张引老师班2020-2021秋冬C语言期中测试1详解

目录

[判断题 1](#_Toc61359587)

[单选题 3](#_Toc61359588)

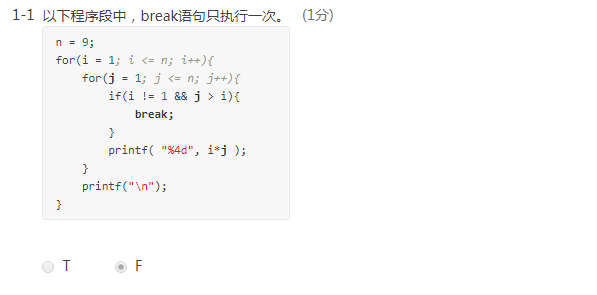
[多选题 8](#_Toc61359589)

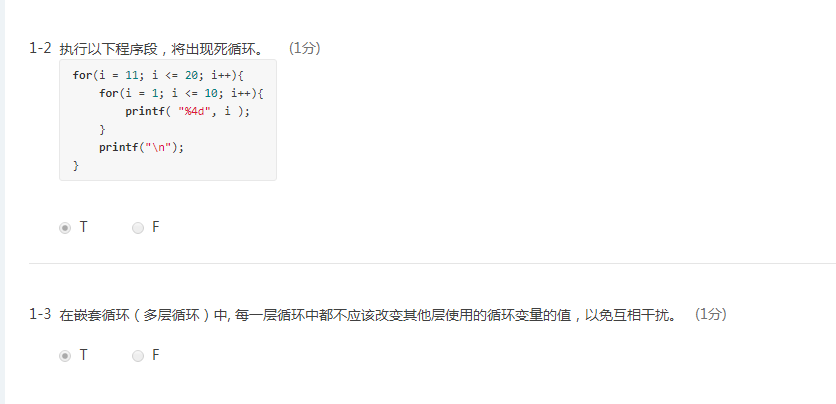
[填空题 11](#_Toc61359590)

[程序填空题 14](#_Toc61359591)

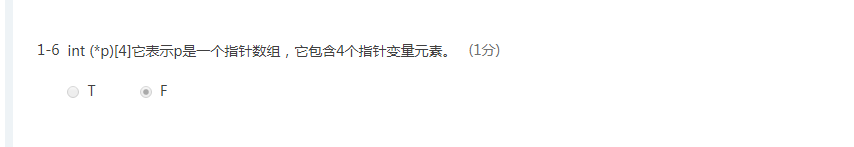
[编程题 17](#_Toc61359592)

