Name-Amrit Aanand

Reg No- RA1911003010917

Lexical Analyzer Implementation

//CODE//

#include <bits/stdc++.h>

using namespace std;

string keywords[] = {"void", "using", "namespace", "int", "include", "iostream", "cin", "cout", "return", "float", "double", "string"};

bool isKeyword(string a)

{

for (string c : keywords)

if (c == a)

return true;

return false;

}

int main()

{

fstream file;

string s, filename;

filename = "./file.cpp";

file.open(filename.c\_str());

int k = 0, i = 0, l = 0, p = 0, o = 0, c = 0;

while (file >> s)

{

if (s == "#include<iostream>")

{

cout << s << " is a header" << endl;

}

else if (s == "+" || s == "-" || s == "" || s == "/" || s == "^" || s == "&&" || s == "||" || s == "=" || s == "==" || s == "&" || s == "|" || s == "%" || s == "++" || s == "--" || s == "+=" || s == "-=" || s == "/=" || s == "=" || s == "%=")

{

cout << s << " is an operator" << endl;

o++;

s = "";

}

else if (isKeyword(s))

{

cout << s << " is a keyword\n";

k++;

s = "";

}

else if (s == "(" || s == "{" || s == "[" || s == ")" || s == "}" || s == "]" || s == "<" || s == ">" || s == "()" || s == ";" || s == "<<" || s == ">>" || s == "," || s == "#")

{

cout << s << " is a punctuator" << endl;

p++;

}

else if (isdigit(s[0]))

{

int x = 0;

if (!isdigit(s[x++]))

continue;

else

{

cout << s << " is a constant" << endl;

c++;

s = "";

}

}

else

{

cout << s << " is an identifier" << endl;

i++;

s = "";

}

}

cout << k << " keywords\n"

<< c << " constants\n"

<< i << " identifiers\n"

<< p << " punctuators\n"

<< o << " operators\n";

}

//OUTPUT//

