

做好闭环（一）：不看答案可能就白学了

2020-01-18 胡光

人人都能学会的编程入门课

[进入课程 >](#)



讲述：胡光

时长 15:56 大小 12.78M



你好，我是胡光。

不知不觉，语言基础篇已学习过半，我非常高兴，看到很多同学都在坚持学习。并且，还有一些同学，每每都能在专栏上线的第一时间里，给我留言，提出疑惑。当面对一些知识点的时候，如果在我的观念中它是不说自明，而对于新手的你来说，可能十分难理解的时候，我也很希望你能指出来，我会在留言区中给你解答的。因为，我知道这种讨论，肯定能够帮助到更多的人。




大部分留言，我都在相对应的文章中回复过了，而对于文章中的思考题呢，由于要给你留足思考时间，所以我选择，一起留在今天这样一篇文章中，给你进行一一的解答。

看一看我的参考答案，和你的思考结果之间，有什么不同吧。也欢迎你在留言区中，给出一些你感兴趣的题目的思考结果，我希望我们能在在这个过程中，碰撞出更多智慧的火花。在这里呢，@rocedu 用户在第一篇留言区中给大家推荐的《程序设计实践》一书，也是非常优秀的书籍。有兴趣的小伙伴，也可以去到他提到的豆瓣读书主页中去游览一番。

第一个程序：教你输出彩色的文字


在这一篇里面呢，我们接触了如何在 Linux 环境下输出彩色文字的编程知识。初步学习了 scanf 和 printf 函数的基础用法，两者一个负责读入，一个负责输出。如果你对这篇文章的内容有点陌生，可以再回去看看 [🔗 《第一个程序：教你输出彩色的文字》](#)。最后围绕着这两个函数，给你出了两个思考题。这两个思考题做的怎么样？下面来看看我的参考答案吧。

思考题（1）：位数输出

 复制代码

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6     printf(" has %d digits\n", printf("%d", n)); // 有多余输出
7     char output[50];
8     int ret = sprintf(output, "%d", n);
9     printf("%d\n", ret); // 无多余输出
10    return 0;
11 }
```

运行如上程序，如果输入 123，程序会输出如下两行内容：

 复制代码

```
1 123 has 3 digits
2 3
```

你会看到，第 1 行除了数字的位数信息以外，还有多余的输出，第 2 行则是没有多余的输出。而两个信息，都是单纯利用 printf 一族函数完成的。这个问题的解题关键是，理解 printf 函数是有返回值的，而其返回的含义是打印了多少个字符。

那么，当我们使用 printf 打印数字 n 的时候，printf 函数的返回值，就是代表了 n 的位数。类似的，sprintf 也是 printf 一族函数中的一员，它的返回值与 printf 含义相同。

思考题（2）：读入一行字符串

 复制代码

```
1 #include <stdio.h>
2 char str[100];
3 int main() {
4     scanf("%[^\n]s", str);
5     printf("%s\n", str);
6     return 0;
7 }
```


这段代码展现了如何使用 scanf 读入一行包含空格的字符串信息。其中，要读入字符串，就需要使用 %s 格式占位符。可是这道题目中，在 % 和 s 中间有一对中括号[]，这个[] 代表了一个集合，用来控制 %s 在读入过程中可以读入的字符集合的，例如：%[a-z]s，是可以输入小写字母 a 到 z，那么一旦遇到了非小写字母，就会停止。

而上述代码中的 ^ 上尖号，读作非，“^\\n” 就是非换行符，也就是说，只要不是换行符，就可以继续读入。这也就达到了我们想要用 scanf 读入一行的功能要求。你可以自己试一下换成 %[a-z]s，然后输入 “abcd12efeee”，看看程序的输出，你就能明白了。

判断与循环：给你的程序加上处理逻辑

在这篇文章 [《判断与循环：给你的程序加上处理逻辑》](#) 中呢，我们学习了除了顺序结构以外的两种程序执行结构：分支结构和循环结构。知识点的话，主要涉及：**条件表达式、if 语句、for 语句等知识内容**。我们说到，任何表达式都有返回值，条件表达式的值，就是 1 或者 0 代表“真”或者“假”，“成立”或者“不成立”。并且，介绍了条件判断的时候，实际上遵循的原则是“非零即为真”。最后呢，给你留了一个和循环相关的思考题“打印乘法表”，下面就看看我的参考答案吧。

思考题：打印乘法表

 复制代码

