

**2020 年夏季学期**  
**《程序设计技能训练》**  
**实验指导书**

**沙金**  
**信息学院**

**2020/06/30~2020/07/09**

## 说明

《程序设计技能训练》是《C 语言程序设计》和《面向对象程序设计》两门课程的后继实践类课程，目的是通过 10 天的集中时间，训练综合程序设计能力。上课时必须注意几点：

1. 6 月 30 日到 7 月 9 日。晚上 19 点到 22 点集中讲题、答疑、答辩，要求都要按时参加，不能无故旷课、迟到、早退，否则会扣除成绩，有事及时向老师请假，旷课三分之一该次课程设计成绩为 0；白天要保证用 3 到 6 个小时编程准备，画流程图，讨论，查资料等。
2. 程序代码不能抄袭，对于雷同程序，不论抄与被抄都按 0 分计算；
3. 编程任务在 PTA 中提交**源程序**，并在**超星学习通**上提交**实验报告**，缺一不可。实验报告中包括：实验任务；数据流程图；源代码（要有详细注释）；心得体会等等。

# 第 1 天

## 1. 数组元素循环右移问题

一个数组  $A$  中存有  $N$  ( $>0$ ) 个整数，在不允许使用另外数组的前提下，将每个整数循环向右移  $M$  ( $\geq 0$ ) 个位置，即将  $A$  中的数据由  $(A_0 \ A_1 \ \dots \ A_{N-1})$  变换为  $(A_{N-M} \ \dots \ A_{N-1} \ A_0 \ A_1 \ \dots \ A_{N-M-1})$  (最后  $M$  个数循环移至最前面的  $M$  个位置)。如果需要考虑程序移动数据的次数尽量少，要如何设计移动的方法？

### 输入格式:

每个输入包含一个测试用例，第 1 行输入  $N$  ( $1 \leq N \leq 100$ ) 和  $M$  ( $\geq 0$ )；第 2 行输入  $N$  个整数，之间用空格分隔。

### 输出格式:

在一行中输出循环右移  $M$  位以后的整数序列，之间用空格分隔，序列结尾不能有多余空格。

### 输入样例:

```
6 2
1 2 3 4 5 6
```

### 输出样例:

```
5 6 1 2 3 4
```

## 2. 水仙花数

水仙花数是指一个  $N$  位正整数 ( $N \geq 3$ )，它的每个位上的数字的  $N$  次幂之和等于它本身。例如：  
 $153 = 1^3 + 5^3 + 3^3$ 。本题要求编写程序，计算所有  $N$  位水仙花数。

### 输入格式:

输入在一行中给出一个正整数  $N$  ( $3 \leq N \leq 7$ )。

### 输出格式:

按递增顺序输出所有  $N$  位水仙花数，每个数字占一行。

### 输入样例:

3

### 输出样例:

153

370

371

407

## 3. 求最大值及其下标

本题要求编写程序，找出给定的  $n$  个数中的最大值及其对应的最小下标（下标从 0 开始）。

### 输入格式:

输入在第一行中给出一个正整数  $n$  ( $1 < n \leq 10$ )。第二行输入  $n$  个整数，用空格分开。

### 输出格式:

在一行中输出最大值及最大值的最小下标，中间用一个空格分开。

### 输入样例:

6

2 8 10 1 9 10

### 输出样例:

10 2

## 4. 找完数

所谓完数就是该数恰好等于除自身外的因子之和。例如： $6=1+2+3$ ，其中 1、2、3 为 6 的因子。本题要求编写程序，找出任意两正整数  $m$  和  $n$  之间的所有完数。

### 输入格式：

输入在一行中给出 2 个正整数  $m$  和  $n$  ( $1 < m \leq n \leq 10000$ )，中间以空格分隔。

### 输出格式：

逐行输出给定范围内每个完数的因子累加形式的分解式，每个完数占一行，格式为“完数 = 因子 1 + 因子 2 + ... + 因子 k”，其中完数和因子均按递增顺序给出。若区间内没有完数，则输出“None”。

