**FSM to find identifiers/keywords**

#include <iostream>

#include <string>

#include <list>

using namespace std;

class IdentifierFSM

{

bool CheckFirstCharacter(char d)

{

if ((d >= 'a' && d <= 'z') ||

(d >= 'A' && d <= 'Z') ||

d == '\_')

{

return true;

}

return false;

}

bool isNumber(char d)

{

if (d >= '0' && d <= '9')

{

return true;

}

return false;

}

bool isStringStarted = false;

bool isString(char d)

{

if (d == '\"')

{

isStringStarted = !isStringStarted;

}

}

public:

void DifferentiateTokens(string input)

{

string tempToken;

for (int i = 0; i < input.length(); i++)

{

char t = input[i];

if (isString(t))

{

continue;

}

if (tempToken.length() > 0)

{

if (CheckFirstCharacter(t) || isNumber(t))

{

tempToken += t;

}

else

{

cout << tempToken << ",";

tempToken.clear();

}

}

else

{

if (CheckFirstCharacter(t))

{

tempToken += t;

}

}

}

}

};

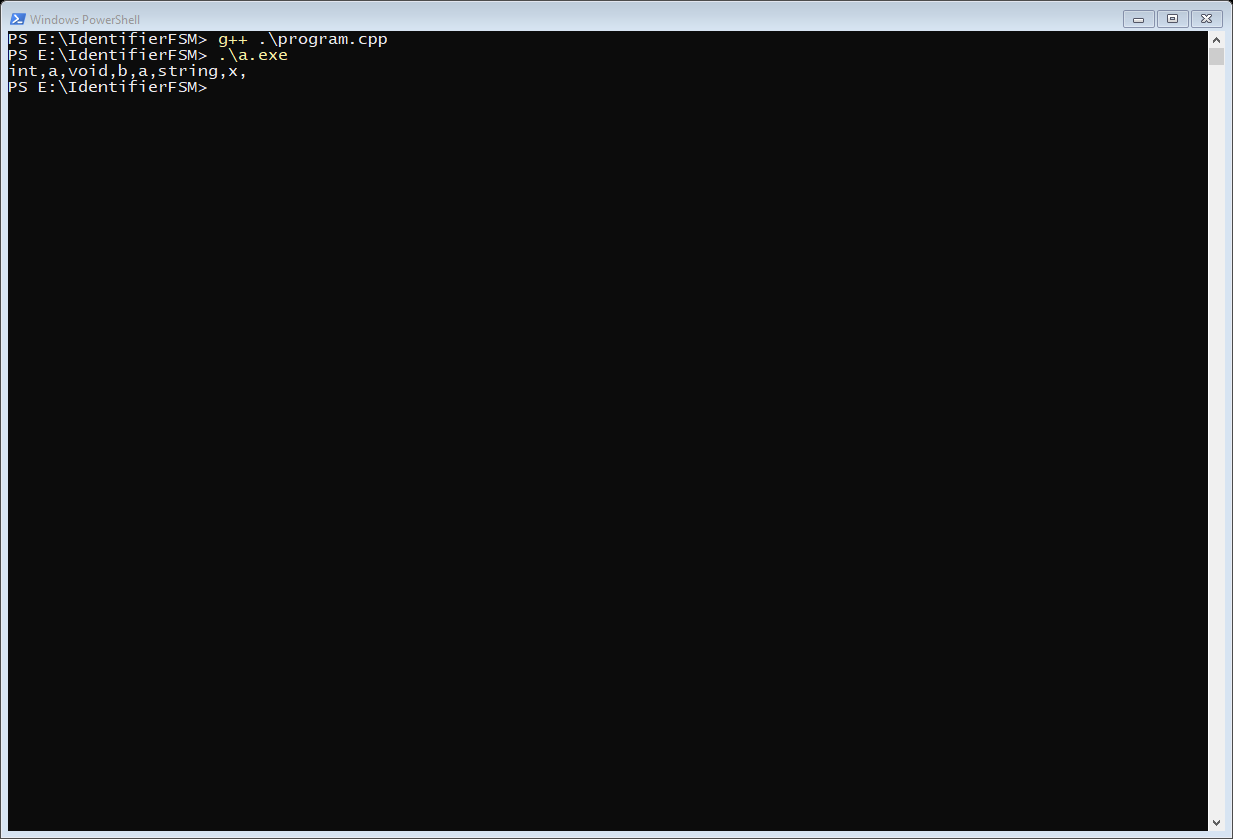
int main()

