



# 大学计算机基础 80 题编程题

数学学院励志书院数学试验班师生联合党支部

主编：陈嵘汉（23）

2025.12



数学与统计学院  
SCHOOL OF MATHEMATICS AND STATISTICS



励志书院  
LIZHI COLLEGE

西安交通大学数学学院励志书院  
数学试验班师生联合党支部



## 写在前面的话

本讲义是数学学院励志书院数学试验班师生联合党支部的助学讲义之一，由支部党员陈嵘汉同志编写，所有代码基本均为陈嵘汉同志大一学习该课程时保留的代码（除 8、13、14 周部分题目没有保留，参考了支部入党积极分子胡可瑜同志的代码）。数试支部党员同志及助学组同志们负责审校。本讲义涵盖了大计基 80 题作业题全部题目的详解。每个题目除了给出了源代码，还附上了较为详细的解题思路，重点语句给出了批注。

特别注意：大部分题目我没有写

1

```
#define _CRT_SECURE_NO_WARNINGS
```

因为我本地用的是 dev cpp 进行编译，无需这句话。如果读者用的本地编译器是 visual studio 的话，是需要加这句话的。

此外，在不引起歧义的情况下，笔者有时候习惯使用下标从 1-n 而非 0-n-1，本质上是一样的，请读者明辨。笔者还习惯于开全局变量，这样默认变量的初值为 0，可以避免非常多的问题。

## 目录

|          |                                          |           |
|----------|------------------------------------------|-----------|
| <b>1</b> | <b>第 7 周</b>                             | <b>4</b>  |
| 1.1      | C005-编写程序完成对应数的输出                        | 4         |
| 1.2      | C006-编写程序，输入用户当月利润，输出发放奖金总数              | 4         |
| 1.3      | C017-按要求输出两个数的和                          | 5         |
| 1.4      | C042-平均数计算                               | 5         |
| 1.5      | C043-3 个小数求和                             | 6         |
| 1.6      | C044-和差积商                                | 6         |
| 1.7      | C046-时、分、秒转换                             | 7         |
| 1.8      | C048-计算银行存款本息                            | 7         |
| 1.9      | C049-输出相邻的字母串                            | 8         |
| 1.10     | C053-输出字母的 ASCII 码和后继字符                  | 8         |
| <b>2</b> | <b>第 8 周</b>                             | <b>9</b>  |
| 2.1      | C001-输出平方根                               | 9         |
| 2.2      | C003-输出对应的英文星号                           | 9         |
| 2.3      | C004-编写程序完成数据的加密                         | 10        |
| 2.4      | C007-找出 m 到 n 之间的所有同构数                   | 10        |
| 2.5      | C011-判断任意一个正整数各位数字之和是奇数还是偶数              | 11        |
| 2.6      | C012-求级数前 n 项的和                          | 11        |
| 2.7      | C014-找出给定范围内是 7 的倍数或带 7 的全部正整数           | 12        |
| 2.8      | C016-求勾股数                                | 13        |
| 2.9      | C018-计算一个正整数各位数字的平方和                     | 13        |
| 2.10     | C050-检验由三边能否构成三角形                        | 14        |
| <b>3</b> | <b>第 9 周</b>                             | <b>15</b> |
| 3.1      | C060-统计整数的个数                             | 15        |
| 3.2      | C013-将一个大于 4 的正整数分解为连续的正整数之和             | 15        |
| 3.3      | C031-求 $a+aa+aaa+aaaa+\cdots+aa\ldots a$ | 16        |
| 3.4      | C032-水仙花数                                | 17        |
| 3.5      | C102-求一组数中的最大值并统计出现次数                    | 18        |
| 3.6      | C105-按字典顺序寻找最大和最小单词                      | 19        |
| 3.7      | C088-将字符串首部的 * 号全部移到字符串的尾部               | 20        |



|          |                                          |           |
|----------|------------------------------------------|-----------|
| 3.8      | C052-将整数分解为 1 和各个质因子的相乘 . . . . .        | 21        |
| 3.9      | C054-输出分数序列前 n 项之和 . . . . .             | 22        |
| 3.10     | C056-月份天数计算 . . . . .                    | 23        |
| <b>4</b> | <b>第 10 周</b>                            | <b>25</b> |
| 4.1      | C126-判断一个正整数是否为回文数 . . . . .             | 25        |
| 4.2      | C129-统计其中各个英文字母的出现次数 . . . . .           | 25        |
| 4.3      | C077-找到连续出现最多的字符 . . . . .               | 26        |
| 4.4      | C038-闰年计算 . . . . .                      | 27        |
| 4.5      | C081-按规则输出倒序字符串 . . . . .                | 28        |
| 4.6      | C030-单词加密问题 . . . . .                    | 29        |
| 4.7      | C086-按规则去掉字符串末尾多余的星号 . . . . .           | 30        |
| 4.8      | C144-字符串中查找字符位置 . . . . .                | 31        |
| 4.9      | C115-统计选票 . . . . .                      | 31        |
| 4.10     | C156-除首尾字符外降序排序 . . . . .                | 32        |
| <b>5</b> | <b>第 11 周</b>                            | <b>34</b> |
| 5.1      | C059-数组循环移动 . . . . .                    | 34        |
| 5.2      | C122-计算函数 $\text{ack}(m,n)$ 的值 . . . . . | 34        |
| 5.3      | C061-最大公约数 . . . . .                     | 35        |
| 5.4      | C127-求多项式的值 . . . . .                    | 36        |
| 5.5      | C069-将数字替换为字母 . . . . .                  | 36        |
| 5.6      | C070-求数组交集 . . . . .                     | 37        |
| 5.7      | C095-输出数组中长度最大的等值数列段的始末下标 . . . . .      | 38        |
| 5.8      | C009-求小球第 n 次落地的距离和反弹高度 . . . . .        | 39        |
| 5.9      | C039-电梯运行时间计算 . . . . .                  | 40        |
| 5.10     | C041-小明的计算 . . . . .                     | 40        |
| <b>6</b> | <b>第 12 周</b>                            | <b>42</b> |
| 6.1      | C120-去掉一个字符串中除头部和尾部空格外的所有空格 . . . . .    | 42        |
| 6.2      | C067-整数三位分节 . . . . .                    | 43        |
| 6.3      | C093-字符串中奇数下标的字符顺序颠倒 . . . . .           | 43        |
| 6.4      | C096-删除串中的重复字符 . . . . .                 | 44        |
| 6.5      | C072-删除字符 . . . . .                      | 45        |
| 6.6      | C084-统计第 2 个字符串在第 1 个字符串中出现的次数 . . . . . | 46        |
| 6.7      | C085-按规则去掉星号 . . . . .                   | 47        |
| 6.8      | C113-求给定区间中的孪生数的数量 . . . . .             | 48        |
| 6.9      | C119-二进制形式的 IP 地址转为十进制 . . . . .         | 48        |
| 6.10     | C066-打点滴时间计算 . . . . .                   | 50        |
| <b>7</b> | <b>第 13 周</b>                            | <b>51</b> |
| 7.1      | C123-电话簿排序 . . . . .                     | 51        |
| 7.2      | C124-学生成绩排序 . . . . .                    | 52        |
| 7.3      | C064-翻译数字 . . . . .                      | 53        |
| 7.4      | C074-统计字母次数 . . . . .                    | 54        |
| 7.5      | C076-按规则排序 . . . . .                     | 55        |
| 7.6      | C134-最大最小和平均值 . . . . .                  | 56        |
| 7.7      | C110-删除字符串中所有非小写字母字符 . . . . .           | 57        |



|          |                                    |           |
|----------|------------------------------------|-----------|
| 7.8      | C145-词频统计 . . . . .                | 58        |
| 7.9      | C148-按字母序升序输出字符串 . . . . .         | 59        |
| 7.10     | C158-按要求输出奇数位置的字符排序后的字符串 . . . . . | 60        |
| <b>8</b> | <b>第 14 周</b>                      | <b>62</b> |
| 8.1      | C036-K 进制数的转换 . . . . .            | 62        |
| 8.2      | C037-乘积中各位数字组成的最大数 . . . . .       | 62        |
| 8.3      | C136-签到和签退 . . . . .               | 63        |
| 8.4      | C103-出圈问题 . . . . .                | 65        |
| 8.5      | C087-按规则删除星号 . . . . .             | 66        |
| 8.6      | C106-新兵报数问题 . . . . .              | 67        |
| 8.7      | C107-输出不是两个数组共有的元素 . . . . .       | 68        |
| 8.8      | C142-分数求和 . . . . .                | 70        |
| 8.9      | C111-提取字符串中的整数并输出 . . . . .        | 71        |
| 8.10     | C146-按 ASCII 排序输出字符串 . . . . .     | 73        |



## 1 第 7 周

### 1.1 C005-编写程序完成对应数的输出

**题目描述 1.** 请输入一个不含 0 的 8 位的十进制整数，编写程序取出该整数的中间 4 位数，分别输出取出的这 4 位数以及该 4 位数加上 1024 的得数。

输入：一个整数。

输出：两个整数，用空格分隔。

**思路.** 考察拆分数字思想，具体来说，对于一个 8 位数  $n$ ，如果我们要取出从右往左第一位，那就直接  $n\%10$ ；如果是从右往左第二位，先除以 10 再模 10（即  $n/10\%10$ ）；如果是从右往左第三位，先除以 100 再模 10（即  $n/100\%10$ ）……以此类推即可，具体实现见代码。 □

```
1 #include<stdio.h>
2 int main()
3 {
4     int n; scanf("%d",&n);
5     int a3=n/100000%10;
6     int a4=n/10000%10;
7     int a5=n/1000%10;
8     int a6=n/100%10;
9     int a=a3*1000+a4*100+a5*10+a6;
10    printf("%d%d%d%d %d",a3,a4,a5,a6,a+1024);
11    return 0;
12 }
```

### 1.2 C006-编写程序，输入用户当月利润，输出发放奖金总数

**题目描述 2.** 企业发放的奖金根据利润提成。利润低于或等于 10 万元时，奖金可提 10%；

利润高于 10 万元，低于等于 20 万元时，低于 10 万元的部分按 10% 提成，高于 10 万元的部分，可提成 8%；

20 万到 40 万之间时，高于 20 万元的部分，可提成 5%；40 万到 60 万之间时高于 40 万元的部分，可提成 3%；

60 万到 100 万之间时，高于 60 万元的部分，可提成 2%，高于 100 万元时，超过 100 万元的部分按 1% 提成。

编写程序，输入用户当月利润，输出发放奖金总数。

输入：一个整数。

输出：一个整数。

**思路.** 一个暴力 if 判断的题，没有很特别的技巧，此类题需要仔细研读题目意思并一点点耐心地解答，注意细节，防止出错。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int m,ans;
4 int main()
```



```
5 {  
6     scanf("%d",&m);  
7     if(m<=100000) ans=m*0.1;  
8     else if(m<=200000) ans=100000*0.1+(m-100000)*0.08;  
9     else if(m<=400000) ans=100000*0.18+(m-200000)*0.05;  
10    else if(m<=600000) ans=100000*0.18+200000*0.05+(m-400000)*0.03;  
11    else if(m<=1000000) ans=100000*0.18+200000*0.08+(m-600000)*0.02;  
12    else ans=100000*0.18+200000*0.08+400000*0.02+(m-1000000)*0.01;  
13    printf("%d",ans);  
14    return 0;  
15 }
```

### 1.3 C017-按要求输出两个数的和

**题目描述 3.** 输入整数  $a$  和  $b$ ，若  $a*a+b*b$  大于等于 100，则输出  $a*a+b*b$  百位及以上的数字，否则输出  $a*a+b*b$ 。例如，输入 25 25，它们的平方和为  $625+625=1250$ ，百位及以上数为 12。

输入：两个整数，中间用空格隔开

输出：一个整数。

样例输入：13 50 样例输出：26

思路. 这题也是比较容易的 if 语句的应用，见代码。

□

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int a,b,c;  
5     scanf("%d %d",&a,&b);  
6     if(a*a+b*b>=100) c=(a*a+b*b)/100;  
7     else if(a*a+b*b<100) c=a*a+b*b;  
8     printf("%d",c);  
9     return 0;  
10 }
```

### 1.4 C042-平均数计算

**题目描述 4.** 从键盘输入任意 3 个整数，然后输出这 3 个数的平均值。

输入：3 个整数，用空格分隔。

输出：平均数（实数，提示：输出平均值时请使用格式符 “%g”）。

样例输入：4 5 6 样例输出：5

思路. 这题考察的是变量的类型。题目告诉我们三个数是整数（%d），但是最后的答案是实数（%f），所以涉及到一个非常非常非常重要的事情：在 C 语言中， $1/2 = 0$ ，但是  $1.0/2.0 = 0.5$

对应到这个题目，题目要求我们求平均值，也就是说  $(a+b+c)/3$ ，但是由于  $a, b, c$  都是 int 类型的整数，所以这个表达式也是一个整数（自动强制取整了），所以事实上我们要写的是  $(a+b+c)/3.0$ ，这才表示一个实数，可以赋值给 float 类型的变量

还有一个小知识点，%g 表示的是“自动删除小数末尾的 0”，由于是小数，所以自然是 float 类型才能使用 %g

□



```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     double a,b,c;
6     scanf("%lf%lf%lf",&a,&b,&c);
7     printf("%g", (a+b+c)/3.0);
8     return 0;
9 }
```

### 1.5 C043-3 个小数求和

**题目描述 5.** 编写程序，输入任意的 3 个小数；将这 3 个小数相加，将相加的结果以及按四舍五入方法转换成整数后的结果显示输出。

输入：3 个小数，空格分隔；

输出：和，四舍五入后的整数，1 行显示，空格分隔。

样例输入：1.1 2.2 3.3

样例输出：6.6 7

思路. x 四舍五入的技巧是：(int)(x+0.5)。

可以自己代入具体数字理解一下（分别代小数部分小于 5、等于 5、大于 5 三类型的数字，如 1.3,2.5,5.6） □

```
1 #include<stdio.h>
2 int main()
3 {
4     float a,b,c;
5     scanf("%f %f %f",&a,&b,&c);
6     float d;
7     d=a+b+c;
8     float e;
9     e=(int)(a+b+c+0.5);
10    printf("%g %g",d,e);
11    return 0;
12 }
```

### 1.6 C044-和差积商

**题目描述 6.** 输入 2 个整数，求：和，差，积，商

输入：2 个整数（第二个数是非零整数）

输出：4 个整数，依次为和、差、积和商，数据间用空格分隔。

样例输入：2 3

样例输出：5 -1 6 0

思路. 注意在整数做商的时候是整除，比如 2/3 是 0。 □

```
1 #include<stdio.h>
2 int main()
```



```
3 {
4     int a,b;
5     scanf("%d%d",&a,&b);
6     printf("%d %d %d %d",a+b,a-b,a*b,a/b);
7     return 0;
8 }
```

## 1.7 C046-时、分、秒转换

**题目描述 7.** 输入总秒数，转换为相应的时、分、秒。提示：在输出时，请使用下面的格式（变量名不要求一样）`printf("`

输入：整数

输出：时分秒，整数，西文冒号分隔。

思路. 没什么特别的难点，这题就是一个简单的时分秒转换。

□

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,h,m,s;
5     scanf("%d",&a);
6     h=a/3600;
7     m=(a-h*3600)/60;
8     s=a-h*3600-m*60;
9     printf("%d:%.2d:%.2d\n", h, m, s);
10 }
```

## 1.8 C048-计算银行存款本息

**题目描述 8.** 计算银行存款本息。输入存款金额 *money*（单位：元），存期 *years*，年利率 *rate*，计算到期存款本息（保留 2 位小数）

输入：存款金额，存期，年利率。均为浮点数，且用逗号分隔

输出：存款本息（保留 2 位小数）

思路. 这题考察浮点数的运算。一般开 `double`（高精度，保留小数位数较多）会相对比 `float` 更保险。不过要注意，前者用 `%ld`，后者用 `%d`。

□

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     double a,sum,money,year,rate;
6     scanf("%lf,%lf,%lf",&money,&year,&rate);
7     a=pow(1+rate,year);
8     sum=a*money;
9     printf("%.2lf",sum);
10     return 0;
11 }
```