## 判断题

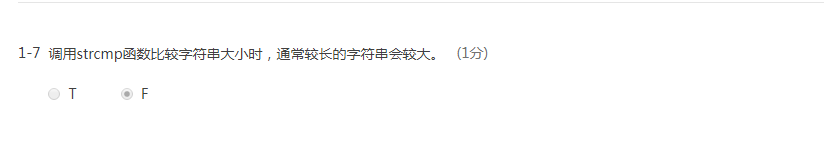








(int\*) p[4] 才是。本题略超纲。

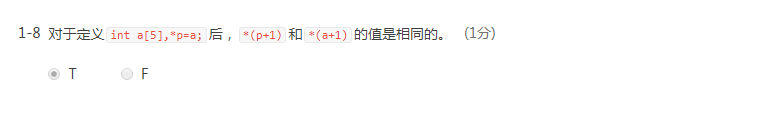


strcmp函数，首先左对齐两个字符串，之后比较每个字符的ASCII码大小。

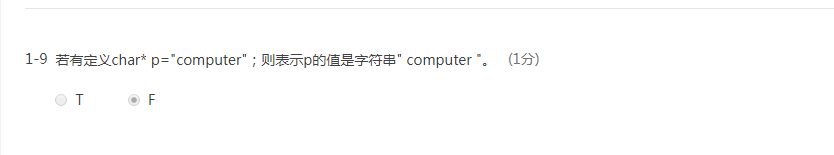
如： aaa > aa

aaaa > aaba

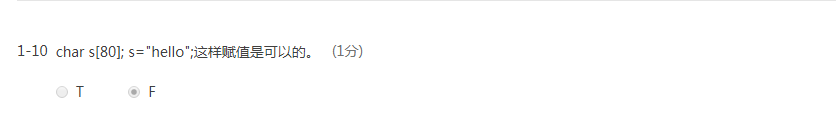
aaaaaaa < b



数组名a也是一个char\*指针，只不过是常量，不能被修改。



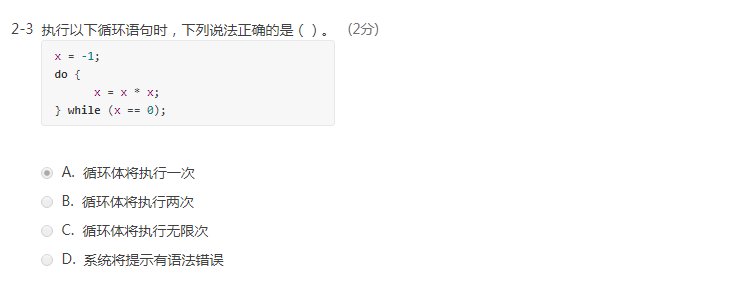
p的值是computer字符串首字母的地址（十六进制整数）。

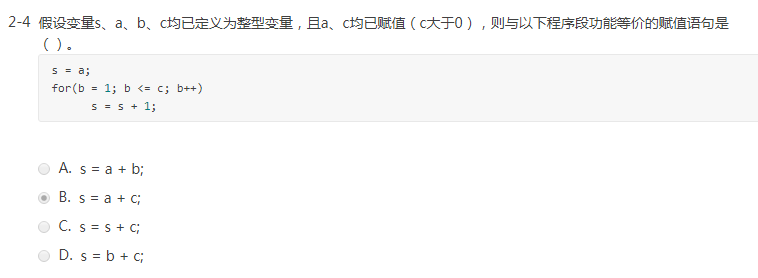


只有初始化的时候能使用 char s[80] = “hello”。后面不能赋值整个数组。

## 单选题





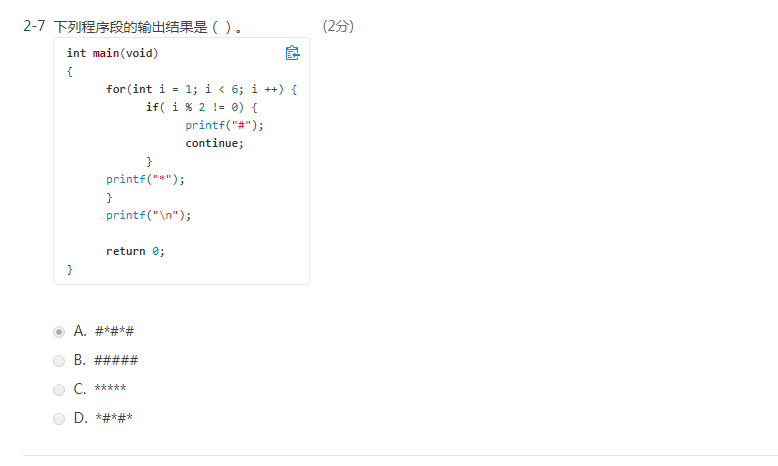


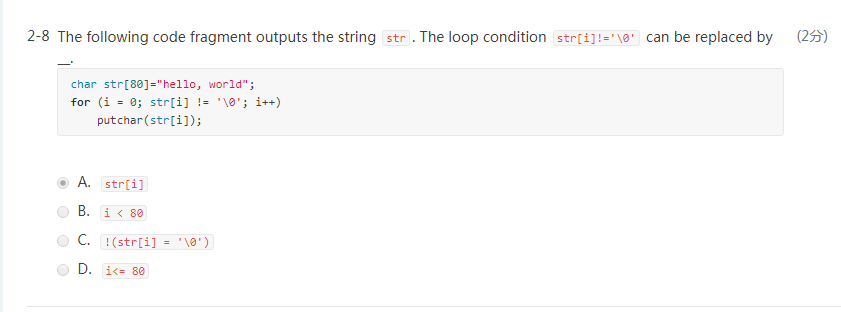




这个方法叫做“更相减损术”，和辗转相除法一样。

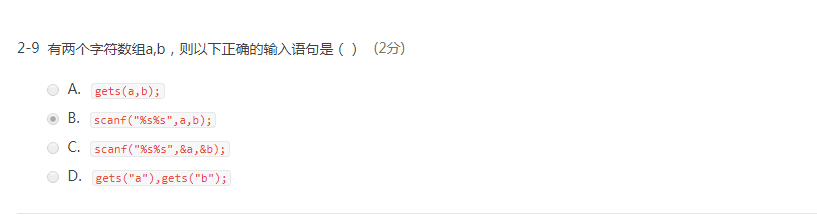
（不知道什么是辗转相除法？？回去看ppt吧，期末有可能考）



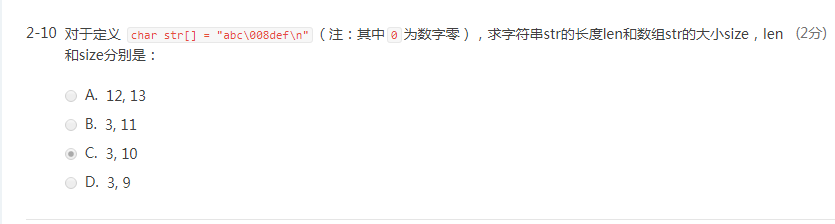


注意：字符’\0’ 的ASCII码就是数字0。

C选项是赋值，不是判断等于。



scanf(“%s”) 后面不加地址符号&。



len 指的是strlen()函数的返回值，就是找到第一个\0为止。

对于size来说，转义字符是一个字符。

即：

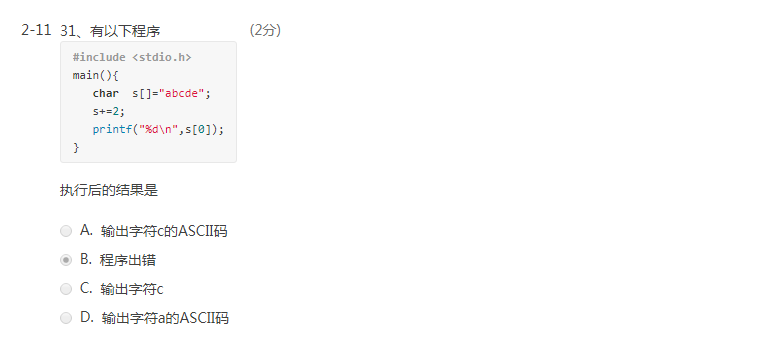
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| a | b | c | \00 | 8 | d | e | f | \n | \0 |  |

注意，这个数组的大小在读入时就固定了。最后还多补了一个0.

