VISHWAKARMA UNIVERSITY
Maximising Human Potential

T. Y. B. Tech Computer Engineering

Student Name	Pranav Dambe (Nikam)
SRN No	202201704
Roll No	68
PRN	2280030506
Division	D(D3)
Subject	System Programming
Year	Third Year

Assignment - 3

QUE 1 :

Implement Pass-II of a two pass assembler designed in assignment 1. Use the output of Assignment-1 (intermediate code file , MOT and symbol table) as input for this assignment.

Output will be machine code for the IC.Submit a single .pdf / .doc file containing IC, Symbol table and machine code with LC and your souce code in that sequence.

OUTPUT 1 :

source.asm

```
1    START 200
2    MOV AREG BREG
3    MOV DREG NUM
4    MOVEM AREG DREG
5    ADD AREG 40
6    MUL DREG BREG
7    DIV BREG Y
8    NUM DC 10
9    SUB DREG BREG
10   Y DS 25
11   END
```

```
PS E:\TY-LAB\SP - Lab\Assignment-3> cd "e:\TY-LAB\SP
Assignment_3 }

---------------------------------

Symbol Table:

1       NUM     212
2       Y       215

---------------------------------

Intermediate Table:

0       (AD, 20)        (C, 200)
200     (IS, 89)        (R, 01) (R, 02)
202     (IS, 89)        (R, 04) (S, 1)
204     (IS, 05)        (R, 01) (R, 02)
206     (IS, 01)        (R, 03) (C, 40)
208     (IS, 03)        (R, 04) (R, 02)
210     (IS, 08)        (R, 02) (S, 2)
212     (DL, 01)        (C, 10)
213     (IS, 02)        (R, 04) (R, 02)
215     (DL, 02)        (C, 25)
240     (AD, 21)

---------------------------------

Machine Code:

200  89  01  02
202  89  04  212
204  05  01  02
206  01  03  40
208  03  04  02
210  08  02  215
212      10
213  02  04  02
215  --
240

---------------------------------
```

OUTPUT 2 :

```
Symbol Table:

1       NUM     418
2       X       410
3       Z       415
4       Y       421

---------------------------------

Intermediate Table:

0       (AD, 20)        (C, 400)
400     (IS, 89)        (R, 01) (R, 02)
402     (IS, 89)        (R, 04) (S, 1)
404     (IS, 04)        (R, 03) (S, 2)
406     (IS, 05)        (R, 01) (R, 04)
408     (IS, 89)        (R, 04) (S, 3)
410     (DL, 01)        (C, 15)
411     (IS, 01)        (R, 01) (C, 40)
413     (IS, 03)        (R, 04) (R, 02)
415     (DL, 01)        (C, 22)
416     (IS, 08)        (R, 02) (S, 4)
418     (DL, 01)        (C, 10)
419     (IS, 02)        (R, 04) (R, 02)
421     (DL, 02)        (C, 25)
446     (AD, 21)

---------------------------------

Machine Code:

400  89  01  02
402  89  04  418
404  04  03  410
406  05  01  04
408  89  04  415
410      15
411  01  01  40
413  03  04  02
415      22
416  08  02  421
418      10
419  02  04  02
421  --
446

---------------------------------
```

source.asm

```
1    START 400
2    MOV AREG BREG
3    MOV DREG NUM
4    MOVER CREG X
5    MOVEM AREG DREG
6    MOV DREG Z
7    X DC 15
8    ADD AREG 40
9    MUL DREG BREG
10   Z DC 22
11   DIV BREG Y
12   NUM DC 10
13   SUB DREG BREG
14   Y DS 25
15   END
```

SOUCE Code :

```
class Pass2 extends Pass1 {

  public void pass2() {
     System.out.println("\nMachine Code:\n");
     for (String[] row : intermediateTable) {
        String location = row[0];
        String opcode = row[1];
        String operand1 = row.length > 2 ? row[2] : "";
        String operand2 = row.length > 3 ? row[3] : "";

        if (opcode.equals("(AD, 20)")) {
           continue;
        }

        if (opcode != null) {
           if(opcode.equals("(AD, 21)") || opcode.equals("(DL, 01)")){
              opcode = "  ";
           }
           else{
              opcode = opcode.equals("(DL, 02)") ? "(DL, 02)" : opcode.replaceAll("[^0-9]", "");
           }
        }
        if (operand1 != null) {
           operand1 = opcode.equals("(DL, 02)") || opcode.equals("(AD, 20)") ? "  " :
     processOperand(operand1);
        }
        if (operand2 != null) {
           operand2 = opcode.equals("(DL, 02)") ? "  " : processOperand(operand2);
        }
        if(opcode.startsWith("(DL, 02)")){
           opcode = "--";
        }

        String machineCode = (opcode == null ? "" : opcode) +
                        (operand1.isEmpty() ? "" : "  " + operand1) +
                        (operand2.isEmpty() ? "" : "  " + operand2);

        System.out.println(location + "  " + machineCode);
     }
     System.out.println("_____________________________");
  }

  private String processOperand(String operand) {
     if (operand.startsWith("(S,")) {
        int symbolIndex = Integer.parseInt(operand.replaceAll("[^0-9]", "")) - 1;
        String[] symbolEntry = symbolTable.get(symbolIndex);
        return symbolEntry[1];
     }
```

```
    return operand.replaceAll("[^0-9]", "");
  }
}

public class Assignment_3 extends Pass2 {
  public static void main(String[] args) {
    Assignment_3 Assembler = new Assignment_3();
    Assembler.pass1("source.asm");
    if (showTable) {
      Assembler.printSymbolTable();
      Assembler.printIntermediateTable();
      Assembler.pass2();
    }
  }
}
```
