**算法训练 C\*++ Calculations**

时间限制：2.0s   内存限制：64.0MB

问题描述

　　C\*++语言和C++语言非常相似，然而C\*++的程序有时会出现意想不到的结果。比如像这样的算术表达式：  
　　表达式=基本式 / 表达式+基本式 / 表达式-基本式  
　　基本式=增量 / 系数\*增量  
　　增量=a++ / ++a  
　　系数=0/1/2/……/1000  
　　如“5\*a++-3\*++a+a++”是合法的C\*++表达式。  
　　计算这样的表达式的值的方法：首先是每个基本式进行计算，然后按照正常的算术运算法则计算。如果一个基本式包含“a++”，则先进行乘法运算再使变量a权值+1；如果一个基本式包含“++a”，则先使变量a权值+1再进行乘法运算。  
　　然而基本式可以按任意顺序计算，这就是为什么计算结果是完全无法预料的。  
　　你的任务就是去找到最大的可能结果。  
  
　　第一行，一个整数n，表示变量a的初始值。  
　　第二行，一个合法的C\*++表达式。  
  
　　共一行，一个整数ans，表示最大可能结果。

输入格式

　　input 1:  
　　1  
　　5\*a++-3\*++a+a++  
　　input 2:  
　　3  
　　a+++++a

输出格式

　　output 1:  
　　11  
　　output 2:  
　　8

数据规模和约定

　　对于20%的数据，表达式长度<=20。  
　　另有20%的数据，满足n>=0。  
　　对于100%的数据，-1000<=n<=1000，表达式长度<=10000。  
　　注意表达式开头可能有负号！

本题的C参考代码如下：

#include <stdio.h>

#include <string.h>

#define MAXSIZE 10011

int n, ans, k, coe, len, outcome, c[MAXSIZE];

char e[MAXSIZE], s[MAXSIZE];

void sort(int x)

{

int i, j, t;

for(i = 1; i < x; i ++){

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

if(c[j] < c[j-1]){

t = c[j-1];

c[j-1] = c[j];

c[j] = t;

}else{

break;

}

}

}

}

int calculate()

{

int i, j;

k = outcome = 0;

strcpy(s, e);

len = strlen(s);

if(s[0] != '-'){

for(i = len + 1; i > 0; i --){

s[i] = s[i-1];

}

s[0] = '+';

len ++;

}

for(i = 0; i < len; i += 3){

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

coe = 1;

}else{

coe = -1;

}

i ++;

j = 0;

while('0' <= s[i] && s[i] <= '9'){

j \*= 10;

j += s[i++] - '0';

}

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

i ++;

}else{

j = 1;

}

coe \*= j;

c[++k] = coe;

outcome += (n-(s[i] == 'a')) \* coe;

}

sort(k);

for(i = 1; i <= k; i ++){

outcome += i \* c[i];

}

return outcome;

}

int main()

{

scanf("%d %s", &n, e);

ans = calculate();

printf("%d\n", ans);

return 0;

}

本题的C++参考代码如下：

#include <cstdio>

#include <iostream>

#include <algorithm>

#include <cstring>

using namespace std;

bool flag;

int a,n,i,j,k,x,o,ans;

int w[10011],tot;

char s[100011];

int main()

{

scanf("%d\n%s",&a,s+1);

n=strlen(s+1);

if (s[1]!='-')

{

for (i=n+1;i>1;i--) s[i]=s[i-1];

s[1]='+';

n++;

}

for (i=1;i<=n;i+=3)

{

if (s[i++]=='+') k=1;

else k=-1;

j=0;

while (i<=n&&'0'<=s[i]&&s[i]<='9') (j\*=10)+=s[i++]-48;

if (s[i]=='\*') i++;

else j=1;

w[++tot]=(k\*=j);

ans+=(a-(s[i]=='a'))\*k;

}

sort(w+1,w+tot+1);

for (i=1;i<=tot;i++) ans+=i\*w[i];

cout<<ans;

return 0;

}