/\*

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.equals("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:



