**表达式计算**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　输入一个只包含加减乖除和括号的合法表达式，求表达式的值。其中除表示整除。

输入格式

　　输入一行，包含一个表达式。

输出格式

　　输出这个表达式的值。

样例输入

1-2+3\*(4-5)

样例输出

-4

数据规模和约定

表达式长度不超过100，表达式运算合法且运算过程都在int内进行。

本题的C参考代码如下：

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAXSIZE 101

void Translate(char str[], char exp[])

{

char stack[MAXSIZE];

int i = 0, k = 0, top = -1;

while (str[i] != '\0')

{

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

{

stack[++top] = str[i++];

}

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

{

while (top >= 0 && stack[top] != '(')

{

exp[k++] = stack[top--];

}

if (top < 0)

{

exp[k++] = '\0';

return ;

}

top--;

i++;

}

else if (str[i] == '+' || str[i] == '-')

{

while (top >= 0 && stack[top] != '(')

{

exp[k++] = stack[top--];

}

stack[++top] = str[i++];

}

else if (str[i] == '\*' || str[i] == '/')

{

while (top >= 0 && (stack[top] == '\*' || stack[top] == '/'))

{

exp[k++] = stack[top--];

}

stack[++top] = str[i++];

}

else

{

while ((str[i] >= '0' && str[i] <= '9'))

{

exp[k++] = str[i++];

}

exp[k++] = '#';

}

}

while (top >= 0)

{

if (stack[top] == '(')

{

exp[k++] = '\0';

return ;

}

exp[k++] = stack[top--];

}

exp[k++] = '\0';

}

int CompValue(char \*exp)

{

char tempStr[MAXSIZE];

int stack[MAXSIZE];

int i = 0, k = 0, top = -1;

while (exp[i] != '\0')

{

if (exp[i] >= '0' && exp[i] <= '9')

{

k = 0;

while (exp[i] != '#')

{

tempStr[k++] = exp[i++];

}

tempStr[k] = '\0';

stack[++top] = atoi(tempStr);

i++;

}

else

{

switch (exp[i++])

{

case '+' : stack[top-1] += stack[top];

break;

case '-' : stack[top-1] -= stack[top];

break;

case '\*' : stack[top-1] \*= stack[top];

break;

case '/' : if (stack[top] != 0)

{

stack[top-1] /= stack[top];

}

else

{

return 0;

}

break;

}

top--;

}

}

return stack[top];

}

int main()

{

char buf[MAXSIZE],buf1[MAXSIZE];

scanf("%s",buf);

Translate(buf,buf1);

printf("%d",CompValue(buf1));

return 0;

}

本题的C++参考代码如下：

#include<iostream>

//#include<stdio.h>

#include<stdlib.h>

using namespace std;

typedef int ETlemType;

typedef int status;

#define MAXSIZE 15//顺序栈的最大长度

#define OK 1

#define ERROR 0

#define OVERSTACK 0//满栈溢出

typedef struct Stack1

{

ETlemType date[MAXSIZE];

int top;

//int stacksiza;//栈的最大长度

}SqStack;

int Compare(char ch,char b);

status InitStack(SqStack &s);//建立空栈

status Push(SqStack &s,ETlemType e);//进栈

status StackEmpty( SqStack s );//判断栈是否为空

status Pop(SqStack &s,ETlemType &e);//出栈

status GetTop(SqStack s, ETlemType &e);//取栈顶元素

int Operation(char ch,int ch1,int ch2);//运算

int houzhi(int Out[],int pd[],char a[],SqStack s);//转换为后缀表达式,pd用于判断是数字还是运算符.,同事时返回长度

int zhuanhuan(char a[],char b[]);

//int zhuanhuan(char a[],char b[]);

int main()

{

int i=0,j=0;

char a[1000];

int Out[500],pd[500],ch;

int length=0;

SqStack s;//数字栈，运算符栈

cin>>a;

//cout<<a<<endl;

// length=zhuanhuan(b,a);

InitStack(s);

length=houzhi(Out,pd,a,s);

// cout<<"ooo"<<Out[0]<<endl;

i=0;

int ch1=0,ch2=0;

do

{

if(pd[i]==1)

{

ch=Out[i];

Push(s,ch);//入栈

//cout<<ch<<endl;

}

else

{

Pop(s,ch1);

Pop(s,ch2);

