

```
1 //日期: 2018/ 时间:
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 //用字符串读入, 再反转着存到d[]数组中。
7 struct bign{
8     int d[1000];
9     int len;
10    bign(){ //“构造函数”初始化结构体
11        memset(d,0,sizeof(d));
12        len = 0;
13    }
14 };
15
16 void print(bign a){
17     for(int i=a.len-1;i>=0;i--){
18         printf("%d",a.d[i]);
19     }
20 }
21 //用字符串读入, 再把字符串存放到bign结构体中
22 bign change(char str[]){ //将整数转换为bign
23     bign a;
24     a.len= strlen(str); //bign的长度就是字符串的长度
25     for(int i=0;i<a.len;i++){
26         if(str[a.len-1-i]>='0' && str[a.len-1-i]<='9'){
27             a.d[i] = str[a.len-1-i] - '0';
28         }
29         else{
30             a.d[i] = str[a.len-1-i];
31         }
32     }
33     return a;
34 }
35 //比较两个结构体: 1)len不一样的话, 直接判断, 2)len一样, 从d[len-1]往d[0]看, 一位大 则整体都大
36 int compare(bign a,bign b){ //比较a和b大小, a大、相等、a小分别返回1、0、-1
37     if(a.len > b.len) return 1; //a大
38     else if(a.len < b.len) return -1; //a小
39     else{
40         for(int i=a.len-1;i>=0;i--){
41             if(a.d[i] > b.d[i]) return 1; //只要有一位a大, 则a大
42             else if(a.d[i] < b.d[i]) return -1;
43         }
44         return 0; //两数相等
45     }
46 }
47 //加法 。需要指出的是, 这种写法要确保两个数都是非负整数, 如果一方为负数要转为减法
48 bign add(bign a, bign b){
49     //print(a); print(b);
50     bign res;
51     int carry=0; //进位标志
52     for(int i=0;i<a.len || i<b.len ;i++){
53         int temp = a.d[i] + b.d[i] +carry;
54         carry = temp / 10;
55         res.d[res.len++] = temp % 10;
```

```

56     }
57     if(carry != 0){           //考虑最高位是否有进位
58         res.d[res.len++] = carry;
59     }
60     return res;
61 }
62
63 //减法
64 bign minus(bign a,bign b){    //a,b为非负整数 计算a-b的绝对值, 正负号由自己判断
65     //如果不够减, 则令被减位的高位减一, 被减位加10再进行减法运算。
66     //最后一步要注意减法后高位可能有多余的0, 要除去它们, 但是也要保证结果至少有一位
67     bign res;
68     if(compare(a,b)==-1){      //如果a<b, 则交换位置
69         bign temp = a;
70         a = b;
71         b = temp;
72     }
73
74     for(int i=0;i<a.len || i<b.len;i++){
75         if(a.d[i] < b.d[i]){
76             a.d[i+1]--;
77             a.d[i] += 10;
78         }
79         res.d[res.len++] = a.d[i] - b.d[i];
80     }
81     while(res.len-1 >=1 && res.d[res.len-1] == 0)
82         res.len--;           //while(res.len -1 >=1)是至少保留一位为最低位, 去除
                                //高位的0
83     return res;
84 }
85
86 //取bign得某位与int型整体相乘, 再与进位相加, 所得的结果的个位数作为该位的结果, 高位
    //部分作为进位
87 bign multi(bign a,int b){      //高精度乘法
88     bign res;
89     int carry = 0;
90     for(int i=0;i<a.len;i++){
91         int temp = a.d[i] * b + carry;
92         res.d[res.len++] = temp%10;    //个位作为该位的结果
93         carry = temp/10;              //高位部分作为新的进位
94     }
95     while(carry != 0){           //和加法不一样, 乘法的进位可能不止一位,
        //因此用while
96         res.d[res.len++] = carry % 10;
97         carry /= 10;
98     }
99     return res;
100 }
101
102 bign divide(bign a,int b,int& r){    //高精度除法, r为余数
103     bign res;
104     res.len = a.len;    //被除数的每一位和商的每一位是一一对应的, 因此先令长度相等
105     for(int i=a.len-1; i>=0;i--){    //从高位开始
106         r = r * 10 + a.d[i];          //和上一位遗留的余数进行组合
107         if(r < b)    res.d[i] = 0;
108         else{

```

```
109         res.d[i] = r / b;           //商
110         r %= b;                     //获得新余数
111     }
112 }
113
114 //再去除高位的0,同时至少保留一位
115 while(res.len - 1 >= 1 && res.d[res.len-1] == 0){
116     res.len--;
117 }
118
119 return res;
120 }
121
122 int main(){
123     char str1[1000],str2[1000];
124     scanf("%s%s",str1,str2);
125     bign a = change(str1);
126     bign b = change(str2);
127
128     printf("a+b=");           //加法
129     print(add(a,b));
130     printf("\n");
131     /* if(a.d[a.len-1] != '-' && b.d[b.len-1] != '-'){ //两个非负整数相加
132         print(add(a,b));
133     }else if(a.d[a.len-1] == '-' && b.d[b.len-1] == '-'){ //两个负数相加
134         printf("-");           //先打印符号, 再将正整数部分相加
135         a.d[--a.len]=0; b.d[--b.len]=0;           //需要将减去的一位置零, 否则会出
136         错
137         bign temp = add(a,b);
138         print(temp);
139     }else if(a.d[a.len-1] == '-' && b.d[b.len-1] != '-'){ //输出b-(-a)
140         a.d[--a.len]=0;
141         //b-a
142         if(compare(b,a)==-1) //b<a
143             printf("-"); //输出负号
144         print(minus(a,b));
145     }else if(a.d[a.len-1] != '-' && b.d[b.len-1] == '-'){ //输出b-(-a)
146         b.d[--b.len]=0;
147         //a-b
148         if(compare(a,b)==-1) //a<b
149             printf("-"); //输出负号
150         print(minus(a,b));
151     }
152 */
153     printf("a-b=");
154     print(minus(a,b));
155     printf("\n");
156
157     int m = 10;
158     printf("a*10=");
159     print(multi(a,m));
160     printf("\n");
161
162     printf("a/10=");
163     int r=0;
164     print(divide(a,m,r));
```

---

```
164     printf(".....%d\n",r);
165
166     return 0;
167 }
168
169
```