### 输入样例：

```
2 30
```

### 输出样例：

```
6 = 1 + 2 + 3
28 = 1 + 2 + 4 + 7 + 14
```

## 5. 梅森数

形如  $2^n - 1$  的素数称为梅森数 (Mersenne Number)。例如  $2^2 - 1 = 3$ 、 $2^3 - 1 = 7$  都是梅森数。1722 年，双目失明的瑞士数学大师欧拉证明了  $2^{31} - 1 = 2147483647$  是一个素数，堪称当时世界上“已知最大素数”的一个记录。

本题要求编写程序，对任一正整数  $n$  ( $n < 20$ )，输出所有不超过  $2^n - 1$  的梅森数。

### 输入格式：

输入在一行中给出正整数  $n$  ( $n < 20$ )。

### 输出格式：

按从小到大的顺序输出所有不超过  $2^n - 1$  的梅森数，每行一个。如果完全没有，则输出“None”。

### 输入样例：

6

### 输出样例：

3

7

31

## 6. 查验身份证

一个合法的身份证号码由 17 位地区、日期编号和顺序编号加 1 位校验码组成。校验码的计算规则如下：

首先对前 17 位数字加权求和，权重分配为：{7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2}；然后将计算的和对 11 取模得到值  $Z$ ；最后按照以下关系对应  $Z$  值与校验码  $M$  的值：

Z: 0 1 2 3 4 5 6 7 8 9 10

M: 1 0 X 9 8 7 6 5 4 3 2

现在给定一些身份证号码，请你验证校验码的有效性，并输出有问题的号码。

### 输入格式：

输入第一行给出正整数  $N$  ( $\leq 100$ ) 是输入的身份证号码的个数。随后  $N$  行，每行给出 1 个 18 位身份证号码。

### 输出格式：

按照输入的顺序每行输出 1 个有问题的身份证号码。这里并不检验前 17 位是否合理，只检查前 17 位是否全为数字且最后 1 位校验码计算准确。如果所有号码都正常，则输出 `All passed`。

### 输入样例 1：

4

320124198808240056

12010X198901011234

```
110108196711301866
37070419881216001X
```

### 输出样例 1：

```
12010X198901011234
110108196711301866
37070419881216001X
```

### 输入样例 2：

```
2
320124198808240056
110108196711301862
```

### 输出样例 2：

```
All passed
```

## 第 2 天

### 1. 简易连连看

本题要求实现一个简易连连看游戏模拟程序。

给定一个  $2N \times 2N$  的方阵网格游戏盘面，每个格子中放置一些符号。这些符号一定是成对出现的，同一个符号可能不止一对。程序读入玩家给出的一对位置  $(x_1, y_1)$ 、 $(x_2, y_2)$ ，判断这两个位置上的符号是否匹配。如果匹配成功，则将两个符号消为 “\*” 并输出消去后的盘面；否则输出 “Uh-oh”。若匹配错误达到 3 次，则输出 “Game Over” 并结束游戏。或者当全部符号匹配成功，则输出 “Congratulations!”，然后结束游戏。

### 输入格式：

输入在一行中给一个正整数  $N$  ( $\leq 5$ )。随后  $2N$  行，每行  $2N$  个大写英文字母（其间以 1 个空格分隔），表示游戏盘面。盘面之后给出一个正整数  $K$ ，随后  $K$  行，每行按照格式

" $x_1$   $y_1$   $x_2$   $y_2$ " 给出一个玩家的输入。注意格子的行、列编号是从 1 到  $2N$ 。

### 输出格式：

根据玩家的每一步输入，输出相应的结果。输出盘面时注意，每行字符间以 1 个空格分隔，行末不得有多余空格。

### 输入样例 1：

```
2
I T I T
Y T I A
T A T Y
I K K T
11
1 1 1 3
4 2 4 3
3 1 4 2
2 2 1 2
3 1 2 4
4 4 3 1
2 1 3 4
3 3 1 4
4 1 2 3
2 4 3 2
1 1 2 2
```

### 输出样例 1：



\* T \* T

Y T I A

T A T Y

I K K T

\* T \* T

Y T I A

T A T Y

I \* \* T

Uh-oh

\* \* \* T

Y \* I A

T A T Y

I \* \* T

Uh-oh

\* \* \* T

Y \* I A

\* A T Y

I \* \* \*

\* \* \* T

\* \* I A

\* A T \*

I \* \* \*

\* \* \* \*

\* \* I A

\* A \* \*

I \* \* \*

\* \* \* \*

\* \* \* A

\* A \* \*

\* \* \* \*

Congratulations!

