表达式求值——栈的运用

#include<iostream>

#include<string>

#include<stack>

#include<cmath>

#include<iomanip>

using namespace std;

int getprior(char op)

{

switch (op)

{

case '+':

case '-':

return 1;

case '\*':

case '/':

case '%':

return 2;

case '^':

return 3;

case '(':

return 0;

case '#':

return -1;

}

}

double calculate(double a,char op,double b)

{

double c;

switch (op)

{

case '+':

{c = b + a; break; }

case '-':

{c = b - a; break; }

case '\*':

{c = b \* a; break; }

case '/':

{c = b / a; break;}

case '%':

{c = (int)b % (int)a; break; }

case '^':

{c = pow(b, a); break; }

}

return c;

}

int main()

{

string str;

cin >> str;

stack<double> stack1\_num;

stack<char> stack2\_op;

stack2\_op.push('#');

for (int i = 0; i < str.length(); )

{

if ((int)str[i] >= 0 && (int)str[i] <= 9)

{

stack1\_num.push((double)str[i]);

}

else if (str[i] == '(')

{

stack2\_op.push('(');

i++;

}

else if (str[i] == ')')

{

i++;

while (stack2\_op.top() != '(' && stack2\_op.size() > 1)

{

double a = stack1\_num.top();

stack1\_num.pop();

double b = stack1\_num.top();

stack1\_num.pop();

char op = stack2\_op.top();

stack2\_op.pop();

double result = calculate(a, op, b);

stack1\_num.push(result);

}

stack2\_op.pop();

if (stack2\_op.size() < 1)

{

cout << "ERROR IN INFIX NOTATION" << endl;

return 0;

}

}

else if (str[i] == '+' || str[i] == '-' || str[i] == '\*' || str[i] == '/' || str[i] == '^' || str[i] == '%')

{

while (getprior(str[i]) <= getprior(stack2\_op.top()) && stack1\_num.size() >= 2 && str[i] != '^')

{

double a = stack1\_num.top();

stack1\_num.pop();

double b = stack1\_num.top();

stack1\_num.pop();

char op = stack2\_op.top();

stack2\_op.pop();

double result = calculate(a, op, b);

stack1\_num.push(result);

}

stack2\_op.push(str[i]);

i++;

}

else

{

cout << "ERROR IN INFIX NOTATION" << endl;

return 0;

}

}

while (stack1\_num.size() >= 2)

{

double a = stack1\_num.top();

stack1\_num.pop();

double b = stack1\_num.top();

stack1\_num.pop();

char op = stack2\_op.top();

stack2\_op.pop();

double result = calculate(a, op, b);

stack1\_num.push(result);

stack2\_op.pop();

}

stack2\_op.pop();

if (!stack2\_op.empty())

{

cout << "ERROR IN INFIX NOTATION" << endl;

return 0;

}

else if (stack1\_num.size() != 1)

{

cout << "ERROR IN INFIX NOTATION" << endl;

return 0;

}

else

{

cout << fixed << setprecision(2) <<stack1\_num.top() << endl;

return 0;

}

return 0;

}