```
1 #include <stdio.h>
2 int main() {
3     for (int i = 1; i <= 6; i++) {
4         for (int j = 1; j <= i; j++) {
```

```

5         j == 1 || printf("\t");
6         printf("%d * %d = %d", j, i, i * j);
7     }
8     printf("\n");
9 }
10 return 0;
11 }

```

这段代码中，采用两层循环，外层循环控制行数，内层循环控制每一行的列数，第 i 行应该有 i 列，所以内层循环是从 1 循环到 i 为止。其中最值得琢磨的是 “ $j == 1 \parallel$ printf(“\t”);” 这句代码，其实这句代码就是用来实现行尾无多余 \t 字符这个要求的。代码中采用了在每一列的前面输出一个 \t 字符，可是在第一列的前面不输出 \t 字符，这样就保证了行尾无 \t 字符。

那么 “ $j == 1 \parallel$ printf(“\t”);” 这句代码是如何工作的呢？首先看 \parallel 条件或运算符。 \parallel 运算符的工作逻辑是，左右两侧只要有一个条件成立，那么最终结果就是成立的。这个工作逻辑，还值得细细思考， \parallel 运算符，从左到右依次判断两个条件是否成立，那么如果第一个左边的条件就成立了呢？作为一个聪明人，还需要判断第二个右边的条件么？你会发现，根本不需要再判断右边的条件了，也就是说不需要执行右边的代码了。

看完了条件“或”的这个特性之后，我们再看看 “ $j == 1 \parallel$ printf(“\t”);” 这句代码，也就是说，当 $j == 1$ 成立时，也就是第一列的时候，右边的 printf(“\t”) 代码就根本不会执行。这也就意味着，第一列前面不会多输出一个 \t 字符。而其他的情况呢，均会执行 printf(“\t”) 代码，这也就实现了题目中的要求。

随机函数：随机实验真的可以算 π 值嘛？


这一篇文章里面 [🔗 《随机函数：随机实验真的可以算 \$\pi\$ 值嘛？》](#)，我们介绍了程序里面随机函数的基本原理，说明了“真随机”和“伪随机”的本质区别。看了一些留言以后，我来给你总结一下，所谓“真随机”与“假随机”，只要你不清楚下一个产生的值是什么，那么对于你来说，就是随机的，而“真”或者“假”，讨论的是随机方法的本质。如果随机过程可以保证，下一次产生的每个值都有一定的概率，那么这个就是“真随机”，如果不能保证，那就是“伪随机”。

理解程序中的“伪随机”，你需要在你的脑袋中，构建一个由值组成的环形序列图，设置随机种子，就是选择图中的某个点作为起始点，在我们一次次地获得随机值的过程中，其实程

序就是依次地输出了这个环形序列中的每个状态的值。

最后呢，给你留了一个设计随机函数过程的思考题，关于这个思考题，我要提前先跟你道歉，因为这个思考题，并不是想让你做出来的。下面来看看我的参考答案吧。

思考题：设计迷你随机函数

 复制代码

