代码链接

首先要说的是这道题的难点是如何把数字输出加入逗号，毕竟数据范围并没有超过Long。当然这个难点也不是问题，将数字转为字符串,C中就有这样的函数，然后再用 %3==0 这样来控制输出 。

但我最近出于想要建立自己的代码库的想法，而且这道题我的第一反应其实是高精度运算，所以一是想练练手，二是以后想用高精度可以直接复制了，我就写的复杂一点。

解题思路

程序的实现需要四个函数。

函数一： `char* highAdd(char* a,char* b,char *c)` //c=a+b

原谅我蹩脚的英语，这个词我的本意是无负数的高精度加法，将数字以字符串读入（负号另处理，不传入），然后逐位相加，可以理解小学的竖式加法。将结果存在c中，因为不确定位数，所以需要除去0位，返回非零位的指针。比如三位数加一位数答案有可能是四位和三位（999+1和100+1），所以我预设的是四位，最后再除零。

函数二： `char* highDel(char* a,char* b,char *c)` //c=a-b

这个函数用在存在负号的情况，其中a的值必大于b的值，否则会出错，实现过程同 `highAdd` ，竖式计算。

函数三： `int compare(char* a,char* b)`

这个compare就是为了上面的 `highDel` 服务的，判断a和b的值谁大，当然不能处理有负号的情况。负号的处理我放在了主函数中。实现的话就是字符串的字典序比较。

函数四： `void formatPrint(char* a)`

这个函数就是将最后的结果进行规范输出，因为是对字符串进行操作，实现比较简单，就不细讲了。

主函数

主函数中读取两个数，依据负号分为三种情况依次处理。

1.没有负号

直接两数相加，规范输出

2.都有负号

将两数的非负号位进行相加，最后输出前加个 - 就行了。

3.一个负号

分两种情况，正数大于负数的绝对值的话可以直接调用highDel，然后输出。正数小于负数的绝对值的话将负数的绝对值减去正数，然后在输出结果前加个负号就可以了。

过程

思路有了话，过程也就没什么可说的的，就是照着自己的设想实现，只要保证函数的功能能够实现，那么程序的主体就可以工作，要非要说什么调试的经历的话，实现逗号的输出算一个，虽然知道要用 %3 来实现逗号输出的控制，但用什么来求余就需要判断，是加1还是加2还是不用加，我是逐个试过去，然后还要注意头和尾不能输出逗号，需要再添加一个条件来控制。

其次要说的就是加法函数了，本来我是没打算有返回值的，我想将c的指针后移来除零，但发现没用，后来想了想，指针指向的内容在函数内可以改变，但指针本身并不可以改变，所以我就返回一个指针来指向第一个非零位，对了，还有答案为0的话还要将0保留，需要添加条件判断，不能直接跳过。

提交列表

简单测试通过后直接AC了

```
                else{
                        if(s1[0]=='-'){
                                if (compare(&s1[1],s2)>0){
                                        printf("-");
                                        formatPrint(highDel(&s1[1],s2,s3));
                                }
                                else{
                                        formatPrint(highDel(s2,&s1[1],s3));
                                }
                        }
                        else{
                                if (compare(s1,&s2[1])>0){

                                        formatPrint(highDel(s1,&s2[1],s3));
                                }
                                else{
                                        printf("-");
                                        formatPrint(highDel(&s2[1],s1,s3));
                                }
                        }
                }
        }

        return 0;
```

```
137 }
```

提交代码

评测结果

时间	结果	得分	题目	语言	用时(ms)	内存(kB)	用户
1月24日 09:46	答案正确	20	1001	C++ (g++ 4.7.2)	9	384	ucj

测试点

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案正确	2	384	9/9
1	答案正确	4	384	1/1
10	答案正确	4	296	1/1
11	答案正确	9	384	1/1
2	答案正确	4	384	1/1
3	答案正确	4	292	1/1
4	答案正确	7	384	1/1
5	答案正确	4	384	1/1
6	答案正确	2	376	1/1
7	答案正确	6	384	1/1
8	答案正确	7	296	1/1
9	答案正确	4	384	1/1

查看代码

最后

写完后回看自己的代码总觉的实现的不是很好，感觉有些不必要的操作，比如对负号的处理好像有点过于复杂，和我想象中的高精度运算有点差距。但我目前也没什么更好的想法，就先这样吧。