

当你打开这个小锦囊的时候，可能正处于被编程作业折磨的走投无路的状态。
如果是这样，请把脑子里乱糟糟的情绪放空，放空，放空，深呼吸。
当你处于一个相对祥和的状态时，再来看一下，我们的提示，能不能帮助你理清一些头绪。
如果它依然效果有限，也请不要绝望，带上你的截图和疑问，大胆私信我吧。

下面的提示分为初级提示、中级提示和高级提示。

初级提示一般仅仅提示基础思路，适合大脑一片空白的小朋友；

中级提示会提示到一些易错点，适合写出了程序框架但是觉得似乎有哪里不对的小朋友；

高级提示大概就很接近剧透了，适合差一点点就豁然开朗的小朋友。

三类提示会分别放在三页上，大家按需查看。

初级提示：

第 1 题：

基础思路：二分查找的前提是数组元素已经按顺序排列好，因此在 `search` 函数中要先做排序，然后再进行二分查找，并根据它的返回值来决定输出信息。

第 2 题：

基础思路：因为 n 的取值范围未知，因此需在用户输入 n 的值后定义一个大小为 n 的动态数组。通过特定处理，使编号为 n 的人报数后下一个报数的是编号为 1 的人。每次报“1”之前检查是否只剩下最后一个人（有可能刚开始就只有一个人）。

第 3 题：

基础思路：在 `Julian` 函数中，定义一个二维字符数组来存储表示 12 个月份的常量字符串，定义一个一维整型数组来存储每个月的天数。写一个循环语句来将天数换算为月数和日数。以月数为下标，把表示月份的常量字符串复制到表示结果的字符数组中。把表示日数的每个数字字符赋值给表示结果的字符数组元素。

第 4 题：

本题没有初级提示。

中级提示：

第 1 题：

易错点：

- ① 没有记录各个数组元素在排序前的下标，错误地输出了排序后的下标。
- ② 写错了排序算法或二分查找算法。
- ③ 在 main 中调用了两次 search，导致第二次 search 把第一次排序后的下标当作原下标。

第 2 题：

易错点：

- ① 习惯性地从 0 开始报数，而非从 1 开始。
- ② 忘了人的编号和对应的数组元素的下标之间相差 1。
- ③ 某些写法如果未考虑只剩下两个人的情况则会出错。

第 3 题：

易错点：

- ④ 误以为返回字符串的函数的返回类型是 char。
- ⑤ 返回了一个局部数组的地址。
- ⑥ 因为各种原因漏掉了月份后面的空格。
- ⑦ 直接将日数的整型数字赋值给字符数组元素，而没有加上 '0'。
- ⑧ 没给字符串结尾添加空字符。
- ⑨ 因为没考虑两位数的日期或忽略了字符串结尾的空字符，导致数组太小。
- ⑩ 没考虑闰年的 2 月有 29 天。

第 4 题：

本题没有中级提示。

高级提示：

第 1 题：

本题没有高级提示。

第 2 题：

定义三个整型变量用来计数，分别对应人的下标、报的数、剩下的总人数。写一个 while 循环，在每个循环周期中判断当前人是否还在圈内、报的数是否为 3、剩下的总人数是否为 1、下一个人的下标是否为 n。

第 3 题：

函数的返回类型是 `char *`，这样才能返回字符串的起始地址。这个字符串应该存储在一个动态数组而不是局部数组中，否则数组会在函数返回后消亡。

第 4 题：

让字符指针指向第一个数组中的某个元素，指针与数组名的差就是第二个数组中对应元素的地址与数组名的差。