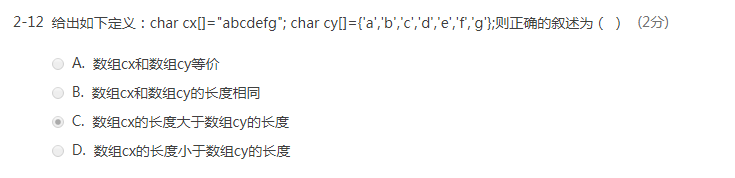
此处的\00 是八进制字符转义。 \0开头的字符为八进制，如\017，等于1\*8+7 = 15;

\0x开头的是十六进制字符转义，如\0x12，等于1\*16 + 2 = 18

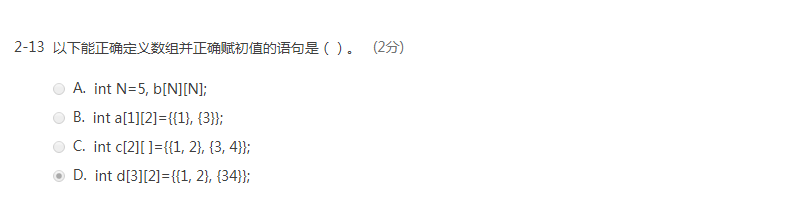
此处考的偏难。一般考察点只要知道\0等转义字符占一个字符就可以。



**数组名**是**常量**指针，不能进行赋值或修改操作。



cx这种赋值方式，由于字符串“”的最后默认有\0，因此较长。

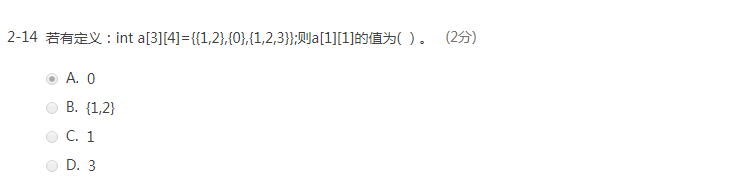


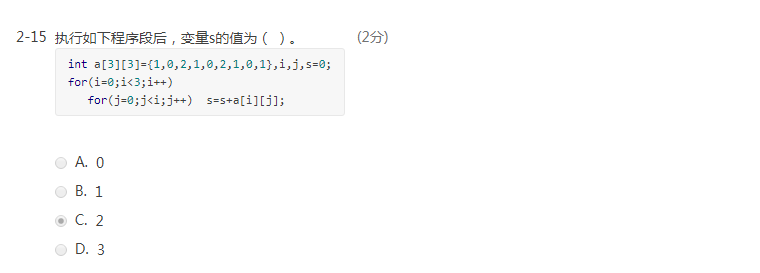
A 是我一直强调的错误方式，很容易出问题。

B 的正确赋值应该是 int a[1][2] = {{1,3}}.

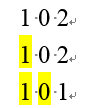
C 数组初始化时，只能省略第一个维度，如int c[][2] = {{1,2},{3,4}}.

