

**Ahsanullah University of Science & Technology**

**Department of Computer Science & Engineering**

**Course No : CSE4130**

**Course Title : Formal Languages and Compilers Lab**

**Assignment No : 02**

**Date of Performance : 15/12/2022**

**Date of Submission : 29/12/2022**

**Submitted To : Mr. Aminur Rahman & Mr. Al Hasib Mahamud**

**Submitted By-**

**Group : B1**

**Name : Debopriya Deb Roy**

**Id : 190104065**

**Section : B**

#include <iostream>

#include <fstream>

#include <string>

#include <map>

#include <vector>

using namespace std;

typedef long long int lli;

lli i, j = 0;

#define pii pair<lli, lli>

ofstream output\_file;

ifstream input\_file("input.txt");

lli flag = 0;

map<string, int> k\_map;

map<char, int> op\_map;

map<char, int> paren\_map;

map<char, int> sperator\_map;

vector<string> keyWord{

    "auto", "break", "case", "char",

    "const", "continue", "default", "do",

    "double", "else", "enum", "extern",

    "float", "for", "goto", "if",

    "int", "long", "register", "return",

    "short", "signed", "sizeof", "static",

    "struct", "switch", "typedef", "union",

    "unsigned", "void", "volatile", "while"};

vector<char> parenthesis{

    '(', ')', '{', '}', '[', ']'};

vector<char> operato{'+', '-', '\*', '/', '%', '=', '<', '>'};

vector<char> separator{',', ';', '\'', '\"', ':'};

void sep\_key\_set()

{

    for (int m = 0; m < separator.size(); m++)

        sperator\_map[separator[m]] = 1;

}

void paren\_key\_set()

{

    for (int m = 0; m < parenthesis.size(); m++)

        paren\_map[parenthesis[m]] = 1;

}

void map\_key\_set()

{

    for (int m = 0; m < keyWord.size(); m++)

        k\_map[keyWord[m]] = 1;

}

void map\_op\_set()

{

    for (int m = 0; m < operato.size(); m++)

        op\_map[operato[m]] = 1;

}

// method------

void isKeyWord(string s)

{

    if (k\_map[s] == 1)

    {

        cout << "[kw " << s << "]"

             << " ";

        output\_file << "[kw " << s << "]"

                    << " ";

        flag = 1;

    }

}

void isParenthesis(string *s*)

{

    lli cnt = 0;

    for (int m = 0; m < s.size(); m++)

    {

        if (paren\_map[s[m]] == 1)

            cnt++;

    }

    if (cnt == s.size() && cnt > 0)

    {

        flag = 1;

        cout << "[par " << s << "]"

             << " ";

        output\_file << "[par " << s << "]"

                    << " ";

    }

    else

        flag = 0;

}

void isOperator(string *s*)

{

    lli cnt = 0;

    for (int m = 0; m < s.size(); m++)

    {

        if (op\_map[s[m]] == 1)

            cnt++;

    }

    if (cnt == s.size())

    {

        flag = 1;

        cout << "[op " << s << "]"

             << " ";

        output\_file << "[op " << s << "]"

                    << " ";

    }

    else

        flag = 0;

}

void isSeparator(string s)

{

    lli cnt = 0;

    for (int m = 0; m < s.size(); m++)

    {

        if (sperator\_map[s[m]] == 1)

        {

            cnt++;

        }

    }

    if (cnt == s.size() && cnt > 0)

    {

        flag = 1;

        cout << "[sep " << s << "]"

             << " ";

        output\_file << "[sep " << s << "]"

                    << " ";

    }

    else

        flag = 0;

}

void isNumber(string *s*)

{

    lli cnt = 0;

    for (int m = 0; m < s.size(); m++)

    {

        lli x = s[m] - '0';

        if (x >= 0 && x <= 9 || s[m] == '.')

        {

            cnt++;

        }

    }

    if (cnt == s.size())

    {

        cout << "[num " << s << "]"

             << " ";

        j++;

        output\_file << "[num " << s << "]"

                    << " ";

        flag = 1;

    }

    else

        flag = 0;

}

void isIdentifier(string *s*)

{

    if (k\_map[s] == 1)

    {

        return;

    }

    lli cnt = 0;

    if (s[0] >= 'A' && s[0] <= 'Z' || s[0] >= 'a' && s[0] <= 'z' || s[0] == '\_')

    {

        cnt++;

        for (int k = 1; k < s.size(); k++)

        {

            if (s[k] >= 'A' && s[k] <= 'Z' || s[k] >= 'a' && s[k] <= 'z' || s[k] == '\_' || s[k] >= '0' && s[k] <= '9')

                cnt++;

        }

        if (cnt == s.size())

        {

            flag = 1;

            cout << "[id " << s << "]"

                 << " ";

            output\_file << "[id " << s << "]"

                        << " ";

        }

        else

            flag = 0;

    }

}

void check(string s)

{

    if (!flag)

        isKeyWord(s);

    if (!flag)

        isIdentifier(s);

    if (!flag)

        isNumber(s);

    if (!flag)

        isOperator(s);

    if (!flag)

        isParenthesis(s);

    if (!flag)

        isSeparator(s);

    if (!flag)

    {

        cout << "[Unkn " << s << "]"

             << " ";

        output\_file << "[Unkn " << s << "]"

                    << " ";

    }

}

void main\_code(string line)

{

    string str = "";

    for (i = 0; i < line.size(); i++)

    {

        if (line[i] != ' ')

            str += line[i];

        if (line[i] == ' ')

        {

            flag = 0;

            check(str);

            str = "";

        }

    }

    if (i == (line.size()))

    {

        flag = 0;

        check(str);

    }

}

void solve()

{

    map\_key\_set();

    map\_op\_set();

    paren\_key\_set();

    sep\_key\_set();

    string line;

    lli cnt = 0;

    output\_file.open("output.txt");

    if (!input\_file.is\_open())

        cout << "Failed to Open" << endl;

    else

    {

        cout << "INPUT: " << endl;

        while (getline(input\_file, line))

        {

            cout << line << endl

                 << endl;

            output\_file << line << endl

                        << endl;

            cout << "OUTPUT " << endl;

            output\_file << "OUTPUT " << endl;

            main\_code(line);

        }

        input\_file.close();

        output\_file.close();

    }

}

int main()

{

    ios\_base::sync\_with\_stdio(false);

    cin.tie(NULL);

    cout.tie(NULL);

    solve();

    return 0;

}