## 1.9 C049-输出相邻的字母串

**题目描述 9.** 输入一个大写英文字母，输出与其相邻的字母串（将字母 A-Z 看作是收尾相接的环状排列）。

输入：大写英文字母

输出：该字母前面的字母，该字母，该字母后面的字母。（三字母间无间隔）

思路：先把 A 和 Z 特殊判断一下，其他的用 +1 和 -1 控制即可

□

```
1 #include<stdio.h>
2 int main()
3 {
4     char a, b,c;
5     b=getchar();
6     if(b==A) printf("ZAB");
7     else if(b==Z) printf ("YZA");
8     else
9     {
10         a=b-1;
11         c=b+1;
12         printf("%c%c%c",a,b,c);
13     }
14     return 0;
15 }
```

## 1.10 C053-输出字母的 ASCII 码和后继字符

**题目描述 10.** 输入由 4 个字母构成的英文单词，按如下格式输出字母，该字母的 ASCII 码以及该字母的后继字符。每行输出数据之间用一个空格分隔。

输入：单词

输出：共 4 行，每行输出字母，该字母的 ASCII 码，该字母后继字符（空格分隔，每行的最后没有空格）

思路：这道题考察如何输出字符对应的 ascll 码：对于字符类型（%c）变量，直接使用%d，即可输出 ASCLL 码。至于后一位，使用%c 类型，输出内容为 s[i]+1 即可（注意与 s[i+1] 区分！s[i+1] 指的是字符串中第 i+1 位置的字符，而题目要求的后继字符是指比如 a 的后继是 b，d 的后继是 e）

□

```
1 #include <stdio.h>
2 #include<string.h>
3 char s[10];
4 int main()
5 {
6     scanf("%s",s+1);
7     int n =strlen(s+1);
8     for(int i=1;i<=n;i++)
9         printf("%c %d %c\n",s[i],s[i],s[i]+1);
10    return 0;
11 }
```



## 2 第 8 周

### 2.1 C001-输出平方根

**题目描述 11.** 从键盘输入一个整数  $x$ ，输出其平方根（为整数）。若输入数大于 1000 或小于等于 0，则输出 0，并继续接受下一个输入直至小于等于 1000。

**思路.** 计算平方根的函数是 sqrt，在头文件 math.h 下面。这题需要注意的一个点是，12,13 两行不能合起来写成 printf("%d",sqrt(a)); 不然会报错!!（建议自己试一下加深印象） □

```
1 #include<stdio.h>
2 #include<math.h>
3 int main ()
4 {
5     int a,b;
6     scanf("%d",&a);
7     while(a>1000 || a<=0)
8     {
9         printf("0\n");
10        scanf("%d",&a);
11    }
12    b=sqrt(a);
13    printf("%d",b);
14    return 0;
15 }
```

### 2.2 C003-输出对应的英文星号

**题目描述 12.** 从键盘输入 3 个整数（三个数取值均在闭区间  $[1,10]$ ），每行输入一个  
每读取一个值，程序打印出该值个数的英文星号 (\*)。

输入：3 个整数，3 行。

输出：3 行星号。

**思路.** 简单 for 循环控制即可 □

```
1 #include<stdio.h>
2 int n;
3 int main()
4 {
5     for(int j=1; j<=3; j++)
6     {
7         scanf("%d",&n);
8         for(int i=1; i<=n; i++)
9             printf("*"); //输出 n 个 *
10        printf("\n");
11    }
12    return 0;
13 }
```



## 2.3 C004-编写程序完成数据的加密

**题目描述 13.** 某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：

1. 每位数字都用自身加 5 后的和除以 10 的余数代替，如 1 用 6 代替，8 用 3 代替；如 3768 就转换为 8213；

2. 再将第一步生成的数字的第一位和第四位交换，第二位和第三位交换。如 8213 转换为 3128，即为最终的加密密码。

编写程序完成数据的加密。输入：一个整数。输出：一个整数。

**思路.** 把每一位数字取出来，然后 +5 后再 mod10 即可。至于交换，只要在输出的时候改变输出顺序即可。 □

```
1 #include<stdio.h>
2 int a;
3 int main()
4 {
5     scanf("%d", &a);
6     int e = (a/1000 + 5) % 10;
7     int f = (a/100 + 5) % 10;
8     int g = (a/10 + 5) % 10;
9     int h = (a%10 + 5) % 10;
10    printf("%d%d%d%d\n", h, g, f, e);
11    return 0;
12 }
```

## 2.4 C007-找出 m 到 n 之间的所有同构数

**题目描述 14.** 找出整数  $m$  到整数  $n$  (闭区间) 之间的所有同构数 ( $m < n$ )。

说明：一个正整数  $x$ ，如果是它平方数的尾部，则称  $x$  为同构数。例如，6 是其平方数 36 的尾部，25 是其平方数 625 的尾部，那么 6 和 25 都是同构数。

输入： $m$  和  $n$  两个正整数，用空格分隔。

输出：连续输出同构数，数据间用空格分隔 (最后一个同构数后面无空格)。

**思路.** 先循环遍历  $m$  至  $n$  之间的每个数，判断是否为同构数。具体判断方法：对于数字  $i$ ，首先要知道  $i$  有多少位，比如  $i$  有三位，那么就要用  $i*i$  模掉 1000，如果  $i$  有五位，就要用  $i*i$  模掉 100000，以此类推  $i$  有  $k$  位，就要用  $i*i$  模掉  $10^k$ 。然后判断模完之后的数字是否等于  $i$ ，相等的话就说明  $i$  是一个同构数。 □

```
1 #include<stdio.h>
2 #include<math.h>
3 int a,b,ans;
4 int get_num(int x)//计算x有多少位数
5 {
6     int res=0;
7     while(x){
8         x = x/10;
9         res+=1;
10    }
11    return res;
```



```
12 }
13 int main()
14 {
15     scanf("%d %d", &a, &b);
16     for(int i=a; i<=b; i++)
17     {
18         int s=i*i;
19         int k=get_num(i);
20         if(s%(int)pow(10,k)==i) //平方模掉 10 的 k 次方等于 i, 说明是平方数
21         {
22             ans+=1;
23             if(ans==1) printf("%d",i); //第一次直接输出, 第二次开始先输出空格再输出数字
24             else printf(" %d",i); //这样可以方便地实现最后一个无空格的效果
25         }
26     }
27     return 0;
28 }
```

## 2.5 C011-判断任意一个正整数各位数字之和是奇数还是偶数

题目描述 15. 编程判断任意一个正整数各位数字之和是奇数还是偶数。

如果和是奇数输出 1, 偶数输出 0。

思路. 取出每一位然后求和即可。

□

```
1 #include<stdio.h>
2 int a, sum;
3 int main()
4 {
5     scanf("%d", &a);
6     while(a > 0)
7     {
8         sum += a % 10;
9         a /= 10;
10    }
11    printf("%d",sum%2);
12    return 0;
13 }
```

## 2.6 C012-求级数前 n 项的和

题目描述 16. 编程求以下级数前 n 项之和:

$$s=1-1/3+1/5-1/7+1/9-1/11+1/13-1/15+.....$$

输入: 一个正整数 n

输出: 前 n 项和的值, 超出小数点后 4 位的, 保留到小数点后 4 位

思路. 一定一定要注意 9,10 两行写的是 1.0 而不是 1! 在浮点数意义下, 前者是会报错的。

□



```
1 #include<stdio.h>
2 int k;
3 double sum;
4 int main()
5 {
6     scanf("%d",&k);
7     for(int i=1;i<=k;i++)
8     {
9         if (i%2==1) sum=sum+1.0/(2*i-1);
10        else sum=sum-1.0/(2*i-1);
11    }
12    printf ("%lf",sum);
13    return 0;
14 }
```

## 2.7 C014-找出给定范围内是 7 的倍数或带 7 的全部正整数

题目描述 17. 编写程序，找出  $[m,n]$  范围内是 7 的倍数或带 7 的全部正整数 (注意：如果一个既是 7 的倍数又是带 7 的数，就输出 2 次，先输出倍数，再输出带 7 的整数，如：7 is a multiple of 7 7 contains 7)。  
其中， $m$  和  $n$  为正整数。

思路. 按照题意解答即可。

□

```
1 #include<stdio.h>
2 int m,n;
3 int contain(int num)
4 {
5     while(num>0)
6     {
7         if(num%10==7) return 1;//某一位是7，直接返回真，后面的while不执行了
8         num/=10;
9     }
10    return 0;
11 }
12 int main()
13 {
14     scanf("%d %d",&m,&n);
15     for(int i=m; i<=n; i++)
16     {
17         if(i%7==0) printf("%d is a multiple of 7\n", i);
18         if(contain(i)) printf("%d contains 7\n", i);
19     }
20    return 0;
21 }
```



## 2.8 C016-求勾股数

**题目描述 18.** 若三个正整数  $a, b, c$  满足  $a^2 + b^2 = c^2$ , 则它们是一组勾股数。

编写程序, 求给定区间  $[m, n]$  中的勾股数的数量 (设一组勾股数满足  $a < b < c$ )。

例如  $[1, 10]$  中的勾股数有  $(3, 4, 5)$  和  $(6, 8, 10)$ , 则  $[1, 10]$  中勾股数的数量为 2。

输入: 正整数  $m, n$ , 空格分隔

输出:  $[m, n]$  中的勾股数的数量

思路. 暴力使用三重循环枚举  $a$ 、 $b$ 、 $c$ , 然后判断是否满足勾股方程即可。

□

```
1 #include<stdio.h>
2 int m,n,ans;
3 int main()
4 {
5     scanf("%d %d",&m,&n);
6     for(int a=m; a<=n; a++)
7     {
8         for(int b=a+1; b<=n; b++)
9             for(int c=b+1; c<=n; c++)
10                 if(a*a + b*b == c*c)
11                     ans += 1;
12     }
13     printf("%d\n", ans);
14     return 0;
15 }
```

## 2.9 C018-计算一个正整数各位数字的平方和

**题目描述 19.** 输入任意一个正整数, 计算各位数字的平方和。

如: 1234 则: 计算  $1*1+2*2+3*3+4*4=30$

输入: 一个正整数

输出: 平方和

思路. 按照题意解答即可。

□

```
1 #include<stdio.h>
2 int n,ans;
3 int main()
4 {
5     scanf("%d", &n);
6     while(n)//n 指的是 n>0, 这两种写法等价
7     {
8         int a = n%10;
9         ans += a*a;
10        n /= 10;
11    }
12    printf("%d\n", ans);
13    return 0;
14 }
```



14 }

## 2.10 C050-检验由三边能否构成三角形

**题目描述 20.** 编写程序检验由三边能否构成三角形，检验方法是任意两边和均要大于第三边。

输入：三边长度（一行输入，空格分隔）

输出：如果可以构成三角形，输出 *YES*，否则输出 *ERROR DATA*

思路：注意输入可以是实数。

□

```
1 #include <stdio.h>
2 float a, b, c;
3 int main()
4 {
5     scanf("%f %f %f", &a, &b, &c);
6     if(a+b>c && a+c>b && b+c>a) printf("YES\n");
7     else printf("ERROR DATA\n");
8     return 0;
9 }
```



### 3 第 9 周

#### 3.1 C060-统计整数的个数

**题目描述 21.** 输入若干个整数，其值在 0 至 4 的范围内，统计每个整数的个数。

输入：[0,4] 内的整数组成的序列，用空格隔开，以 -1 终止。

输出：0,1,2,3,4 的统计个数，用一个空格隔开（最后一个输出后面没有空格）。

样例输入：3 3 3 2 2 1 0 3 4 4 -1

样例输出：1 1 2 4 2

**思路.** 注意审题，题目中说输入将以 -1 终止，但是何时出现 -1 我们是不知道的。这就符合了使用 while 循环的条件之一（即不知道循环的次数时，往往使用 while 循环）。所以我们使用 while 循环，退出循环的条件为  $x == -1$ ，所以 while 括号里面的条件应该写  $x != -1$ ，意思是只要满足  $x != -1$ ，while 循环就会一直执行下去。当然不要忘记，最开始需要先输入一次 x 的值（第 5 行），不然的话，第一次就无法判断 x 与 -1 的关系了。 □

```
1 #include<stdio.h>
2 int x,k,b,c,d,e;
3 int main()
4 {
5     scanf("%d",&x);
6     while(x!=-1)
7     {
8         if(x==0) k=k+1;
9         if(x==1) b=b+1;
10        if(x==2) c=c+1;
11        if(x==3) d=d+1;
12        if(x==4) e=e+1;
13        scanf("%d",&x);
14    }
15    printf("%d %d %d %d %d",k,b,c,d,e);
16    return 0;
17 }
```

#### 3.2 C013-将一个大于 4 的正整数分解为连续的正整数之和

**题目描述 22.** 将一个大于 4 的正整数分解为连续的正整数之和，请显示全部分解结果。

如数字 15，可以分解为：

$$15=1+2+3+4+5$$

$$15=4+5+6$$

$$15=7+8$$

样例输入：6

样例输出：6=1+2+3

**思路.** 题目说要求输出所有的把给定的数字 n 分解成连续几个数字之和的结果。很自然的一个想法是，我们可以枚举这连续一串数字中最小的那个。比如我们用 i 表示循环变量（当然这里是 for 循环，因为起点和终点都知道了），我们尝试把 i 开始的连续 k 个数字  $i, i+1, i+2, \dots, i+k-1$  全部加起来，看看它会



不会等于  $n$ 。

但是这里有一个显然的问题： $k$  是多少？何以得知  $k$  是多少？——这就涉及到一个很重要的思想，人不知道的事情，就交给 `while` 循环。具体实现方法见下方代码的批注。□

```
1 #include<stdio.h>
2 int n,sum;
3 int main()
4 {
5     scanf("%d",&n);
6     for(int i=1; i<=n-1; i++)//枚举一串数字的起点 i，思考：为啥是 n-1?
7     {
8         sum=0;//这个 sum 每次都要清空！因为对于不同的 i，我们每次都要重新求这个 sum
9         int k=i;//用 k 先记录一下 i 的值，后续 k 要在 while 循环里面不断增加
10        //注意这里不能直接让 i 去增加，所以我们需要用临时变量 k 来记录
11        while(sum<n) //循环将在 sum 大于等于 n 的时候结束
12        {
13            sum=sum+k;//sum 每次加上 k
14            k=k+1;//k 每次增加 1，表示从 i 开始后面连续的一堆数字
15        }
16        if(sum==n) //退出 while 循环时，要么 sum 大于 n，要么 sum==n，
17            //后者就是我们想要的
18        {
19            printf("%d=",n);
20            for(int j=i; j<=k-1; j++)//思考一下为啥是 k-1，这个很重要
21            {
22                if(j==i) printf("%d",j);//类似于之前用 cnt 判断是否为第一个数
23                else printf("+%d",j);
24            }
25            printf("\n");
26        }
27    }
28    return 0;
29 }
```

### 3.3 C031-求 $a+aa+aaa+aaaa+\cdots+aa\dots a$

**题目描述 23.** 求  $a+aa+aaa+aaaa+\cdots+aa\dots a$  ( $n$  个)，其中  $a$  为 1 9 之间的整数。

例如：当  $a=1, n=3$  时，求  $1+11+111$  之和为 123；

输入：组成序列的数字  $a$  和求和项的数量  $n$

输出：算式和结果。

样例输入：2 7

样例输出：2+22+222+2222+22222+222222+2222222=2469134

**思路.** 这个题在写的时候最关键的一个点是，如何从 22 变成 222？

首先一个自然的想法，肯定是要用循环的，不然的话一个一个去写实在是太麻烦了

其实可以这样考虑：我们记  $a=22, b=222$ ，那么其实  $b=a*10+2$

同理，有  $2222=222*10+2, 22222=2222*10+2$ ，



也就是说，每次下一个数，等于当前数字  $\times 10 + 2$

如果用代码来写，就是  $t = t \times 10 + a$ ，这里的  $a$  在样例里面就是 2， $t$  用来记录当前的数字是多少，具体看代码

□

```
1 #include<stdio.h>
2 int a,t,n,ans;
3 int main()
4 {
5     scanf("%d%d",&a,&n);
6     t=a; ans=a;
7     printf("%d",a);
8     for(int i=1; i<=n-1; i++)
9     {
10         t=t*10+a;//这句话是精髓，具体见上面的分析
11         printf("+%d",t);
12         ans+=t;//ans 用于记录我们要的和，是 answer 的简写
13     }
14     printf("=%d",ans);//注意格式，别忘了这里要输出一个 = 符号
15     return 0;
16 }
```

