**算法训练 删除多余括号**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　从键盘输入一个含有括号的四则运算表达式，要求去掉可能含有的多余的括号，结果要保持原表达式中变量和运算符的相对位置不变，且与原表达式等价,不要求化简。另外不考虑'+'　　'-'用作正负号的情况，即输入表达式不会出现(+a)或(-a)的情形。

输入格式

　　表达式字符串，长度不超过255,　　并且不含空格字符。表达式中的所有变量都是单个小写的英文字母, 运算符只有加+减-乘\*除/等运算符号。

输出格式

　　去掉多余括号后的表达式

样例输入

样例一：

a+(b+c)-d

样例二：

a+b/(c+d)

样例三：

(a\*b)+c/d

样例四：

((a+b)\*f)-(i/j)

样例输出

样例一：

a+b+c-d

样例二：

a+b/(c+d)

样例三：

a\*b+c/d

样例四：

(a+b)\*f-i/j

本题的C++参考代码如下：

#include "iostream"

#include "string"

#include "stdio.h"

#include "ctype.h"

#include "algorithm"

#include "stack"

using namespace std;

int cacluExprePriority(string str,bool &hasC)

{

int left=0;

int right=0;

bool bfind=false;

for(int i=0;i<str.size();i++)

{

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

left++;

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

right++;

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

hasC=true;

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

{

if(left==right)

return 2;

}

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

{

bfind=true;

}

}

return bfind?1:-1;

}

bool vis[1000];

int safe(int i,int n)

{

if(i<0)

return 0;

if(i>=n)

return n-1;

return i;

}

void findIndexOfBrackets(string str)

{

stack<int>q;

for(int i=0;i<str.size();i++)

{

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

{

q.push(i);

}

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

{

int s=q.top();

int t=i;

bool hasC=false;

int priority=cacluExprePriority(str.substr(s+1,t-s-1),hasC);

q.pop();

bool temp=false;

if(s-1>=0&&cacluExprePriority(str.substr(safe(s-1,str.size()),1),temp)>=cacluExprePriority(str.substr(safe(t+1,str.size()),1),temp))

{

char op=str[s-1];

{

if(op=='+')

{

vis[s]=vis[t]=true;

}

if(op=='-')

{

if(priority==2)

vis[s]=vis[t]=true;

}

if(op=='\*')

{

if(priority==2&&hasC==false)

vis[s]=vis[t]=true;

}

}

}

else if(t+1<str.size())

{

char op=str[t+1];

if(op=='+'||op=='-')

{

vis[s]=vis[t]=true;

}

if(op=='\*'||op=='/')

{

if(priority==2)

vis[s]=vis[t]=true;

}

}

}

}

}

int main()

{

string exper;

cin>>exper;

findIndexOfBrackets(exper);

for(int i=0;i<exper.size();i++)

if(vis[i]==false)

cout<<exper[i];

cout<<endl;

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

int q(char \*ch)

{

int i=0,z=0;

ch[i] = '#';

while (ch[i] != ')'||z!=0)

{

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

{

z++;

}

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

{

z--;

}

i++;

}

ch[i] = '#';

return i;

}

int f(char \*ch, char a)

{

int i = 0,jj=0;

if (a == '+')

{

while (ch[i] != ')')

{

if (ch[i+1] == '(')

if (f(&ch[i], ch[i - 1]) == 0)

i += q(&ch[i]);

else

{

while (ch[i] != ')')

{

i++;

}

}

i++;

}

if (ch[i + 1] == '\*' || ch[i + 1] == '/')

{

return 1;

}

else

{

return 0;

}

}

if (a == '-')

{

while (ch[i] != ')')

{

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

if (f(&ch[i+1], ch[i - 1]) == 0)

i += q(&ch[i]);

else

{

while (ch[i] != ')')

{

i++;

}

}

if (ch[i] == '+' || ch[i] == '-')

return 1;

i++;

}

if (ch[i + 1] == '\*' || ch[i + 1] == '/')

{

return 1;

}

else

{

return 0;

}

}

if (a == '\*')

{

while (ch[i] != ')')

{

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

if (f(&ch[i+1], ch[i - 1]) == 0)

q(&ch[i]);

else

{

while (ch[i] != ')')

{

i++;

}

}

if (ch[i] == '+' || ch[i] == '-')

return 1;

i++;

}

return 0;

}

if (a == '/')

{

while (ch[i] != ')')

{

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

if (f(&ch[i+1], ch[i - 1]) == 0)

q(&ch[i]);

else

{

while (ch[i] != ')')

{

i++;

}

}

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

return 1;

i++;

}

}

while (ch[i] != ')')

{

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

if (f(&ch[i+1], ch[i - 1]) == 0)

i += q(&ch[i]);

else

{

while (ch[i] != ')')

{

i++;

}

}

if (ch[i] == '+' || ch[i] == '-')

jj=1;

i++;

}

if ((ch[i + 1] == '\*' || ch[i + 1] == '/')&&jj==1)

{

return 1;

}

else

{

return 0;

}

}

void g(char \*a)

{

int i = 0;

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

{

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

if (f(&a[i + 1], a[i - 1]) == 0)

q(&a[i]);

else

{

while (a[i] != ')')

{

i++;

}

}

i++;

}

}

int main()

{

int l = 0, i = 0;

char ch[100];

scanf("%s", ch);

g(ch);

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

{

if (ch[i]!='#')

{

printf("%c", ch[i]);

}

i++;

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main{

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

char[] chs = br.readLine().toCharArray();

List<Character> list = new ArrayList<Character>();

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

list.add(chs[i]);

}

String s = "";

for (int i = 0; i < func(list).size(); i++) {

s += list.get(i);

}

System.out.println(s);

}

public static List<Character> func(List<Character> list) {

for (int i = 0; i < list.size(); i++) {

if (list.get(i) == '+' || list.get(i) == '-') {

if (list.get(i - 1) == ')' && list.get(i + 1) == '(') {

list.remove(i - 1);

list.remove(i);

for (int j = i - 1; j > -1; j--) {

if (list.get(j) == '(') {

list.remove(j);

break;

}

}

for (int k = i + 1; k < list.size(); k++) {

if (list.get(k) == ')') {

list.remove(k);

break;

}

}

}

if (list.get(i - 1) == ')') {

list.remove(i - 1);

for (int j = i - 1; j > -1; j--) {

if (list.get(j) == '(') {

list.remove(j);

break;

}

}

}

if (list.get(i + 1) == '(') {

if (list.get(i) == '+')

for (int k = i + 1; k < list.size(); k++) {

if (list.get(k) == ')' && !list.contains('/')

&& !list.contains('\*')) {

list.remove(k);

list.remove(i + 1);

break;

}

}

}

}

}

return list;

}

}