**package** spos2;

**import** java.io.BufferedReader; **import** java.io.BufferedWriter; **import** java.io.FileReader; **import** java.io.FileWriter; **import** java.io.IOException; **import** java.util.ArrayList;

**public class** pass2 {

**public static void** main(String args[]) **throws** IOException { ArrayList<TableRow> symtab =

*readSymtab*("C:\\Users\\SNEHAL\\Desktop\\symtabb.txt");

*processInstructions*(symtab, "C:\\Users\\SNEHAL\\Desktop\\iic.txt", "C:\\Users\\SNEHAL\\Desktop\\MachineCode.txt");

System.***out***.println("DONE!!");

}

**private static** ArrayList<TableRow> readSymtab(String symtabFilePath)

**throws** IOException {

BufferedReader br = **new** BufferedReader(**new** FileReader(symtabFilePath));

String line;

ArrayList<TableRow> symtab = **new** ArrayList<>();

**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();

**return** symtab;

}

**private static void** processInstructions(ArrayList<TableRow> symtab, String inputFilePath, String outputFilePath)

**throws** IOException {

BufferedReader br = **new** BufferedReader(**new** FileReader(inputFilePath));

BufferedWriter bw = **new** BufferedWriter(**new** FileWriter(outputFilePath));

String line;

**while** ((line = br.readLine()) != **null**) { String parts[] = line.split("\\s+");

**if** (parts[0].contains("AD") || parts[0].contains("(DL,02)")) { bw.write("\n");

} **else if** (parts[0].contains("DL,01")) {

String[] opcode = parts[1].split(",");

opcode[1] = opcode[1].replace(")", ""); bw.write("+ 00 0 00" + opcode[1] + "\n");

} **else if** (parts[0].contains("IS")) { String op = "+ ";

String[] opcode = parts[0].split(",");

opcode[1] = opcode[1].replace(")", "");

op = op + opcode[1] + " ";

**if** (parts.length == 2) {

String[] opc = parts[1].split(",");

opc[1] = opc[1].replace(")", "");

**int** oc = Integer.*parseInt*(opc[1]);

**int** add = symtab.get(oc - 1).getAddress();

op = op + add + " ";

} **else if** (parts.length == 3) { String[] rg = parts[1].split(",");

rg[1] = rg[1].replace(")", ""); op = op + rg[1] + " ";

String[] opc = parts[2].split(",");

opc[1] = opc[1].replace(")", "");

**int** oc = Integer.*parseInt*(opc[1]);

**int** add = symtab.get(oc - 1).getAddress();

op = op + add + " ";

}

bw.write(op + "\n");

}

}

bw.close();

br.close();

}

}

**class** TableRow { String symbol;

**int** address, index;

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

**this**.index = index; **this**.symbol = symbol; **this**.address = address;

}

**public** String getSymbol() {

**return** symbol;

}

**public int** getIndex() {

**return** index;

}

**public int** getAddress() {

**return** address;

}

}

Input for code IC:

(AD,01) (C,3) (IS,02) (R,1) (S,2) (DL,01) (C,1)

SYMBOL TABLE:

1 A 400

2 B 402

3 C 404

OUTPUT:

+ 02 1 402

+ 00 0 001