

【注意:】

- 1、本次作业**不允许**使用后续课程的知识点，包括但不限于指针、引用、结构体、类等概念!!!
- 2、除明确要求外，已学过的知识中，不允许使用 goto，不允许使用全局变量
- 3、cstdio 及 cmath 中的系统函数**可以**直接使用，包括课上未介绍过的，具体可自行查阅相关资料
- 4、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
- 5、多编译器下均要做到“0 errors, 0 warnings”
- 6、部分题目要求 C 和 C++ 两种方式实现，具体见网页要求
- 7、输出为浮点数且未指定格式的，均要求为 double 型，C++ 为 cout 缺省输出，C 为%lf 的缺省输出
- 8、认真阅读格式要求及扣分说明!!!

【输出格式要求:】

- 1、为方便机器自动判断正确性，作业有一定的输入输出格式要求（但不同于竞赛的无任何提示）
- 2、每个题目见具体说明，必须按要求输入和输出，不允许有偏差
- 3、没有特别说明的情况下，最后一行有效输出的最后一个 endl

补充:

- 1、一批已排好序的数组存放在一个数组中，现要求输入一个数，插入到数组中，使数组仍保持有序

【要求:】 1、假设均为正整数，升序排列

- 2、已排序的数组的元素个数不定，键盘输入时，结束条件为满 20 个或输入<0 的数为止
- 3、不考虑输入错误，包括非升序、输入非法字符等
- 4、提供 5-b1-demo.exe 供参考（cmd 下运行）
- 5、输入输出的具体格式要求如下

输出格式要求：多行

Line1: 原数组输入提示

Line2: 键盘输入的若干正整数

- 如果超过 20 个，则清空缓冲区，以方便后续输入要插入的数
- 如果一个数都没有，则输出“无有效输入”后结束

Line3: 原数组输出提示

Line4: 数组的原内容

Line5: 输入要插入数字的提示

Line6: 键盘输入要插入的正整数

Line7: 输出首行提示

Line8: 数组的新内容

Microsoft Visual Studio 调试控制台

请输入任意个正整数（升序，最多20个），以-1结束

2 8 11 23 26 28 34 37 39 41 42 46 48 51 53 57 58 60 62 64 -1

原数组为：

2 8 11 23 26 28 34 37 39 41 42 46 48 51 53 57 58 60 62 64

请输入要插入的正整数

33

插入后的数组为：

2 8 11 23 26 28 33 34 37 39 41 42 46 48 51 53 57 58 60 62 64

Microsoft Visual Studio 调试控制台

```
请输入任意个正整数（升序，最多20个），以-1结束
2 8 11 23 26 28 34 37 39 41 42 46 48 51 53 57 58 60 62 64 69 73 84 95 -1
原数组为：
2 8 11 23 26 28 34 37 39 41 42 46 48 51 53 57 58 60 62 64
请输入要插入的正整数
33
插入后的数组为：
2 8 11 23 26 28 33 34 37 39 41 42 46 48 51 53 57 58 60 62 64
```

Microsoft Visual Studio 调试控制台

```
请输入任意个正整数（升序，最多20个），以-1结束
-1
无有效输入
```

- 2、有 100 盏灯，编号 1-100，初始状态为全灭；现有 100 个人，编号也是 1-100，每个人只对自己编号整数倍的灯按一次开关，100 人依次操作完成后，给出所有依旧处于亮灯状态的灯的编号。

（例：编号为 1 的人操作后所有灯全亮；编号为 2 的操作后，所有偶数灯全灭；…）

【要求：】此题为小学奥数题（答案为 1-100 间的完全平方数），但不允许用数学方法解题，只允许用最基本的循环+数组方法模仿 100 个人按开关的操作来实现

本题无输入，输入如左图
(最后不允许出空格)

Microsoft Visual Studio 调试控制台

```
1 4 9 16 25 36 49 64 81 100
```

```
D:\WorkSpace\高级语言程序设计\部分作业
按任意键关闭此窗口...
```

- 3、用函数+数组方式重新完成 3-b5(输入年月日求是该年的第几天)

【要求：】1、断闰年、求天数分别用不同函数完成，主程序只负责输入及输出

2、每月的天数/总天数不能用 if-else/swicth 语句来填写（类似下面的语句都不允许）

```
if (m==1 || m==3 ...)  
    d = 31;  
else if (m==4 || m==6...)  
    d = 30;  
else  
    ...
```

