

## 【注意:】

- 1、本次作业不允许使用尚未讲授过的任何后续课程中的知识点，包括但不限于指针、结构体、类等相关概念!!!
- 2、除明确要求外，已学过的知识中，不允许使用 goto 和全局变量，不允许使用 C++ 的 string 变量
- 3、除明确要求外，所有 cpp 源程序不允许使用 scanf/printf 进行输入/输出
- 4、在 cstdio、cmath、cstring 中的系统函数，即使未讲过，也允许使用
- 5、所有题目均需要考虑输入错误的情况，包括同型数据不在指定范围内（例：要求输入[1..12]但输入-2/13 等）以及输入了异型数据（例：需要正整数但输入字符）的情况
- 6、作业必须符合相应的缩进格式，格式分占 10%
- 7、多编译器下均要做到“0 errors, 0 warnings”
- 8、部分题目要求 C 和 C++ 两种方式实现，具体见网页要求
- 9、任何题目使用打表方式输出结果，则除本题分数为 0 外，再扣除总分 20 分

书: P.152 17 (姓名最长四个汉字，学号最长 7 位字符，均不含空格，成绩为整数，信息由键盘输入，输入顺序为依次输入姓名、学号、成绩，不考虑输入错误)

输出格式要求：三行

Line1: 第 1 个人的输入提示，任意

Line2: 键盘输入的第 1 个人的信息

Line3-20: 重复第 2-10 人

Line21: 空行

Line22: 输出提示

Line23~: 输出结果

本题分为两小题:

5-17-1: 用 C 方式，数组完成

5-17-2: 用 C++ 方式，string 类变量完成  
(本小题允许使用 string)



```

C:\WINDOWS\system32\cmd.exe
请输入第1个人的姓名、学号、成绩
江浩 1751375 76
请输入第2个人的姓名、学号、成绩
江浩 1751376 54
请输入第3个人的姓名、学号、成绩
宋安迪 1751396 82
请输入第4个人的姓名、学号、成绩
石佳琛 1751456 35
请输入第5个人的姓名、学号、成绩
宋子帆 1751533 49
请输入第6个人的姓名、学号、成绩
潘昊昊 1751561 87
请输入第7个人的姓名、学号、成绩
潘泓舟 1751691 78
请输入第8个人的姓名、学号、成绩
杨君临 1751739 34
请输入第9个人的姓名、学号、成绩
刘鲲鹏 1751740 53
请输入第10个人的姓名、学号、成绩
陈威 1751763 71

不及格名单:
江浩 1751376 54
石佳琛 1751456 35
宋子帆 1751533 49
杨君临 1751739 34
刘鲲鹏 1751740 53
请按任意键继续. . .
  
```

补充:

- 12、 用二维字符数组的方式实现 P. 148-149 的例 5.11 (假设每个字符串长度不超过 29 字节, 格式与例题相同即可, **不考虑输入错误**)
- 13、 用二维字符数组的方式实现 P. 149-150 的例 5.12 (姓名/学号是字符串, 长度不超过 9 字节, 格式与例题相同即可, **不考虑输入错误, 可继续使用全局变量**)
- 14、 用函数+数组方式重新完成 3-b10 (人民币转大写), 要求如下:  
【要求:】1、所有的大写数字均放在全局一维只读字符数组 chistr 中, 具体形式为 const char chistr[]="零壹贰叁肆伍陆柒捌玖";  
凡需输出“零-玖”的地方, **只允许**从此数组中取值  
2、其它大写内容 (拾佰仟万亿圆角分整) 可自行取值  
3、转换后的内容 **不允许逐次输出**, 最后的输出 **只允许**用一句 cout << result << endl 来完成, result 数组有两种要求 (两个小题)  
3.1 **全局字符数组** char result[256];  
3.2 **全局 C++ 的 string 类变量** string result; (本小题允许使用 string 类变量)  
(可通过其它输出语句进行输入提示、错误提示等个性化输出, 但输出大写转换结果的语句只能是一句)  
4、保证 3-b10 中的所有测试数据均通过  
【提示:】1、根据分解的各位数字从 chistr 中取部分内容  
2、各位数字要输出的内容依次放入 result 中, 最后输出这个字符串即可  
3、保证 3-b10 中的所有测试数据均通过, 输入输出格式要求同 3-b10
- 15、 写一个密码生成程序, 要求按要求生成指定长度的密码, 放在一个数组中并输出  
【要求:】1、构成密码的所有字符为基本 ASCII 码中的所有可见字符 (33-126 之间)  
2、密码的长度由键盘输入, 在 12-16 之间  
3、密码必须包含大写字母、小写字母、数字、其他符号等四类, 每类字符的最少数目由键盘输入, 不少于 2 个  
4、各类字符出现的位置不允许固定 (例: 首字母永远小写, 第 4 个永远是数字等)  
5、为便于检查效果, 每次运行生成 10 组密码并打印在屏幕上  
6、输出形式如下, 给出 5-b15.exe 供参考 (在 cmd 下运行)