D 是正确的。

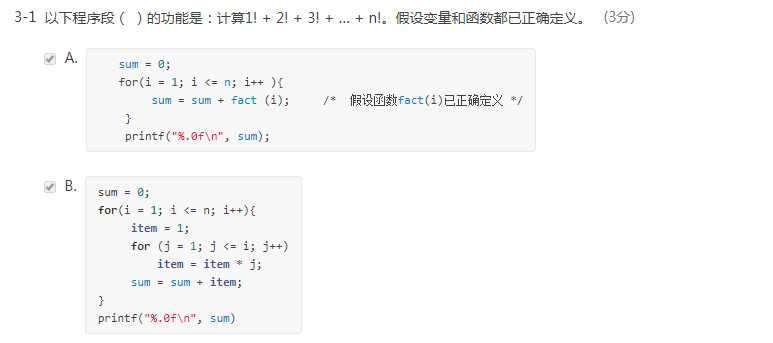


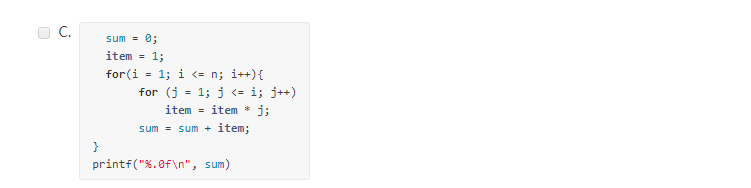


示意图见下图。



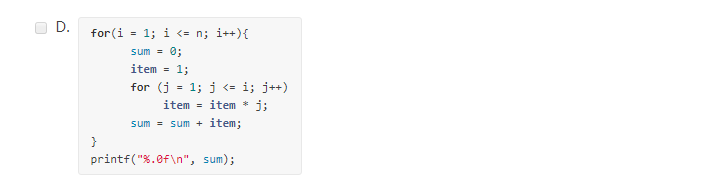
## 多选题





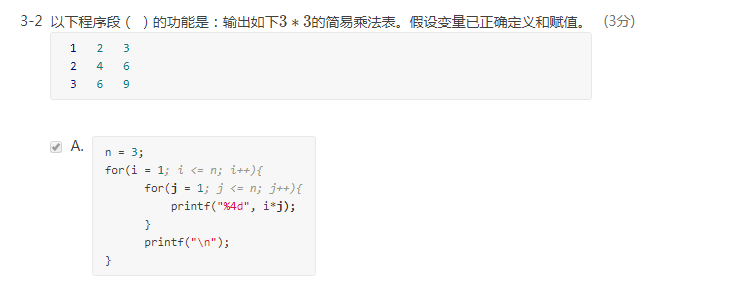
C错，错在item没有每次都初始化。

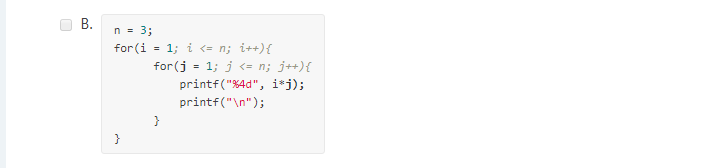
问问自己，这个应该怎么改？？写在旁边。 如果不会，那你期末危险了，赶快问！

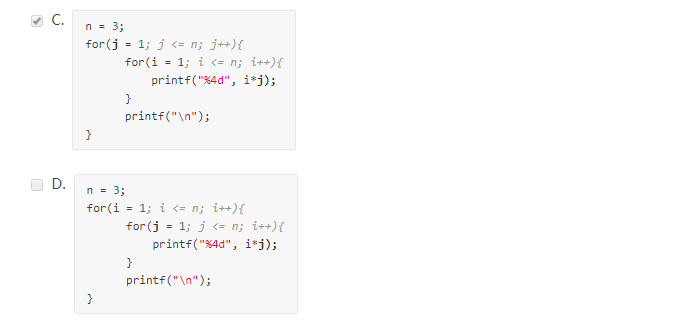


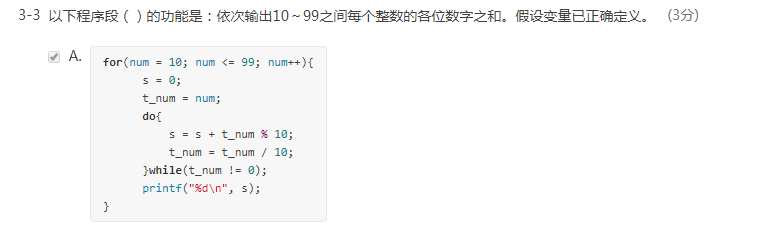
D错。

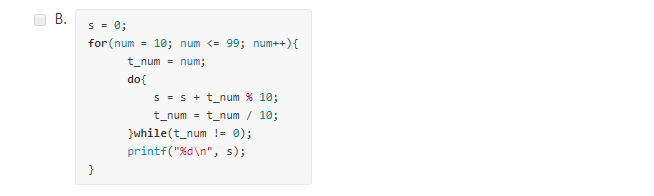


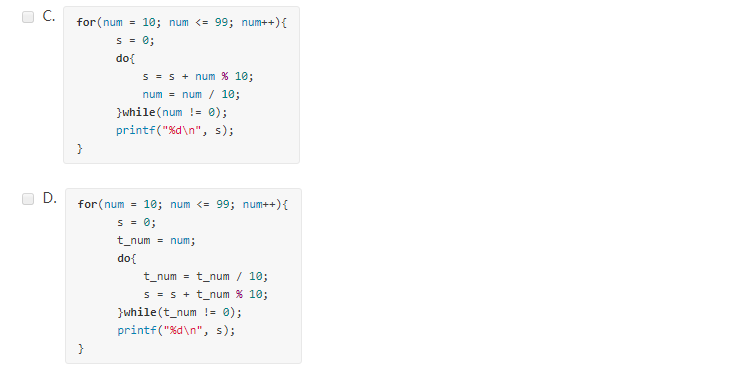




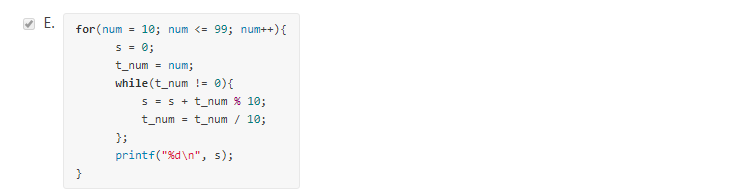




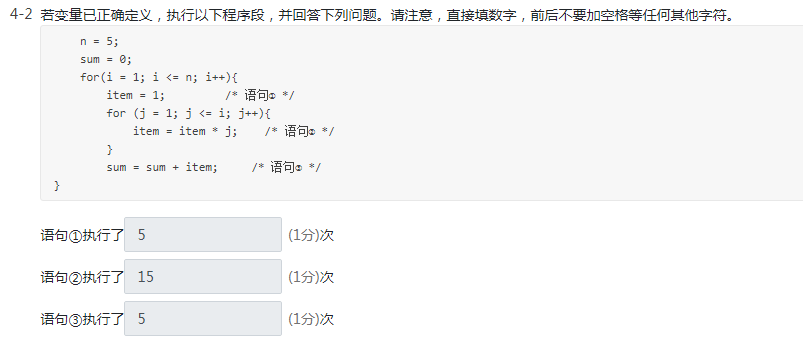