```
1 #include <stdio.h>
2 int main() {
3     int n = 5;
4     for (int i = 1; i <= 100; i++) {
5         printf("%2d ", n);
6         if (i % 10 == 0) printf("\n");
7         n = (n * 3) % 101;
8     }
9     return 0;
10 }
```

当你运行这个程序的时候，就会看到程序的输出，正如原文中我给你的样例输出一样。要是想理解这段程序，你需要一些数论方面的基础知识，其中包括：欧拉函数，欧拉定理、费马小定理、取余循环节等知识。

在这里，我要再次因为设置这个你可能做不出来这个题，而向你道歉。不过，当你看到上面的那些知识以后，你会发现，这是一道初学者很大概率不可能完成的题目，尽管代码很简单，可背后的原理却看似不简单。其实，我就是想跟你说明，程序的灵魂在算法，算法的灵魂在数学。

数组：一秒钟，定义 1000 个变量

这一篇中，我们学习了数组的基本用法，学会了定义一组数据存储区的方法。并且，围绕着数组知识，完成了“计算数字二进制表示中 1 的个数”的递推程序的设计与实现。

相关的课后思考题呢，也是希望你使用数组来完成相关任务，我看到用户 @奔跑的八戒，完成的就很好，他的思路描述与参考答案一致。也非常感谢 @梅利奥猪猪 和 @毛丽莎肉酱（根据这位用户的名称，我猜可能是漫画《七大罪》的爱好者）和 @Geek_And_Lee00 给出的修改建议以及指正出文章中的笔误，再次感谢二位。如果有好奇的朋友，可以到原文章及留言区看看 [《数组：一秒钟，定义 1000 个变量》](#)。

最后让我们来看看这篇文章的参考答案吧。

思考题：去掉倍数

 复制代码

```
1 #include <stdio.h>
2 int check[1005] = {0};
3 int main() {
4     int n, m, num;
5     scanf("%d%d", &n, &m);
6     for (int i = 0; i < n; i++) {
7         scanf("%d", &num);
8         for (int j = num; j <= m; j += num) {
9             check[j] = 1;
10        }
11    }
12    for (int i = 1; i <= m; i++) {
13        if (check[i] == 1) continue;
14        printf("%d ", i);
15    }
16    return 0;
17 }
```

这段代码中，使用一个 check 数组作为标记，check[i] 等于 0，代表 i 这个数字不是 n 个数字中的任何一个数字的倍数。check[i] 等于 1，代表 i 这个数字能够被 n 个数字中的某个数字整除。其中第 7 行到第 10 行代码，是需要特别关注的。这段代码中，首先读入 n 个数字中的某一个，存储在 num 变量中，之后循环 m 以内所有 num 的倍数，把每个数字的 check 值标记为 1。最后我们循环把 1 到 m 中没有被标记的数字输出，就是符合题目要求的所有数字。


字符串：彻底被你忽略的 printf 的高级用法

这篇 [《字符串：彻底被你忽略的 printf 的高级用法》](#) 的文章中，我们认识了 scanf 和 printf 家族中的两员猛将：sscanf 函数和 sprintf 函数。这两者操作的是字符串，可以理解其本质，就是以字符串为中介做数据类型之间的转换。并且我们还介绍了字符串的相关知识，字符串的相关知识中，比较重要的就是那个 \0 字符，这是一个标记字符串结束的字符，虽然看不到，可作用非常重要，并且这个 \0 字符，也是需要占用存储空间的。

这篇文章中的两个思考题，都是帮助你打开脑洞的，主要就是想告诉你，知识点是死的，而理解知识点和应用知识点是活的，也就是我们常说的活学活用。下面就来看看这篇文章中的

两个思考题的参考答案吧。

思考题 (1) : 体验利器

 复制代码