## 输入样例 2 :

2

I T I T

Y T I A

T A T Y

I K K T

5

1 1 4 4

1 1 2 3

1 1 2 3

2 2 4 1

2 2 3 3

## 输出样例 2 :

Uh-oh

\* T I T

Y T \* A

T A T Y

I K K T

Uh-oh

Uh-oh

Game Over

## 2. 大炮打蚊子

现在，我们用大炮来打蚊子：蚊子分布在一个  $M \times N$  格的二维平面上，每只蚊子占据一格。向该平面的任意位置发射炮弹，炮弹的杀伤范围如下示意：

```
0
0X0
0
```

其中， $X$  为炮弹落点中心， $O$  为紧靠中心的四个有杀伤力的格子范围。若蚊子被炮弹命中（位于  $X$  格），一击毙命，若仅被杀伤（位于  $O$  格），则损失一半的生命力。也就是说，一次命中或者两次杀伤均可消灭蚊子。现在给出蚊子的分布情况以及连续  $k$  发炮弹的落点，给出每炮消灭的蚊子数。

## 输入格式:

第一行为两个不超过 20 的正整数  $M$  和  $N$ ，中间空一格，表示二维平面有  $M$  行、 $N$  列。

接下来  $M$  行，每行有  $N$  个  $0$  或者  $\#$  字符，其中  $\#$  表示所在格子有蚊子。

接下来一行，包含一个不超过 400 的正整数  $k$ ，表示发射炮弹的数量。

最后  $k$  行，每行包括一发炮弹的整数坐标  $x$  和  $y$ （ $0 \leq x < M$ ， $0 \leq y < N$ ），之间用一个空格间隔。

## 输出格式:

对应输入的  $k$  发炮弹，输出共有  $k$  行，第  $i$  行即第  $i$  发炮弹消灭的蚊子数。

## 输入样例:

```
5 6
00#00#
000###
00#000
000000
00#000
2
1 2
1 4
```

### 输出样例:

```
0
2
```

### 3. 验证“哥德巴赫猜想”

数学领域著名的“哥德巴赫猜想”的大致意思是：任何一个大于 2 的偶数总能表示为两个素数之和。比如：24=5+19，其中 5 和 19 都是素数。本实验的任务是设计一个程序，验证 20 亿以内的偶数都可以分解成两个素数之和。

#### 输入格式：

输入在一行中给出一个(2, 2 000 000 000]范围内的偶数 N。

#### 输出格式：

在一行中按照格式“N = p + q”输出 N 的素数分解，其中  $p \leq q$  均为素数。又因为这样的分解不唯一（例如 24 还可以分解为 7+17），要求必须输出所有解中 p 最小的解。

#### 输入样例：

```
24
```

#### 输出样例：

```
24 = 5 + 19
```

### 4. 到底是不是太胖了

据说一个人的标准体重应该是其身高（单位：厘米）减去 100、再乘以 0.9 所得到的公斤数。真实体重与标准体重误差在 10%以内都是完美身材（即  $|\text{真实体重} - \text{标准体重}| < \text{标准体重} \times 10\%$ ）。已知市斤是公斤的两倍。现给定一群人的身高和实际体重，请你告诉他们是否太胖或太瘦了。

### 输入格式：

输入第一行给出一个正整数  $N$  ( $\leq 20$ )。随后  $N$  行，每行给出两个整数，分别是一个人的身高  $H$  ( $120 < H < 200$ ；单位：厘米) 和真实体重  $W$  ( $50 < W \leq 300$ ；单位：市斤)，其间以空格分隔。

### 输出格式：

为每个人输出一行结论：如果是完美身材，输出 `You are wan mei!`；如果太胖了，输出 `You are tai pang le!`；否则输出 `You are tai shou le!`。