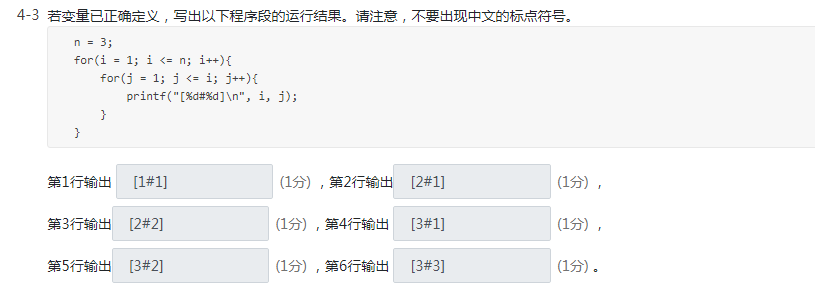


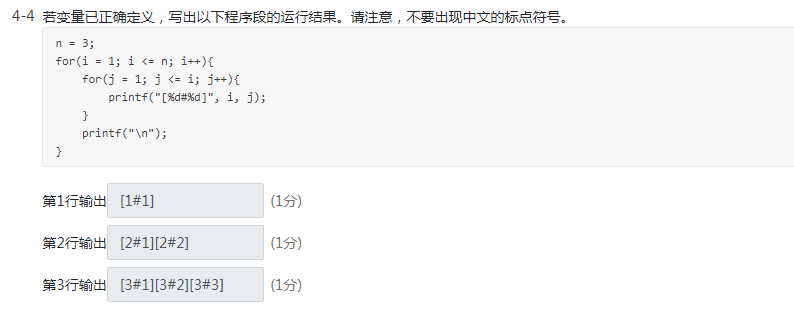
好好观察一下D和A的区别。如果不看选项，你们能否自己写出，依次输出10-99间每个数字各位之和的程序？如果不能，那期末危险了！赶紧看！

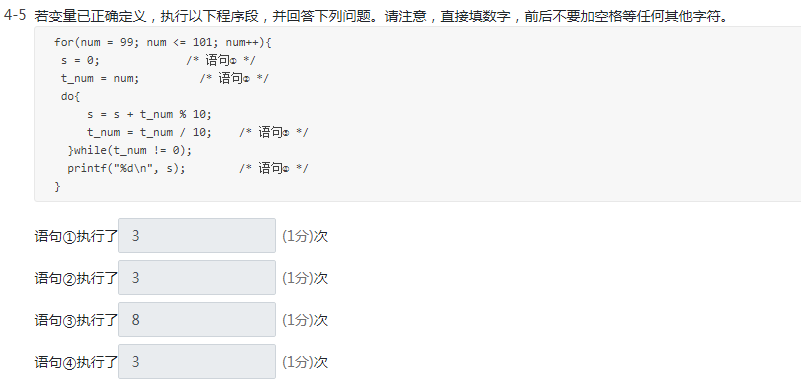


## 填空题

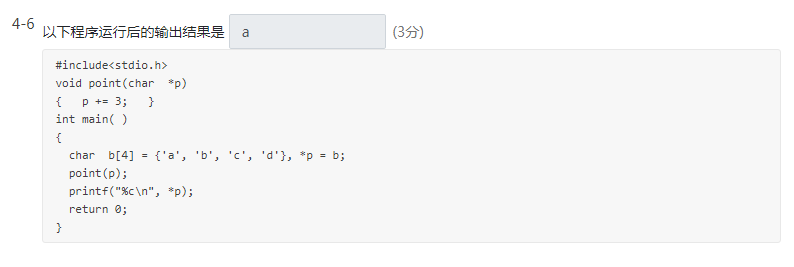








99、100、101 加起来一共是2+3+3=8位数字。



point()函数修改的是形参，形参是point()函数的局部变量，与main函数中的p无关。

不知道形参、局部变量是什么？赶紧回去看函数这一章的ppt





这套卷子里面唯一稍微复杂点的填空题。实际上，期末卷比这套卷子的难度高至少两个层次。

静态变量(如G()函数里面的变量p)，程序刚开始运行时就分配好了存储空间，仅进行一次初始化（也就是static int p=1里面的p=1只执行一次）。同时，每次调用函数G()时，使用的p都是同一个变量，修改的结果会被保存下来。

