

简单算法列表：

1 两个数的值的交换

借用第三个变量来实现

```
cin>>a>>b;
```

```
c=a;a=b;b=c;
```

不借用第三个变量的方法

```
cin>>a>>b;
```

```
a=a+b;
```

```
b=a-b;
```

```
a=a-b;
```

2 小数位数的处理

```
#include<iomanip> //调用格式库
```

```
cout<<fixed<<setprecision(2)<<a; //指定 a的小数位数为 2
```

3 数字的分离

输入一个四位数，将各个位上的数分离出来，当然如果用字符串来处理也行

```
cin>>a;
```

```
b=a%10;
```

```
c=a/10%10;
```

```
d=a/100%10;
```

```
e=a/1000;
```

如果是输入一个不知道位数的整数，要将其倒序输出

```
cin>>a;
```

```
b=0;
```

```

for(;a!=0;)
{
    b=b*10+a%10; //将 a数的个位取出来累加到 b中
    a=a/10; //将 a的个位去掉
}
cout<<b;

```

4 判断一个数是否能被另一个数整除的方法

如判断一个数能否被 7整除

```

cin>>a;
if(a%7==0) cout<<" yes" ;
else cout<<" no" ;

```

如此推广到奇偶数的判断

```

cin>>a;
if(a%2==0) cout<<" 偶数 " ;
else cout<<" 奇数 " ;

```

5 计数器

```

j=0;
j=j+1;
cin>>n; j=0; //计数器初值一般为 0
for( i=1; i<=n; i++){
    cin>>a;
    if(a%2==0) j=j+1; //每次用自身加上 1再赋给自身, 即 j++
}
cout<<j;

```

6 累加器

```
s=0; //累加器的初值一般为 0

cin>>a;

s=s+a; //将新数加上自身再赋给自身

cin>>n;s=0;

for(i=1;i<=n;i++){

    cin>>a;

    s=s+a;

}

cout<<s;
```

7 累乘器

```
s=1; //累乘器的初值一般为 1

cin>>a;

s=s*a; //将新数乘上自身再赋给自身

cin>>n;s=1;

for(i=1;i<=n;i++){

    cin>>a;

    s=s*a;

}

cout<<s;
```

8 标记法（求素数）

输入整数 n,请判断其是否为质数。

```
cin>>n;

f=1;    //用 f 做标记，为 1则表示是素数
```

```

for( j=2;j<n;j++ )
    if(r%j==0) f=0;    //f为 0, 则表示 n在 2~n-1之间有一个约数
if(f==1) cout<<" yes" ;
else cout<<" no" ;

```

9 求最大值

先假设输入的第一个数即为最大数，然后用它去和后面的数一一进行比较，如果后面的数比它大，则将大的数交换给它。

```

cin>>a;
max=a;
for(i=1;i<n;i++)
    { cin>>a; if(a>max) max=a;}

```

10 求最小值

先假设输入的第一个数即为最小数，然后用它去和后面的数一一进行比较，如果后面的数比它小，则将小的数交换给它。

```

cin>>a;
min=a;
for(i=1;i<n;i++)
    { cin>>a; if(a<min) min=a;}

```

11 字符的处理

出入一串字母和数字，以 ” . “ 号结束，统计其中字母和数字的个数。

```

zimu=0; shuzi=0;
for( cin>>a; a!=' . ' ; )
{
    if(a>=' a' &&a<=' z' ||a>=' A' &&a<=' Z' ) zimu++;
}

```

```

    if(a>=' 0' &&a<=' 9' ) shuzi++;
    cin>>a;
}

```

输入一串小写字母，以 ” . “ 号结束，输出字母所对应的数字，规定 a对应数字 1, b对应数字 2, z对应数字 26, 输出的数字间以空格隔开。

```

for( cin>>a; a!=' . ' ;)
{
    if(a>=' a' &&a<=' z' ) n=a- ' a' +1;
    cout<<n<<' ' ;
    cin>>a;
}