不允许

```
switch(m) {  
    case 1:  
    case 3:  
    case ...  
        d = 31;  
    }  
    ...
```

不允许

```
switch(m) {  
    case ...:  
    case 5:  
        d=31+28+leap+31+30+day  
    case ...  
    }  
    ...
```

不允许

3、其余要求（含输入错误处理）同 3-b5 要求

4、其余位置允许使用分支语句

4、数组中有一批数据，代表学生成绩（[0~100]，有相同值，无序），请统计各分数的人数

- 【要求：】
- 1、数据从键盘读入，个数不定，最多 1000 个，以-1 为输入结束（-1 不计入有效数据）
 - 2、不考虑输入错误（分数为负数或大于 100，输入字母，输入超过 1000 个等）
 - 3、给出 5-b4-demo.exe 供参考
 - 4、给出示例的 5-b4-makedat.exe，运行一次，可生成一个 score.dat 文件，内含符合要求的测试数据，可采用输入输出重定向方式运行并查看结果，方法为 cmd 下输入 5-b4-demo.exe < score.dat > count.txt，则输出结果在 count.txt 中

输出格式要求：多行（有不清楚之处请参考 5-b4-demo.exe 的运行结果）

Line1：输入提示

Line2：键盘输入的若干非负整数，以负数结束

- 这部分内容不会出现在输出重定向的结果文件中，因此多行也可以

Line3：数组输出提示

Line4[~]x：数组的内容

- 每 10 个数字一行，最后允许有空格，最后一行无论是否满 10 个都要换行
- 如果第一个数就是负数，输出“无有效输入”

Line x+1：“分数与人数的对应关系为：”

Line x+2[~]：按分数从高到低的人数统计

```
Microsoft Visual Studio 调试控制台
请输入成绩（最多1000个），以-1结束
87 86 56 76 87 92 76 96 98 23 55 56 34 23 -1
输入的数组为：
87 86 56 76 87 92 76 96 98 23
55 56 34 23
分数与人数的对应关系为：
98 1
96 1
92 1
87 2
86 1
76 2
56 2
55 1
34 1
23 2

D:\WorkSpace\高级语言程序设计\部分作业(VS2019)
按任意键关闭此窗口...
```

```
Microsoft Visual Studio 调试控制台
请输入成绩（最多1000个），以-1结束
87 86 56 76 87 92 76 96 98 23 -1
输入的数组为：
87 86 56 76 87 92 76 96 98 23
分数与人数的对应关系为：
98 1
96 1
92 1
87 2
86 1
76 2
56 1
23 1

D:\WorkSpace\高级语言程序设计\部分作
按任意键关闭此窗口...
```

5、数组中有一批数据，代表学生成绩（[0~100]，有相同值，无序），请逆序输出各成绩对应的名次

【要求：】1、成绩相同则名次相同，下一名次要跳过并列名次

2、数据从键盘读入，个数不定，最多 1000 个，以-1 为输入结束（-1 不计入有效数据）

3、不考虑输入错误（分数为负数或大于 100，输入字母，输入超过 1000 个等）

4、给出 5-b5-demo.exe 供参考

5、可继续使用 5-b4-makedat.exe 生成的 score.dat 文件，可采用输入输出重定向方式运行并查看结果，方法为 cmd 下输入 5-b5-demo.exe < score.dat > sort.txt，则输出结果在 sort.txt 中

输出格式要求：多行（有不清楚之处请参考 5-b5-demo.exe 的运行结果）

Line1：输入提示

Line2：键盘输入的若干非负整数，以负数结束

● 这部分内容不会出现在输出重定向的结果文件中，因此多行也可以

Line3：数组输出提示

Line4~x：数组的内容

● 每 10 个数字一行，最后允许有空格，最后一行无论是否满 10 个都要换行

● 如果第一个数就是负数，输出“无有效输入”

Line x+1：“分数与名次的对应关系为：”（英文冒号）

Line x+2~：按名次排列的成绩，每个一行，成绩与名次之间一个空格

Microsoft Visual Studio 调试控制台

请输入成绩（最多 1000 个），以-1 结束

87 86 56 76 87 92 76 96 98 23 55 56 34 23 -1

输入的数组为：

87 86 56 76 87 92 76 96 98 23

55 56 34 23

分数与名次的对应关系为：

98 1

96 2

92 3

87 4

87 4

86 6

76 7

76 7

56 9

56 9

55 11

34 12

23 13

23 13

D:\WorkSpace\高级语言程序设计\部分作业(VS2019)

按任意键关闭此窗口...