与之相对的，函数F()里面的变量p是局部（动态）变量，每次调用F()都会重新生成变量p并将其赋值为1；函数中对其修改的结果不会被保存。

因此：

在运行这四句之前，G()函数中的p已经被赋值为1

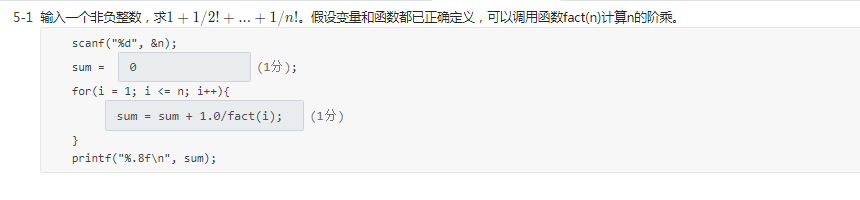
a = F(2) 此时F()函数中运行 p =1 和 p \*=2 两句，最后返回2

b = F(3) 此时F()函数中运行 p =1 和 p \*=3 两句，最后返回3

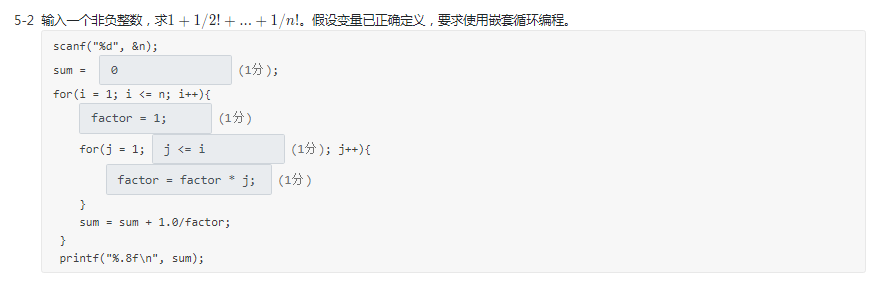
c = G(2) 此时G()函数中运行 p \*=2 一句，最后返回2，且p的值变为2

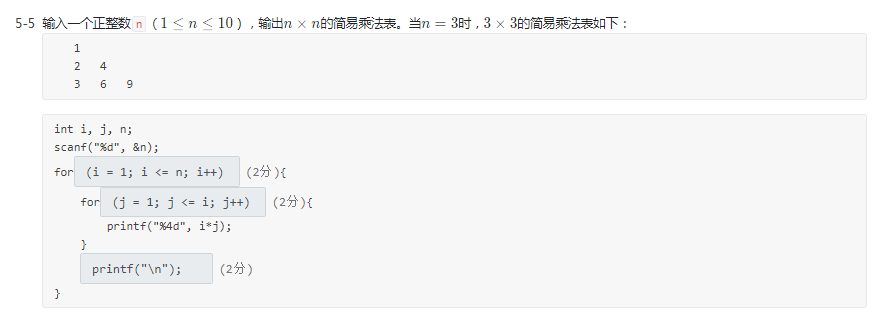
d = G(3) 此时G()函数中运行 p \*=3 一句，最后返回6，且p的值变为6

## 程序填空题

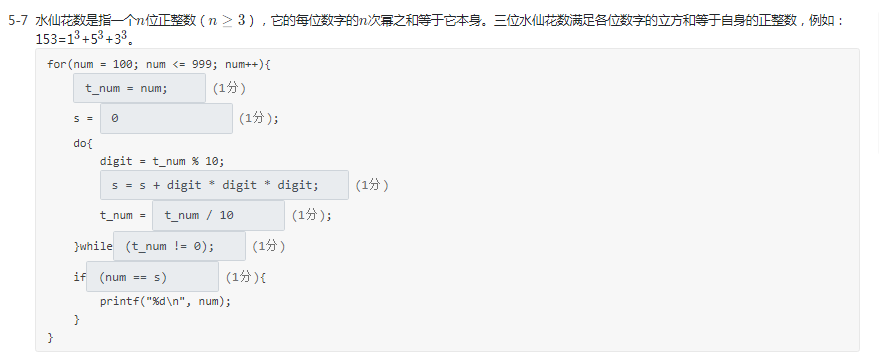


注意1.0将计算转为double防止整除

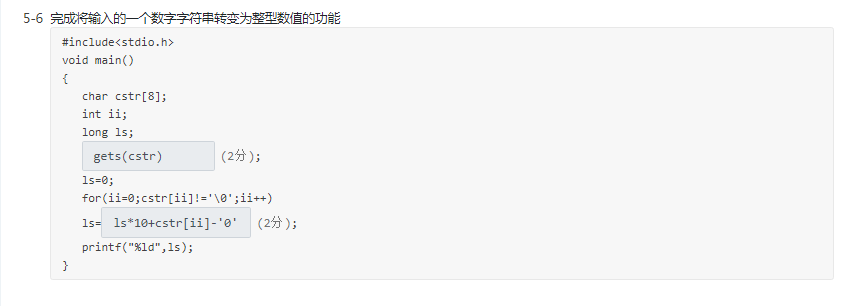




思考一下：这个题给写成编程题给你会不会做？？如果不会，那期末不是危险，基本完蛋了！赶紧过来找我！



这个题主要考察的还是大家对取整数每一位数字的熟练程度。必须会！不会也得背下来！



1. gets()函数必须会用。

2. 这个转换必须会做，期末必考。

3. 单个字符转数字：x = c – ‘0’ 单个大写转小写： c2 = c1 - ‘A’ + ‘a’