{

IdentifierFSM fsm;

fsm.DifferentiateTokens(string("int a; void b=a+5;string x=\"aishwarya\""));

}



**FSM to check if number is integer, real or exponential**

#include <iostream>

#include <string>

using namespace std;

class FSM

{

bool isReal(string n)

{

if (n.find('.') != string::npos)

{

return true;

}

return false;

}

bool isExponential(string n)

{

if (n.find('e') != string::npos)

{

return true;

}

return false;

}

public:

void GetNumberType(string number)

{

if (isExponential(number))

{

cout << "Exponential ";

}

else if (isReal(number))

{

cout << "Real ";

}

else

{

cout << "Integer";

}

cout << "\n";

}

};

int main()

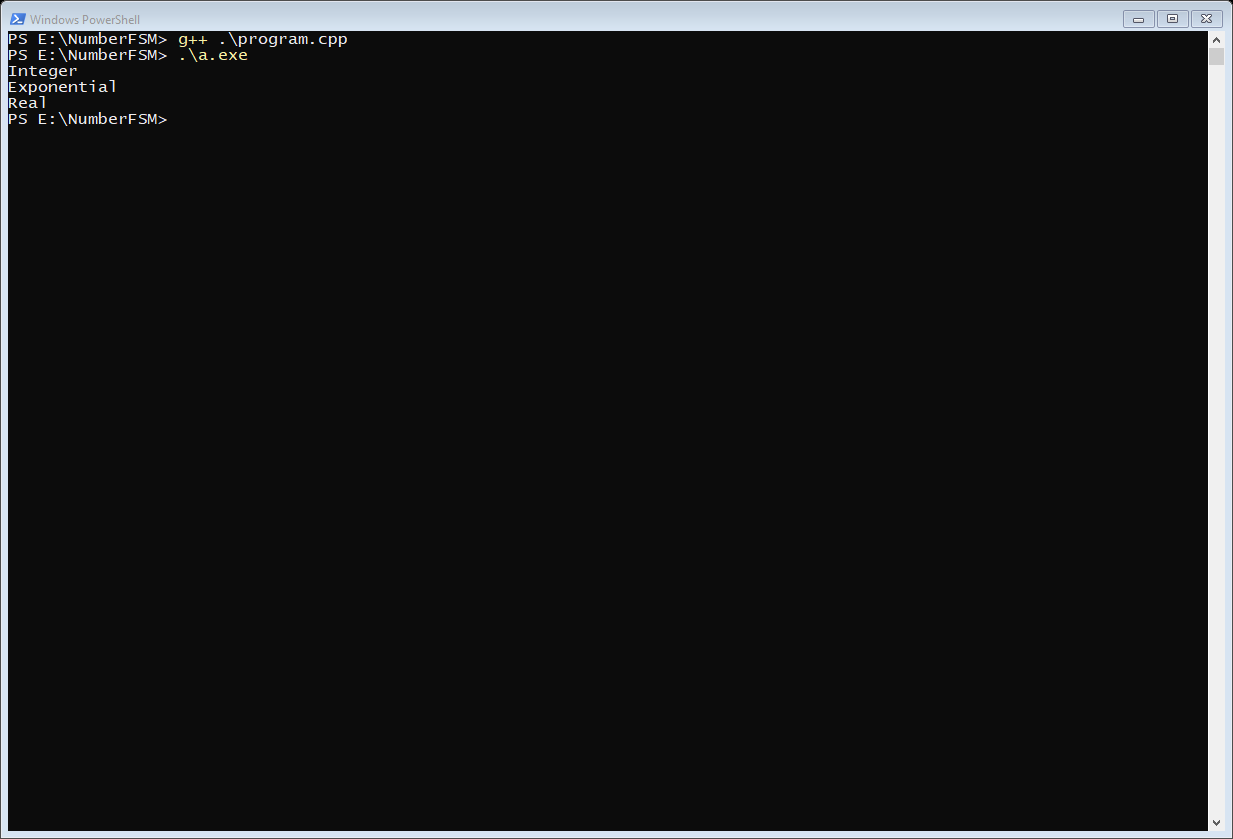
{

FSM fsm;

fsm.GetNumberType("12");

fsm.GetNumberType("12.4e14");

fsm.GetNumberType("12.0");

}