```
/*
Name: Rohit Saini
RollNo: PC41
PRN: 1032200897
*/
Code:
import java.util.ArrayList;

class argument_list {
    String index;
    String argument;
}

class macro_name {
    int Index;
    String Name;
    int definition_pointer;
    ArrayList<argument_list> A = new ArrayList<argument_list>();
}

class macro_definition {
    int index;
    String definition;
}

public class lab3 {

    public static String mapper(String s, ArrayList<argument_list> arg,
ArrayList<macro_name> macro_list) {
        String ans = "";
        String[] temp = string_token.token(s, " ");
        boolean is_definition = true;
        for (int i = 0; i < temp.length; i++) {
            /* all the argument in arg */
            for (int j = 0; j < macro_list.size(); j++) {
                if (macro_list.get(j).Name.equals(temp[i])) {
                    ans += temp[i] + " ";
                    i++;
```

```
                    is_definition = false;
                    break;
                }
            }
            if (temp[i].charAt(0) == '&' && is_definition == true) {
                for (int j = 0; j < arg.size(); j++) {
                    if (temp[i].equals(arg.get(j).argument)) {
                        temp[i] = arg.get(j).index;
                    }
                }
            }

            ans += temp[i] + " ";
        }

        return ans;
    }

    public static int Macro_position(String s, ArrayList<macro_name>
macro_name) {
        for (int i = 0; i < macro_name.size(); i++) {
            if (s.equals(macro_name.get(i).Name)) {
                return i + 1;
            }
        }
        return -1;
    }

    public static String argument_match(String data, String definition,
ArrayList<argument_list> arg)
    {
        String[] data_token = string_token.token(data, " ");
        String ans = "";
        String[] temp = string_token.token(definition, " ");
        for (int i = 0; i < temp.length; i++)
        {
            if(temp[i].charAt(0) == '&')
            {
```

```
                // System.out.println("Argument: "+temp[i]);
                for (int j = 0; j < arg.size(); j++) {
                    if (arg.get(j).argument.equals(temp[i])) {
                        temp[i] = data_token[j + 1];
                        break;
                    }
                }
            }
            ans += temp[i]+" ";
        }
        return ans+"\n";
    }

    public static String expand_macro(String Expanded_code,
ArrayList<macro_definition> macro_definition_table,
        ArrayList<macro_name> macro_name, String temp, int macro_pos) {
        String[] temp1 = string_token.token(temp, " ");
        // int total_arguments = temp1.length-1;
        // System.out.println(total_arguments);
        int definition_pointer = 0;
        int macro_name_pointer = 0;
        for (int i = 0; i < macro_name.size(); i++) {
            if (temp1[0].equals(macro_name.get(i).Name)) {
                definition_pointer = macro_name.get(i).definition_pointer;
                macro_name_pointer = i;
                break;
            }
        }
        while(!macro_definition_table.get(definition_pointer).definition.e
quals("MEND"))
        {
            // System.out.println("data: " +
macro_definition_table.get(definition_pointer).definition);
            Expanded_code += argument_match(temp,
macro_definition_table.get(definition_pointer).definition,
                    macro_name.get(macro_name_pointer).A);
            definition_pointer++;
        }
```

```
        return Expanded_code;
    }

    public static void main(String[] args) {
        try {
            String input_data = reader.read();
            String[] output = string_token.token(input_data, "\n");
            String output_file = "";
            ArrayList<macro_name> macro_name = new
ArrayList<macro_name>();
            ArrayList<macro_definition> macro_definition_table = new
ArrayList<macro_definition>();
            ArrayList<argument_list> arg = new ArrayList<argument_list>();
            int count_macro = 0;
            for (int i = 0; i < output.length; i++) {
                String[] temp = string_token.token(output[i], " ");
                for (int j = 0; j < temp.length; j++) {
                    if (temp[0].equals("MACRO")) {
                        count_macro++;
                    }
                }
            }
            int count = 1;
            for (int i = 0; i < output.length; i++) {