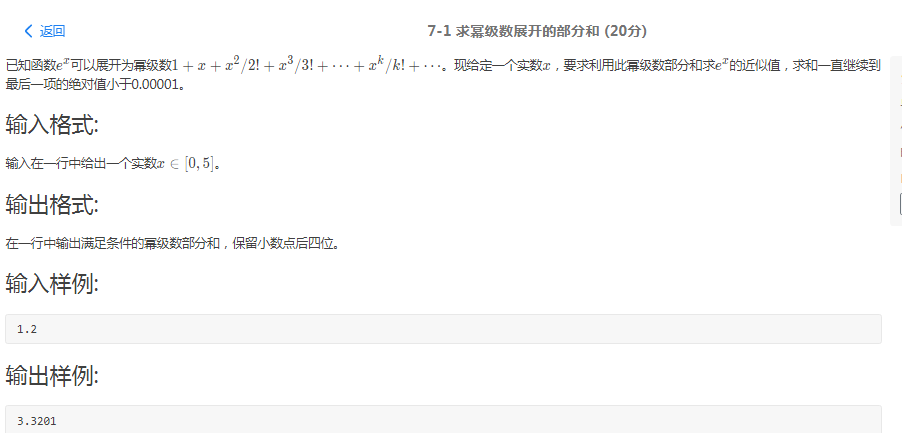
另外，ord(‘\0’) = 0, ord(‘0’) = 48, ord(‘A’) = 65, ord(‘a’) = 97 , chr(48) = ‘0’, chr(65) = ‘A’

