#include<iostream>

#include<cstdlib>

using namespace std;

const int maxsize = 100;

template<class T>

class Stack {

public:

int top;

T stack[maxsize];

Stack() {

top = 0;

}

bool empty() {

if (top == 0) { return true; }

else { return false; }

}

void push(T n) {

top++;

stack[top] = n;

}

void pop() {

if (top == 0) return;

else { top--; }

}

T get() {

return stack[top];

}

};

int main() {

Stack<double>number;

Stack<char>Symbol;

char A[50];

cin >> A;

double x = 0;

double y = 0;

double z = 0;

for (int i = 0;; i++) {

if (A[i] == '\0') {

break;

}//到字符串末尾则跳出循环

if (A[i] == '+' || A[i] == '-' || A[i] == '\*' || A[i] == '/' || A[i] == '(' || A[i] == ')') {

if (Symbol.empty()) {

//考虑负数

if ((A[i] == '-' && i == 0) || (i > 0 && A[i - 1] == '(')) {

for (int j = i + 1;; j++) {

if (A[j] >= 48 && A[j] <= 57) {

y = y \* 10 + A[j] - 48;

}

else {

if (A[j] == '.') {

while (1) {

j++;

if (A[j] >= 48 && A[j] <= 57) {

double key = A[j] - 48;//防止先去尾再转换成double

z = z / 10 + key / 10;

}

else { break; }

}

}

number.push(-(y + z));

i = j - 1;

z = 0;

y = 0;

break;

}

}

continue;

}

else {

Symbol.push(A[i]);

continue;

}

}//符号栈为空，则直接进栈

if(A[i] == '('){ Symbol.push(A[i]); continue;}//左括号直接进栈

if (A[i] == ')') {

for (int k =Symbol.top;; k--) {

if (A[k] == '(') {

Symbol.pop();

break;

}//不断寻找左括号，然后跳出循环

if (Symbol.stack[k] == '+' ) {

Symbol.pop();

x = number.stack[number.top - 1] + number.stack[number.top];//将操作数栈前两个进行运算

number.pop();

number.pop();

number.push(x);

}

if (Symbol.stack[k] == '-') {

Symbol.pop();

x = number.stack[number.top - 1] - number.stack[number.top ];

number.pop();

number.pop();

number.push(x);

}

if (Symbol.stack[k] == '\*') {

Symbol.pop();

x = number.stack[number.top - 1] \* number.stack[number.top ];

number.pop();

number.pop();

number.push(x);

}

if (Symbol.stack[k] == '/') {

Symbol.pop();

x = number.stack[number.top - 1] / number.stack[number.top ];

number.pop();

number.pop();

number.push(x);

}

}

continue;

}//若是右括号，就一直弹出符号，直至左括号

if (A[i] == '\*') {

if (Symbol.stack[Symbol.top ] == '/') {

x = number.stack[number.top - 1] / number.stack[number.top ];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('\*');

continue;

}

if (Symbol.stack[Symbol.top] == '\*') {

x = number.stack[number.top - 1] \* number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('\*');

continue;

}

else{ Symbol.push('\*'); }

continue;

}//乘号

if (A[i] == '/') {

if (Symbol.stack[Symbol.top] == '\*') {

x = number.stack[number.top - 1] \* number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('/');

continue;

}

if (Symbol.stack[Symbol.top] == '/') {

x = number.stack[number.top - 1] / number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('/');

continue;

}

else{ Symbol.push('/'); }

continue;

}//除号

if (A[i] == '+') {

if (Symbol.stack[Symbol.top] == '\*') {

x = number.stack[number.top - 1] \* number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('+');

continue;

}

if (Symbol.stack[Symbol.top] == '/') {

x = number.stack[number.top - 1] / number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('+');

continue;

}

if (Symbol.stack[Symbol.top] == '+'|| Symbol.stack[Symbol.top] == '-') {

Symbol.push('+');

continue;

}

Symbol.push('+');

continue;

}//加号

if (A[i] == '-') {

//考虑负数

if ((A[i] == '-' && i == 0) || (i > 0 && A[i - 1] == '(')) {

for (int j = i+1;; j++) {

if (A[j] >= 48 && A[j] <= 57) {

y = y \* 10 + A[j] - 48;

}

else {

if (A[j] == '.') {

while (1) {

j++;

if (A[j] >= 48 && A[j] <= 57) {

double key = A[j] - 48;//防止先去尾再转换成double

z = z / 10 + key / 10;

}

else { break; }

}

}

double h = -(y + z);

number.push(h);

i = j - 1;

z = 0;

y = 0;

break;

}

}

continue;

}

if (Symbol.stack[Symbol.top] == '\*') {

x = number.stack[number.top - 1] \* number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('-');

continue;

}

if (Symbol.stack[Symbol.top] == '/') {

x = number.stack[number.top - 1] / number.stack[number.top];

number.pop();

number.pop();

number.push(x);

Symbol.pop();

Symbol.push('-');

continue;

}

if (Symbol.stack[Symbol.top] == '+' || Symbol.stack[Symbol.top] == '-') {

Symbol.push('-');

continue;

}

Symbol.push('-');

continue;

}//减号

}//符号

else {

for (int j=i;; j++) {

if (A[j] >= 48 && A[j]<= 57) {

y = y \* 10 + A[j] - 48;

}

else {

//小数实现

if (A[j] == '.') {

while (1) {

j++;

if (A[j] >= 48 && A[j] <= 57) {

double key = A[j] - 48;//防止先去尾再转换成double

z = z / 10 + key/10;

}

else { break; }

}

}

number.push(y+z);

i = j - 1;

z = 0;

y = 0;

break;

}

}

}

}

if (Symbol.empty()) {

x = number.stack[number.top];

cout << x;

system("pause");

return 0;

}

else {

for (int i = Symbol.top; i!=0; i--) {

if (Symbol.stack[i] == '\*') {

x = number.stack[number.top - 1] \* number.stack[number.top];

number.pop();

number.pop();

number.push(x);

}//乘号

if (Symbol.stack[i] == '/') {

x = number.stack[number.top - 1] / number.stack[number.top];

number.pop();

number.pop();

number.push(x);

}//除号

if (Symbol.stack[i]=='+') {

x = number.stack[number.top - 1] + number.stack[number.top];

number.pop();

number.pop();

number.push(x);

}//加号

if (Symbol.stack[i] == '-') {

x = number.stack[number.top - 1] - number.stack[number.top];

number.pop();

number.pop();

number.push(x);

}//减号

}

}

x = number.stack[number.top];

cout << x;

system("pause");

return 0;

}