### 输入样例：

```
3
169 136
150 81
178 155
```

### 输出样例：

```
You are wan mei!
You are tai shou le!
You are tai pang le!
```

## 5. 求整数的位数及各个数字之和

对于给定的正整数  $N$ ，求它的位数及其各位数字之和。

### 输入格式：

输入在一行中给出一个不超过  $10^9$  的正整数  $N$ 。

### 输出格式：

在一行中输出  $N$  的位数及其各位数字之和，中间用一个空格隔开。

### 输入样例：

321

### 输出样例：

3 6

## 6. 谁先到

划拳是古老中国酒文化的一个有趣的组成部分。酒桌上两人划拳的方法为：每人口中喊出一个数字，同时用手比划出一个数字。如果谁比划出的数字正好等于两人喊出的数字之和，谁就输了，输家罚一杯酒。两人同赢或两人同输则继续下一轮，直到唯一的赢家出现。

下面给出甲、乙两人的酒量（最多能喝多少杯不倒）和划拳记录，请你判断两个人谁先倒。

### 输入格式：

输入第一行先后给出甲、乙两人的酒量（不超过 100 的非负整数），以空格分隔。下一行给出一个正整数  $N$  ( $\leq 100$ )，随后  $N$  行，每行给出一轮划拳的记录，格式为：

甲喊 甲划 乙喊 乙划

其中喊是喊出的数字，划是划出的数字，均为不超过 100 的正整数（两只手一起划）。

### 输出格式：

在第一行中输出先倒下的那个人：A 代表甲，B 代表乙。第二行中输出没倒的那个人喝了多少杯。题目保证有一个人倒下。注意程序处理到有人倒下就终止，后面的数据不必处理。

### 输入样例：

```
1 1
6
8 10 9 12
5 10 5 10
3 8 5 12
12 18 1 13
```

```
4 16 12 15
```

```
15 1 1 16
```

### 输出样例：

```
A
```

```
1
```

## 第 3 天

### 1. 约瑟夫环

要求用循环链表实现，N 个人围成一圈顺序编号，从 1 号开始按 1、2、3.....顺序报数，报 p 者退出圈外，其余的人再从 1、2、3 开始报数，报 p 的人再退出圈外，以此类推。请按退出顺序输出每个退出人的原序号。

#### 输入格式:

输入只有一行，包括一个整数 N( $1 \leq N \leq 3000$ )及一个整数 p( $1 \leq p \leq 5000$ )。

#### 输出格式:

按退出顺序输出每个退出人的原序号，数据间以一个空格分隔，但行尾无空格。

#### 输入样例:

在这里给出一组输入。例如：

```
7 3
```

#### 输出样例:

```
3 6 2 7 5 1 4
```

## 2. 井字棋

井字棋，英文名叫 Tic-Tac-Toe，是一种在 3\*3 格子上的连珠游戏，和五子棋类似，由于棋盘一般不画边框，格线排成井字故得名。游戏需要的工具仅为纸和笔，然后由分别代表 O 和 X 的两个游戏者轮流在格子里留下标记（一般来说先手者为 X），任意三个标记形成一条直线，则为获胜。（来自百度百科）

请你判断后手者是否存在任意三个标记形成一条直线？

### 输入格式:

第一行输入一个整数  $t(1 \leq t \leq 1000)$  代表  $t$  组输入。接下来每组输入一个井字棋游戏局面。

### 输出格式:

若后手者存在任意三个标记形成一条直线输出 "yes"，否则输出 "no"。

### 输入样例:

在这里给出一组输入。例如：

```
2
000
XXX
000

000
XX0
0XX
```

### 输出样例:

在这里给出相应的输出。例如：

```
yes
yes
```



### 3. 猜数字游戏

猜数字游戏是令游戏机随机产生一个 100 以内的正整数，用户输入一个数对其进行猜测，需要你编写程序自动对其与随机产生的被猜数进行比较，并提示大了（“Too big”），还是小了（“Too small”），相等表示猜到了。如果猜到，则结束程序。程序还要求统计猜的次数，如果 1 次猜出该数，提示 “Bingo!”；如果 3 次以内猜到该数，则提示 “Lucky You!”；如果超过 3 次但是在 N（>3）次以内（包括第 N 次）猜到该数，则提示 “Good Guess!”；如果超过 N 次都没有猜到，则提示 “Game Over”，并结束程序。如果在到达 N 次之前，用户输入了一个负数，也输出 “Game Over”，并结束程序。

#### 输入格式:

输入第一行中给出两个不超过 100 的正整数，分别是游戏机产生的随机数、以及猜测的最大次数 N。最后每行给出一个用户的输入，直到出现负数为止。

#### 输出格式:

在一行中输出每次猜测相应的结果，直到输出猜对的结果或 “Game Over” 则结束。

#### 输入样例:

```
58 4
70
50
56
58
60
-2
```

#### 输出样例:

```
Too big
Too small
```

Too small  
Good Guess!