这两个函数是在字符和ASCII码之间转换的。

回过头想想字符串比较大小， “0” 和 “A” 和 “a” 哪个大？”A0”和”0AA” 呢？不会的返回去看这份卷子的填空题！



## 编程题





这个题非常重要！

很多同学估计到现在也不知道自己为啥答案错误或者超时。

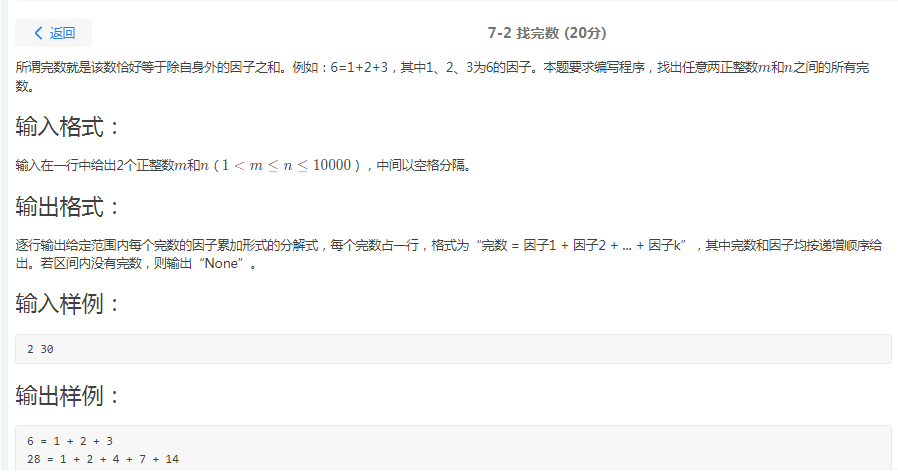


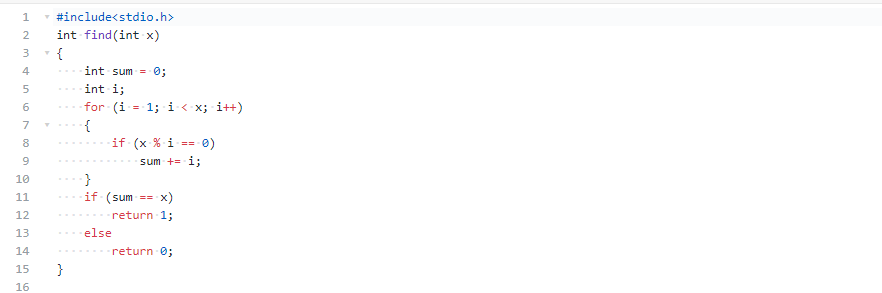
这句话就是这个算法的精髓。每次我只乘一项、除一项。有这么几个好处：

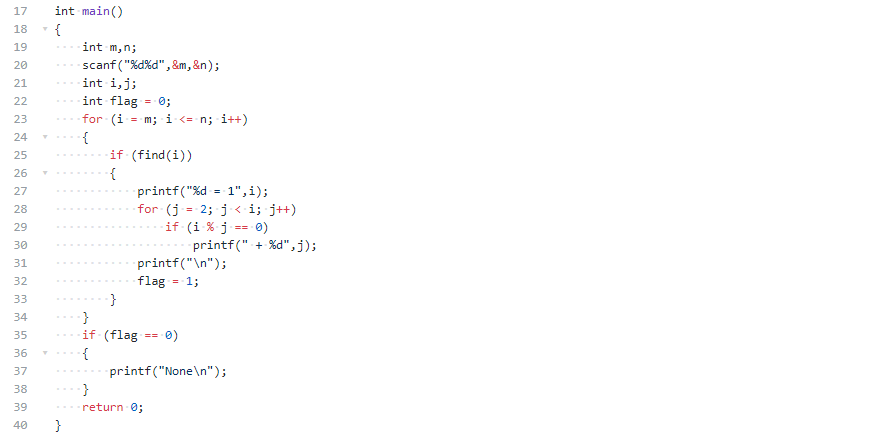
1. 不会由于每次计算都要调用fact()等函数导致超时。

2. 不会由于阶乘增长过快导致精度（答案错误）问题。（阶乘增长实在太快了，别说int会溢出，double都顶不住！）

3. 我猜你们写不出比我这个更简单精炼的代码。这种方法处理分子、分母都很大的情况是非常有用的。





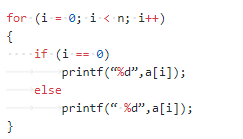


这种题很常见，“完数、水仙花数、素数”….等等。做这种题，一定要写一个函数来判断一个数是不是题目要求的这种特殊的数，这样的话整个程序逻辑会非常清晰。

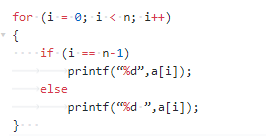
另外，完数是“除了自己之外的因子”，因此1不是完数，注意。见到新的题目，更要好好考虑一下边界情况，0,1,2到底是否是满足他要求的数，是否需要特殊处理。

此外，如果你现在还不会“每行输出n个数，中间空格分开，行末没有多余空格”的方法的话，我现在教你。一共有两种方案：

1. 第一个元素直接输出，后面元素的都在前面加空格

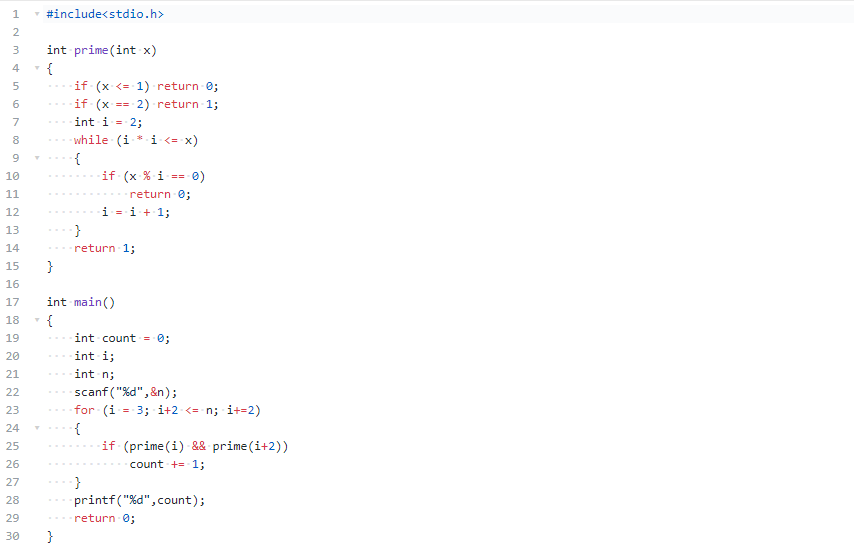


2. 前面元素都在后面加空格，最后一个元素直接输出



这两种必须会，非常容易考程序填空题。

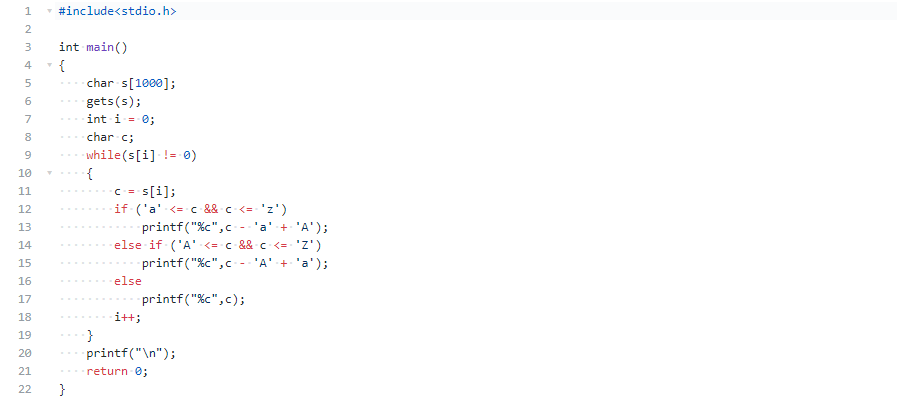




如果有同学超时，说明你根本不会使用根号N的素数判断方法，一定会超时。

我上面的prime()函数建议大家全文背诵（说的就是你们！所有这个题错了的同学或者以没时间为借口没做这个题的同学！）





可谓是字符串处理中基础的基础了，期末填空必考。建议对照程序填空5-6去背。

另外，如果你要另开一个char[] 数组并使用%s输出，那么必须在最后先补\0才能使用%s输出。