**2. Pass\_2**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.util.ArrayList;

class TableRow {

String symbol;

int address;

int index;

public TableRow(String symbol, int address) {

super();

this.symbol = symbol;

this.address = address;

}

public TableRow(String symbol, int address, int index) {

super();

this.symbol = symbol;

this.address = address;

this.index = index;

}

public int getIndex() {

return index;

}

public void setIndex(int index) {

this.index = index;

}

public String getSymbol() {

return symbol;

}

public void setSymbol(String symbol) {

this.symbol = symbol;

}

public int getAddress() {

return address;

}

public void setAddress(int address) {

this.address = address;

}

}

public class Pass\_2 {

ArrayList<TableRow> SYMTAB, LITTAB;

public Pass\_2() {

SYMTAB = new ArrayList<>();

LITTAB = new ArrayList<>();

}

public static void main(String[] args) {

Pass\_2 pass2 = new Pass\_2();

try {

pass2.generateCode("IC.txt");

} catch (Exception e) {

e.printStackTrace();

}

}

public void readtables() {

BufferedReader br;

String line;

try {

br = new BufferedReader(new FileReader("SYMTAB.txt"));

while ((line = br.readLine()) != null) {

String parts[] = line.split("\\s+");

SYMTAB.add(new TableRow(parts[1], Integer.parseInt(parts[2]), Integer.parseInt(parts[0])));

}

br.close();

br = new BufferedReader(new FileReader("LITTAB.txt"));

while ((line = br.readLine()) != null) {

String parts[] = line.split("\\s+");

LITTAB.add(new TableRow(parts[1], Integer.parseInt(parts[2]), Integer.parseInt(parts[0])));

}

br.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

public void generateCode(String filename) throws Exception {

readtables();

BufferedReader br = new BufferedReader(new FileReader(filename));

BufferedWriter bw = new BufferedWriter(new FileWriter("PASS2.txt"));

String line, code;

while ((line = br.readLine()) != null) {

String parts[] = line.split("\\s+");

if (parts[0].contains("AD") || parts[0].contains("DL,02")) {

bw.write("\n");

continue;

} else if (parts.length == 2) {

if (parts[0].contains("DL")) {

parts[0] = parts[0].replaceAll("[^0-9]", "");

if (Integer.parseInt(parts[0]) == 1) {

int constant = Integer.parseInt(parts[1].replaceAll("[^0-9]", ""));

code = "00\t0\t" + String.format("%03d", constant) + "\n";

bw.write(code);

}

} else if (parts[0].contains("IS")) {

int opcode = Integer.parseInt(parts[0].replaceAll("[^0-9]", ""));

if (opcode == 10) {

if (parts[1].contains("S")) {

int symIndex = Integer.parseInt(parts[1].replaceAll("[^0-9]", ""));

code = String.format("%02d", opcode) + "\t0\t"

+ String.format("%03d", SYMTAB.get(symIndex - 1).getAddress()) + "\n";

bw.write(code);

} else if (parts[1].contains("L")) {

int symIndex = Integer.parseInt(parts[1].replaceAll("[^0-9]", ""));

code = String.format("%02d", opcode) + "\t0\t"

+ String.format("%03d", LITTAB.get(symIndex - 1).getAddress()) + "\n";

bw.write(code);

}

}

}

} else if (parts.length == 1 && parts[0].contains("IS")) {

int opcode = Integer.parseInt(parts[0].replaceAll("[^0-9]", ""));

code = String.format("%02d", opcode) + "\t0\t" + String.format("%03d", 0) + "\n";

bw.write(code);

} else if (parts[0].contains("IS") && parts.length == 3) // All OTHER IS INSTR

{

int opcode = Integer.parseInt(parts[0].replaceAll("[^0-9]", ""));

int regcode = Integer.parseInt(parts[1]);

if (parts[2].contains("S")) {

int symIndex = Integer.parseInt(parts[2].replaceAll("[^0-9]", ""));

code = String.format("%02d", opcode) + "\t" + regcode + "\t"

+ String.format("%03d", SYMTAB.get(symIndex - 1).getAddress()) + "\n";

bw.write(code);

} else if (parts[2].contains("L")) {

int symIndex = Integer.parseInt(parts[2].replaceAll("[^0-9]", ""));

code = String.format("%02d", opcode) + "\t" + regcode + "\t"

+ String.format("%03d", LITTAB.get(symIndex - 1).getAddress()) + "\n";

bw.write(code);

}

}

}

bw.close();

br.close();

}

}