### 3.4 C032-水仙花数

**题目描述 24.** “水仙花数”是指一个三位正整数，其各位数字的立方和刚好等于该数本身，如： $153 = 1^3 + 5^3 + 3^3$

输入  $n, m$ ， $100 \leq n, m < 1000$ ，求出  $[n, m]$  之间的水仙花数。若该区间没有水仙花数，输出 -1。

输入： $n, m$ ，用空格隔开。

输出：若干水仙花数，用空格隔开（最后一位输出后面无空格）。

样例输入：100 500

样例输出：153 370 371 407

**思路.** 这题的关键还是如何取出一个三位数的最低位，中间位，最高位。分别是  $x \% 10$ ， $x / 10 \% 10$ ， $x / 100$ ，当然，还需要掌握的是  $n$  位数的分割，之前已经出现过了，不妨复习一下。

□

```
1 #include <stdio.h>
2 int n,m,flag,cnt;
3 int pd(int i)
4 {
5     int a,b,c;
6     a=i%10;
7     b=i/10%10;
8     c=i/100;//这三行分别取出个位，十位，百位
9     if(i==a*a*a+b*b*b+c*c*c)
10     {
11         flag=1;
12         return 1;
13     }
```



```
14     return 0;
15 }
16 int main ()
17 {
18     scanf("%d %d",&n,&m);
19     for(int i=n; i<=m; i++)
20     {
21         if (pd(i)==1)
22         {
23             cnt++;
24             if (cnt==1) printf("%d",i);//用 cnt 判断是不是第一个，很重要
25             else printf(" %d",i);
26         }
27     }
28     if (flag!=1) printf("-1");//flag 标记思想，复习一下
29     return 0;
30 }
31 }
```

### 3.5 C102-求一组数中的最大值并统计出现次数

**题目描述 25.** 求一组数 (整数数组) 中的最大值, 然后统计该最大值在这组数中出现的次数。

输入: 元素个数  $n$  和  $n$  个整数。 $n < 200$ 。(要先输入个数  $n$ , 然后再输入  $n$  个数)

输出: 最大值及出现的次数。数据间以一个逗号隔开。

样例输入: 1 2

样例输出: 2,1

**思路.** 这个题还是相对比较容易的。我们用  $\text{maxv}$  表示最大值, 每次把输入的  $a[i]$  与  $\text{maxv}$  做比较, 如果说  $a[i]$  超过了  $\text{maxv}$ , 那么就更新  $\text{maxv}$  的值 (即把  $a[i]$  的值赋值给  $\text{maxv}$ )。当全部输入完之后, 我们再做一次 for 循环, 遍历整个数组, 用  $\text{cnt}$  记录  $a[i] == \text{maxv}$  的  $a[i]$  的个数即可。□

```
1 #include<stdio.h>
2 int n,a[105],maxv,cnt;
3 int main()
4 {
5     scanf("%d",&n);
6     for(int i=1; i<=n; i++)
7     {
8         scanf("%d",&a[i]);
9         if(a[i]>maxv) maxv=a[i]; //超过最大值, 就更新最大值
10    }
11    for(int i=1; i<=n; i++)
12    {
13        if(a[i]==maxv) cnt+=1;
14    }
15    printf("%d,%d",maxv,cnt); //cnt 记录最大值的个数
```



```
16     return 0;
17 }
```

### 3.6 C105-按字典顺序寻找最大和最小单词

**题目描述 26.** 定义包含 5 个英文单词的字符数组 (即二维字符数组), 键盘输入 5 个单词 (一个单词最多 80 个字符), 按字典顺序寻找最大和最小单词并输出。

输入: 5 个单词

输出: 最大和最小单词 (按字典序)

如: *one two three four five*

*max:two min:five* (注: *max* 和 *min* 后是英文冒号)

样例输入: *this is a book end* 样例输出: *max:this min:a*

**思路.** 本题涉及的知识点有点小多, 一个一个分开来看:

首先当然还是读题, 它说定义了一个二维字符数组, 那这时候就要复习一下二维数组的相关内容。题干已经告诉我们了一共是 5 个单词, 也就是说二维数组的第一维的长度是 5。又告诉我们每个单词的长度不超过 80, 也就是说第二维的长度是 80, 那么我们就可以定义一个二维数组叫做 `a[6][85]`, 注意我特地都写的大一些, 防止报错。

接着看题目, 要求按照字典序作比较, 这里又涉及到一个知识点: C 语言有一个函数叫做 `strcmp`, 它属于头文件 `string.h` 下面的一个函数, 它的用法是: 格式为 `strcmp(a,b)`, 表示比较 `a` 和 `b` 两个字符串。返回值为一个整数, `a` 大于 `b` (按照字典序比较) 时返回值大于 0, `a` 小于 `b` 时返回值为小于 0, `a==b` 时返回值为 0。

还有一个难点是, 字符串不方便直接赋值, 所以我们只能先记录最大的那个字符串的位置 (1-5 中某一个数字), 最小的那个字符串的位置 (1-5 中某一个数字), 具体看代码里面的注释更清晰。

另外, 还要注意一点: 字符串的下标默认从 0 开始, 所以在用 `for` 循环遍历字符串的时候, 往往是 `for(i=0; i<=len(字符串)-1; i++)` □

```
1  #include<stdio.h>
2  #include<string.h>
3  char maxpos,minpos;//分别用来记录最大和最小的字符串的下标 (0 至 4 某个数)
4  char a[6][85];
5  int main()
6  {
7      for(int i=0; i<=4; i++) scanf("%s",a[i]);//输入的 a[i] 是一个字符串
8      maxpos=0;//maxposition 的缩写
9      minpos=0;//先让 maxpos=minpos=0
10     for(int i=1; i<=4; i++)//接下去判断 1 到 4 这几个下标的字符串有没有比 1 更大 (小) 的
11     {
12         if(strcmp(a[i],a[maxpos])>0)//当前位置比最大位置还大
13             maxpos=i;
14         if(strcmp(a[i],a[minpos])<0)//当前位置比最小位置还小
15             minpos=i;
16     }
17     printf("max:");
18     for(int i=0; i<=strlen(a[maxpos])-1; i++) printf("%c",a[maxpos][i]);
19     printf(" ");//别忘了中间要输出一个空格
20     printf("min:");
```



```
21     for(int i=0; i<=strlen(a[minpos])-1; i++) printf("%c",a[minpos][i]);
22     return 0;
23 }
```

### 3.7 C088-将字符串首部的 \* 号全部移到字符串的尾部

**题目描述 27.** 将字符串首部的 \* 号全部移到字符串的尾部，中间若有 \* 号，保持中间的 \* 号不动。字符串长度不超过 200。

样例：

\*\*\*street\*\*\*\*\*music\*\*\*

street\*\*\*\*\*music\*\*\*\*\*

样例输入：stree\*\*\*\*\*tmusic

样例输出：stree\*\*\*\*\*tmusic

**思路.** 这个题比上个题简单多了，我们只要知道最前面到底有几个 \*，然后先输出这之后的所有内容，再输出一堆 \* 就可以了。

具体来说，我们先 for 循环遍历字符串，如果当前位置是 \*，那么计数器 ++；否则就直接 break（break 的用法下一段说）。由于字符串下标从 0 开始，所以假设起 cnt 最终的值为 3，那么 0,1,2 三个位置是 \*，第 3 个位置开始不是 \*，所以我们让 pos 表示“第一个不为 \* 的位置”，则有 pos=cnt

这里要特别提一下 break，break 意思是强制退出 for 循环，比如 i=3 的时候，满足了第一个不是 \*，那么这时候记录完 pos 和 cnt，就会执行 break，也就是说 i=4 开始的循环就不会继续了，直接结束整个循环

循环结束之后就只需要从 pos 这个位置开始到字符串末尾，先把这一段的字符输出，再输出 cnt 个 \*，即可达到题目的要求。 □

```
1  #include<stdio.h>
2  #include<string.h>
3  char a[205]; //长度不超过 200，所以开 205
4  int pos,cnt;
5  int main()
6  {
7      scanf("%s",a);
8      int n=strlen(a);
9      for(int i=0; i<=n-1; i++) //先找到第一个满足 a[i] 不为 * 的位置
10     {
11         if(a[i]==*) cnt++;
12         else break;
13     }
14     pos=cnt; //pos 记录第一个位置的下标
15     for(int i=pos; i<=n-1; i++) //从 pos 位置开始，把后面的全输出
16     {
17         printf("%c",a[i]);
18     }
19     for(int i=1; i<=cnt; i++) //再输出 cnt 个 *
20         printf("*");
21     return 0;
22 }
```



再来看一种更好的思路。

**思路.** 考虑一个一个字符读入，然后来判断有多少个 \*。这样就很好地省去了全部读入之后，再用 for 循环判断有多少个 \* 的环节。我们只需要用 getchar() 一个个读入（配合 while 循环），就可以很快知道到底有多少个 \* 了。具体看代码里面的注释 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int cnt;
4 char s[105];
5 int main()
6 {
7     char c=getchar();//getchar() 是读入一个字符，赋值给 c
8     while(c==*)//注意这里 * 上面是有引号的，但是不知道为什么 latex 里面一时半会打不出来
9     {
10         cnt++;//cnt 用来记录前面 * 的数量
11         c=getchar();//继续输入后面的字符，赋值给 c
12     }
13     printf("%c",c);//想一下为啥这里要单独输出一个 c（提示：举例子模拟一下 while 循环）
14     gets(s);//后面剩下的直接整行读入
15     for(int i=0; i<strlen(s); i++) printf("%c",s[i]);//剩下的全部一口气输出
16     for(int i=1; i<=cnt; i++) printf("*");//最后输出 cnt 个 *
17     return 0;
18 }
```

### 3.8 C052-将整数分解为 1 和各个质因子的相乘

**题目描述 28.** 从键盘输入一个正整数 ( $> 1$ )，然后将该整数分解为 1 和各个质因子的相乘，如果输入的整数本身就是质数，则应分解为 1 和该数本身相乘。

如：18

$1*2*3*3$

**思路.** 这个题有点数学知识的背景，根据数论中的算数基本定理：任意一个正整数，都可以被唯一分解为一堆素数的幂次的乘积举个例子，

$$12 = 2^2 * 3$$

$$48 = 2^4 * 3$$

所以我们这个题可以这么来做：for 循环遍历 2 至  $n-1$  的所有素数  $i$ （注意是素数，这里涉及到判断素数的那个函数怎么写，建议熟练掌握甚至如有必要可以看考虑背会），并且如果这个  $i$  还满足  $n\%i==0$ ，那么这个  $i$  就是所谓的  $n$  的“素因子”。接下去我们只要让  $n$  不断地除以  $i$ ，直至把  $i$  这个素因子全部除尽，具体的实现用的是 while 循环， □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n,flag;
4 int judge(int x) //判断素数函数，强烈建议熟练理解并默写
5 {
6     for(int i=2; i<=x-1; i++)
```



```
7         if(x%i==0) return 0;
8         //如果有某一个 i 被 x 整除了,说明 x 肯定不是素数了,直接返回 0。
9         //函数调用中的 return 是指直接返回 return 后面的值并退出函数,后面都不执行了
10        return 1; //如果直到循环结束都还没有返回值,那就说明确实是素数,返回 1
11    }
12    int main()
13    {
14        scanf("%d",&n);
15        printf("1");//先输出一个 1,这是题目的格式要求
16        for(int i=2; i<=n-1; i++)//注意范围,从 2 开始,到根号 n
17        {
18            if(judge(i) && n%i==0)
19            {
20                flag=1;//flag 标记为 1,说明 n 是一个合数
21                int k=n;//想一下,为啥要用 k 先记录一下 n?
22                while(k%i==0)//让 n 一直除以 i,直至除不了为止
23                {
24                    printf("%d",i);
25                    k=k/i;
26                }
27            }
28        }
29        if(flag==0) //如果 n 是素数,还需要输出 n
30            printf("%d",n);
31        return 0;
32    }
```

### 3.9 C054-输出分数序列前 n 项之和

**题目描述 29.** 有一分数序列如下:  $2/1, 3/2, 5/3, 8/5, 13/8, 21/13, \dots$  即后一项的分母为前一项的分子,后项的分子为前一项分子与分母之和,求其前  $n$  项之和(从第 1 项算起)。

输入: 项数  $n$  (整数)

输出: 前  $n$  项之和(实数,注:保留 4 位小数)

样例输入: 6

样例输出: 10.0071

**思路.** 注意一些细节: 一个是定义  $a$  和  $b$  的初值的时候,应该是 1.0 和 2.0 (所以开的是 float),而不是 1 和 2。否则第 12 行就会错。

还有一个是  $sum$  作为一个求和遍历,必须必须必须初值为 0!!! (两种写法,一种是定义成全局变量,默认初值为 0,还有一种是定义为局部变量,人为写上  $float\ sum=0;$ )。□

```
1 #include<stdio.h>
2 int main ()
3 {
4     int n;
5     float a,b,c;
6     a=1.0;
```



```
7      b=2.0;//注意是 1.0 和 2.0!!
8      float sum=0,k;//注意初值一定要是 0!!
9      scanf("%d",&n);
10     while(n>0)
11     {
12         k=b/a;
13         c=a;
14         a=b;
15         b=c+b;
16         n=n-1;
17         sum=sum+k;
18     }
19     printf("%.4f",sum);
20     return 0;
21 }
```

### 3.10 C056-月份天数计算

**题目描述 30.** 输入公元年份和月份，输出该月份的天数。

输入：年，月（逗号分隔）

输出：天数  
样例输入：2018,8 样例输出：31

**思路.** 这个题就一个核心：判断闰年。这和“判断素数”一样，也是必须必须掌握（甚至可以背代码）的内容。就一句话：如果一个年份，它（是 4 的倍数且不是 100 的倍数）或（是 400 的倍数）那么它就是一个闰年。下面的代码没有单独写一个判断闰年函数，建议可以单独写，这样更清晰。 □

```
1  #include<stdio.h>
2  int main ()
3  {
4      int year,month;
5      scanf ("%d,%d",&year,&month);
6      if((year%4==0&&year%100!=0&&month==2)|| (year%400==0&&month==2))//闰年的 2 月
7      {
8          printf("29");
9      }
10     else if(month==2)//非闰年的 2 月
11     {
12         printf("28");
13     }
14     else if(month==1||month==3||month==5||month==7||month==8||month==10||
15             month==12)
16         //非 2 月，且为 31 天的月份
17     {
18         printf("31");
19     }
20     else printf("30");//非 2 月，且为 30 天的月份
21     return 0;
```



|    |   |
|----|---|
| 21 | } |
|----|---|



## 4 第 10 周

### 4.1 C126-判断一个正整数是否为回文数

**题目描述 31.** 编写一个函数来判断一个正整数是否为回文数，若是则返回 1，否则返回 0。所谓回文数是指各位数字左右对称的数，例如 1221、3553 等。该函数的原型为：

`int ispalindrome(int n);`

其中参数  $n$  是待判断的正整数，该函数有返回结果。

使用该函数找出 1000  $n$  (包括 1000 和  $n$ ,  $1000 \leq n < 10000$ ) 之间的所有回文数，按从小到大的次序在屏幕上显示输出，每个数之间用一个空格分隔，最后一个数后面没有空格。

