

1001. A+B Format (20)

Calculate $a + b$ and output the sum in standard format -- that is, the digits must be separated into groups of three by commas (unless there are less than four digits).

解题思路

将 $a+b$ 的值先存在 sum 中，再逐个分解 sum 的数字，将组成 sum 的所有数字存到数组中，由于题目给出 a 和 b 的范围是 -1000000 及 1000000 之间，于是数组只用开到7就行，接着是处理输出格式问题，首先判断 sum 共有几位数，采用从高位往低位遍历寻找第一个不为0的数，该数即为最高位，然后从此为开始往低位输出，输出的时候判断一下需不需要输出“，”，判断依据是下一位位数是否为3的整数倍，是则输出。

调试过程

问题1：数组越界

时间	结果	得分	题目	语言	用时(ms)	内存(kB)
1月22日 23:29	答案错误	0	1001	C (gcc 4.7.2)	11	456

测试点

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案错误	4	256	0/9
1	答案错误	4	256	0/1
10	答案错误	9	384	0/1
11	答案错误	11	456	0/1
2	答案错误	2	256	0/1
3	答案错误	2	384	0/1
4	答案错误	5	256	0/1
5	答案错误	4	256	0/1
6	答案错误	3	256	0/1
7	答案错误	6	256	0/1
8	答案错误	7	384	0/1
9	答案错误	11	256	0/1

[查看代码](#)

我的思路是设置一个数组number来保存sum各位上的数字，经分析sum最多只有7位也就是七个数字，所以我不假思索的写上了 `int number[7];`，这样定义也不是不可以只要接下来从0开始存就可以了，但我又不假思索的从 `number[1]` 开始存到了 `number[7]`，于是问题就出现了，导致最后一位全错。（其实我事先在DC上编译运行的时候是没问题的，后来提交后才发现了错误，至今没懂为什么自己运行的时候竟然是过得去的。）这个小问题告诉我们，细心非常之重要。

再次划重点：数组下标从0开始

问题2：负号的输出问题

```
E:\c\calculate a+b.exe
-10000 9
-9, -9-9-1
-----
Process exited after 3.043 seconds with return value 0
请按任意键继续. . .
```

由于第一版代码我把负数的判断放在了最后，所以每输出一个数字就判断一次正负的，最终导致出现了多个负号的情况。于是改进办法就是从一开始就先判断正负，判断完后直接输出负号，再取绝对值进行下面的运算。

问题3：flag忘记初始化

评测结果

时间	结果	得分	题目	语言	用时(ms)	内存(kB)
1月26日 21:57	部分正确	19	1001	C (gcc 4.7.2)	219	2292

测试点

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案正确	7	256	9/9
1	答案正确	5	256	1/1
10	答案正确	4	256	1/1
11	答案正确	6	256	1/1
2	答案正确	5	256	1/1
3	答案正确	5	256	1/1
4	段错误	219	2292	0/1
5	答案正确	6	256	1/1
6	答案正确	2	256	1/1
7	答案正确	2	256	1/1
8	答案正确	5	256	1/1
9	答案正确	5	256	1/1

[查看代码](#)

最后一次提交，出现了一个错误，查了帮助提示是数组越界。于是自己看了半天代码，尝试了各种情况后还是没找出是哪一组数据出现了问题，最后经高人指点发现其实就是漏了全为0的情况.....这还是一个细节问题。初始化了flag之后就全AC了。

时间	结果	得分	题目	语言	用时(ms)	内存(kB)
1月26日 22:12	答案正确	20	1001	C (gcc 4.7.2)	6	452

测试点

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案正确	4	384	9/9
1	答案正确	2	452	1/1
10	答案正确	5	384	1/1
11	答案正确	4	256	1/1
2	答案正确	2	384	1/1
3	答案正确	6	256	1/1
4	答案正确	4	384	1/1
5	答案正确	4	256	1/1
6	答案正确	4	256	1/1
7	答案正确	4	128	1/1
8	答案正确	4	256	1/1
9	答案正确	5	384	1/1

[查看代码](#)

提交记录

PAT (Advanced Level) Practise - 提交

[显示所有用户提交](#)

时间	结果	得分	题目	语言	用时(ms)	内存(kB)	用户
1月26日 22:12	答案正确	20	1001	C (gcc 4.7.2)	6	368	Syxxx
1月26日 21:57	部分正确	19	1001	C (gcc 4.7.2)	219	2292	Syxxx
1月22日 23:29	答案错误	0	1001	C (gcc 4.7.2)	11	456	Syxxx

后记

虽然作业只是很简单的一道题目，但还是可以从中收获许多经验。这道题带给我的，就是细心的重要性。细节无疑在编程中是不可忽视的部分，也是最容易出错的地方。往往有时过不去就是一个地方的小细节出了错，有时是格式，有时甚至是单词错了，这些问题还好，编译的时候还会提示你，但如果是编译器找不出来的问题那就有点麻烦了。所以说，从现在开始就要养成注意细节的习惯了。

附录（代码）

```
#include <stdio.h>
#include <math.h>
int main()
{
    int a,b,sum,number[8]={0},i,flag=0;
    scanf("%d %d",&a,&b);
    sum=a+b;
    /*处理负号*/
    if (sum<0)
    {
        printf("-");
        sum=abs(sum);
    }
    /*将sum中每个数依次存入数组中*/
    for (i=1;i<=7;i++)
    {
        if (sum!=0)
        {
            number[i]=sum%10;
            sum=sum/10;
        }
        else break;
    }
    /*寻找最高位*/
    for (i=7;i>0;i--)
    {
        if (number[i]!=0)
        {
            flag=i;
            break;
        }
    }
    /*依次输出数字*/
    for (i=flag;i>0;i--)
    {
        printf("%d",number[i]);
        if ((i>3)&&((i-1)%3==0))
        {
            printf(",");
        }
    }
    if (flag==0) printf("0"); //处理为0的情况
    return 0;
}
```