红线上方为输入提示 (任意) 及要输入的内容

红线下方为输出:

第 1 行: 5 个整数, 分别表示密码长度、**本次输入要求的大写/小写/数字/其它字符的最少数量**  
后续 10 行: 10 组密码串, 每组均符合刚才的要求

```
C:\WINDOWS\system32\cmd.exe
请输入密码长度<12-16>
16
请输入大写字母个数<至少2个>
3
请输入小写字母个数<至少2个>
2
请输入数字个数<至少2个>
3
请输入其它字符个数<至少2个>
4
16 3 2 3 4
94c"6AS.?`q"FL1B
zf*qR7eJ1=$Z2Rc1
P6_PHH;Y99o.u:7"
WB2F6H-0eY1!?'`!l
<Gs2_2>M?9P95W.g
P/9au9*R<\h1T*+>
9nN6#6JT\Nn#Y!0r
1n8P`oF3$^aK(Rx=
Tp&+L>9uYL/*9z9X
9!q<uy>q\4G5XM5+
请按任意键继续. . .
```

- 16、从文件中读取随机生成的密码串，来验证上一题的答案是否正确
- 【要求:】** 1、将上一题的输出，重定向到文件中，再做为本题的输入重定向文件（红线下的内容），不考虑输入错误
- 2、读取方法为：首先读取 5 个整数，表示密码长度，大写/小写/数字/其它的最小个数  
再读取后续 10 行，每行为指定长度个字母当做密码
- 2、检查条件为：长度是否符合要求、各种类型字符是否符合各自的最小要求
- 3、程序的输入不需要任何提示，输出只有“正确”/“错误”两种，带一个换行
- 4、本题需要相互验证（甲的本题去验证乙的上一题生成的数据文件），每人需要验证至少 5 人的上一题，将名单放在源程序的第 2 行用注释说明即可（**正常情况双向查验都应该正确，如果查验不正确则要连环扣分，即生成者检查者同步扣分**）

```

/* 1751234 张三 计算机 2 班 */
/* 已验证 1751111 李四 1752222 王五 ... 的密码串生成 5 */

```

5、也可以在上一题输出重定向结果文件中故意改错部分数据，使结果为“错误”

17、用一维字符数组方式实现下列函数

函数原型	功能说明	返回值
int tj_strlen(const char str[]);	求字符串 str 的长度	字符串长度
int tj_strcat(char s1[], const char s2[]);	将字符串 s2 追加到 s1 后面，含\0	0
int tj_strcpy(char s1[], const char s2[]);	将字符串 s2 复制到 s1 中，覆盖 s1 中原内容，复制时包含\0	0
int tj_strncpy(char s1[], const char s2[], const int len);	将字符串 s2 的前 len 个字符复制到 s1 中，复制时不含\0 ★ 若 len 比 s2 的长度大，复制 s2 长度个字符即可(不含\0)	0
int tj_strcmp(const char s1[], const char s2[]);	比较字符串 s1 和 s2 的大小，英文字母要区分大小写	相等为 0，不等则为第 1 个不相等字符的 ASCII 差值
int tj_strcasecmp(const char s1[], const char s2[]);	比较字符串 s1 和 s2 的大小，英文字母不分大小写	相等为 0，不等则为第 1 个不相等字符的 ASCII 差值
int tj_strncmp(const char s1[], const char s2[], const int len);	比较字符串 s1 和 s2 的前 len 个字符的大小，英文字母要区分大小写 ★ 若 len 大于 s1/s2 中长度短的串，则比较到短串的\0 即结束	相等为 0，不等则为第 1 个不相等字符的 ASCII 差值

int tj_strcasencmp(const char s1[], const char s2[], const int len);	比较字符串 s1 和 s2 的前 len 个字符的大小，英文字母不分大小写 ★ 要求同 tj_strncmp	相等为 0，不等则为第 1 个不相等字符的 ASCII 差值
int tj_strupr(char str[]);	将字符串 str 中所有小写字母均转为大写，其它字符不变，转换后放在原串中	0
int tj_strlwr(char str[]);	将字符串 str 中所有大写字母均转为小写，其它字符不变，转换后放在原串中	0
int tj_strchr(const char str[], const char ch);	在字符串 str 中寻找字符 ch 第 1 次出现的位置，顺序是从左到右	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
int tj_strstr(const char str[], const char substr[]);	在字符串 str 中寻找字符串 substr 第 1 次出现的位置，顺序是从左到右	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
int tj_strrchr(const char str[], const char ch);	在字符串 str 中寻找字符 ch 第 1 次出现的位置，顺序是从右到左 ★ 例：tj_strrchr("abcdab", 'a') 返回 5	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
int tj_strrstr(const char str[], const char substr[]);	在字符串 str 中寻找字符串 substr 第 1 次出现的位置，顺序是从右到左 ★ 例：tj_strrstr("abcdab", "ab") 返回 5	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
int tj_strrev(char str[]);	字符串反转，放入原串中	0