Microsoft Visual Studio 调试控制台

请输入成绩（最多 1000 个），以-1 结束

87 86 56 76 87 92 76 96 98 23 -1

输入的数组为：

87 86 56 76 87 92 76 96 98 23

分数与名次的对应关系为：

98 1

96 2

92 3

87 4

87 4

86 6

76 7

76 7

56 9

23 10

D:\WorkSpace\高级语言程序设计\部分作业

按任意键关闭此窗口...

6、题目同 4-b7（汉诺塔），要求给出移动过程中每根柱子上现有的圆盘数量及编号

【要求：】1、假设圆盘最大数量为 10，其余输入格式要求同前

2、要求打印初始状态下，起始圆柱拥有的圆盘数及每个圆盘的编号，在随后的每个移动步骤中，打印移动完成后每个圆柱的现有的圆盘数及编号（效果如下图）

请输入汉诺塔的层数(1-10)：

4

请输入起始柱(A-C)：

A

请输入目标柱(A-C)：

C

要求同之前题目

初始:	A: 4 3 2 1	B:	C:
第 1 步(1): A-->B	A: 4 3 2	B: 1	C:
第 2 步(2): A-->C	A: 4 3	B: 1	C: 2
第 3 步(1): B-->C	A: 4 3	B:	C: 2 1
第 4 步(3): A-->B	A: 4	B: 3	C: 2 1
第 5 步(1): C-->A	A: 4 1	B: 3	C: 2
第 6 步(2): C-->B	A: 4 1	B: 3 2	C:
第 7 步(1): A-->B	A: 4	B: 3 2 1	C:
第 8 步(4): A-->C	A:	B: 3 2 1	C: 4
第 9 步(1): B-->C	A:	B: 3 2	C: 4 1
第 10 步(2): B-->A	A: 2	B: 3	C: 4 1
第 11 步(1): C-->A	A: 2 1	B: 3	C: 4
第 12 步(3): B-->C	A: 2 1	B:	C: 4 3
第 13 步(1): A-->B	A: 2	B: 1	C: 4 3
第 14 步(2): A-->C	A:	B: 1	C: 4 3 2
第 15 步(1): B-->C	A:	B:	C: 4 3 2 1

D:\WorkSpace\高级语言程序设计\部分作业(VS2019)\Debug\第05章-用数组模拟汉诺塔的移动
按任意键关闭此窗口...

3、从初始行开始，每行的具体输出格式要求（下面截图为 10 层的部分步骤）

初始:	A:10 9 8 7 6 5 4 3 2 1	B:	C:
第 1 步(1): A-->B	A:10 9 8 7 6 5 4 3 2	B: 1	C:
第 2 步(2): A-->C	A:10 9 8 7 6 5 4 3	B: 1	C: 2
第 510 步(2): C-->B	A:10 1	B: 9 8 7 6 5 4 3 2	C:
第 511 步(1): A-->B	A:10	B: 9 8 7 6 5 4 3 2 1	C:
第 512 步(10): A-->C	A:	B: 9 8 7 6 5 4 3 2 1	C:10
第1021 步(1): A-->B	A: 2	B: 1	C:10 9 8 7 6 5 4 3
第1022 步(2): A-->C	A:	B: 1	C:10 9 8 7 6 5 4 3 2
第1023 步(1): B-->C	A:	B:	C:10 9 8 7 6 5 4 3 2 1

- 步数宽度为 4，右对齐，再多一个空格到“步”
- () 内为当前移动的盘号，宽度为 2，右对齐
- ABC 柱内容打印时，每个数字间一个空格，ABC 间宽度预留正好 10 个数字+空格
- 所有空白均为空格，不是其它符号（例：“初始:”和“A:”之间是多个空格）
- 每行最后允许多出空格，所有冒号均为英文符号
- 输出行数过多，可以用输出重定向方式进行检查
- 打印时，每行必须完整地从左到右输出，不允许采用之前的 gotoxy 之类的函数改变光标位置，也不允许使用退格键
- 如果屏幕宽度过小，不足以容纳一行，则需要调整 cmd 窗口的宽度（具体方法参考之前的文档）

