

**【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，包括但不限于二维/多维数组、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
- 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、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、从本次作业（含本次）开始，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.151 4 (输入任意个正整数，升序，最多 20 个有效数字，以-1 结束，打印数组的原内容后再输入要插入的正整数，再打印数组的新内容即可，本题不考虑输入错误，包括超数量、非升序、输入负数/字母等)

输出格式要求：多行

Line1: 原数组输入提示，任意

Line2: 键盘输入的若干正整数

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提示，任意

Line6: 键盘输入要插入的正整数

Line7: 输出首行提示，任意

Line8: 数组的新内容

```

C:\WINDOWS\system32\cmd.exe
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
请按任意键继续. . .
  
```

**补充:**

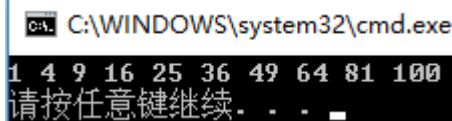
- 1、了解程序运行时的系统分配（动态数据区、静态数据区）

**【要求:】** 1、仿照第 10 周上课时的演示程序，用不断变换 char 数组定义大小的方法来分析出四种编译器生成的程序的动态数据区、静态数据区的大小

- 2、具体项目列在 5-b1.docx 中，完成该表格并转换为 PDF 后提交

- 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 个人的开关操作来实现

本题无输入，输入如左图  
 (最后不允许多出空格)



- 3、用函数+数组方式重新完成 3-b9(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/switch 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、年份限定在 2000-2030 之间，其余输出格式要求同 3-b9

- 4、假设数组中有一批数据，代表学生成绩（有相同值且无序），请排序后输出成绩对应的名次

假设 int a[5]={85,90,90,73,85};

则输出为：

```
90  1
90  1
85  3
85  3
73  5
```

**【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，个数不限（最大 1000），-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（可以采用输入输出重定向方式输入数据并查看结果，方法为 cmd 下输入 5-b4.exe < score.dat > sort.txt，具体参考之前的课件）

输出格式要求：多行

Line1: 输入提示，任意

Line2: 键盘输入的若干正整数，-1 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）

Line x+1: “分数与名次的对应关系为:”（中文冒号）

Line x+2~: 按名次排列的成绩，每个一行

```

C:\WINDOWS\system32\cmd.exe
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
请按任意键继续. . .

```

```

C:\WINDOWS\system32\cmd.exe
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
请按任意键继续. . .

```

- 5、题目同第4章 P.123 习题9（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):
b
请输入目标柱(A-C):
a

```

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

请按任意键继续. . .

- 3、递归方式完成，给出 5-b5.exe 供参考
- 5、本题的三个数组的内容打印**不允许**采用之前的 gotoxy 之类的函数改变光标位置
- 6、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:      B:10 9 8 7 6 5 4 3 2 1 C:
第1步(1): B-->A A: 1 B:10 9 8 7 6 5 4 3 2 C:
第2步(2): B-->C A: 1 B:10 9 8 7 6 5 4 3 C: 2
第510步(2): C-->A A: 9 8 7 6 5 4 3 2 B:10 1 C:
第511步(1): B-->A A: 9 8 7 6 5 4 3 2 1 B:10 C:
第512步(10): B-->C A: 9 8 7 6 5 4 3 2 1 B: C:10
第1022步(2): B-->C A: 1 B: C:10 9 8 7 6 5 4 3 2
第1023步(1): A-->C A: B: C:10 9 8 7 6 5 4 3 2 1
请按任意键继续. . .

```

- 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，跨度为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
- 可以用输出重定向方式进行检查

【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  $\text{int top}$  做为栈顶指针，指向元素即将插入的位置