#### 4. 稳赢

大家应该都会玩“锤子剪刀布”的游戏：两人同时给出手势，胜负规则如图所示：



现要求你编写一个稳赢不输的程序，根据对方的出招，给出对应的赢招。但是！为了不让对方输得太惨，你需要每隔  $K$  次就让一个平局。

#### 输入格式：

输入首先在第一行给出正整数  $K$  ( $\leq 10$ )，即平局间隔的次数。随后每行给出对方的一次出招：

`ChuiZi` 代表“锤子”、`JianDao` 代表“剪刀”、`Bu` 代表“布”。`End` 代表输入结束，这一行不要作为出招处理。

#### 输出格式：

对每一个输入的出招，按要求输出稳赢或平局的招式。每招占一行。

#### 输入样例：

```
2
ChuiZi
```

JianDao

Bu

JianDao

Bu

ChuiZi

ChuiZi

End

## 输出样例：

Bu

ChuiZi

Bu

ChuiZi

JianDao

ChuiZi

Bu

## 5. 德才论

宋代史学家司马光在《资治通鉴》中有一段著名的“德才论”：“是故才德全尽谓之圣人，才德兼亡谓之愚人，德胜才谓之君子，才胜德谓之小人。凡取人之术，苟不得圣人，君子而与之，与其得小人，不若得愚人。”

现给出一批考生的德才分数，请根据司马光的理论给出录取排名。

## 输入格式：

输入第一行给出 3 个正整数，分别为： $N$  ( $\leq 10^5$ )，即考生总数； $L$  ( $\geq 60$ )，为录取最低分数线，即德分和才分均不低于  $L$  的考生才有资格被考虑录取； $H$  ( $< 100$ )，为优先录取线——德分和才分均不低于此线的被定义为“才德全尽”，此类考生按德才总分从高到低排序；才分不到但德分到线的一类考生属于“德胜才”，也按总分排序，但排在第一类考生之后；德才分均低于  $H$ ，但是德分不

低于才分的考生属于“才德兼亡”但尚有“德胜才”者，按总分排序，但排在第二类考生之后；其他达到最低线 L 的考生也按总分排序，但排在第三类考生之后。

随后 N 行，每行给出一位考生的信息，包括：准考证号、德分、才分，其中准考证号为 8 位整数，德才分为区间 [0, 100] 内的整数。数字间以空格分隔。

### 输出格式：

输出第一行首先给出达到最低分数线的考生人数 M，随后 M 行，每行按照输入格式输出一位考生的信息，考生按输入中说明的规则从高到低排序。当某类考生中有多人总分相同时，按其德分降序排列；若德分也并列，则按准考证号的升序输出。

### 输入样例：

```
14 60 80
10000001 64 90
10000002 90 60
10000011 85 80
10000003 85 80
10000004 80 85
10000005 82 77
10000006 83 76
10000007 90 78
10000008 75 79
10000009 59 90
10000010 88 45
10000012 80 100
10000013 90 99
10000014 66 60
```

### 输出样例：

```
12
10000013 90 99
```

```
10000012 80 100
10000003 85 80
10000011 85 80
10000004 80 85
10000007 90 78
10000006 83 76
10000005 82 77
10000002 90 60
10000014 66 60
10000008 75 79
10000001 64 90
```

## 第 4、5 天

**简单通讯录程序：**编程实现简单通讯录程序。通讯者的基本信息包括编号、姓名、性别、年龄、宅电、手机电话等信息。要求用**结构体和类**分别实现。具体实现的管理功能如下：

- (1) 输入并显示多个通讯者的信息；
- (2) 可实现通讯者信息的添加；
- (3) 查询（至少提供按姓名和手机查询两种方式）；
- (4) 修改（只提供按照姓名进行修改）；
- (5) 删除（依据指定的信息删除如姓名、编号等）；
- (6) 文件的导入和导出（从文件中读取若干条通讯录，或者将通讯录输出到文件中）。