4、要求使用递归方式完成，为了在各层递归中共用变量，**本题允许使用全局变量**，具体要求为两种方式，**不准违规多使用全局变量**

- 用三个全局简单变量表示三个栈的栈顶指针、三个全局一维数组来记录三根圆柱中的圆盘数及编号、一个计数用的全局变量
- 用一个全局一维数组（大小为[3]）、一个全局二维数组（大小为[3][10]）来记录三根圆柱中的圆盘数及编号、一个计数用的全局变量

5、给出 5-b6-demo.exe 供参考

6、递归函数**不允许**使用循环，处理输入错误及打印横向数组时，**允许**使用循环

【提示：】

1、之前表达式求值、函数嵌套调用等多处都接触到了数据结构中“栈”的概念，学习了数组后，我们可以用一个一维数组来模拟一个栈，完成栈的基本操作，基本方式为：假设数组大小为 n，则把[0]做为栈底，用一个 int top 做为栈顶指针，指向元素即将插入的位置

(请思考：为什么 top 不指向栈顶，而是栈顶+1，有什么好处？)

在此基础上，栈(用数组 stack 表示)的基本操作示意如下：

init: top=0

push: 元素入栈，stack[top++]=入栈元素的值 (思考：为什么是后缀++)，如果 top 已经是数组大小，则入栈失败

pop : 元素出栈，存放出栈元素值的变量=a[--top] (思考：为什么是前缀--)，如果 top 已经是 0，则出栈失败

例：下图分别表示了不同情况下栈的情况及栈顶指针的值

下标	值	下标	值	下标	值	下标	值
[9]	0	[9]	0	[9]	0	[9]	11
[8]	0	[8]	0	[8]	0	[8]	23
[7]	0	[7]	0	[7]	0	[7]	15
[6]	0	[6]	0	[6]	0	[6]	2
[5]	0	[5]	0	[5]	0	[5]	9
[4]	0	[4]	0	[4]	0	[4]	7
[3]	0	[3]	0	[3]	3	[3]	3
[2]	0	[2]	0	[2]	8	[2]	8
[1]	0	[1]	0	[1]	12	[1]	12
[0]	0	[0]	5	[0]	5	[0]	5

空栈 一个元素 四个元素 栈满

top=0 top=1 top=4 top=10

2、柱子编号为 A、B、C，对应 ABC 三个长度为 10 的数组，数组中的值表示圆盘的编号（1-最小的盘子 10-最大的盘子）

例如：4 层，初始在 B 柱，要求移动到 A 柱，则三个数组的初值如左下所示

	A	B	C		A	B	C
[9]	0	0	0		0	0	0
[8]	0	0	0		0	0	0
[7]	0	0	0		0	0	0
[6]	0	0	0		0	0	0
[5]	0	0	0		0	0	0
[4]	0	0	0		0	0	0
[3]	0	1	0		0	0	0
[2]	0	2	0		0	0	0
[1]	0	3	0		0	0	0
[0]	0	4	0		0	0	1

完成第 1 步
最小的圆盘(1)从 B->C 后，
数组变为

- 3、递归方式，为了在各层递归中共用变量，**允许**用最多三个全局简单变量、最多三个全局一维数组来记录三根圆柱中的圆盘数及编号、一个计数用的全局变量（与题目要求 4 对应）
- 4、递归方式，为了在各层递归中共用变量，**允许**用最多一个全局一维数组、最多一个全局二维数组来记录三根圆柱中的圆盘数及编号、一个计数用的全局变量（与题目要求 4 对应）

【编译器要求：】

		编译器VS	编译器Dev
5-b1.cpp	有序数组中插入数据后仍保持有序	Y	Y
5-b2.cpp	100盏灯的亮灭	Y	Y
5-b3.c	输入日期求天数	Y	Y
5-b4.c	含相同值的成绩计数	Y	Y
5-b5.cpp	含相同值的成绩排序	Y	Y
5-b6-1.c	汉诺塔-每步详细(横向)-3个变量+3个一维	Y	Y
5-b6-2.cpp	汉诺塔-每步详细(横向)-1个一维+1个二维	Y	Y

【作业要求：】

- 1、**11月15日前**网上提交本次作业
- 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、**友情提醒：下周作业恢复正常的一周限制，即截止时间也是11月15日**