题目：

输入一个中缀算术表达式，计算其结果。对输入的表达式，做如下假设：

(1)只考虑+、-、\*、/这四种运算符；

(2)输入的中缀表达式中数字只有整数，没有小数；

(3)假定输入表达式是合法的。

代码：

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

int main()

{

int a[10];//存储优先级

int STACK1[10];//STACK1存储操作数

char STACK2[10];//STACK2存储运算符

a[0] = 0;

STACK1[0] = 0;

STACK2[0] = '\0';

int x = 0;

int t = 1;

FILE\* p, \* q;

char p\_1, q\_1;

p = fopen("data.txt", "r");

q = fopen("data.txt", "r");

p\_1 = fgetc(p);

q\_1 = fgetc(q);

while (q\_1 != EOF)

{

int num = 1;

int sum = 0;//sum记录数字大小

if (t == 1)

p\_1 = fgetc(p);

else

{

p\_1 = fgetc(p);

p\_1 = fgetc(p);

q\_1 = fgetc(q);

}

if (p\_1 >= '0' && p\_1 <= '9' && q\_1 >= '0' && q\_1 <= '9')

{

num++;

while (1)

{

p\_1 = fgetc(p);

if (p\_1 >= '0' && p\_1 <= '9')

num++;

else

break;

}

for (int i = num; i > 0; i--)

{

sum = sum + (q\_1 - '0') \* pow(10, (i - 1));

q\_1 = fgetc(q);

}

}

else

{

sum = q\_1 - '0';

q\_1 = fgetc(q);

}

x++;

if (p\_1 == '+' || p\_1 == '-')

{

a[x] = 1;

if (p\_1 == '+')

STACK2[x] = '+';

else

STACK2[x] = '-';

}

else if (p\_1 == '\*' || p\_1 == '/')

{

a[x] = 2;

if (p\_1 == '\*')

STACK2[x] = '\*';

else

STACK2[x] = '/';

}

STACK1[x] = sum;

t++;

}

int j = 1;

while (1)

{

if (j == x)

break;

if (j + 1 == x)

{

if (STACK2[j] == '+')

STACK1[j + 1] = STACK1[j] + STACK1[j + 1];

else if (STACK2[j] == '-')

STACK1[j + 1] = STACK1[j] - STACK1[j + 1];

else if (STACK2[j] == '\*')

STACK1[j + 1] = STACK1[j] \* STACK1[j + 1];

else if (STACK2[j] == '/')

{

if (STACK1[j + 1] == 0)

printf("error!");

else STACK1[j + 1] = STACK1[j] / STACK1[j + 1];

}

}

else if (a[j] >= a[j + 1])

{

if (STACK2[j] == '+')

STACK1[j + 1] = STACK1[j] + STACK1[j + 1];

else if (STACK2[j] == '-')

STACK1[j + 1] = STACK1[j] - STACK1[j + 1];

else if (STACK2[j] == '\*')

STACK1[j + 1] = STACK1[j] \* STACK1[j + 1];

else if (STACK2[j] == '/')

{

if (STACK1[j + 1] == 0)

printf("error!");

else

STACK1[j + 1] = STACK1[j] / STACK1[j + 1];

}

}

else if (a[j] < a[j + 1])

{

if (STACK2[j + 1] == '\*')

{

STACK1[j + 2] = STACK1[j + 1] \* STACK1[j + 2];

a[j + 1] = a[j];

}

else if (STACK2[j + 1] == '/')

{

if (STACK1[j + 2] == 0)

printf("error!");

else

STACK1[j + 2] = STACK1[j + 1] / STACK1[j + 2];

a[j + 1] = a[j];

}

STACK1[j + 1] = STACK1[j];

STACK2[j + 1] = STACK2[j];

}

j++;

}

int result = STACK1[x];

FILE\* fp;

fp = fopen("new.txt", "w+");

fprintf(fp, "结果为:%d", result);

fclose(fp);

return 0;

}

运行结果：



总结：定义两个栈，stack1存储数字，stack2存储运算符，将字符串str元素一个个扫描，遇到数字型则进栈stack1，遇到运算符型，则要看看栈stack2栈顶元素运算符优先级是否比自己大或等于，如果真比自己大，那么那个运算符出栈，假设出栈是运算符a，那么此时从stack1中出栈两个数字b、c参与运算，把运算结果进栈stack1，此时此字符还不能进栈，如果栈顶优先级还比自己大或等于，那么那个栈顶运算符还要拿出来运算，直到有小于自己的自己才进栈；遇到‘（’直接进stack2，遇到’）’，则就要把这一对括号之间运算符都一个个拿出来运算，当str[i]读到’\0’那么扫描结束，结束后还要注意stack2里应该还有一个运算符，于是还要多加一步运算，最终stack1中剩一个数，那就是最后结果