样例输入：1200 样例输出：1001 1111

**思路.** 注意看题目中  $n$  的范围，这个范围告诉我们  $n$  一定是一个四位数，那么这个问题就变得非常简单。我们用 for 循环遍历 1000 到  $n$  的所有整数，对于每一个整数，判断它是否为回文数只要判断是否满足“第一位 == 第四位并且第二位 == 第三位”即可 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n,cnt;
4 int ispalindrome(int n)//根据题意，n 一定是一个四位数
5 {
6     int a=n%10;
7     int b=n/10%10;
8     int c=n/100%10;
9     int d=n/1000;
10    if(a==d && b==c) return 1;
11    else return 0;
12 }
13 int main()
14 {
15     scanf("%d",&n);
16     for(int i=1000; i<=n; i++)
17     {
18         if(ispalindrome(i))
19         {
20             cnt++;
21             if(cnt==1) printf("%d",i);
22             else printf(" %d",i);
23         }
24     }
25     return 0;
26 }
```

### 4.2 C129-统计其中各个英文字母的出现次数

**题目描述 32.** 输入一个可能带空格字符的字符串 (长度不超过 100)，统计其中各个英文字母的出现次数，不区分大小写。输出字母  $a \sim z$  的出现次数，数据间以英文逗号分隔。非英文字母不统计。

输入：可能带空格的字符串。



输出：26 个整数，以英文逗号分隔。

样例输入：World's oldest drink enjoys new success

样例输出：0,0,2,3,4,0,0,0,1,1,1,2,0,3,3,0,0,2,6,1,1,0,2,0,1,0

思路. 这题的最关键的地方是，用  $s[i] - 'a' + 1$  这个表达式，可以把一个小写字母  $s[i]$  变成  $1 - 26$  中的一个数字。具体的可以代入 'b', 'c', 'd' 来看看结果。当然，也可以不做这个平移，直接写  $\text{int } x = s[i]$ ；那么这样得到的  $x$  将是  $97 - 122$  中的一个数字（ASCLL 码），这样好处是避开了抽象的  $s[i] - 'a' + 1$ ，坏处是要记住  $a - z$  的 ASCLL 码为  $97 - 122$ 。下面给出的是前一种方法的代码

还有一个小细节，题干里说了字符串可能会包含空格，那么就不能使用  $\%s$  读入（ $\%s$  只能读入一个单词，一旦出现空格，后面的就读入不聊了）。这种情况别忘了我们还有非常厉害的  $\text{gets}$ ，可以整行读入。

注意：不知道为啥代码 11,13,16,19 行的单引号被吞掉了，应该为 'a', 'z', 'A', 'Z'。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105];
4 int n,cnt[30];
5 int main()
6 {
7     gets(s);
8     n=strlen(s);
9     for(int i=0; i<=n-1; i++)
10    {
11        if((s[i]>=a && s[i]<=z))//如果 s[i] 是小写字母
12        {
13            int id=s[i]-a+1;
14            cnt[id]+=1;
15        }
16        if(s[i]>=A && s[i]<=Z)//如果 s[i] 是大写字母
17        {
18            char c=s[i]+32;//大写转小写
19            int id=c-a+1;
20            cnt[id]+=1;
21        }
22    }
23    printf("%d",cnt[1]);
24    for(int i=2; i<=26; i++)
25        printf(",%d",cnt[i]);
26    return 0;
27 }
```

### 4.3 C077-找到连续出现最多的字符

题目描述 33. 编写程序，要求在一个字符串中查找连续出现次数最多的一个字符，并显示其所在的开始下标和次数。（如果出现最多的字符不止一个，输出最靠前的字符）

输入：一个字符串（字符个数不超过 100，无空格）

输出：出现最多的字符，次数，开始索引（这三个值之间用空格分隔，末尾无空格）



样例输入: *xyxyzzzqqq1q122334*

样例输出: *z 3 4*

思路. 如果当前的字符等于上一个,  $\text{cnt}++$  (它等价于  $\text{cnt}+=1$ )。如果  $\text{cnt}$  超过了最大值  $\text{maxv}$ , 那就做一系列更新: 更新  $\text{maxv}$ , 更新  $\text{lastpos}$ , 更新字符  $\text{cc}$ 。(这三条是题目要求输出的东西, 所以需要这时候更新) 否则, 如果当前字符不等于上一个,  $\text{cnt}$  变成 1 (重新开始)。□

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105],cc;
4 int n,cnt,maxv,lastpos;
5 int main()
6 {
7     gets(s);
8     n=strlen(s);
9     cc=s[0]; cnt=1;
10    for(int i=1; i<=n-1; i++)
11    {
12        if(s[i]==s[i-1])
13        {
14            cnt++;
15            if(cnt>maxv)//如果 cnt 超过了 maxv 最大值 maxv, 那就做三个更新
16            {
17                cc=s[i];
18                maxv=cnt;
19                lastpos=i;
20            }
21        }
22        else cnt=1;//否则, 如果当前字符不等于上一个, cnt 变成 1 (重新开始)。
23    }
24    printf("%c %d %d",cc,maxv,lastpos-maxv+1);
25    return 0;
26 }
```

#### 4.4 C038-闰年计算

题目描述 34. 闰年计算。程序输入一个正整数  $Y$ , 以及另一个正整数  $N$ , 以一个空格分隔。计算从  $Y$  年开始后的第  $N$  个闰年是哪一年 (如果  $Y$  本身是闰年, 则  $Y$  之后的第一个闰年是  $Y$ , 闰年不超过 6240 年)。

输入格式: 两个整数:  $Y$  和  $N$ 。用空格分隔

输出个数: 一个整数

思路. 闰年计算, 说来说去还是那一句话  
(是 4 的倍数且不是 100 的倍数) 或 (是 400 的倍数) □

```
1 #include<stdio.h>
2 int pd(int Y)//这次单独写了判断闰年的函数
3 {
```



```
4      if((Y%4==0 && Y%100!=0) || (Y%400==0) )
5          return 1;//建议不要赌各个符号的运算优先级，老老实实加括号不会错
6      else return 0;
7  }
8  int main()
9  {
10     int Y,N;scanf("%d %d",&Y,&N);
11     if(pd(Y)==1)
12     {
13         N=N-1;
14         while(N>0)
15         {
16             Y=Y+1;
17             if(pd(Y)==1) N=N-1;
18         }
19         printf("%d",Y);
20     }
21     else
22     {
23         while (N>0)
24         {
25             Y=Y+1;
26             if(pd(Y)==1) N=N-1;
27         }
28         printf("%d",Y);
29     }
30     return 0;
31 }
```

#### 4.5 C081-按规则输出倒序字符串

**题目描述 35.** 编写程序，按下列规则倒序输出子字符串。先输出最后的一个字符，再输出最后两个字符串，在再出后面三个字符...，最后输出整个字符串。输入的字符串长度不超过 100。

输入：一个字符串（无空格，字符个数不超过 100）

输出：空格隔开的子字符串，用一个空格间隔。

输入输出样例：

*student*

*t nt ent dent udent tudent student*

思路. 用 while 循环枚举起点 k，注意字符串的下标是 0 n-1

□

```
1 #include<stdio.h>
2 #include<string.h>
3 int n,k;
4 int main()
5 {
```



```
6   char s[105];
7   gets(s); //使用整行读入
8   n=strlen(s);
9   while(k<n)
10  {
11      k=k+1;
12      if(k==1) printf("%c",s[n-1]);
13      else
14      {
15          printf(" ");
16          for(int i=n-k; i<=n-1; i++) printf("%c",s[i]);
17      }
18  }
19  return 0;
20 }
```

#### 4.6 C030-单词加密问题

**题目描述 36.** 单词加密。输入一个字符串和一个非负整数  $k$ ，对字符串中的每一个字母，用字母表中其后的第  $k$  个字母代替，不够  $k$  个时再从字母  $a$  循环计数。例如  $k=3$  是， $a$  用  $d$  代替， $A$  用  $D$  代替， $x$  用  $a$  代替， $y$  用  $b$  代替，保持大小写不变。字符串中的非字母字符不变。字符串的长度不超过 100。

输入：一个字符串（无空格）和非负整数  $k$ ，之间用空格分隔

输出：加密的字符串。

样例输入：Beijing 9

样例输出：Knrswrp

**思路.** 大写字母 ASCII 从 65 开始到 90 结束（不是 91），小写字母从 97 开始到 122 结束（不是 123），不要记错了。还是之前的问题，latex 引用代码的时候好像一下子打不出来引号，我回头研究研究。11 行的 'a'、'z'，17 行的 'A'、'Z' 都是有引号的。 □

```
1  #include<stdio.h>
2  #include<string.h>
3  char s[105];
4  int k,i,n;
5  int main ()
6  {
7      scanf("%s %d",s,&k);
8      n=strlen(s);
9      for(i=0;i<=n-1;i++)
10     {
11         if(s[i]>=a&&s[i]<=z)
12         {
13             int id=s[i];
14             if((id+k)<=122) printf("%c",id+k);
15             //判断是否溢出，若溢出的话要从头开始，所以下一行减去 26 就是这个意思
16             else printf("%c",id+k-26);
```



```
17     }
18     else if(s[i]>=A&& s[i]<=Z)
19     {
20         int id=s[i];
21         if((id+k)<=90) printf("%c",id+k);
22         else printf("%c",id+k-26);
23     }
24     else printf("%c",s[i]);
25 }
26 return 0;
27 }
```

#### 4.7 C086-按规则去掉字符串末尾多余的星号

**题目描述 37.** 编写程序，去掉字符串末尾多余的星号。输入带星号（\*）的字符串和  $n$ ，使字符串尾部的 \* 号不得多于  $n$  个；若多于  $n$  个，则删除多余的 \* 号；若少于或等于  $n$  个，则什么也不做，字符串中间和前面的 \* 号不删除。字符串的长度不超过 200。字符串中的星号是英文星号。

输入：一个字符串（无空格，字符串长度不超过 100）和一个非负整数，中间用空格隔开。

输出：去掉多余 \* 号的字符串

**思路.** 千万注意这题的输入方法，注意到第一行有一个空格，空格前面是一个字符串，后面是一个整数，所以不能使用 gets 读入（因为 gets 是整行读入），所以这里只能使用 scanf 读入这个字符串。这里要特别注意 scanf 读入字符串的时候，是不能使用 “&” 的。

然后这题整体的思路是简单的，先通过倒着循环（选择用 while），来算出末尾有几个 \*，记为  $k$ 。再分  $k$  大于等于  $a$  和  $k$  小于  $a$  两种情况讨论，分别根据题意输出即可 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int a,n,k,i;
4 int main ()
5 {
6     char s[105];
7     scanf("%s %d",s,&a);
8     n=strlen(s);
9     i=n-1;
10    while(s[i]==*)//注意这里的写法
11    {
12        k++;
13        i--;
14    }
15    if(k>=a)
16    {
17        for(i=0;i<=n-1-k+a;i++)
18        {
19            printf("%c",s[i]);
```



```
20     }
21 }
22 else for(i=0;i<=n-1;i++)printf("%c",s[i]);
23 return 0;
24 }
```

#### 4.8 C144-字符串中查找字符位置

**题目描述 38.** 查找一个字符在字符串中出现的第一个位置并输出这个位置。位置从 0 开始。

输入输出格式：

输入：待查找的字符串（字符个数不超过 100）和需要查找的字符，字符串中可能含有空格。

（输入时，待查找的字符串与所需查找的字符用 \* 号隔开）如：“待查找字符串 \* 需要查找的字符”

输出：字符的位置（如有多个相同的字符，只查找第一个，如果没有输出 -1。）样例输入：fbhby Wml\*W

样例输出：6

**思路.** 这题根据题意，不得不使用 gets 输入，输入进来后，最后一位是要查的，倒数第二位是 \*，1 至倒数第三位是字符串。如果找到了就 break。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105];
4 int n,flag=0;
5 int main ()
6 {
7     gets(s);
8     n=strlen(s);
9     for(int i=0; i<=n-3; i++)
10         if(s[i]==s[n-1])
11         {
12             printf("%d",i);
13             flag=1; break;
14         }
15     if(flag==0) printf("-1");
16     return 0;
17 }
```

#### 4.9 C115-统计选票

**题目描述 39.** 统计选票。三个候选人分别是 Li、Zhang 和 Wang，Li 的代号是 1；Zhang 的代号是 2；Wang 的代号是 3。

依次输入代表得票人代号的数字（即投票），直到输入 -1 则投票结束。然后统计每个人的得票数 and 废票数。不是 -1,1,2,3 的数字为废票。

输入：若干整数，最后一个数是 -1。（输入的每个数之间有空格）

输出：四个整数，以空格隔开，代表三个人的 Li、Zhang、Wang 的得票数以及废票数。

输出格式：“%d %d %d %d”

**思路.** 注意这题使用的一个小技巧：一直输入直到输入 -1 结束，注意看这题的代码是怎么实现这个功能的。 □



```
1 #include<stdio.h>
2 int a,e,b,c,d,i;
3 int main()
4 {
5     scanf("%d",&a);
6     while(a!=-1)//一直输入直到输入-1 结束
7     {
8         if(a==1) e++;
9         else if(a==2) b++;
10        else if(a==3) c++;
11        else d++;
12        scanf("%d",&a);//这一行很关键，不要漏下了
13    }
14    printf("%d %d %d %d\n",e,b,c,d);
15    return 0;
16 }
```

#### 4.10 C156-除首尾字符外降序排序

**题目描述 40.** 输入一个字符串（长度不超过 200，不包含空格，至少有 1 个字符），除首尾字符外，将其余的字符按 *ascii* 码降序排列，然后输出。

样例输入: *announced*

样例输出: *auonnnecd*

**思路.** 首先，注意到这个题整行读入了一个字符串，所以我们使用 *gets* 进行读入，“去掉首尾”的方法很简单，只需要让 *i* 从 1 开始，到 *n*-2 结束即可

然后我们来考虑如何把中间这段按照 *ASCLL* 码倒序输出，这里用到了一个“计数思想”，我们先用 *cnt* 数组把每个字符出现的次数都记录下来，比如：*cnt*[105] 表示 *ascll* 码值为 105 的那个字符出现的次数，更一般地，*cnt*[*i*] 表示 *ascll* 码值为 *i* 的那个字符出现的次数。这样干完之后，我们得到了去掉首尾这段里面，每一个字符出现的次数。然后我们从 200 开始倒着做一遍 *for* 循环，把 *ASCLL* 码从 305（只要大于 256 理论上都是可以的，多写一点保险）开始倒着走一遍，如果当前 *cnt*[*i*] 是 3，就说明 *i* 这个数字对应的字符出现了三次，我们就输出三次这个字符。更一般的，如果当前 *cnt*[*i*]=*k*，我们就输出 *k* 次 *i* 对应的字符。具体代码如下： □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n,k;
4 char s[205];
5 int cnt[305];
6 int main()
7 {
8     gets(s);
9     int n=strlen(s);
10    for(int i=1; i<=n-2; i++)
11    {
```



```
12     int id=s[i]; //id 是 s[i] 对应的 ASCLL 码值
13     cnt[id]+=1; //表示数字 id (也就是字符 s[i]) 出现的次数 +1
14 }
15 printf("%c",s[0]);
16 for(int i=305; i>=1; i--)
17 //倒着遍历一遍 305 1, 因为 ASCLL 码一共有 256 个, 多一点保险一点
18 {
19     if(cnt[i]>0) //如果当前字母出现次数不为 0
20     {
21         for(int j=1; j<=cnt[i]; j++) //输出 cnt[i] 个 i 对应的字符
22             printf("%c",i); //直接用%c 就能输出数字 i 对应的字符
23     }
24 }
25 printf("%c",s[n-1]);
26 return 0;
27 }
```



## 5 第 11 周

### 5.1 C059-数组循环移动

**题目描述 41.** 设数组  $A$  中存有  $n$  ( $n > 0$ ) 个整数, 在不允许使用另外数组的前提下, 将  $A$  中的每个整数循环右移  $m$  ( $m \geq 0$ ) 个位置。即: 将  $A$  中的数据由  $(A_0 A_1 \cdots A_{n-1})$  变换为  $(A_{n-m} \cdots A_{n-1} A_0 A_1 \cdots A_{n-m-1})$ 。

要求: 输入  $n$  ( $1 \leq n \leq 100$ )、 $m$  ( $m \geq 0$ ) 及  $n$  个整数, 输出循环右移  $m$  位以后的整数序列。

输入: 占二行, 其中

第 1 行:  $n$  和  $m$  两个数,  $n$  和  $m$  之间用空格分隔。

第 2 行: 数组  $A$  中的  $n$  个数, 各数之间用空格分隔 (第一个数之前和最后一个数之后无空格)。

输出: 占一行, 数据间用一个空格分隔 (第一个数之前和最后一个数之后无空格)。

样例:

10 5

1 2 3 4 5 6 7 8 9 0

6 7 8 9 0 1 2 3 4 5

**思路.** 这题出现一些小 bug。题目里说的是“右移  $m$  位”, 但按照答案反推 (申明: 根据 23 年的题库, 我不确定现在修正 bug 了没), 这题事实上是“左移”。左移  $m$  位之后, 可以发现, 只要先输出  $m+1$  至  $n$ , 再输出 1 至  $m$  即可 (举例子 1 2 3 4 5 6, 左移 2 位, 模拟一下即可), 因此代码如下:  $\square$

```
1 #include<stdio.h>
2 int n,m,cnt;
3 int a[105];
4 int main ()
5 {
6     scanf("%d%d",&n,&m);
7     for(int i=1; i<=n; i++) scanf("%d",&a[i]);
8     for(int i=m+1; i<=n; i++)
9     {
10         cnt++;
11         if(cnt==1) printf("%d",a[i]);
12         else printf(" %d",a[i]);
13     }
14     for(int i=1; i<=m; i++)
15     {
16         cnt++;
17         if(cnt==1) printf("%d",a[i]);
18         else printf(" %d",a[i]);
19     }
20     return 0;
21 }
```

### 5.2 C122-计算函数 $ack(m,n)$ 的值

**题目描述 42.** 编程计算函数  $ack(m,n)$  的值。 $m,n$  的定义域是非负整数 ( $m \leq 3, n \leq 9$ )。

当  $m=0$  时,  $ack(0,n)=n+1$



当  $n=0$  时,  $ack(m,0)=ack(m-1,1)$

其它情况下,  $ack(m,n)=ack(m-1,ack(m,n-1))$

思路. 递归就是自己调用自己。这个题只要按照题意一步步往下写就可以了

□

```
1 #include<stdio.h>
2 int m,n;
3 int ack(int m,int n)
4 {
5     if(m==0) return n+1;
6     if(n==0) return ack(m-1,1);
7     return ack(m-1,ack(m,n-1));
8 }
9 int main ()
10 {
11     scanf("%d%d",&m,&n);
12     printf("ack(%d,%d)=%d",m,n,ack(m,n));
13     return 0;
14 }
```

### 5.3 C061-最大公约数

题目描述 43. 求  $n$  个数的最大公约数。其中:  $2 \leq n < 50$

输入:  $n$  个正整数, 以 0 作为数的结束。用空格隔开。

输出: 最大公约数和这  $n$  个数, 用一个空格隔开。

思路. 首先, 最大公约数是有结合律的, 我们只要不断地两两求最大公约数即可。其次, 最大公约数的求法是“辗转相除法”: 我们做带余数除法

$$a = b * q_1 + r_1$$

其中  $r_1$  是余数,  $a$  是被除数,  $b$  是除数。那么下一轮我们用当前的  $b$  放到  $a$  的位置, 当前的  $r_1$  放到  $b$  的位置, 继续做带余数除法:

$$b = r_1 * q_2 + r_2$$

, 以此不断地做下去, 直到某一次的余数  $r=0$ , 那么此时的  $b$  就是“最大公约数”

□

```
1 #include<stdio.h>
2 int ans,n,cnt,a[105];
3 int gcd(int a,int b) //强烈建议理解并背会这个求解最大公约数的函数代码
4 {
5     if(!b) return a;//其中!b 等价于 b!=0
6     return gcd(b,a%b);//递归
7 }
8 int main ()
9 {
10     scanf("%d",&n); ans=n;
11     while(n!=0)
12     {
```



```
13     a[++cnt]=n;//这个操作要学会，如何把一堆以 0 结尾的数字存到一个数组里面
14         //++cnt 表示先 cnt+=1，再来存储，当然也可以分两行写，更清晰
15     ans=gcd(ans,n);//在 C 语言中两两比较，或者两两干啥别的事情，这个方法也要记住
16     scanf("%d",&n);
17 }
18 printf("%d",ans);
19 for(int i=1; i<=cnt; i++) printf(" %d",a[i]);
20 return 0;
21 }
```

## 5.4 C127-求多项式的值

**题目描述 44.** 编写程序，计算下列多项式的值：

$poly(n, x) = 1$ , 当  $n = 0$ ;

$poly(n, x) = x$ , 当  $n = 1$ ;

$poly(n, x) = ((2 * n - 1) * x * poly(n - 1, x) - (n - 1) * poly(n - 2, x)) / n$ , 当  $n > 1$ ;

输入： $n$  和  $x$ ，格式：“ $\%d\%lf$ ”。( $n < 20$ )

输出：第  $n$  个多项式在  $x$  处的值，格式：“ $\%lf$ ”。

**思路.** 这题需要注意的一个点是，函数的返回值应当为 double，也就是说要写 `double poly(int n, double x)` □

```
1 #include<stdio.h>
2 int n;
3 double x;
4 double poly(int n,double x)
5 {
6     if(n==0) return 1;
7     if(n==1) return x;
8     return ((2*n-1)*x*poly(n-1,x)-(n-1)*poly(n-2,x))/n;
9 }
10 int main ()
11 {
12     scanf("%d %lf",&n,&x);
13     printf("%lf",poly(n,x));
14     return 0;
15 }
```

## 5.5 C069-将数字替换为字母

**题目描述 45.** 将数字替换为字母。输入一个整数，将它除以 2；再将除以 2 后所得数字的整数部分的各位用相应序号的字母替代。

替换原则：0 换为 a，1 换为 b，2 换为 c，...，以此类推，9 换为 j。

例如，1234，除以 2 等于 617，替换结果为 gbh

输入：一个正整数

输出：除 2 后的整数部分和字符串，中间用一个空格隔开。

样例：1234



617 gbh

思路. 依然是考察了如何把一个整数一位位分离出来, 分离出来之后再转化为字符就不是什么难事了。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 int a[105];
4 int n,k,cnt;
5 int main ()
6 {
7     scanf("%d",&n);
8     n=n/2;
9     k=n;
10    while(k>0)
11    {
12        cnt++;
13        a[cnt]=k%10;
14        k=k/10;
15    }
16    printf("%d ",n);
17    for(int i=cnt; i>=1; i--)
18        printf("%c",a[i]+97); //思考: 为啥是加 97? 97 是谁的 ASCLL 码?
19    return 0;
20 }
```

## 5.6 C070-求数组交集

**题目描述 46.** 定义最大下标为 50 的整型数组  $a$  和  $b$ , 编程实现: 查找同时在数组  $a$  和数组  $b$  中的全部元素。首先输入数组  $a$  的元素个数, 再输入数组  $a$  每个元素的值; 接着输入数组  $b$  的元素个数, 再输入数组  $b$  中各元素的值。

输出同时在两个数组的全部元素, 各元素间用空格隔开 (第 1 个元素前和最后 1 个元素后无空格)。

样例输入:

```
4
1 2 3 1
5
1 3 5 7 9
```

样例输出 1 3

思路. 用双重循环分别枚举  $a$  和  $b$  数组中的元素, 如果两个相等 (记为  $m$ ), 并且之前没有出现过 (这个地方具体的实现小技巧就是用搞一个  $flag$  数组, 每次判断  $flag[m]$  是否还为 0. 如果还是 0 就表示  $m$  这个数字之前没出现过), 那么就可以输出  $m$ , 同时要把  $flag[m]$  标记成 1

□

```
1 #include<stdio.h>
2 int x,y,a[55],b[55],i,j,m,n,cnt,flag[5000];
3 int main ()
4 {
```



```
5   scanf("%d",&x);
6   for(i=1;i<=x;i++) scanf("%d",&a[i]);
7   scanf("%d",&y);
8   for(i=1;i<=y;i++) scanf("%d",&b[i]);
9   for(i=1;i<=x;i++)
10  {
11      m=a[i];
12      for(j=1;j<=y;j++)
13      {
14          n=b[j];
15          if(m==n && flag[m]==0)
16          {
17              cnt++;
18              if(cnt==1) printf("%d",m);
19              else printf(" %d",m);
20              flag[m]=1;
21          }
22      }
23  }
24  return 0;
25 }
```

## 5.7 C095-输出数组中长度最大的等值数列段的始末下标

**题目描述 47.** 如果一个数列中某一段（至少有两个元素）的各元素值均相同，则称其为等值数列段。等值数列段中元素的个数叫做等值数列段的长度。

输入：分两行输入序列长度  $N$  和  $N$  个整数（其中  $1 \leq N \leq 50$ ）， $N$  个整数的每个数之间以空格隔开。

输出：数组中长度最大的等值数列段的始末下标（ $0 \sim N-1$ ），用英文逗号分隔。如果没有等值数列段，则输出  $NO$

说明：如果有多个同等长度的等值数列，只输出第一个等值数列的起止下标。

样例输入 7

3 1 2 1 1 2 2

样例输出：3,4

**思路.** 主要的思路就是，判断第  $i$  个数和它前面的那个数是否相等，如果相等那么  $cnt+=1$ ，然后同时记录这一段相等数字的起点和终点，并且同时更新最大值  $k$ 。如果不相等那么就让  $cnt$  回到 1，这样一遍 for 循环结束就可以得到最后的答案。 □

```
1  #include<stdio.h>
2  int n,a[55],cnt,flag,t,b,i,k;
3  int main()
4  {
5      scanf("%d",&n);
6      for(i=1;i<=n;i++) scanf("%d",&a[i]);
7      for(i=1;i<=n;i++)
8      {
9          if(a[i-1]==a[i])
```



```
10     {
11         flag=1;
12         cnt++;
13         if(cnt>k)
14         {
15             t=i-cnt+1;
16             b=i;
17             k=cnt;
18         }
19     }
20     else cnt=1;
21 }
22 if(flag==1) printf("%d,%d",t-1,b-1);
23 else printf("NO");
24 return 0;
25 }
```

### 5.8 C009-求小球第 $n$ 次落地的距离和反弹高度

**题目描述 48.** 一个球从 100 米高度自由落下，每次落地后反弹回原高度的一半，再落下，再反弹。求其到第  $n$  次落地（不考虑第  $n$  次反弹），共经过了多少米？第  $n$  次反弹高度是多少米？

输入： $n$  值。 $n$  为整数，取值范围： $0 < n \leq 20$

输出：分 2 行输出。第一行为小球到第  $n$  次落地经历的路程长度，第二行为小球第  $n$  次反弹的高度。长度值和高度值均为实数。

样例输入 1

样例输出

100.000000

50.000000

**思路.** 这题是相对简单的模拟题，只要根据题意，用 while 循环模拟每次反弹的过程即可。

□

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     double sum,h;
6     scanf("%d",&n);
7     h=50;sum=100;
8     while(n>1)
9     {
10         sum=sum+h*2;
11         h=h/2;
12         n--;
13     }
14     printf("%lf\n%lf",sum,h);
15     return 0;
16 }
```



## 5.9 C039-电梯运行时间计算

**题目描述 49.** 某城市最高的楼有一部电梯，该电梯依照输入楼层数的先后次序运行。电梯最初在 0 层。运行完一个输入序列后就停止在该楼层，不返回 0 层。编写程序计算电梯运行一个序列的时间。电梯每上 1 层需要 6 秒。每下 1 层需要 4 秒。如在某层停留，无论上下人多少，均停留 5 秒。楼层值大于等于 1，小于 100，0 在序列中表示结束序列输入。

输入：电梯运行序列（即：需要停靠的各楼层），用 1 个空格分隔，以 0 表示结束；

输出：电梯运行时间（秒）。

样例输入：2 1 0 样例输出：26

**思路.** 这个题也属于“模拟题”。首先输入是一个重点，会一直输入直到出现 0。这是一个十分常见的操作，需要熟记，具体见下面的代码：（先输入第一个 x，然后用 while 去控制 x 不等于 0，把 x 存到 a 数组中，然后继续输入 x，直至 x 等于 0 的时候结束 while 循环）。

接下来就是一个简单的根据题意的模拟。

□

```
1 #include<stdio.h>
2 int a[1005],sum,x,i,cnt=0;
3 int main ()
4 {
5     scanf("%d",&x);
6     while(x!=0)
7     {
8         cnt++;
9         a[cnt]=x;
10        scanf("%d",&x);
11    }
12    if(cnt==0)
13    {
14        printf("0");
15        return 0;
16    }
17    sum=6*a[1]+5;
18    for(int i=2;i<=cnt;i++)
19    {
20        if(a[i-1]>a[i])
21            sum=sum+4*(a[i-1]-a[i])+5;
22        else if(a[i-1]<a[i])
23            sum=sum+6*(a[i]-a[i-1])+5;
24    }
25    printf("%d",sum);
26    return 0;
27 }
```

## 5.10 C041-小明的计算

**题目描述 50.** 叛逆期的小明什么都喜欢反着做，连看数字也是如此（负号除外），比如：小明会把 1234 它看成 4321；把 -1234 看成 -4321；把 230 看成 032 (032=32)；把 -230 看成 -032 (-032=-32)。现在，小明做了一些  $a+b$  和  $a-b$  的题目 ( $a, b$  为整数且不含前导 0)，如果给你这些题目的正确答案，你能猜出小明会得到



什么答案吗?

输入: 两个整数  $x, y$  ( $-100000 \leq x, y \leq 100000$ ),  $x$  表示  $a+b$  的正确答案,  $y$  表示  $a-b$  的正确答案。输入保证合法, 且不需考虑  $a$  或  $b$  是小数的情况。

输出: 输出两个整数  $s$  和  $t$ , 两数之间用 1 个空格分开。其中:  $s$  表示小明将得到的  $a+b$  答案,  $t$  表示小明将得到的  $a-b$  答案。

样例输入: 20 6

样例输出: 38 24

**思路.** 这题关键在于如何把一个数字反过来。一个很直接的想法是, 我们在之前学过的“把一个整数一位一位分开”的时候, 其实是用 while 循环, 从最低位开始一个一个分出来的。所以我们如果在这个 while 过程中, 把最低位开始的数字一个个存到数组里, 那么这个数组的从左往右顺序自然就是原来的那个数字的倒序了。具体见代码。 □

```
1 #include<stdio.h>
2 #include<math.h>
3 int x,y,a,b,f[50],k,t,cnt,i,n,s,t,m;
4 int change(int t)
5 {
6     n=0;cnt=0;
7     m=t;
8     if(t==0) return 0;
9     t=abs(t);
10    while(t>0)
11    {
12        k=t%10;
13        cnt++;
14        f[cnt]=k;//从最低位开始一位位存到 f 数组里面
15        t=t/10;
16    }
17    for(i=1;i<=cnt;i++)
18    {
19        n=f[i]+n*10;//再从左往右一位位把 f 数字里面的元素加起来
20    }
21    n=(m/abs(m))*n;
22    return n;
23 }
24 int main ()
25 {
26     scanf("%d %d",&x,&y);
27     a=(x+y)/2;
28     b=(x-y)/2;
29     s=change(a)+change(b);
30     t=change(a)-change(b);
31     printf("%d %d",s,t);
32     return 0;
33 }
```



## 6 第 12 周

### 6.1 C120-去掉一个字符串中除头部和尾部空格外的所有空格

**题目描述 51.** 去掉一个字符串中除头部和尾部空格外的所有空格。

输入：占一行（注：输入的字符串长度不会超过 80）

输出：占一行

例如：

*abc de*

*abcde*

**思路.** 去掉首尾空格，也就是要知道从左往右第一个不是空格的位置 st（start 的缩写），以及从右往左第一个不是空格的位置 ed（end 的缩写），这可以用 while 循环来实现。然后把原字符串从 st 到 ed 这一段输出即可 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n,i,st,ed;
4 char a[100];
5 int main()
6 {
7     gets(a);
8     n=strlen(a);
9     i=0;
10    while(a[i]== ) //依旧是用 while 循环来处理“不知道循环上下界的情况”
11    {
12        i++;
13        st=i;
14        printf(" ");
15    }
16    i=n-1;
17    ed=n-1;
18    while(a[i]== )
19    {
20        i--;
21        ed=i;
22    }
23    for(i=st;i<=ed;i++)
24    {
25        if(a[i]!= )printf("%c",a[i]);//这一行其实等号后面是一个单引号，里面是一个空格
26        //也就是“（中文输入法的引号能打出来，但这是错的
27        //我们需要英文输入法的引号）。
28    }
29    for(i=ed+1;i<=n-1;i++) printf(" ");
30    return 0;
31 }
```



## 6.2 C067-整数三位分节

**题目描述 52.** 写出正整数的三位分节格式。如，当用户输入 82668634 时，程序应该输出 82,668,634。

输入：正整数

输出：三位分解格式。

样例：

82668634

82,668,634

**思路.** 三个三个分开来，那么肯定是要考虑  $n$  除以 3 的余数  $k$  的，先输出  $k$  个数字，再剩下的三个三个输出就好。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[100];
4 int cnt,n,k;
5 int main()
6 {
7     gets(s);
8     n=strlen(s);
9     k=n%3;
10    if(n<3) puts(s);
11    else
12    {
13        if(k>0)
14        {
15            for(int i=0;i<k;i++) printf("%c",s[i]);
16            printf(",");
17        }
18        for(int i=k;i<=n-2;i++)
19        {
20            cnt++;
21            printf("%c",s[i]);
22            if(cnt%3==0)printf(",");
23        }
24        printf("%c",s[n-1]);
25    }
26    return 0;
27 }
```

## 6.3 C093-字符串中奇数下标的字符顺序颠倒

**题目描述 53.** 将字符串中奇数下标（从 0 开始）的字符顺序颠倒。如字符 12345678，结果为 18365472

输入：字符串，不含空格。

输出：结果字符串。

样例输入：1234567890

样例输出：1038567492



思路. 把奇偶分开来考虑。先一个 for 循环轮一遍，把奇数下标的都存下来，然后用 i 和 cnt 两个变量来控制奇偶数的输出。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105],c,t[105];
4 int i,cnt,n;
5 int main()
6 {
7     gets(s);
8     n=strlen(s);
9     for(i=0;i<=n-1;i++)
10    {
11        if(i%2!=0)
12        {
13            cnt++;
14            t[cnt]=s[i];
15        }
16    }
17    i=0;
18    while(i<n-1)
19    {
20        printf("%c%c",s[i],t[cnt]);
21        cnt--;
22        i+=2;
23    }
24    if(i==n-1)printf("%c",s[i]);
25    return 0;
26 }
```

## 6.4 C096-删除串中的重复字符

题目描述 54. 输入一个长度不超过 100 的字符串，删除串中的重复字符。

输入：输入要检查的字符串，长度不超过 100 个字符。例如：abacaeedabcdcd。

输出：删除重复字符后的字符串。例如：abcded。

思路. 类似于之前出现过 11 周的的一个 flag 数组标记的题目，我们用一个 cnt 标记之前出现过的数字，若是第一次出现，我们才输出这个数字，否则就不用输出了，这样就很好的实现了“删除重复数字”的功能。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105];
4 int n,cnt[30];
5 int main()
6 {
```



```
7     gets(s);
8     n=strlen(s);
9     for(int i=0;i<=n-1;i++)
10    {
11        int id=s[i]-a+1;
12        cnt[id]++;
13        if(cnt[id]==1) printf("%c",s[i]);
14    }
15 }
```

## 6.5 C072-删除字符

**题目描述** 55. 编写程序，输入字符串到字符数组  $s$  中，再输入一个字符给变量  $c$ ，判断  $s$  中是否有  $c$ ，若有则将该字符从  $s$  中删除后输出。若没有，则输出字符串 “*NotFound*”。

输入 2 行，1 个字符串和 1 个字符：

输出：删除后的字符串

样例 1:

输入：

*smiles*

*s*

输出： *mile*

样例 2:

输入：

*smiles*

*a*

输出： *NotFound*

**思路.** 先一个 for 循环轮一遍，判断是否存在我们需要找的字符  $a$ 。如果不存在直接输出 *NotFound*，如果存在再轮第二遍，把不等于  $a$  的输出出来，这样就自动的完成了“删除”。

这里可以做一个小小的总结，很多时候我们“删除”“插入”，其实不需要非得真的删除或者插入，这样会很大的改动整个数组的结构，写起来不是很方便，其实我们只要在最后输出的时候做删除或者插入的操作就可以了。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[100],c;
4 int i,flag,n,k;
5 int main ()
6 {
7     gets(s);
8     scanf("%c",&c);
9     n=strlen(s);
10    for(i=0;i<=n-1;i++)
11    {
12        if(s[i]==c)
13        {
14            flag=1;
```



```
15         }
16     }
17     if(flag==1)
18     {
19         for(i=0;i<=n-1;i++)
20         {
21             if(s[i]!=c) printf("%c",s[i]);
22         }
23     }
24     else printf("NotFound");
25     return 0;
26 }
```

## 6.6 C084-统计第 2 个字符串在第 1 个字符串中出现的次数

**题目描述 56.** 输入两个字符串，统计第 2 个字符串（至少含 2 字符）在第 1 个字符串中出现的次数。没有出现，次数为 0。

输入：两个字符串，以空格分隔。

输出：出现次数，整数。

样例 1:

*banana an*

*2*

样例 2:

*bnnnnn nn*

*3*

**思路.** 为方便起见，我们记第一个字符串为 s，第二个为 t。先一个 for 循环轮一遍 s，如果 s[i] 和 t[0] 相等，那么我们从 s[i] 位置开始，判断后面的一串是不是和 t 完全相等（写一个子函数），如果是的话，cnt+=1。最后 cnt 就是答案 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105],a[20];
4 int cnt,flag,m,n,i,j;
5 int pd(char s[],char a[],int k)
6 {
7     j=0;
8     while(j<=n-1)
9     {
10         if(s[k]==a[j])
11         {
12             j++;
13             k++;
14         }
15         else return 1;
16     }
17     return 0;
```



```
18 }
19 int main()
20 {
21     scanf("%s %s",s,a);
22     m=strlen(s);
23     n=strlen(a);
24     for(i=0;i<=m-1;i++)
25     {
26         if(pd(s,a,i)==0)
27         {
28             cnt++;
29             flag=1;
30         }
31     }
32     if(flag==1) printf("%d",cnt);
33     else printf("0");
34     return 0;
35 }
```

## 6.7 C085-按规则去掉星号

**题目描述 57.** 输入字符串，字符串中有若干星号 \*，去掉字符串首尾的星号 (\*)，保留中间的星号。字符串长度不大于 200。

输入：带有 \* 号的字符串。

输出：首尾不带 \* 号的字符串。

样例：

*\*\*street\*\*music\*\*\*\**

*street\*\*music*

**思路.** 和第 12 周第一题类似，我们需要知道从左往右第一个不为 \* 的位置 st，和从右往左第一个不为 \* 的位置 ed，然后把原字符串的 st 到 ed 位置的字符输出来。(下面代码中的 k 就是我这里说的 st，b 就是我说的 ed)

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char a[205];
4 int k,b,n,i;
5 int main()
6 {
7     gets(a);
8     n=strlen(a);
9     for(i=0; i<=n-1; i++)
10     {
11         if(a[i]==*) k++;
12         else break;
13     }
14     b=n-1;
```



```
15     for(i=n-1; i>=0; i--)
16     {
17         if(a[i]==*) b--;
18         else break;
19     }
20     for(i=k; i<=b; i++)
21     printf("%c",a[i]);
22     return 0;
23 }
```

## 6.8 C113-求给定区间中的孪生数的数量

**题目描述 58.** 所谓孪生素数是指间隔为 2 的相邻素数，例如最小的孪生素数是 3 和 5, 5 和 7 也是孪生素数。编写程序，求给定区间  $[m,n]$  中的孪生数的数量。例如  $[2,10]$  中的孪生数有 (3,5) 和 (5,7)，则  $[2,10]$  中孪生数的数量为 2。

输入：正整数  $m,n$ ，其中  $m,n > 1$

输出： $[m,n]$  中的孪生的数量

样例输入：2 100

样例输出：8

**思路.** 本题考察的是素数的判断，正好可以回顾一下这个知识点，以及这个函数该怎么写（建议会熟练默写，当然是在理解的基础上） □

```
1  #include<stdio.h>
2  int n,m,ans;
3  int judge(int x)
4  {
5      for(int i=2; i<=x-1; i++)
6          if(x%i==0) return 0;
7      return 1;
8  }
9  int main()
10 {
11     scanf("%d%d",&m,&n);
12     for(int i=m; i<=n-2; i++)
13     {
14         if(judge(i)==1 && judge(i+2)==1) ans++;
15     }
16     printf("%d",ans);
17     return 0;
18 }
```

## 6.9 C119-二进制形式的 IP 地址转为十进制

**题目描述 59.** 从键盘输入一个 32 位的二进制形式的 IP 地址，将其转换为点分十进制的表示，如果输入的字符串中含有 1 和 0 以外的字符，则输出“该字符串不是正确的 IP 地址”。

输入：0、1 二进制序列，不会超过 32 位，位数不足时程序自动在前面补 0。



输出：点分 IP 地址。

样例输入：11111111100000011111111110001111

样例输出：255.129.255.143

思路. 我们要 8 位 8 位分开来，把 32 位数字分成 4 组，然后分别做二进制转十进制（加权和），比如最低位权值是 2 的 0 次方，倒数第二位是 2 的 1 次方，最高位是 2 的 7 次方，以此类推。

注：第 37 行不知道为啥 latex 里面写 `printf(" 该字符串不是正确的 IP 地址")` 就报错了，用 `printf("A")` 替代了一下 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n;
4 char s[105],t[105];
5 int get(int st,int ed)
6 {
7     int res=0,j=0;
8     for(int i=ed; i>=st; i--)
9     {
10         res+=(t[i]-0)*(1<<j);
11         j++;
12     }
13     return res;
14 }
15 int main()
16 {
17     gets(s);
18     n=strlen(s);
19     if(n<32)
20     {
21         int cha=32-n;
22         for(int i=0; i<=cha-1; i++) t[i]=0;
23         int j=0;
24         for(int i=cha; i<=31; i++)
25         {
26             t[i]=s[j];
27             j++;
28         }
29     }
30     else
31     {
32         for(int i=0; i<=31; i++) t[i]=s[i];
33     }
34     for(int i=0; i<=31; i++)
35         if(t[i]!=0 && t[i]!=1)
36         {
37             printf("A");
38             return 0;
```



```
39     }
40     printf("%d.%d.%d.%d",get(0,7),get(8,15),get(16,23),get(24,31));
41     return 0;
42 }
```

## 6.10 C066-打点滴时间计算

**题目描述 60.** 在医院打点滴（吊针）的时候，如果滴起来有规律，先是滴一滴，停一下；然后滴二滴，停一下；再滴三滴，停一下...，现在有一个问题：这瓶盐水一共有  $v$  毫升，每一滴是  $d$  毫升，每一滴的速度是一秒（假设最后一滴不到  $d$  毫升，则花费的时间也算一秒），停一下的时间也是一秒，这瓶水什么时候能滴完呢？（ $0 < d < v < 6000$ ）

输入：一滴是多少毫升和一瓶盐水有多少毫升，中间用空格隔开。

输出：滴完需要多少时间。

样例输入：0.5 1.7

样例输出：6

思路. 这其实也是一个模拟题，下面的 `ceil` 函数表示向上取整，要用的头文件 `math.h`

□

```
1 #include<stdio.h>
2 #include<string.h>
3 #include<math.h>
4 double d,v;
5 int n;
6 int main()
7 {
8     scanf("%lf%lf",&d,&v);
9     n=ceil(v/d);
10    int i,s=0;
11    for(i=1; ;i++)
12    {
13        if(s+i>n) break;
14        s+=i;
15    }
16    if(s<n) printf("%d",n+i-1);
17    else printf("%d",n+i-2);
18    return 0;
19 }
```



## 7 第 13 周

### 7.1 C123-电话簿排序

**题目描述 61.** 编写一个程序，读入  $n$  个用户姓名和电话号码，按姓名的字典顺序排列后，输出用户的姓名和电话号码， $n$  从键盘输入。

样例：

输入：

3

zhang 122

wang 233

li 567

输出：

li 567

wang 233

zhang 122

**思路.** 考察结构体排序。结构体可以理解成  $n$  个元素的的捆绑，或者也可以简单粗暴理解成一个向量，只不过这边的向量各个维度的元素可以是不同类型的（比如，可以有 int，可以有 char，可以有 double）。

结构体的排序，重点就是自己根据题目要求写出合适的比较函数 Cmp，很多时候这种题目会有第一关键字和第二关键字（即先根据第一关键字排序，如果一样再根据第二关键字排序），所以会用到非常多的 if 和 else 嵌套，需要有清晰的逻辑思维。如果题目复杂，可以先在草稿纸上面梳理一下。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n;
4 struct PhoneList
5 {
6     char name[30];
7     char phoneNum[20];
8 }plist[100];
9 void cmp(PhoneList *plist, int n)//compare 缩写，表示排序过程中的比较
10 {
11     for (int i=0; i<n; i++)//本质上是冒泡排序
12     {
13         for (int j=n-1; j>i; j--)
14         {
15             if (strcmp(plist[j].name, plist[j-1].name)<0) //自定义比较的是 name
16             {
17                 PhoneList temp = plist[j];
18                 plist[j] = plist[j-1];
19                 plist[j-1] = temp;
20             }
21         }
22     }
23 }
24 int main()
25 {
```



```
26     scanf("%d", &n);
27     for (int i=0; i<n; i++)
28     {
29         scanf("%s%s", plist[i].name, plist[i].phoneNum);
30     }
31     cmp(plist, n); //调用自定义函数排序
32     for (int i=0; i<n; i++)
33     {
34         printf("%s %s\n", plist[i].name, plist[i].phoneNum);
35     }
36     return 0;
37 }
```

## 7.2 C124-学生成绩排序

**题目描述 62.** 有  $n$  名学生，每个学生的数据包含学号、姓名、三门课的成绩。可以从键盘输入  $n$  个学生的数据，按总成绩从小到大排序，打印包含学号、姓名、三门课成绩和总成绩的成绩单。（测试时，数据从键盘输入。）

输入：第 1 行为整数  $n$ ，后面  $n$  行表示  $n$  个人的信息，包括学号、姓名、和三门课的成绩，每行的数据间用空格隔开。

输出： $n$  行，表示  $n$  个人的信息，包括学号、姓名、三门课的成绩和总成绩，数据间一个空格，末尾无空格。

样例：

输入：

2

2021001 li 60 80 70

2021002 wang 100 90 80

输出：

2021001 li 60 70 80 210

2021002 wang 100 90 80 270

**思路.** 仍旧是结构体排序，本题的排序关键字是成绩，具体看代码（和上题区别本质上就只有 cmp 函数里面冒泡排序的某一行不一样） □

```
1 #include<stdio.h>
2 #include<string.h>
3 int n;
4 struct GradeList
5 {
6     char id[20];
7     char name[30];
8     int course01;
9     int course02;
10    int course03;
11    int total;
12 }glist[100];
13 void GradeBubble(GradeList* glist, int n)
```



```
14 {
15     for (int i=0; i<n; i++)
16     {
17         for (int j=n-1; j>i; j--)
18         {
19             if (glist[j].total<glist[j-1].total)//按总成绩排序
20             {
21                 GradeList temp = glist[j];
22                 glist[j] = glist[j-1];
23                 glist[j-1] = temp;
24             }
25         }
26     }
27 }
28 int main()
29 {
30     scanf("%d", &n);
31     for (int i = 0; i < n; i++)
32     {
33         scanf("%s%s%d%d%d",
34             glist[i].id, glist[i].name, &glist[i].course01,
35             &glist[i].course02, &glist[i].course03);
36         glist[i].total = glist[i].course01 + glist[i].course02 + glist[i].course03;
37     }
38     GradeBubble(glist, n);
39     for (int i = 0; i < n; i++)
40     {
41         printf("%s %s %d %d %d %d\n",
42             glist[i].id, glist[i].name, glist[i].course01,
43             glist[i].course02, glist[i].course03, glist[i].total);
44     }
45     return 0;
46 }
```

### 7.3 C064-翻译数字

**题目描述 63.** 编写一个程序，从键盘上读入一个数字串，把数字转化为对应的小写英语数字单词输出。例如：输入 234，输出 two three four。

输入：一串数字

输出：用空格隔开英文数字单词（英文字母都是小写）。

**思路.** 原则上用 Switch 语句当然是可以的，但是我个人其实不会太推荐，因为要写 break，要写 default，如果到时候记错了会比较麻烦（当然如果能记对，当然也没有问题）。

当然，本题还有一种更简便的写法，需要先把 1 至 9 对应的单词记成一个数组，要用到一个技巧：减去单引号 0，表示 char 转为 int，具体看下面的代码。 □

```
1 #include<stdio.h>
```



```
2 #include<string.h>
3 char word[][30] = {
4     "zero", "one", "two", "three",
5     "four", "five", "six", "seven", "eight", "nine"
6 };
7 char s[100];
8 int main()
9 {
10     scanf("%s",s);
11     printf("%s",word[s[0]-0]);//先输出第一个，减去单引号 0，表示 char 转 int
12     for (int i=1; i<strlen(s); i++)//i=1 其实已经是第二个了
13         printf(" %s", word[s[i]-0]);//从第二个开始要先输出一个空格
14     return 0;
15 }
```

#### 7.4 C074-统计字母次数

**题目描述 64.** 输入一个可能含空格的字符串（其长度不超过 81），分别统计其中 26 个英文字母出现的次数（不区分大、小写字母），并按字母出现的次数，从高到低进行排序。若次数相同，按字母顺序排列。字母输出格式举例，例如：A-3，表示字母 A 出现 3 次，C-0 表示字母 C 没有出现过。

输入：

第一行为输入，占一行

输出：

第二行为输出，占一行。按照字母输出格式从高到低输出，各字母输出之间用一个空格字符进行分隔。

思路. 还是考察结构体排序。



```
1 #include<stdio.h>
2 #include<string.h>
3 char s[100];
4 struct Count
5 {
6     char alphabet;
7     int c;
8 }count[26];
9 void BubbleAlpha(Count count[])
10 {
11     for (int i=0; i<26; i++)
12     {
13         for (int j=25; j>i; j--)
14         {
15             if (count[j].c > count[j-1].c)
16             {
17                 Count temp = count[j];
18                 count[j] = count[j-1];
19                 count[j-1] = temp;
20             }
21         }
22     }
23 }
```



```
21     }
22 }
23 }
24 int main()
25 {
26     for (int i=0; i<26; i++)
27     {
28         count[i].alphabet = A + i;
29         count[i].c = 0;
30     }
31     gets(s); //整行读入
32     int n=strlen(s);
33     for(int i=0; i<n; i++) //统一转为大写字母
34         if(s[i]>=a && s[i]<=z) s[i]-=32; //97-65=32
35     for(int i=0; i<n; i++)
36         if (s[i] >= A && s[i] <= Z)
37             count[s[i]-A].c++;
38     BubbleAlpha(count);
39     printf("%c-%d", count[0].alphabet, count[0].c);
40     for (int i = 1; i < 26; i++)
41     {
42         printf(" %c-%d", count[i].alphabet, count[i].c);
43     }
44     return 0;
45 }
```

## 7.5 C076-按规则排序

**题目描述** 65. 编写程序，有一个包含奇数个项的整数序列，按照以下规则进行排序：

最大值排在中间，最小值排在最左，中值排在最右，其它值清为 0。

输入 2 行，第一行  $n$  表示要输入的数的个数， $n$  是奇数。第二行输入的  $n$  个数。

输出：排序后的数据，数据之间用空格分隔，最后一个数据后面没有空格。

样例 1：

输入

```
3
66 10 30
```

输出

```
10 66 30
```

样例 2：

输入

```
5
12 20 9 88 32
```

输出

```
9 0 88 0 20
```

**思路.** 考察冒泡排序。先排序，然后输出的时候按照题目要求输出即可。

□



```
1 #include<stdio.h>
2 int n;
3 int a[100] = {0};
4 void Bubble(int a[], int n)
5 {
6     for (int i = 0; i < n; i++)
7     {
8         for (int j = n - 1; j > i; j--)
9         {
10             if (a[j] < a[j - 1])
11             {
12                 int t = a[j];
13                 a[j] = a[j - 1];
14                 a[j - 1] = t;
15             }
16         }
17     }
18 }
19 int main()
20 {
21     scanf("%d", &n);
22     for (int i = 0; i < n; i++)
23         scanf("%d", &a[i]);
24     Bubble(a, n); //先排序
25     int min = a[0];
26     int mid = a[n / 2];
27     int max = a[n - 1];
28     //清0并输出
29     printf("%d", a[0]);
30     for (int i = 1; i < n; i++)
31     {
32         if (i == n/2) a[i] = max;
33         else if (i == n-1) a[i] = mid;
34         else a[i] = 0;
35         printf(" %d", a[i]);
36     }
37     return 0;
38 }
```

## 7.6 C134-最大最小和平均值

**题目描述 66.** 编写函数，求有  $n$  个元素的一维数组中的最大值和最小值，并返回数组元素的平均值，函数原形如下：

`double fun(int a[],int n,int *max,int *min);`

输入输出格式：



输入：输入  $n+1$  个数，各个数之间用空格分隔。第一个数为数组元素的个数  $n$ 。

输出：3 个数字。第一个数字为数组中的最大元素值，第二个数字为数组中的最小元素值，第三个数字为数组元素的平均值（保留小数点后 6 位有效数字）。3 个数之间用空格分开。

样例：

输入：3 1 5 9

输出：9 1 5.000000

思路. 题目要求使用给定的函数进行解答，这个函数调用了指针，所以需要稍微熟悉一下指针的用法（函数传值的时候要加一个取地址符号，函数内要在变量名前面加星号） □

```
1 #include<stdio.h>
2 int n,maxv,minv;
3 int a[100];
4 double fun(int a[], int n, int* maxv, int* minv)
5 {
6     double sum = 0; //求和
7     *maxv = *minv = a[0]; //最大最小
8     for (int i = 0; i < n; i++)
9     {
10         sum += a[i];
11         if (a[i] > *maxv) *maxv = a[i];
12         if (a[i] < *minv) *minv = a[i];
13     }
14     return sum / n;
15 }
16 int main()
17 {
18     scanf("%d", &n);
19     for (int i = 0; i < n; i++)
20         scanf("%d", &a[i]);
21     double average = fun(a, n, &maxv, &minv);
22     printf("%d %d %lf", maxv, minv, average);
23     return 0;
24 }
```

## 7.7 C110-删除字符串中所有非小写字母字符

题目描述 67. 输入一个英文字符串（长度  $<81$ ）可能包含空格，删除其中所有非小写字母的字符，并输出删除后的字符串（小写字母的相对位置保持不变）。

思路. 判断是否为小写字母，如果是的就输出。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[100];
4 int main()
```



```
5 {
6     gets(s);
7     int n = strlen(s);
8     for(int i=0; i<n; i++)
9         if ((s[i] >= a && s[i] <= z))
10            printf("%c", s[i]);
11     return 0;
12 }
```

## 7.8 C145-词频统计

**题目描述 68.** 输入一系列英文单词，单词之间用空格隔开（一到多个空格），用“###”表示输入结束，统计输入过哪些单词以及各单词出现的次数，统计时区分大小写字母。本题假定：不重复的单词数不超过 100 个；每个单词长度限制在 20 以内。

**思路.** 这种要求以某某为输入结束的题目，可以再重点复习一下，可以用我们这份讲义前面说的，先输入一个，再用 while，当然也可以直接用 do while。尤其这道题最后输入结束语句是一个字符串的时候，还要会 strcmp 这个函数。其他关于这题，只需要用结构体就可以解决。 □

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int n;
5 char s[30];
6 struct WordCount
7 {
8     char word[30];
9     char count;
10 }wordCount[100];
11 int main()
12 {
13     scanf("%s", s);
14     while (strcmp(s, "###"))
15     {
16         int flag = 0; //是否重复
17         //查找是否已经统计过（重复）
18         for (int i = 0; i < n; i++)
19         {
20             if (strcmp(s, wordCount[i].word) == 0)
21             {
22                 wordCount[i].count++;
23                 flag = 1;
24                 break;
25             }
26         }
27         if (flag == 0) //新单词
28         {
```



```
29         strcpy(wordCount[n].word, s);
30         wordCount[n++].count = 1;
31     }
32     scanf("%s", s);
33 }
34 printf("%s-%d", wordCount[0].word, wordCount[0].count);
35 for (int i = 1; i < n; i++)
36 {
37     printf(" %s-%d", wordCount[i].word, wordCount[i].count);
38 }
39 return 0;
40 }
```

## 7.9 C148-按字母序升序输出字符串

**题目描述 69.** 定义字符串数组 *str*，其数组长度为 81，键盘读入一个字符串（少于 81 个字符）。将该字符串中出现的所有小写字母按字母序升序输出（字母序为：*abcdefghijklmnopqrstuvwxyz*）。

输入输出格式要求：

输入输出各占一行，第一行是输入，第二行是输出。

样例：

输入：AaBbCc

输出：abc

思路. 按照题意进行排序即可。



```
1 #include<stdio.h>
2 #include<string.h>
3 char low[100];
4 char s[200];
5 void StrBubble(char* a)
6 {
7     int len = strlen(a);
8     for (int i = 0; i < len; i++)
9     {
10         for (int j = len - 1; j > i; j--)
11         {
12             if (a[j] < a[j - 1])
13             {
14                 char temp = a[j];
15                 a[j] = a[j - 1];
16                 a[j - 1] = temp;
17             }
18         }
19     }
20 }
21 int main()
```



```
22 {
23     scanf("%s", s);
24     int n=strlen(s);
25     int j = 0;
26     for(int i=0; i<n; i++)
27     {
28         if (s[i] >= a && s[i] <= z)
29         {
30             low[j] = s[i];
31             j++;
32         }
33     }
34     StrBubble(low);
35     printf("%s", low);
36     return 0;
37 }
```

### 7.10 C158-按要求输出奇数位置的字符排序后的字符串

**题目描述 70.** 用户输入一个字符串，将字符串中的奇数位置上的字符按字母表中的顺序排序，仍按顺序保存在奇位置上，其他字母位置不变。例如 *teacher*, *t,a,h,r* 排序后结果为 *a,h,r,t*, 则结果字符串为: *achcret*。

输入格式: 长度小于 100 的字符串,

输出: 奇数位置的字符排序后的字符串

思路. 也是按照题意排序即可。



```
1  #include<stdio.h>
2  #include<string.h>
3  char s[200];
4  int main()
5  {
6      scanf("%s", s);
7      int len = strlen(s);
8      if(len % 2 == 0) len -= 1;
9      for (int i = 0; i < len; i += 2)
10     {
11         for (int j = len - 1; j > i; j -= 2)
12         {
13             if (s[j] < s[j - 2])
14             {
15                 int temp = s[j];
16                 s[j] = s[j - 2];
17                 s[j - 2] = temp;
18             }
19         }
20     }
```



```
21     printf("%s", s);  
22     return 0;  
23 }
```



## 8 第 14 周

### 8.1 C036-K 进制数的转换

**题目描述 71.**  $K$  进制数的转换。输入非负整数  $a$  将其转换为  $K$  进制数,  $2 \leq K \leq 32$ 。

$K$  进制数使用的符号是  $0 \sim 9, A, \dots V$  来表示 (字母是大写)

输入: 一个十进制非负整数  $a$  和一个正整数  $K$ ,  $2 \leq K \leq 32$ 。空格分隔。

输出:  $K$  进制表示的整数。

**思路.** 非常经典的进制转换知识点, 只需要一直对进制  $k$  取模, 然后逆序输出即可。

□

```
1 #include <stdio.h>
2 #include <string.h>
3 char symbols[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"; //K 进制符号表
4 int a,K;
5 int index; //记录 result 数组的存储位置
6 char result[100];
7 int main()
8 {
9     scanf("%d %d", &a, &K);
10    if (a == 0)
11    {
12        printf("0\n");
13        return 0;
14    }
15    while (a > 0)
16    {
17        result[index++] = symbols[a%K];
18        a = a/K;
19    }
20    for(int i=index-1; i>=0; i--)
21        { // 逆序输出结果
22            printf("%c", result[i]);
23        }
24    printf("\n");
25    return 0;
26 }
```

### 8.2 C037-乘积中各位数字组成的最大数

**思路.** 思路没啥困难的, 就是把乘以三之后的数字用 while 循环一位位存到数组里面, 然后冒泡排序。比较关键的是这个冒泡, 一定要掌握 (理解原理 + 背代码), 尤其是内外两层循环下标要特别注意。感觉我当年同学中抽到冒泡的非常多。

**提醒:** 尽量不要用 max 和 min 作为变量名 or 函数名, 计算机内部有用这个取名的函数, 理论上是不会出事的, 但是保不齐考试的时候机房电脑有一些奇怪的 bug, 所以还是尽量不要用。所以我选择用 maxv 和 minv

□



```
1 #include<stdio.h>
2 #include<math.h>
3 #include<string.h>
4 int n, out, maxv[100], i;
5 int main()
6 {
7     scanf("%d", &n);
8     out = 3 * n;
9     while (out / 10 != 0 || out % 10 != 0)
10    {
11        maxv[i] = out % 10;
12        out = out / 10;
13        i++;
14    }
15    if(i == 0) maxv[i] = 0;
16    for(int j=0; j<i; j++)
17    {
18        for(int k=i-1; k>0; k--)
19        {
20            if (maxv[k] > maxv[k-1])
21            {
22                int term = maxv[k];
23                maxv[k] = maxv[k-1];
24                maxv[k-1] = term;
25            }
26        }
27    }
28    printf("%d ", 3*n);
29    for (int j=0; j<i; j++)
30        printf("%d", maxv[j]);
31    return 0;
32 }
```

### 8.3 C136-签到和签退

思路. 考察了结构体的排序。

□

```
1 #include<stdio.h>
2 #include<math.h>
3 #include<string.h>
4 typedef struct Time
5 {
6     int HH;
7     int MM;
8     int SS;
9 }Ti; //自定义结构体
```



```
10 typedef struct Record
11 {
12     char ID[15];
13     Ti intime;
14     Ti outtime;
15 }Rec; //自定义结构体
16 int Cmp(Ti t1, Ti t2) //自己定义一个 compare 函数，可以自己定义比较规则
17 {
18     if (t1.HH < t2.HH) return -1; //第一关键字可以直接比较出来小于
19     else if (t1.HH > t2.HH) return 1; //第一关键字可以直接比较出来大于
20     else //第一关键字无法直接比较出来，即比较结果是等于
21     {
22         if (t1.MM < t2.MM) return -1; //比较第二关键字
23         else if (t1.MM > t2.MM) return 1;
24         else
25         {
26             if (t1.SS < t2.SS) return -1;
27             else if (t1.SS > t2.SS) return 1;
28         }
29     }
30 }
31 int main()
32 {
33     int m;
34     Rec rec[100];
35     scanf("%d", &m);
36     for (int i=0; i<m; i++)
37     {
38         scanf("%s %d:%d:%d %d:%d:%d", rec[i].ID, &rec[i].intime.HH, &rec[i].
39             intime.MM, &rec[i].intime.SS, &rec[i].outtime.HH, &rec[i].outtime.
40             MM, &rec[i].outtime.SS);
41     }
42     for (int i=0; i<m; i++)
43     {
44         for (int j=m-1; j>0; j--)
45         {
46             if (Cmp(rec[j].intime, rec[j-1].intime)<0) //调用自己定义的比较函数
47             {
48                 Rec term = rec[j];
49                 rec[j] = rec[j-1];
50                 rec[j-1] = term;
51             }
52         }
53     }
54     printf("%s", rec[0].ID);
55     for (int i=0; i<m; i++)
```