                String[] temp = string_token.token(output[i], " ");
                /* Macro */
                if (temp[0].equals("MACRO")) {
                    i++;
                    temp = string_token.token(output[i], " ");
                    macro_name m = new macro_name();
                    m.Index = count++;
                    m.Name = temp[0];
                    m.definition_pointer = macro_definition_table.size() +
1;
                    macro_name.add(m);
                    macro_definition md = new macro_definition();
                    md.index = i;
                    md.definition = output[i];
```

```
                    macro_definition_table.add(md);
                    for (int j = 1; j < temp.length; j++) {
                        argument_list al = new argument_list();
                        al.index = "#" + j;
                        al.argument = temp[j];
                        m.A.add(al);
                        arg.add(al);
                    }
                } else if (!temp[0].equals("MEND") && count_macro > 0) {
                    macro_definition md = new macro_definition();
                    md.index = i;
                    md.definition = output[i];
                    macro_definition_table.add(md);
                } else if (temp[0].equals("MEND")) {
                    count_macro--;
                    macro_definition md = new macro_definition();
                    md.index = i;
                    md.definition = output[i];
                    macro_definition_table.add(md);
                } else {
                    output_file = output_file + '\n' + output[i];
                }
            }
            System.out.println("\nMacro Name Table:");
            for (int i = 0; i < macro_name.size(); i++) {
                System.out.println(macro_name.get(i).Index + " " +
macro_name.get(i).Name + " "
                        + macro_name.get(i).definition_pointer);
            }
            for (int i = 0; i < macro_name.size(); i++) {
                System.out.println("\nArgument List Table for " +
macro_name.get(i).Name);
                for (int j = 0; j < macro_name.get(i).A.size(); j++) {
                    System.out.println(macro_name.get(i).A.get(j).index +
" " + macro_name.get(i).A.get(j).argument);
                }
            }
            System.out.println("\nMacro Definition Table:");
```

```
            for (int i = 0; i < macro_definition_table.size(); i++) {
                System.out
                        .println(i + 1 + " "
                                +
mapper(macro_definition_table.get(i).definition, arg, macro_name));
            }
            System.out.println("\nPass-1 Ouput File:");
            System.out.println(output_file);

            /* Code for expantion of macro calling */
            String[] output_data = string_token.token(output_file, "\n");
            String Expanded_code = "";
            for (int i = 0; i < output_data.length; i++) {
                String[] temp = string_token.token(output_data[i], " ");
                int macro_pos = Macro_position(temp[0], macro_name);
                if (macro_pos != -1) {
                    /* get definition and replace the arguments from
argument table */
                    // System.out.println("Macro-Position: " + macro_pos);
                    Expanded_code = expand_macro(Expanded_code,
macro_definition_table, macro_name, output_data[i], macro_pos);

                } else {
                    Expanded_code += output_data[i];
                }
                Expanded_code += "\n";
            }
            System.out.println("\nExpantion of Macro: \n" +
Expanded_code);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
Input:
MACRO
```

```
M1 &ARG1, &ARG2
SUB AREG, &ARG1
ADD BREG, &ARG2
MEND
MACRO
M2 &ARG3, &ARG4
SUB AREG, &ARG3
ADD BREG, &ARG4
MEND
START
MOVER DREG, S1
M1 D3, D4
M2 D1, D2
S1 DC 5
END
```

Output:

```
PS C:\Users\rohit\Documents\GitHub\sem_7\ssc\Lab3> java lab3
Macro Name Table:
1 M1 1
2 M2 5
Argument List Table for M1
#1 &ARG1
#2 &ARG2
Argument List Table for M2
#1 &ARG3
#2 &ARG4
Macro Definition Table:
1 M1 &ARG1 &ARG2
2 SUB AREG #1
3 ADD BREG #2
4 MEND
5 M2 &ARG3 &ARG4
6 SUB AREG #1
7 ADD BREG #2
8 MEND
```

```
Ouput File:

START
MOVER DREG  S1
M1 D3  D4
M2 D1  D2
S1 DC 5
END
Expantion of Macro:
START
MOVER DREG  S1
SUB AREG D3
ADD BREG D4

SUB AREG D1
ADD BREG D2

S1 DC 5
END
```