Text

Description automatically generated

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

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

**Lab Assignment No – 1**

**Name: Aniruddha Shende**

**Roll number: PD-05**

**Batch: D1**

**Panel: D**

Text, letter

Description automatically generated

Text, letter

Description automatically generated

Text, letter

Description automatically generated

A sheet of music

Description automatically generated with medium confidence

**Code and Output from next page onwards :**

**Java Code:**

import java.util.HashMap;

import java.io.File;

import java.util.\*;

import java.io.\*;

import java.util.Scanner;

public class assign1 {

public static void main(String[] args) {

HashMap<String, ArrayList<String>> optab = new HashMap<String, ArrayList<String>>();

HashMap<String, String> registertable = new HashMap<String, String>();

HashMap<String, String> bctable = new HashMap<String, String>();

LinkedHashMap<String, ArrayList<String>> symtab\_table = new LinkedHashMap<String, ArrayList<String>>();

//to store the address of the labels/symbols

HashMap<String, Integer> symbol\_addresses = new HashMap<String, Integer>();

//all the location\_counter, symbol\_id, literal\_id

int location\_counter = 0;

int symbol\_id = 0;

int literal\_id = 0;

int start\_pointer = 0;

optab.put("STOP", new ArrayList<String>(Arrays.asList("(IS,00)")));

optab.put("ADD", new ArrayList<String>(Arrays.asList("(IS,01)")));

optab.put("SUB", new ArrayList<String>(Arrays.asList("(IS,02)")));

optab.put("MULT", new ArrayList<String>(Arrays.asList("(IS,03)")));

optab.put("MOVER", new ArrayList<String>(Arrays.asList("(IS,04)")));

optab.put("MOVEM", new ArrayList<String>(Arrays.asList("(IS,05)")));

optab.put("COMP", new ArrayList<String>(Arrays.asList("(IS,06)")));

optab.put("BC", new ArrayList<String>(Arrays.asList("(IS,07)")));

optab.put("DIV", new ArrayList<String>(Arrays.asList("(IS,08)")));

optab.put("READ", new ArrayList<String>(Arrays.asList("(IS,09)")));

optab.put("PRINT", new ArrayList<String>(Arrays.asList("(IS,10)")));

optab.put("DC", new ArrayList<String>(Arrays.asList("(DL,01)")));

optab.put("DS", new ArrayList<String>(Arrays.asList("(DL,02)")));

optab.put("START", new ArrayList<String>(Arrays.asList("(AD,01)")));

optab.put("END", new ArrayList<String>(Arrays.asList("(AD,02)")));

optab.put("ORIGIN", new ArrayList<String>(Arrays.asList("(AD,03)")));

optab.put("EQU", new ArrayList<String>(Arrays.asList("(AD,04)")));

optab.put("LTORG", new ArrayList<String>(Arrays.asList("(AD,05)")));

registertable.put("AREG", "1");

registertable.put("BREG", "2");

registertable.put("CREG", "3");

registertable.put("DREG", "4");

bctable.put("LT", "01");

bctable.put("LE", "02");

bctable.put("EQ", "03");

bctable.put("GT", "04");

bctable.put("GE", "05");

bctable.put("ANY","06");

System.out.println();

File file = new File("input.txt");

Scanner sc = null;

try {

sc = new Scanner(file);

} catch (FileNotFoundException e) {

e.printStackTrace();

}

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

while (sc.hasNextLine()) {

lines.add(sc.nextLine());

}

System.out.println("The given code is : \n");

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

for (int i = 0; i < lines.size(); i++) {

String command = lines.get(i);

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

StringTokenizer tokenizer = new StringTokenizer(command, " ");

while (tokenizer.hasMoreTokens()) {

words.add(tokenizer.nextToken());

}

for (int j = 0; j < words.size(); j++) {

System.out.print(words.get(j)+"\t");

}

for (int j = 0; j < words.size(); j++) {

words.set(j, words.get(j).replace(",", ""));

}

System.out.println();

if (words.get(0).equals("START")) {

if (words.size() == 2) {

ictab.add(new ArrayList<String>(

Arrays.asList("-", "(AD,01)", "-", "(C," + words.get(1) + ")")));

location\_counter = Integer.parseInt(words.get(1)) - 1;

start\_pointer = location\_counter;

} else {

ictab.add(new ArrayList<String>(Arrays.asList("-", "(AD,01)", "-")));

}

}

//end

else if (words.get(0).equals("END")) {

location\_counter++;

ictab.add

(new ArrayList<String>(Arrays.asList(Integer.toString(location\_counter), "(AD,02)", "-", "-")));

continue;

}

else if (words.get(0).equals("ORIGIN")) {

//System.out.println("Inside origin");

if (words.size() == 1) {

ictab.add(

new ArrayList<String>(Arrays.asList("-", "(AD,03)", "-", "-")));

location\_counter = start\_pointer;

} else if (words.size() == 2) {

int flag = 0;//if flag becomes 1 then we can note that a +/- is encountered

for (int i1 = 0; i1 < words.get(1).length(); i1++) {

if (words.get(1).charAt(i1) == '+') {

flag = 1;

break;

} else if (words.get(1).charAt(i1) == '-') {

flag = 2;

break;

}

}

//if the flag is 1 then we have +/- present

if (flag == 1 || flag == 2) {

String str1 = words.get(1);

if (flag == 1) {

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

String temporary = "";

for (int kk = 0; kk < str1.length(); kk++) {

if (str1.charAt(kk) == '+') {

tempo.add(temporary);

temporary = "";

continue;