程序所显示的菜单内容如下：

请选择下面的一个项目：

- 1 创建
- 2 添加
- 3 查询
- 4 修改
- 5 删除
- 6 显示
- 7 从文件导入
- 8 导出到文件
- 9 退出

请选择：

## 第 6 天

**学生基本信息管理程序：**编程实现学生基本信息管理程序。学生基本信息包括：学号、姓名、性别、年龄、班级、学院、专业等。要求用**结构体和类**分别实现。具体实现的管理功能如下：

- (1) 输入并显示多个学生的基本信息；
- (2) 可根据需要实现学生信息的添加；
- (3) 查询（至少提供按学号查询、班级和专业的综合查询两种方式）；
- (4) 修改（只提供按照学号进行修改）；
- (5) 删除（依据指定的信息删除如姓名、学号等）；

（6）文件的导入和导出（从文件中读取若干条学生信息，或者将学生信息输出到文件中）。

程序所显示的菜单内容如下：

请选择下面的一个项目：

- 1 创建
- 2 添加
- 3 查询
- 4 修改
- 5 删除
- 6 显示
- 7 从文件导入
- 8 导出到文件
- 9 退出

请选择：

## 第 7 天

**学生成绩管理程序：**设计一个菜单驱动的学生成绩管理程序。输入学生的基本信息，包括学生的学号、姓名、数学、语文、英语等信息，要求用**结构体**和**类**分别实现。实现如下管理功能：

- （1）能输入并显示n个学生的基本信息。
- （2）计算学生的平均分和总分。
- （3）按总分降序进行排序。
- （4）任意输入一个学号，能显示该学生的基本信息。

(5) 可根据需要添加学生信息到指定位置，亦可在最后追加一批学生信息。

(6) 文件的导入和导出（从文件中读取若干条学生信息，或者将学生信息输出到文件中）。

输出菜单形式如下：

学生成绩管理系统：

- 1 创建
- 2 添加
- 3 汇总
- 4 排序
- 5 查询
- 6 显示
- 7 从文件导入
- 8 导出到文件
- 9 退出

请选择：

## 第 8 天

空间点、线和面：

实现三维解析几何中的点、直线和平面类，



- (1) 能够实现直线的不同创建方式（例如，两个点确定一条直线，两个相交的平面确定一条直线，空间曲线的点斜式）和平面的不同创建方式（例如，三个不共线的点确定一个平面，一个点和一个法向量确定一个平面）；
- (3) 能够计算相应的距离：两点之间的距离，点到直线的距离，点到平面的距离；
- (4) 能够计算空间直线的单位方向向量（长度为1），空间平面的单位法向量（长度为1）；
- (5) 能够判断点和线的关系，线和线的关系，点和平面的关系，线和平面的关系，平面和平面的关系。

#### 提示：

- 点、直线和平面分别用类来封装，每个类的数据为该表示该类需要的参数，例如，对于空间平面的一般方程为 $ax+by+cz+d=0$ ，那么私有数据成员为 $a, b, c$ 和 $d$ ；
- 空间直线和平面的不同创建方式定义为对应类的成员函数；
- 空间点和点、点和直线、点和平面、直线和直线、直线和平面、平面和平面位置关系的判断定义为一般的函数；
- 要在`main`函数中对上述功能进行调用和验证。

## 第9天

**整数链表：**仔细阅读相应的例子，掌握链表的创建，插入节点，删除节点，遍历和查询，并用类进行封装。

4 5 6

1 2 3

## 第 10 天

**简单通讯录程序：**利用**链表实现简单通讯录程序**。通讯者的基本信息包括编号、姓名、性别、年龄、宅电、手机电话等信息。具体实现的管理功能如下：

- (1) 输入并显示多个通讯者的信息；
- (2) 可实现通讯者信息的添加；
- (3) 查询（至少提供按姓名和手机查询两种方式）；
- (4) 修改（只提供按照姓名进行修改）；
- (5) 删除（依据指定的信息删除如姓名、编号等）；
- (6) 文件的导入和导出（从文件中读取若干条通讯录，或者将通讯录输出到文件中）。

程序所显示的菜单内容如下：

请选择下面的一个项目：

- 1 创建
- 2 添加
- 3 查询
- 4 修改
- 5 删除
- 6 显示
- 7 从文件导入
- 8 导出到文件

9 退出

请选择: