1、功能点：[计算器](https://www.baidu.com/s?wd=%E8%AE%A1%E7%AE%97%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YzPj63mHu-njf3PH0YPy7-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH0dnjTvnWf1nWRLrjfvrjc4rf" \t "https://zhidao.baidu.com/question/_blank) 加法功能   
测试点：  
1)是否可以正常输入数据   
2）是否可以修改输入的数据   
3）是否数字相加结果正确   
4）是否可以多个数相加，并且数据正确

2、C++编程源代码  
#include<iostream>  
#include<cmath>  
#include<cstdlib>  
using namespace std;  
[#define](https://www.baidu.com/s?wd=%23define&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YzPANbPWuWn1fLnhckn1R10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EP1R3P1mdrjn" \t "https://zhidao.baidu.com/question/_blank) MAX 1000  
  
struct save1  
{  
float n[MAX];  
int top;  
}stack1;  
  
struct save2  
{  
char n[MAX];  
int top;  
}stack2;  
//stack1存储数字，stack2存储运算符号.  
  
bool stackempty(save1 s)//判断是否为空  
{  
if (s.top== -1)  
return 1;  
else  
return 0;  
}  
  
bool stackempty2(save2 s)//判断是否为空  
{  
if (s.top== -1)  
return 1;  
else  
return 0;  
}  
  
void push(save1 &s,float e)//将e入栈  
{  
if(s.top==MAX-1)  
{  
cout<<"栈已满"<<endl;  
return ;  
}  
s.top++;  
s.n[s.top]=e;  
}  
  
void push2(save2 &s,char e)//将e入栈  
{  
if(s.top==MAX-1)  
{  
cout<<"栈已满"<<endl;  
return ;  
}  
s.top++;  
s.n[s.top]=e;  
}  
  
void pop(save1 &s,float &e)//将栈顶元素出栈，存到e中  
{  
if(s.top==-1)  
{ cout<<"栈为空"<<endl; }  
else  
{e=s.n[s.top]; s.top--; }  
}  
  
void pop2(save2 &s,char &e)//将栈顶元素出栈，存到e中  
{  
if(s.top==-1)  
{ cout<<"栈为空"<<endl; }  
else  
{e=s.n[s.top]; s.top--; }  
}  
  
int in(char e)//e在栈内的优先级别  
{  
if(e=='-' || e=='+') return 2;  
if(e=='\*' || e=='/') return 4;  
if(e=='^') return 5;  
if(e=='(') return 0;  
if(e==')') return 7;  
return -1;  
}  
  
int out(char e)//e在栈外的优先级别  
{  
if(e=='-' || e=='+') return 1;  
if(e=='\*' || e=='/') return 3;  
if(e=='^') return 6;  
if(e=='(') return 7;  
if(e==')') return 0;  
return -1;  
}  
  
void count(float a,char ope,float b)//进行计算并将计算结果入栈  
{  
float sum;  
if(ope=='+') sum=a+b;  
if(ope=='-') sum=a-b;  
if(ope=='\*') sum=a\*b;  
if(ope=='/') sum=a/b;  
if(ope=='^') sum=pow(a,b);  
push(stack1,sum);  
}  
  
int main()  
{  
int i=0,len,j,nofpoint,g=0;//len表示输入式子的长度。 g表示读入的字符是否是字母变量、数字以及运算符。  
float a,b;//a、b用来存储操作数栈中弹出的操作数，便于代入函数中进行计算。  
char line[MAX],operate,temp[20];  
cout<<"请输入表达式"<<endl;  
cin>>line;  
len=strlen(line);  
stack1.top=-1;//将栈置为空  
stack2.top=-1;//将栈置为空  
while(1)  
{  
g=0;  
if(isdigit(line[i]))//若读入的字符为数字，则继续判断下一个字符，直到下一个字符不是数字或者不是小数点，即可保证该操作数是完整的小数，然后将该数入操作数栈。  
{  
j=0; g=1;  
nofpoint=0;//记录所存的数中小数点的个数  
while(isdigit(line[i]) || line[i]=='.')  
{  
if(line[i]=='.') nofpoint++;  
temp[j++]=line[i];  
i++;  
if(i>=len) break;  
}  
if( nofpoint>1 || (i<len&&(line[i]!='-' && line[i]!='+' && line[i]!='\*' && line[i]!='/' && line[i]!='^' && line[i]!=')')) )  
{ cout<<"表达式有错"<<endl; return 0; }//所存数中含有不止一个小数点，或者数字后面跟的不是“+、-、\*、/、^、)”，则为错误  
  
temp[j]='\0';  
b=atof(temp);  
push(stack1,b);  
if(i>=len) break;  
}  
else  
{  
if(line[i]=='-' || line[i]=='+' || line[i]=='\*' || line[i]=='/' ||  
line[i]=='^' || line[i]=='(' || line[i]==')' ) //若读入的字符为运算符的情况  
{  
g=1;  
if(line[i]=='(' && i==len) { cout<<"表达式有错"<<endl; return 0; }// “(”放表达式最后面，错误  
if(line[i]=='-' || line[i]=='+' || line[i]=='\*' || line[i]=='/' || line[i]=='^')  
{  
if(i==len) { cout<<"表达式有错"<<endl; return 0; }//“+、-、\*、/、^”放在表达式最后面，错误  
if( (!isdigit(line[i+1])) && (!isalpha(line[i+1])) && line[i+1]!='(')//“+、-、\*、/、^”后面跟的不是数字或者变量，错误  
{ cout<<"表达式有错"<<endl; return 0; }  
}  
  
if(line[i]=='-' && (i==0 || line[i-1]=='(' ))//运算符是负号  
{  
push(stack1,0);  
push2(stack2,line[i]);  
i++;  
}  
else  
{ //读入的运算符与运算符栈的栈顶元素相比，并进行相应的操作  
if(in(stack2.n[stack2.top])<out(line[i])||stackempty2(stack2)) { push2(stack2,line[i]);i++;}  
else  
if(in(stack2.n[stack2.top])==out(line[i])) {i++; stack2.top--;}  
else  
if(in(stack2.n[stack2.top])>out(line[i]))  
{  
pop(stack1,a);  
pop(stack1,b);  
pop2(stack2,operate);  
count(b,operate,a);  
}  
if(i>=len) break;  
}  
}  
else  
{  
if(isalpha(line[i]))//读入的字符是字母变量的情况  
{  
g=1;  
cout<<"请输入变量";  
while( isalpha(line[i])) { cout<<line[i]; i++; }  
cout<<"的值"<<endl;  
cin>>b;  
push(stack1,b);  
if(i>=len) break;  
if(line[i]!='-' && line[i]!='+' && line[i]!='\*' && line[i]!='/' && line[i]!='^' && line[i]!=')')//变量后面跟的不是“+、-、\*、/、^、)”，则为错误  
{ cout<<"表达式有错"<<endl; return 0; }  
}  
}  
}  
if(g==0) { cout<<"表达式有错"<<endl; return 0; }//g=0表示该字符是不符合要求的字符  
}  
  
while(stack2.top!=-1)//读入结束后，继续进行操作，直到运算符栈为空  
{  
pop(stack1,a);  
pop(stack1,b);  
pop2(stack2,operate);  
if(operate=='(' || operate==')') //括号多余的情况  
{ cout<<"表达式有错"<<endl; return 0; }  
count(b,operate,a);  
}  
cout<<stack1.n[stack1.top]<<endl;  
return 0;  
}