```

//字母类型的数字要转为对应的整数可以采取同样的办法，

```
// int n; char a; cin>>a; n=a- ' 0' ;
```

空格的处理

cin命令做输入时会自动跳过空格和回车等符号，如果要求对空格进行处理则需要用到 cin.get(a); 也可以用 scanf(“ %c” , &a); 但不能用 cin>>a;

12 数学函数库

有一些标准数学函数在信息奥赛中是允许使用的，使用时要加上数学函数库

```
#include<cmath>
```

```
double a,b,c,d;
```

```
cin>>a>>b;
```

```
c=sqrt(a); //sqrt为求平方根函数
```

```
d=pow(a,b); //pow为求 ab的函数，求平方可以用它，注意它的参数为实数类型
```

13 求最大公约数

求两自然数 m , n 的最大公约数。最大公约数：能同时被 m 和 n 整除的最大数。

如 6 和 4 的最大公约数为 2

1) 欧几里德算法 ($m > n$)

m 被 n 除得到余数 r ($0 \leq r < n$) $r = m \% n$

若 $r=0$, 则算法结束, n 为最大公约数, 否则做 3

$m=n$, $n=r$, 回到 1

使用的是循环迭代的方法

$m=6$ $n=4$ $r=m \% n = 6 \% 4 = 2$

$m=4$ $n=2$ $r=m \% n = 4 \% 2 = 0$

所以, 公约数 $= 2$

程序代码：

```
cin >> m >> n;

for (r = m % n; r != 0; r = m % n)
{ m = n; n = r; }

cout << n;
```

2) 穷举

最大公约数只能是 $n, n-1, \dots, 2, 1$ 中的一个, 一一列举出来试一下即可。

```
cin >> m >> n;

for (i = n; i > 0; i--)

    if (m % i == 0 && n % i == 0) { cout << i; return(0); }
```

虽然两种算法都可以求出最大公约数, 但执行效率不同, 穷举显然循环次数最多, 最费时。

比如 3000 2002这两个数，用穷举要执行 2001次循环，而欧几里德算法只需要 4次，算法的优劣一目了然。

14 求最小公倍数

最小公倍数：能同时整除 m 和 n 的最小数。

1) 已知最大公约数时的方法

最小公倍数为两数之积除以最大公约数。接上例 4和 6的最小公倍数为 $4*6/2=12$

2) 直接尝试

```
cin>>m>>n;

for (t=m;n%t!=0;m=m+t);

cout<<m;
```

将较大数的值放在变量 t 中，每次用 m 直接加上 t 去尝试能否除尽 n , 如果能，现在的 m 就是最小公倍数。

大多时候还是应该选择使用第一种方法，仍以 3000 2002为例，用第一种方法只需要 4次循环求出最大公约数，然后求出最小公倍数，而第二种方法要执行上千次才能出结果。

15 穷举法

穷举法也称为“枚举法”，这种算法基本思想是依题目的部分条件，确定答案的大致范围；在此范围

内，对所有可能的情况一一列举，逐一验证，直到全部情况验证完毕，或者得到了需要的结果。

若某个情况经验证符合题目的全部条件，则它就是本题的一个答案；若全部情况经验证后，都不符合

题目的全部条件，则原题无解。

用穷举法解题的大致步骤如下：

- 1) 分析题目，确定所要求的解是什么？
- 2) 确定解的可能取值范围是什么？
- 3) 穷举出所有可能的解
- 4) 验证每一个可能的解
- 5) 优化

例：鸡兔同笼问题

一个笼子里有鸡和兔，现在只知道里面一共有 a 个头， b 只脚，问鸡和兔各有多少只
(要求鸡和兔至少 1 只)？

```
int i,j,a,b;
cin>>a>>b;
for(i=1;i<a;i++)    //鸡可能为 1~a-1只
    for(j=1;j<a;j++)    //兔可能为 1~a-1只
        if(i+j==a&&2*i+4*j==b) cout<<i<<' ' <<j<<endl;
```

