package com.lab1;

import java.util.HashSet;

import java.util.LinkedHashMap;

import java.util.Set;

public class MNTtable {

private static LinkedHashMap<String, String> MNT = new LinkedHashMap<String, String>();

private static Set<String> all\_macros = new HashSet<String>();

public static void add\_to\_MNT(String macro\_name, int index) {

all\_macros.add(macro\_name);

MNT.put(macro\_name,Integer.toString(index));

}

public static void printMNT() {

int mnt\_index = 0;

for (String key : MNT.keySet()) {

mnt\_index++;

System.out.println(mnt\_index + " " + key + " " + MNT.get(key));

}

}

public static boolean isMacro\_present(String macro\_name) {

return all\_macros.contains(macro\_name);

}

}

1. **ALA Table file**

package com.lab1;

import java.util.LinkedHashMap;

public class ALAtable {

private static LinkedHashMap<String, String> ALA = new LinkedHashMap<String, String>();

private static int index = 0;

public static void add(String arguments) {

index++;

ALA.put(Integer.toString(index), arguments);

}

public static void printALA() {

for (String key : ALA.keySet()) {

System.out.println(key + " " + ALA.get(key));

}

}

public static LinkedHashMap<String, String> getALA() {

return ALA;

}

}

1. **Pass1\_MACRO.java file**

package com.lab1;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.Scanner;

import java.util.StringTokenizer;

public class Pass1\_MACRO {

private static LinkedHashMap<String, ArrayList<String>> macro\_with\_their\_params = new LinkedHashMap<String, ArrayList<String>>();

public static void printPass1\_MACRO() {

System.out.println("Pass1 MACRO : \n");

File file = new File("src/com/lab1/input1.txt");

Scanner sc = null;

try {

sc = new Scanner(file);

} catch (FileNotFoundException e) {

e.printStackTrace();

}

while (sc.hasNextLine()) {

String line = sc.nextLine();

StringTokenizer st = new StringTokenizer(line, " ");

String opcode = st.nextToken();

if (opcode.equals("MACRO")) {

String str = sc.nextLine();

MDTtable.add(str);

int counter = 0;

while (!str.equals("MEND")) {

if (counter == 0) {

String[] list1 = str.split(" ");

MNTtable.add\_to\_MNT(list1[0], MDTtable.getLocation\_counter());

ArrayList<String> list2 = new ArrayList<String>();

for (int i = 1; i < list1.length; i++) {

list2.add(list1[i]);

ALAtable.add(list1[i]);

}

macro\_with\_their\_params.put(list1[0], list2);

}

counter++;

str = sc.nextLine();

MDTtable.add(str);

}

} else {

FileWriter abc;

try {

abc = new FileWriter("src/com/lab1/output\_file.txt");

BufferedWriter writer = new BufferedWriter(abc);

writer.write(line);

writer.newLine();

while (sc.hasNextLine()) {

String line1 = sc.nextLine();

writer.write(line1);

writer.newLine();

}

writer.close();

} catch (IOException except) {

except.printStackTrace();

}

}

}

System.out.println("\n\nMDT Table\n");

MDTtable.printMDT();

System.out.println("\n\nMNT Table\n");

MNTtable.printMNT();

System.out.println("\n\nALA Table\n");

ALAtable.printALA();

sc.close();

}

public static ArrayList<String> getMacro\_with\_their\_params(String macro\_name) {

return macro\_with\_their\_params.get(macro\_name);

}

}

**Output of the program:**

Text

Description automatically generated

**Output File generated as a result of Pass 1 :**

Table

Description automatically generated with medium confidence