Name-Nissi Rathod

Class-TE-COM .B

Roll NO.-64

ASSIGNMENT NO.2

CODE:

```
#include <bits/stdc++.h>
using namespace std;

struct MNTEntry {
    string name;
    int mdtIndex;
};

vector<MNTEntry> MNT;         // Macro Name Table
vector<string> MDT;           // Macro Definition Table
vector<string> intermediate;    // Non-macro code lines

vector<string> program = {
    "MACRO",
    "INCR &ARG1, &ARG2",
    "LDA &ARG1",
    "ADD &ARG2",
    "STA &ARG1",
    "MEND",
    "START",
    "INCR A,B",

    "INCR X,Y",

    "END"

};


// ---------------- PASS 1 ----------------

void pass1() {

    bool inMacro = false;

    for (int i = 0; i < program.size(); i++) {

        string line = program[i];

        stringstream ss(line);

        string word; ss >> word;


        if (word == "MACRO") { inMacro = true; continue; }


        if (inMacro) {

            if (line.find("MEND") != string::npos) {

                MDT.push_back("MEND");

                inMacro = false;

                continue;

            }

            if (MNT.empty() || MDT.empty() || MDT.back() == "MEND") {

                // First line after MACRO — macro header

                MNT.push_back({word, (int)MDT.size()});

            }

            MDT.push_back(line);
        }
        else {
            intermediate.push_back(line);
        }
    }
}

// ---------------- PASS 2 ----------------
void pass2() {
    cout << "MNT (Macro Name Table):\n";
    for (auto &m : MNT)
        cout << m.name << " -> MDT Index " << m.mdtIndex << "\n";

    cout << "\nMDT (Macro Definition Table):\n";
    for (int i = 0; i < MDT.size(); i++)
        cout << i << " : " << MDT[i] << "\n";

    cout << "\nExpanded Code (PASS 2 Output):\n";
    for (auto &line : intermediate) {
        stringstream ss(line);
        string word; ss >> word;

        bool expanded = false;
        for (auto &m : MNT) {
            if (word == m.name) {
                string args; getline(ss, args);
                args.erase(remove(args.begin(), args.end(), ' '), args.end());
                vector<string> actualArgs;
                stringstream as(args);
                string temp;
                while (getline(as, temp, ',')) actualArgs.push_back(temp);

                for (int i = m.mdtIndex + 1; i < MDT.size() && MDT[i] != "MEND"; i++) {
                    string expandedLine = MDT[i];
                    for (int j = 0; j < actualArgs.size(); j++) {
                        string formal = "&ARG" + to_string(j + 1);
                        size_t pos = expandedLine.find(formal);
                        if (pos != string::npos)
                            expandedLine.replace(pos, formal.size(), actualArgs[j]);
                    }
                    cout << expandedLine << "\n";
                }
                expanded = true;
                break;
            }
        }
        if (!expanded) cout << line << "\n";
    }
}

// ---------------- MAIN ----------------
int main() {
    pass1();
    cout << "===== PASS 1 OUTPUT =====\n";
    cout << "Intermediate Code:\n";
    for (auto &l : intermediate) cout << l << "\n";
    cout << "\n========================\n\n";
    pass2();
}
```

OUTPUT:

```
===== PASS 1 OUTPUT =====
Intermediate Code:
START
INCR A,B
INCR X,Y
END
========================


MNT (Macro Name Table):
INCR -> MDT Index 0


MDT (Macro Definition Table):
0 : INCR &ARG1, &ARG2
1 : LDA &ARG1
2 : ADD &ARG2
3 : STA &ARG1
4 : MEND
```

Expanded Code (PASS 2 Output):

```
START
LDA A
ADD B
STA A
LDA X
ADD Y
STA X
END
```