```
1 #include <stdio.h>
2 char str1[1000], str2[1000];
3 int main() {
4     scanf("%s%s", str1, str2);
5     printf("str1 = %s\tstr2 = %s\n", str1, str2);
6     sprintf(str1, "%s", str1);    // strlen(str1)
7     sprintf(str1, "%s", str2);    // strcpy(str1, str2)
8     printf("str1 = %s\tstr2 = %s\n", str1, str2);
9     sprintf(str1, "%s%s", str1, str2);    // strcat(str1, str2)
10    printf("str1 = %s\tstr2 = %s\n", str1, str2);
11    return 0;
12 }
```


在这段代码中，首先读入两个字符串，str1 和 str2。然后使用 sprintf 分别替代 strlen、strcpy 以及 strcat 三个函数的功能。具体如下：

首先，使用 sprintf(str1, "%s", str1); 代替 strlen(str1) 的功能，正如你所知道的，sprintf 返回值代表输出了多少个字符，这行代码中也就是 str1 字符串中的字符数量。

其次，使用 sprintf(str1, "%s", str2); 代替 strcpy(str1, str2) 的功能。使用 sprintf 函数，将 str2 中的内容，输出到 str1 的存储空间中，其实就相当于把 str2 的内容复制到了 str1 中。

最后，使用 sprintf(str1, "%s%s", str1, str2); 代替 strcat(str1, str2) 的功能。这里，我们将 str1 和 str2 的值，依次性的输出到 str1 中以后，str1 的内容，就是原 str1 和 str2 内容连接以后的总内容了。

思考题 (2) : 优美的遍历技巧

 复制代码

```
1 #include <stdio.h>
2 int main() {
3     char str[1000];
4     scanf("%s", str);
5     for (int i = 0; str[i]; i++) {
6
```

```
7     printf("%c\n", str[i]);
8 }
9     return 0;
}
```

这段代码中，最值得思考的是循环的终止条件。当循环条件成立的时候，循环会一直执行，不成立的时候，循环就会终止。那么 `str[i]` 你可以看成是字符，也可以看成是一个整型值，因为任何信息在底层都是二进制存储的，那么其余字符均为非零值，也就是代表条件成立。

只有一个字符的值是零值，就是我们之前所说的字符串中的最后一个特殊的，看不见的字符，`\0` 字符，这个字符所对应的整型值就是 0，也就是我们所谓的假值。那么这个循环，就会一直循环到字符串的最后一位，才会停止。

好了，今天的思考题答疑就结束了，如果你还有什么不清楚的，或者有更好的想法的，欢迎告诉我，我们留言区见！



人人都能学会的 编程入门课

>>> 每天 10 分钟，轻松学编程

胡光

原百度高级算法研发工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 06 | 字符串：彻底被你忽略的 `printf` 的高级用法

下一篇 07 | 指针系列（一）：记住，指针变量也是变量

精选留言 (4)

写留言



Geek_Andy_Lee00

2020-01-18

看了老师的答疑，解决了前面不少的困惑。至于那个迷你随机数，是真的不怎么懂。先放着吧，先广后精，嘿嘿。还有，给老师提一个小建议，就是在示例代码中加入一些文字，这样可读性更强，对初学者更友好。不然面对一个黑框框，初学者容易懵，不知道该干嘛了。去掉倍数那个思考题，自己也实现了，代码如下，但是老师的方法应该应用更加广泛。...

展开 ∨

作者回复: 好的，你的建议非常好，我会在后面的文章中采纳，你的代码方法也是对的。d(^_^o)



1



奔跑的八戒

2020-01-18

渐入佳境，跟着学就是了。

展开 ∨

作者回复: d(^_^o)



奔跑的八戒

2020-01-18

```
j == 1 || printf("\t");
```

老师这样语句，c 的编译器一定从左往右一个一个条件判断（计算值）吗？有可能从右往左或者随机吗？

作者回复: 对，你上网搜运算符优先级，其中每一个等级的运算符都有一个计算顺序，条件或的计算顺序是从左向右。这是有规定的，跟编译器无关。



梅利奥猪猪 毛丽莎肉...

2020-01-18

其实有挺多东西想说的，首先我的确是七大罪动漫爱好者，虽然现在动画有点ppt了，哈哈扯远了，先说那个乘法表，那个或的知识应该是短路或者惰性的知识吧，我是个前端工程师，有时候就会写到条件&&函数()，这个其实就是条件为真则执行函数，然后还看到过个类似的基础题，true||true&&false答案是true，新手可能直接回答错，从左往右看，认为是false，稍微厉害点的，看优先级，然后得出答案是true，其实这边在多想一步，优先...
展开 ∨

作者回复: d(^_^o)