```
54 {
55     for (int j=m-1; j>0; j--)
56     {
57         if(Cmp(rec[j].outtime, rec[j-1].outtime)>0)//调用自定义的比较函数
58         {
59             Rec term = rec[j];//临时变量 term, 用于交换
60             rec[j] = rec[j-1];
61             rec[j-1] = term; //这三行在交换 rec[j] 和 rec[j-1]
62         }
63     }
64 }
65 printf(" %s", rec[0].ID);
66 return 0;
67 }
```

#### 8.4 C103-出圈问题

**题目描述 72.** 有  $n$  个小孩围成一圈, 从 1 顺序排号。从第一个人开始报数 (1 至  $k$  报数), 凡报到  $k$  的人退出圈子, 问最后留下的人原来排在第几号。

输入: 人数 (不超过 1000), 以及报数的  $k$  (不超过 3000), 空格分隔。

输出: 最后留下的人的编号

**思路.** 这个题个人感觉本意考察的应该是链表, 如果用了链表的话, 对逻辑的要求就会低很多 (我写了一个, 在下面, 写了批注, 如果想看的话可以看看)。当然, 如果直接用传统的模拟方式, 只要能想得清楚, 当然也是可以的。

要非常注意的一个点是: 这个题目不能用  $k$  去模  $n$ , 这样会导致报数结果和正确答案不一致。实际上, 计算机一般能做的运算规模往往会轻松达到亿级别, 像大学计算机这门课算是比较基础的编程课, 所以实际上考察的数据规模不会太大, 题目中也明确说了是 1000。如果从考试的角度来说, 还是不要刻意去想着简化数据了, 防止没想清楚导致做错。□

```
1  #include<stdio.h>
2  #include<string.h>
3  int n, k, num[1005] = { 0 };//尽量稍微开大一点
4  int le[1005], ri[1005];
5  int main()
6  {
7      scanf("%d %d", &n, &k);
8      for(int i=1; i<=n; i++)//个人习惯, 整数数组下标从 1 开始, 读者按照自己的来就好
9      {
10         le[i]=i-1;//初始每个人的前一个就是比他下标小 1 的人
11         ri[i]=i+1;//初始每个人的后一个就是比他下标大 1 的人
12     }
13     le[1]=n; ri[n]=1;//第一个人的上一个和第 n 个人的下一个是第 1 个, 构成环状
14     int remain = n, cnt = 0, j = 1;
15     if(n == 1)
16     {
17         printf("1");
```



```
18         return 0; +
19     }
20     while(remain > 1)
21     {
22         cnt += 1;
23         if (cnt == k)
24         {
25             le[ri[j]] = le[j]; //当前人前面一个的下一个, 更新为当前人的下一个
26             ri[le[j]] = ri[j]; //当前人后面一个的上一个, 更新为当前人的上一个
27             //这两句是链表删除节点的核心语句
28             cnt = 0;
29             remain -= 1;
30             if(remain == 1) //如果当前人剔除后只剩一个了
31             {
32                 printf("%d", ri[j]);
33                 //直接输出当前人的下一个, 也就是最后的幸存者, 然后直接结束进程
34                 return 0;
35             } //但是这样的风险是, 程序最后没有输出了, 所以 n=1 要特判一下
36         }
37         j = ri[j]; //j 变为 j 的下一个人, 这就是链表的神奇之处, 抛弃了传统的 1234 下标
38     }
39     return 0;
40 }
```

## 8.5 C087-按规则删除星号

**题目描述 73.** 编写程序, 删除字符串中间的所有星号 \*, 如果首尾有 \* 号, 保留不变。字符串长度不超过 200。

如: 输入 `***street*****music***`, 则输出 `***streetmusic***`

输入: 带 \* 号的字符串 (字符串中不含空格)。

输出: 中间没有 \* 号的字符串。

**思路.** 找到第一个不是星号的位置 st, 和倒着书上来第一个不是星号的位置 ed, 然后输出中间这一段中不是星号的内容即可 (无需单独写一个删除函数, 只要输出的时候判断一下是不是星号就行)。 □

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[100];
4 int main()
5 {
6     gets(s);
7     int st=0, ed=strlen(s)-1;
8     while (s[st] == *)
9         st++;
10    //这个 while 结束后, st 指向从前往后第一个不是 * 的位置
11    while (s[ed] == *)
```



```
12         ed--;
13         //这个 while 结束后, ed 指向从后往前第一个不是 * 的位置
14         for(int i=st; i<=ed; i++)
15             if(s[i]!=*) printf("%c",s[i]);
16         return 0;
17     }
```

## 8.6 C106-新兵报数问题

**题目描述 74.** 某部队进行新兵队列训练, 将新兵从 1 开始按顺序依次编号, 并排成一行横队, 训练的规则如下: 从头开始 1 至 2 报数, 凡报到 2 的出列, 剩下的向小序号方向靠拢, 再从头开始进行 1 至 3 报数, 凡报到 3 的出列, 剩下的向小序号方向靠拢, 继续从头开始进行 1 至 2 报数, 以后从头开始轮流进行 1 至 2 报数、1 至 3 报数直到剩下的人数不超过三人为止。编写程序, 输入数  $N$  为最开始的新兵人数 ( $2 < N < 6000$ ), 输出剩下的新兵最初的编号。

思路. 按照题意进行模拟即可。具体实现见代码。

□

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  int n;
4  int a[6005];
5  int Compress(int a[], int n)
6  { // 将数组中的 0 剔除, 同时非 0 项前移
7      for (int i = 0; i < n; i++)
8      {
9          if (a[i] == 0)
10         {
11             for (int k = i; k <= n - 1; k++)
12                 {
13                     a[k] = a[k + 1];
14                 }
15             n = n - 1;
16         }
17     }
18     return n;
19 }
20 int main()
21 {
22     scanf("%d", &n);
23     for (int i = 0; i < n; i++) //初始化数组
24         a[i] = i + 1;
25     int flag = 0; //区分是 1-2 报数还是 1-3 报数
26     while(n > 3) //剩下的人数超过 3 人
27     {
28         if (flag == 0) //1-2 报数
29         {
30             for (int i = 0; i < n; i++)
```



```
31         {
32             if ((i + 1) % 2 == 0)
33             {
34                 a[i] = 0; //出队的标记为 0
35             }
36         }
37         flag = 1;
38     }
39     else //1-3 报数
40     {
41         for (int i = 0; i < n; i++)
42         {
43             if ((i + 1) % 3 == 0)
44             {
45                 a[i] = 0;
46             }
47         }
48         flag = 0;
49     }
50     n = Compress(a, n); //去掉数组中的 0, 所有的数据向前移动
51 }
52 if(n==3) printf("%d %d %d", a[0], a[1], a[2]);
53 else printf("%d %d", a[0], a[1]);
54 return 0;
55 }
```

## 8.7 C107-输出不是两个数组共有的元素

**题目描述 75.** 输入两个整型数组（假设数组的大小为不超过 100）的各个元素，输出不是两个数组共有的元素，并从小到大排序输出。

允许一个数组内具有重复的元素，如该元素在另一个数组中没有（非共有）则所有重复的元素都需要输出。

输入：输入分为 2 行，分别输入 2 个数组中的整数，每行的第一个整数  $N > 0$  指示该数组需要输入的整数个数，随后输入  $N$  个整数。

输出：两个数组的元素中非共有的元素，并且从小到大进行排序。

**思路.** 分别查找在第一个数组但是不在第二个数组中的元素和在第二个数组但是不在第一个数组中的元素，拼接到一个 ans 数组之后进行排序输出即可。 □

```
1 #include<stdio.h>
2 int n,m;
3 int a[30],b[30];
4 int ans[30];
5 int Find(int a[], int b[], int na, int nb, int c[], int nc)
6 { //寻找在数组 a 中，但不在数组 b 中的数，将其从数组 c 的 nc 处开始存放，并返回找到的个数
7     int n = 0;
8     for (int i = 0; i < na; i++)
9     {
```



```
10     int flag = 0;
11     for (int j = 0; j < nb; j++)
12     {
13         if (a[i] == b[j]) //有重复的则跳出循环
14         {
15             flag = 1;
16             break;
17         }
18     }
19     if (!flag) //找到没有重复的
20     {
21         c[nc + n] = a[i];
22         n++;
23     }
24 }
25 return n;
26 }
27 void Bubble(int b[], int n)
28 {
29     for (int i = 0; i < n; i++)
30     {
31         for (int j = n - 1; j > i; j--)
32         {
33             if (b[j] < b[j - 1])
34             {
35                 int t = b[j];
36                 b[j] = b[j - 1];
37                 b[j - 1] = t;
38             }
39         }
40     }
41 }
42 int main()
43 {
44     scanf("%d", &n);
45     for (int i = 0; i < n; i++)
46         scanf("%d", &a[i]);
47     scanf("%d", &m);
48     for (int i = 0; i < m; i++)
49         scanf("%d", &b[i]);
50     int num=Find(a,b,n,m,ans,0); //查找在第一个数组中不在第二个数组中的数
51     num+=Find(b,a,m,n,ans,num); //查找在第二个数组中不在第一个数组中的数
52     Bubble(ans, num); //排序
53     printf("%d", ans[0]);
54     for (int i = 1; i < num; i++)
55         printf(" %d", ans[i]);
```



```
56     return 0;  
57 }
```

## 8.8 C142-分数求和

**题目描述 76.** 给定  $N$  个“分子/分母”形式的有理数，计算这些分数的和。

输入格式：第一行输入一个正整数  $N$  ( $1 < N \leq 100$ )，然后下一行  $N$  个有理数 “ $a_1/b_1 a_2/b_2 \dots$ ”，其中所有的分子和分母都在 “ $int$ ” 的范围内。如果有负数，则符号必须出现在分子的前面。

输出格式：输出最简单形式的和，“整数分子/分母”，其中“整数”是和的整数部分，“分子” $<$ “分母”，分子和分母没有公因数。如果整数部分为 0，则必须仅输出小数部分。如果结果是一个整数，则仅输出该整数且后面没有空格。如：

```
2  
1/2 1/2  
1  
3  
-1/2 -1/2 -1/2  
-1 1/2
```

思路. 这题比较繁琐，需要根据题目意思一点点分情况讨论，具体见代码。

□

```
1 #include<stdio.h>  
2 struct Rational  
3 {  
4     int fenzi; //分子  
5     int fenmu; //分母  
6 };  
7 int n;  
8 Rational sum = {0, 1};  
9 Rational x;  
10 int Gcd(int x, int y)  
11 {  
12     int gcd = x;  
13     while (x % gcd || y % gcd)  
14     {  
15         gcd--;  
16     }  
17     return gcd;  
18 }  
19 Rational Add(Rational x, Rational y)  
20 {  
21     Rational z; //通分相加  
22     z.fenmu = x.fenmu * y.fenmu;  
23     z.fenzi = x.fenmu * y.fenzi  
24         + x.fenzi * y.fenmu;  
25     int gcd = Gcd(z.fenmu, z.fenzi);  
26     //约分
```



```
27     z.fenmu = z.fenmu / gcd;
28     z.fenzi = z.fenzi / gcd;
29     return z;
30 }
31 int main()
32 {
33     scanf("%d", &n);
34     char ch;
35     for (int i = 0; i < n; i++)
36     {
37         scanf("%d%c%d", &x.fenzi, &ch, &x.fenmu);
38         sum = Add(sum, x);
39     }
40     //输出
41     if (sum.fenzi < 0) //负数先化为正数
42     {
43         printf("-");
44         sum.fenzi = -sum.fenzi;
45     }
46     if (sum.fenmu == 1) //只有整数部分
47     {
48         printf("%d", sum.fenzi);
49     }
50     else
51     {
52         if (sum.fenzi > sum.fenmu) //整数分数都有
53         {
54             int integer = sum.fenzi / sum.fenmu;
55             sum.fenzi = sum.fenzi
56                 - sum.fenmu * integer;
57             printf("%d %d/%d", integer, sum.fenzi
58                 , sum.fenmu);
59         }
60         else //只有分数部分
61         {
62             printf("%d/%d", sum.fenzi, sum.fenmu);
63         }
64     }
65     return 0;
66 }
```

## 8.9 C111-提取字符串中的整数并输出

**题目描述 77.** 输入一个带有空格的字符串（包含字母、数字、空格以及其他符号），提取字符串中的数字转换为整数并输出，字符串中的非数字字符视为分割符。输出数据间以空格分隔，末尾没有空格，并换行输出所有整数的和值。若字符串中没有任何数字时，输出“NO”。输入的字符串长度不超过 80。



输入：一行字符串 (含有空格)。

输出格式：第一行：转换后的各整数间用一个空格分隔，第二行：和值。

思路：较为复杂的模拟题，需要一步一步进行模拟。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char split[100][100];
4 int num[100];
5 int SplitString(char str[], char ch, char r[][100])//以 ch 为分隔符，分割 str
6 {
7     int i = 0, m = 0, n = 0;
8     while(str[i]==ch && str[i]!=0)
9         i++;
10    //找到第一个不是 ch 的位置
11    while (str[i] != 0)
12    {
13        while (str[i] != ch && str[i] != 0)
14        {
15            r[m][n] = str[i];
16            n++;
17            i++;
18        }
19        r[m][n] = 0;
20        m++;
21        n = 0;
22        while (str[i] == ch && str[i] != 0)
23        {
24            i++;
25        }
26    }
27    return m;
28 }
29 int Convert(char str[])
30 {
31     int n = 0;
32     int len = strlen(str);
33     for (int i = 0; i < len; i++)
34     {
35         n = n * 10 + (str[i] - 0);
36     }
37     return n;
38 }
39
40 int main()
41 {
```



```
42     char s[1000];
43     gets(s);
44     int i = 0;
45     while (s[i])
46     {
47         if (!(s[i] >= 0 && s[i] <= 9)) //不是数字，用星号替代
48         {
49             s[i] = *;
50         }
51         i++;
52     }
53     int m = SplitString(s, *, split);
54     if (m == 0)
55     {
56         printf("NO");
57         return 0;
58     }
59     int sum=0;
60     for (i = 0; i < m; i++)
61     {
62         num[i] = Convert(split[i]);
63         sum += num[i];
64     }
65     printf("%d", num[0]);
66     for (i = 1; i < m; i++)
67         printf(" %d", num[i]);
68     printf("\n%d", sum);
69     return 0;
70 }
```

## 8.10 C146-按 ASCII 排序输出字符串

**题目描述 78.** 输入两个字符串  $s, t$ , (字符串中没有空格, 包含大小写字母、数字以及其他符号) 按从小到大顺序输出在  $s$  或  $t$  中出现过 (不区分大小写) 的所有字符, 所有字母均按大写输出 重复的字符只输出 1 次。

$s, t$  长度不超过 100。

输入: 1 行, 两个字符串中间用空格隔开, 字符串不含空格

输出: 结果字符串, 按 ASCII 编码从小到大排序。

思路. 按照题意模拟即可。

□

```
1 #include<stdio.h>
2 #include<string.h>
3 char s[105],t[105],u[205];
4 void merge(char* a, char* b)// 将 a 中的字符放入 b 中, 重复的不放入
5 {
```



```
6     int i = 0;
7     int j = strlen(b);
8     while (a[i])
9     {
10         int flag = 0;
11         for (int m = 0; m < j; m++)
12         {
13             if (a[i] == b[m]) //重复
14             {
15                 flag = 1;
16                 break;
17             }
18         }
19         if (flag == 0)
20         {
21             b[j] = a[i];
22             b[++j] = 0;
23         }
24         i++;
25     }
26 }
27 void Bubble(char* a)
28 {
29     int len = strlen(a);
30     for (int i = 0; i < len; i++)
31     {
32         for (int j = len - 1; j > i; j--)
33         {
34             if (a[j] < a[j - 1])
35             {
36                 char temp = a[j];
37                 a[j] = a[j - 1];
38                 a[j - 1] = temp;
39             }
40         }
41     }
42 }
43 int main()
44 {
45
46     scanf("%s%s", s, t);
47     __strupr(s);
48     __strupr(t); //统一大写，因为题目不区分大小写
49     merge(s, u);
50     merge(t, u);
51     Bubble(u);
```



```
52     printf("%s", u);  
53     return 0;  
54 }
```

至此，大学计算机 80 题讲义已经全部完结。这门课或许会让很多刚进入大二的同学（如果高中未曾接触计算机编程）感到迷茫与无所适从。因此，数学学院励志书院数学试验班师生联合党支部特此推出 80 题的详细讲义，希望能够帮助到大家。以上内容由于时间有限和笔者水平有限，难免存在错误。敬请各位读者联系 1272336175@qq.com 指正！