System Software and Compiler Design

Lab Assignment 3

Name: Tushar Mittal

PRN: 1032200956

Roll No: PB68

Panel: B

Batch: B2

Code:

Macro.java

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Hashtable;

public class Macro {
```

```
    ArrayList<String> mdt = new ArrayList<>();
    ArrayList<String> alatab = new ArrayList<>();
    ArrayList<String> alaac = new ArrayList<>();
    Hashtable<String, MNTable> mnt = new Hashtable<>();

    public void passOne(BufferedReader br) throws Exception {
        Integer mntc = 0;
        Integer mdtc = 0;
        String line;
        boolean isMacro = false;
        BufferedWriter wrO = new BufferedWriter(new FileWriter("output.txt"));
        BufferedWriter wrE = new BufferedWriter(new FileWriter("expansion.txt"));

        while ((line = br.readLine()) != null) {
            if (isMacro) {
                if (line.contains("MEND")) {
                    isMacro = false;
                    mdt.add(mdtc, line);
                    BufferedWriter wrMdt = new BufferedWriter(new FileWriter("pass1mdt.txt"));
                    wrMdt.write("Index  Instruction\n");
                    for (int i = 0; i < mdt.size(); i++) {
                        wrMdt.write(String.format("%-6d %s%n", (i + 1), mdt.get(i)));
                        wrMdt.newLine();
                    }
                    wrMdt.close();
                    break;
                }
                mdt.add(mdtc, line);
                mdtc++;
            } else {
                if (line.contains("MACRO")) {
```

```
                isMacro = true;
                line = br.readLine();
                mdt.add(mdtc, line);
                String[] words = line.split(" ");
                String macroName = words[0];
                MNTable mnTable = new MNTable(mntc, macroName, mdtc);
                mnt.put(macroName, mnTable);
                for (int i = 1; i < words.length; i++) {
                    alatab.add(words[i].replace(",", ""));
                    alaac.add("?");
                }
                mntc++;
                mdtc++;
                continue;
            } else {
                wrO.write(line);
                wrO.newLine();
                wrE.write(line);
                wrE.newLine();
            }
        }
    }
    wrE.close();
    wrO.close();
}

public static void main(String[] args) throws Exception {
    Macro macro = new Macro();
    BufferedReader br = new BufferedReader(new FileReader("input.asm"));
    macro.passOne(br);
    br.close();
```

```
    }
}
```

MNTable.java

```java
public class MNTable {
    private Integer mntIndex;
    private String macroName;
    private Integer mdtIndex;

    public MNTable(Integer mntIndex, String macroName, Integer mdtIndex) {
        this.mntIndex = mntIndex;
        this.macroName = macroName;
        this.mdtIndex = mdtIndex;
    }

    public Integer getMntIndex() {
        return mntIndex;
    }

    public void setMntIndex(Integer mntIndex) {
        this.mntIndex = mntIndex;
    }

    public String getMacroName() {
        return macroName;
    }
```

```
    public void setMacroName(String macroName) {
        this.macroName = macroName;
    }

    public Integer getMdtIndex() {
        return mdtIndex;
    }

    public void setMdtIndex(Integer mdtIndex) {
        this.mdtIndex = mdtIndex;
    }
}
```

Input:

input.asm

```
MACRO
INCR &ARG1 &ARG2
ADD AREG &ARG1
ADD BREG &ARG2
MEND
START
MOVER AREG S1
MOVER BREG S1
INCR D1 D2
S1 DC 5
D1 DC 2
D2 DC 3
END
```

Output:

pass1mdt.txt

Index	Instruction
1	INCR &ARG1 &ARG2
2	ADD AREG &ARG1
3	ADD BREG &ARG2
4	MEND

mnt.txt

Index	Macro-Name	MDT-Index
1	INCR	1

alatab.txt

Index	Formal-Argument
#1	&ARG1
#2	&ARG2

output.txt

```
START
MOVER AREG S1
MOVER BREG S1
INCR D1 D2
S1 DC 5
D1 DC 2
D2 DC 3
END
```