}

temporary += str1.charAt(kk);

}

tempo.add(temporary);

int loc = Integer.parseInt(symtab\_table.get(tempo.get(0)).get(0));

int additional = Integer.parseInt(tempo.get(1));

location\_counter = loc + additional - 1;

ictab.add(

new ArrayList<String>(Arrays.asList("-", "(AD,03)", "-",

"(S," + symtab\_table.get(tempo.get(0)).get(2) + ")")));

} else if (flag == 2) {

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

String temporary = "";

for (int kk = 0; kk < str1.length(); kk++) {

if (str1.charAt(kk) == '-') {

tempo.add(temporary);

temporary = "";

continue;

}

temporary += str1.charAt(kk);

}

tempo.add(temporary);

int loc = Integer.parseInt(symtab\_table.get(tempo.get(0)).get(0));

int additional = Integer.parseInt(tempo.get(1));

location\_counter = loc - additional - 1;

ictab.add(

new ArrayList<String>(Arrays.asList("-", "(AD,03)", "-",

"(S," + symtab\_table.get(tempo.get(0)).get(2) + ")")));

}

} else {

String str = words.get(1);

int loc = Integer.parseInt(symtab\_table.get(str).get(0));

location\_counter = loc - 1;

ictab.add(

new ArrayList<String>(

Arrays.asList("-", "(AD,03)", "-", "(S," + symbol\_addresses.get(str) + ")")));

}

}

}

// equ implementation

else if (words.get(0).equals("EQU")) {

String str1 = words.get(0);

String str2 = words.get(2);

// int loc = Integer.parseInt(symtab\_table.get(str1).get(1));

ictab.add(

new ArrayList<String>(

Arrays.asList("-", "(AD,04)", "-", "(S," + symbol\_addresses.get(str2) + ")")));

String str3 = symtab\_table.get(str2).get(0);

symtab\_table.get(str1).set(0, str3);

}

else if (optab.containsKey(words.get(0)) == false) {

if (symtab\_table.containsKey(words.get(0)) == false) {

symbol\_id++;

symtab\_table.put(words.get(0),

new ArrayList<String>(Arrays.asList("-", "-", Integer.toString(symbol\_id))));

symbol\_addresses.put(words.get(0), symbol\_id);

}

if (words.size() == 4) {

location\_counter++;

symtab\_table.get(words.get(0)).set(0, Integer.toString(location\_counter));

symtab\_table.get(words.get(0)).set(1, "1");

words.remove(words.get(0));

if (optab.containsKey(words.get(0))) {

String operand = words.get(1);

String opcode = optab.get(words.get(0)).get(0);

if (registertable.containsKey(operand)) {

operand = registertable.get(operand);

}

if (symtab\_table.containsKey(words.get(2)) == false) {

symbol\_id++;

symtab\_table.put(words.get(2),

new ArrayList<String>(Arrays.asList("-", "-", Integer.toString(symbol\_id))));

symbol\_addresses.put(words.get(2), symbol\_id);

}

ictab.add(new ArrayList<String>(Arrays.asList(Integer.toString(location\_counter),opcode, operand, "(S," +symbol\_addresses.get(words.get(2))+ ")")));

}

}

else if (words.size() == 3) {

if (words.get(1).equals("EQU")) {

}

else if (words.get(1).equals("DS")) {

location\_counter++;

ictab.add(

new ArrayList<String>(Arrays.asList(Integer.toString(location\_counter), "(DL,02)", "-", "(C," + words.get(2) + ")")));

symtab\_table.get(words.get(0)).set(0, Integer.toString(location\_counter));

symtab\_table.get(words.get(0)).set(1, "1");

location\_counter += Integer.parseInt(words.get(2))-1;

}

else if (words.get(1).equals("DC")) {

location\_counter++;

ictab.add(

new ArrayList<String>(Arrays.asList(Integer.toString(location\_counter), "(DL,01)", "-", "(C," + words.get(2) + ")")));

symtab\_table.get(words.get(0)).set(0, Integer.toString(location\_counter));

symtab\_table.get(words.get(0)).set(1, "1");

}

}

else if (words.size() == 2) {

}

}

else if (optab.containsKey(words.get(0))) {

location\_counter++;

String operand = words.get(1);

String opcode = optab.get(words.get(0)).get(0);

if (registertable.containsKey(operand)) {

operand = registertable.get(operand);

}

if (symtab\_table.containsKey(words.get(2)) == false) {

symbol\_id++;

symtab\_table.put(words.get(2),

new ArrayList<String>(Arrays.asList("-", "-", Integer.toString(symbol\_id))));

symbol\_addresses.put(words.get(2), symbol\_id);

}

ictab.add(new ArrayList<String>(Arrays.asList(Integer.toString(location\_counter),opcode, operand, "(S," +symbol\_addresses.get(words.get(2))+ ")")));

}

}

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

for (int i = 0; i < ictab.size(); i++) {

for (int j = 0; j < ictab.get(i).size(); j++) {

System.out.print(ictab.get(i).get(j) + "\t ");

}

System.out.println();

}

System.out.println();

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

for (Map.Entry<String, ArrayList<String>> entry : symtab\_table.entrySet()) {

System.out.println(entry.getValue().get(2) + "\t" + entry.getKey() + "\t" + entry.getValue().get(0) + "\t"

+ entry.getValue().get(1));

}

}

}

**Output of the program:**

**Text

Description automatically generated**