（请思考：为什么  $\text{top}$  不指向栈顶，而是栈顶+1，有什么好处？）

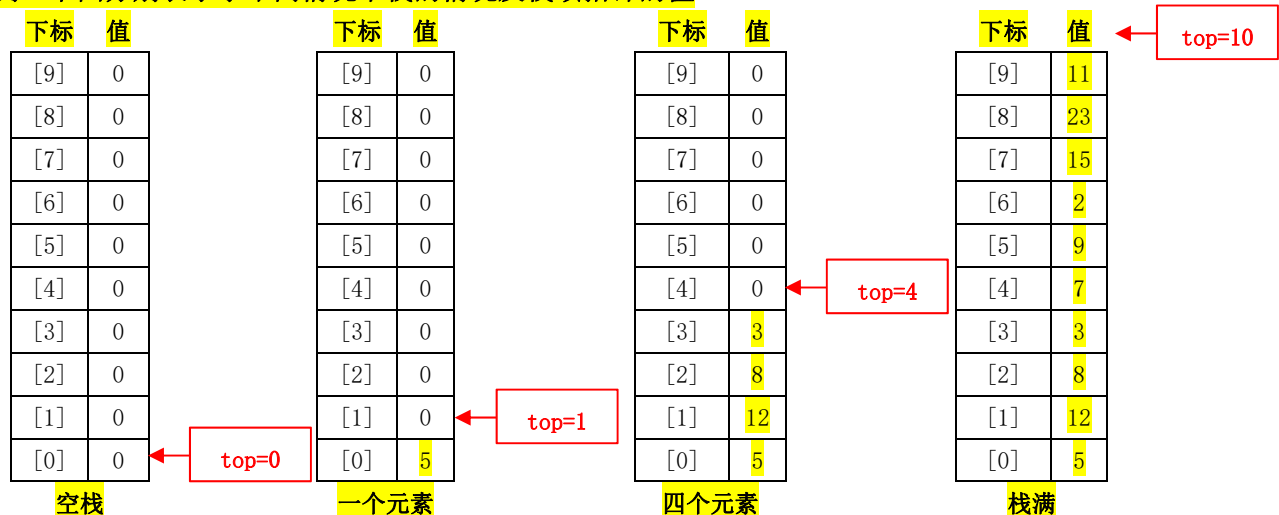
在此基础上，栈（用数组  $\text{stack}$  表示）的基本操作示意如下：

$\text{init: top}=0$

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

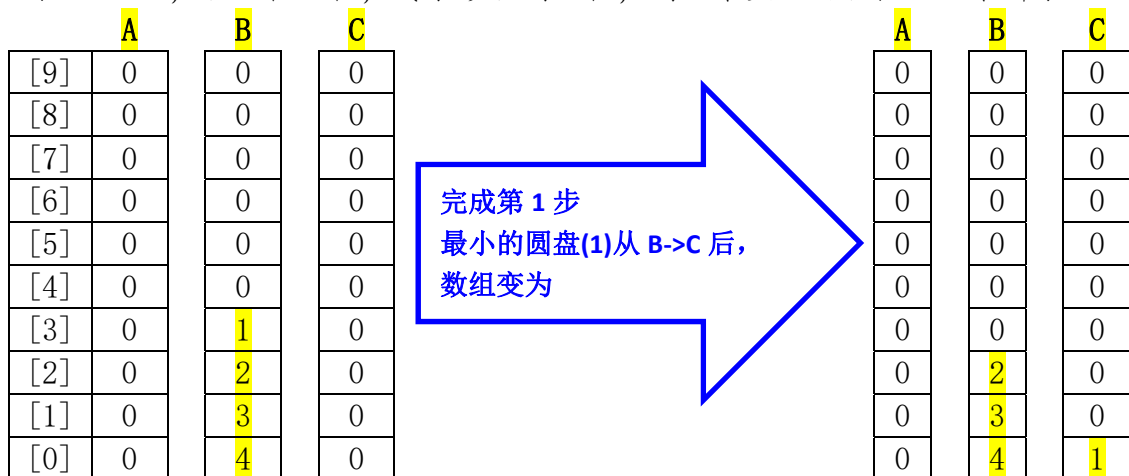
$\text{pop}$ : 元素出栈，存放出栈元素值的变量  $= \text{a}[\text{--top}]$ （思考：为什么是前缀--），如果  $\text{top}$  已经是 0，则出栈失败

例：下图分别表示了不同情况下栈的情况及栈顶指针的值



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



3、递归方式，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

**【作业要求:】**

- 1、本次作业，全部要求4编译器通过（0 error，0 warning）
- 2、**11 月 29 日前**网上提交本次作业
- 3、每题所占平时成绩的具体分值见网页
- 4、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