【要求：】1、**不允许**使用任何系统函数（strlen、strcpy 等），**不允许**使用 C++ 的 string 类，**不允许**借助指针，**不允许**定义全局变量

2、可以用自己定义的函数（例如在其它 tj\_\*\*函数中调用 tj\_strlen）

3、函数实现时不必考虑空间不够的情况（空间由调用函数保证）

4、给出 5-b17.h、5-b17-main.cpp、5-b17-sub.cpp 三个文件共同形成一个可执行文件，5-b17.h 用于函数声明，5-b17-main.cpp 是用于测试的主函数，这两个文件**不准改动，无需提交**；列表中所有函数的具体实现均在 5-b17-sub.cpp 中，每个函数实现时有具体要求，必须按要求实现，提交时只需提交此文件即可。

## 18、生成并打印 Windows 扫雷游戏的内部数组结构

【Windows 扫雷游戏的玩法：】1、开始游戏，以高级难度 16\*30 的位置中 99 颗雷为例，此时虽然屏幕无显示，但 99 颗雷在什么位置内部已知

2、按下鼠标左键，表示玩家确认该位置不是雷，此时若其周围 8 个位置均无雷（**四角位置：1-3，四边位置：1-5，下同**），则屏幕显示空白（会将所有相连的空白位置全部显示），否则会按周围 8 个位置有几颗雷来显示数字 1-8；如果该位置是雷，则给出提示，游戏结束

3、按下鼠标右键，表示玩家确认该位置是雷，此时屏幕会显示小红旗（如果玩家判断错误，此处不应是雷，会导致后续判断错误）

- 【要求:】1、在 10x26 的范围内随机产生 50 颗雷(若生成的位置已有雷,则需要再次生成新位置)
- 2、其它非雷位置分别给出 0-8, 表示其周围 8 个位置的雷数
- 3、输出形式如下, 给出 5-b17.exe 供参考 (在 cmd 下运行)

```
1 2 * 1 0 1 1 2 * * 3 2 * 1 0 0 0 0 0 1 * 1 0 0 0
1 * 3 3 1 2 * 2 3 * * 3 2 2 0 1 1 1 0 0 1 1 1 0 0 0
2 3 * 2 * 2 1 1 1 2 3 4 * 2 1 2 * 1 0 1 1 1 0 1 1 1
* 3 2 3 2 1 0 0 0 0 1 * * 2 1 * 2 1 0 2 * 2 0 1 * 1
* 3 1 * 1 1 1 2 1 1 1 2 2 1 2 2 2 1 1 3 * 2 0 1 2 2
* 2 2 2 2 1 * 4 * 2 0 0 0 0 2 * 4 3 * 3 2 2 0 1 2 *
1 1 2 * 2 1 2 * * 4 1 0 0 0 2 * * * 2 3 * 2 0 1 * 2
0 0 2 * 3 2 3 4 * * 1 0 0 0 1 2 3 2 1 2 * 3 1 1 2 2
1 1 2 2 3 * * 2 3 4 4 2 1 0 0 0 0 0 0 2 3 * 1 0 1 *
* 1 1 * 2 2 2 1 1 * * * 1 0 0 0 0 0 0 1 * 2 1 0 1 1
请按任意键继续. . .
```

19、从文件中读取 Windows 扫雷游戏的内部数组结构, 来验证上一题的答案是否正确

- 【要求:】1、将上一题的输出, 重定向到文件中, 再做为本题的输入重定向文件, 仍然放在 10x26 的字符数组内 (不考虑输入错误)

2、检查条件如下

2.1 星号的个数是否是 50 个, 输出“错误”后程序结束

2.2 在\*个数正确的前提下, 重新计算周围的雷数, 再和读入的内容进行比较, 任一不匹配则输出“错误”后程序结束

2.3 通过 2.1/2.2 的检查则输出“正确”后程序结束

3、程序的输入不需要任何提示, 输出只有“正确”/“错误”两种, 带一个换行

4、本题需要相互验证 (甲的本题去验证乙的上一题生成的数据文件), 每人需要验证至少 5 人的上一题, 将名单放在源程序的第 2 行用注释说明即可 (正常情况双向查验都应该正确, 如果查验不正确则要连环扣分, 即生成者检查者同步扣分)

```
/* 1751234 张三 计算机 2 班 */
/* 已验证 1751111 李四 1752222 王五 ... 的扫雷内部数组 */
```

5、也可以在上一题输出重定向结果文件中故意改错部分数据, 使结果为“错误”

#### 【作业要求:】

1、12 月 13 日前网上提交本次作业 (本次作业, 5-b17 要求除 Linux 外的 3 编译器通过, 其余均为 4 编译器通过, 0 error, 0 warning)

2、每题所占平时成绩的具体分值见网页

3、超过截止时间提交作业会自动扣除相应的分数, 具体见网页上的说明

4、预告: 本周还有综合题及额外加分题, 具体要求另行下发, 完成时间为 2 周