

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： 用 `cin>>` 或 `cin.getline` 来读取这个字符串，并存入字符数组 `ch`。
for 循环：lh 从下标为 0 开始往右遍历，到字符元素为 `\0` 停止
for 循环：rh 从下标为 lh 开始往右遍历，到字符元素为 `\0` 停止
if (`ch[lh]==ch[rh]`) `strcpy(ch+rh, ch+rh+1)`; //删去重复字符
//剩下就是排序

第 2 题：

基础思路：定义一个或两个二维字符数组，每行存储一个摩尔斯码。
在字符和摩尔斯码的行序号之间建立对应联系。

第 3 题：

基础思路：写一个 `main` 函数和一个 `getUChar` 函数。在 `main` 函数中调用 `getUChar` 函数，并输出该函数的返回值。在 `getUChar` 函数中请用户输入字符，判断是否为大写字母，再进行后续处理，最后返回用户输入的第一个大写字母。

第 4 题：

基础思路：将用户的输入看作字符串。如果第一个字符不是 `'0'`，那么它是十进制。如果第一个字符是 `'0'` 而第二个字符是 `'x'`，那么它是十六进制。除以上情况之外，它是八进制。根据剩下的字符串计算对应的整型数 `n`。分别定义函数 `Print10(n)`、`Print8(n)` 和 `Print16(n)` 来计算 `n` 在相应进制下每一位的数字并输出（如果前面有 0 或 x 的话也要输出）。

第 5 题：

基础思路：把合并后的单词存到一个二维数组里。

第 1 行是长度为 1 的单词合并的单词，第 2 行是长度为 2 的单词合并的单词，以此类推。

注意：每行都要加上一个 `'\0'` 来表示字符串的结尾。

然后进行选择排序或者冒泡排序。

中级提示：

第 1 题：

易错点：“给数组 ch 排序后再用 `cout<<ch` 输出，发现字符串里有很多乱码。”

分析：以用 `cin.getline` 读取用户输入的字符串为例，一般我们的数组大小会设置成较大的值，而用户输入的字符串往往达不到上限，因此数组 ch 里除了存储用户输入的字符串之外，还有许多剩下的元素，这些元素未被赋值，因此是随机值。如果这些随机值也参与了排序，字符串里就会出现很多乱码。

第 2 题：

难点提示：可以用 `cstring` 库的函数 `strcmp` 来比较两个字符串是否相同。

当 `strcmp(str1, str2)` 返回 0 时，表示 `str1` 和 `str2` 这两个字符串相同，否则不相同。

第 3 题：

易错点：

- ① 用特别大的数组来存放用户输入的字符串，违反题目要求；
- ② 用 `cin.getline` 或 `cin>>` 来获取用户输入的一串字符，然而这也离不开数组；
- ③ 用 `cin.get` 来获取用户输入的字符时误以为最后一个输入的字符是 `'\0'`；
- ④ 在 `main` 函数中“调用 `getUChar`”和“输出 `getUChar` 的返回值”时重复调用了 `getUChar`；
- ⑤ 误将 `getUChar` 的返回类型写成 `int` 并在 `main` 函数中直接输出 `getUChar` 的调用结果；
- ⑥ 误将 `getUChar` 的返回类型写成 `void` 导致和返回语句冲突；
- ⑦ 再次让用户输入时没有先输出“Input: ”。

第 4 题：

易错点：

- ① 误以为将用户输入的 `021`、`0x21` 用 `cin>>` 存入整型变量后即可得到对应的十进制数；
- ② 误以为输出整型变量的进制与存入该整型变量的进制相同（其实无关）；
- ③ 在“根据剩下的字符串计算对应的整型数”时，忽略或未分清结尾处的 `'\0'` 或 `'\n'`；
- ④ 整型数字和数字字符之间转换错误，忘记 `0` 和 `'0'` 之间相差 48；
- ⑤ 忘记十六进制的部分数字是用字母来表示的。

第 5 题：

技术细节：怎么把一个单词存到二维数组的一行里？

如果该单词是个字符串，即以 `'\0'` 结尾，那么可以用 `strcpy`。

例如对于 `char a[5][3]`，那么 `strcpy(a[0], "Hi")` 就可以把“Hi”这个字符串存到 `a` 的第 0 行里。

如果该单词结尾没有 `'\0'`，只有空格符或回车符，那么：

- ① 可以一个字符一个字符地来赋值；或
- ② 把空格符、换行符替换成 `'\0'` 之后再用 `strcpy`。

高级提示：

第 1 题：

出现乱码的解决方法：在排序时要把扫描范围控制在字符串结尾的'\0'之前。

其它提示：

- ① 怎么避免把重复字符删得一个不剩？答：rh 从 lh+1 开始。
- ② 怎么做到删除连续重复的多个字符？答：strcpy 之后执行 rh--；。

第 2 题：

在摩尔斯码转英文时，逐个扫描摩尔斯码中的字符。

当第 i-1 个字符为空格、第 i 个字符不为空格时，找到一个新的摩尔斯码，记录 i 的值；

当第 i 个字符不为空格、第 i+1 个字符为空格时，找完一个摩尔斯码，再记录 i 的值；

把上述两个 i 之间的字符复制到一个数组中，在其结尾加上空字符，使该摩尔斯码成为字符串，以便与二维字符数组中的摩尔斯码进行比较。

注意不要漏掉连续三个空格的情况。

第 3 题：

在 getUChar 函数中用循环结构逐个读取字符，每读取一个字符就判断它是否为大写字母，如果是的话用 break 退出循环并返回该字符。如果读到'\n'还没有退出循环，表示用户在该次输入中没有大写字母，则再次输出"Input: "让用户继续输入。

第 4 题：

在根据字符串计算整型数时，从左往右扫描字符。在任一循环周期中，设当前已经转换出的整型数为 a，当前扫描到的字符为 c，则通过 $a \times \text{进制} + c - '0'$ 来获得新的 a。

在将整型数转换为指定进制的字符串时，先用循环除以进制的方式来确定转换后的位数 k，再从右往左填写字符数组。

第 5 题：

技术细节：怎么按长度对字符串进行排序？

比大小的时候用 strlen 来取值，交换两个元素的时候则用 strcpy（需要中转平台）。