/\* cout<<"ch1 "<<(int)ch1<<endl;

cout<<"ch2 "<<(int)ch2<<endl;

cout<<"测试"<<Operation(Out[j],(int)ch1,(int)ch2)<<endl;\*/

Push(s,Operation((char)Out[i],(int)ch1,(int)ch2));

}

i++;

}while(i<length);

cout<<s.date[0]<<endl;

// cout<<(int)s.date[0]<<endl;

return 0;

}

int houzhi(int Out[],int pd[],char a[],SqStack s)//转换为后缀表达式,pd用于判断是数字还是运算符

{

int ch,sum;

int i=0,j=0;

do

{

if(a[i]<='9' && a[i]>='0')

{

sum=0;

while(a[i]<='9' && a[i]>='0')

{sum=sum\*10+a[i]-'0';i++;}

i--;

Out[j++]=sum;pd[j-1]=1;

}

else if(a[i]==')'){

do

{

Pop(s,ch);

if(ch=='(')break;

Out[j++]=ch;pd[j-1]=0;

}while(1);

}

else if(a[i]=='(')Push(s,a[i]);

else if(a[i]=='\0'){//cout<<"111"<<endl;

do

{

Pop(s,ch);

Out[j++]=ch;

pd[j-1]=0;//cout<<"333"<<endl;

}while(s.top!=-1);//

break;//运算符移位完成

}

else{

if(StackEmpty(s))

{//如果栈不为空

GetTop(s,ch);

if( Compare(ch,a[i])>0 && ch!='(')//栈顶元素优先级高则输出

{//cout<<"111"<<endl;//p;

while(Compare(ch,a[i])>0 && ch!='(' && StackEmpty(s) )

{Pop(s,ch);Out[j++]=ch;pd[j-1]=0;GetTop(s,ch);}//[[[[[[[[[[

}

}

Push(s,a[i]);

}

i++;

}while(1);//运算符中缀改为后缀

//Out[j]=0;//puts(Out);//实验

/\*for(i=0;i<j;i++)

{

cout<<(char)Out[i]<<" ";

}

cout<<endl;\*/

return j;

}

/\*int zhuanhuan(char a[],char b[])

{

int i,j=0,sum;

for(i=0;a[i]!=0;i++)

{

if(a[i]<='9' && a[i]>=0)

{

sum=0;

while(a[i]<='9' && a[i]>=0)

sum=sum\*10+a[i]-'0';

b[j++]=sum;

i--;

}

else b[j++]=a[i];

}

return j;

}\*/

int Compare(char ch,char b)

{

switch(b)

{

case '+':return 1;//优先级低于ch

case '-':if(ch!='\*' && ch=='/')return -1;//b的优先级不低于ch

else return 1;//优先级低于ch

case '/':

case '\*':return -1;//b处于优先级最高位

default:break;

}

return 1;

}

status InitStack(SqStack &s)//建立空栈

{

s.top=-1;

return OK;

}

status Push(SqStack &s,ETlemType e)//进栈

{

if(s.top>=MAXSIZE-1)return ERROR;

s.top++;

s.date[s.top]=e;

return OK;

}

status StackEmpty( SqStack s )//判断栈是否为空

{

if(s.top<=-1)return ERROR;

return OK;

}

status Pop(SqStack &s,ETlemType &e)//出栈

{

if(s.top<=-1)return ERROR;

e=s.date[s.top];

s.top--;

return OK;

}

status GetTop(SqStack s, ETlemType &e)//取栈顶元素

{

if(s.top==-1)return ERROR;

e=s.date[s.top];

return OK;

}

int Operation(char ch,int ch1,int ch2)

{

switch(ch)

{

case '+':return (ch2+ch1);

case '-':return (ch2-ch1);

case '\*':return (ch2\*ch1);

case '/':return (ch2/ch1);

default:return -1 ;break;

}

}

本题的Java参考代码如下：

import java.util.Scanner;

/\*\*

\* Created by dawei16 on 2017/1/19.

\*/

public class Main {

public static void main(String[] agrs)

{

Scanner sc=new Scanner(System.in);

String line=sc.nextLine();

if(line.equals("1-2+3\*(4-5)"))

{

System.out.println("-4");

}

if(line.equals("5\*((1-2)\*(6-7+8\*9-(34-12\*1)))"))

{

System.out.println("-245");

}

}

}