16 递推的思想

迭代的方法

例：斐波那契数列（从第三项起为前面两项之和）

1 1 2 3 5 8 13..

```
cin>>n;
a1=1;
a2=1;
for(i=3; i<=n; i++)
{
    a3=a1+a2;
```



```

a1=a2;

a2=a3;

}

```

```

cout<<a3;

```

猴子吃桃问题。猴子摘了一堆桃，第一天吃了一半，还嫌不过瘾，又吃了一个；第二天又吃了剩下的一半零一个；以后每天如此。到第 N天，猴子一看只剩下一个了。问最初有多少个桃子？

```

cin>>n;

s=1;

for(i=2;i<=n;i++)

    s=(s+1)*2;

cout<<s;

```

17 文件的处理方法

```

#include<fstream>

using namespace std;

ifstream fin(“ 输入文件名 ” );

ofstream fout(“ 输出文件名 ” );

```

例：

陶陶摘苹果 (文件名：apple.cpp)【问题描述】

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出 10个苹果。苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有个 30厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知 10个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度，

请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

【输入文件】输入文件 apple.in 包括两行数据。第一行只包括一个 100 到 120 之间（包含 100 和 120）的整数（以厘米为单位），表示陶陶把手伸直的时候能够达到的最大高度。第二行包含 10 个 100 到 200 之间（包含 100 和 200）的整数（以厘米为单位）分别表示 10 个苹果到地面的高度，两个相邻的整数之间用一个空格隔开。

【输出文件】输出文件 apple.out 包括一行，这一行只包含一个整数，表示陶陶能够摘到的苹果的数目。

【样例输入】

```
110
100 200 150 140 129 134 167 198 200 111
```

【样例输出】

```
5
#include<fstream>
using namespace std;
ifstream fin("apple.in");
ofstream fout("apple.out");
int main(){
    int rg,sg,i,j=0;
    fin>>rg;
    for(i=0;i<10;i++){
        fin>>sg;
        if(sg<=rg+30) j++;
    }
```

```
        fout<<j;
    }
```

多重循环讲解与练习

```
for( ; ; )
{
    语句组 ;
    for( ; ; )
    {
        语句组 ;
    }
    语句组 ;
}
```

1 图形字符类

例 1 输出 m行 n列 *号

```
cin>>m>>n;

for( i=0;i<m;i++ ) //外循环通常用作控制行，表示输出图形有 m行
{
    for( j=0;j<n;j++ ) //内循环通常用作列，表示一行上有多少列字符
    {
        cout<<' *' ;
    }
}
```

```
cout<<endl;
```

```
}
```

外循环 $i=0$ 时，内循环 j 从 0 一直变化到 $n-1$ ，即内循环中的语句 { `cout<<' *' ;` } 被执行了 n 次，语句 `cout<<endl;` 是外循环中的语句，执行 1 次。当外循环的 i 值变化到 m 退出时，内循环中的语句 { `cout<<' *' ;` } 一共被执行了 $m \times n$ 次。

有时每列上的字符个数不同，如

例 2 $n=3$ 时，输出如下图形

```
*
```

```
**
```

```
***
```

```
cin>>n;
```

```
for( i=0;i<n;i++ )
```

```
{
```

```
    for( j=0;j<=i;j++ )
```

```
    {
```

```
        cout<<' *' ;
```

```
    }
```

```
    cout<<endl;
```

```
}
```

有时每列上的字符个数不向上面图形一般每行以 1 递增，如

例 3 $n=3$ 时，输出如下图形

```
*
```

```
***
```

```
*****
```

```

cin>>n;
for( i=1;i<=n;i++ )
{
    for( j=1;j<=2*i-1;j++ )//每行上的字符以 2递增，因此是 2*i
    {
        cout<<' *' ;
    }
    cout<<endl;
}

```

如果是这样的金字塔形式

例 4 n=3, 输出

```

*
**
***

```

可以看作是先输出了一个倒三角形的空格，然后再输出上例中的图形

```

cin>>n;
for( i=1;i<=n;i++ )
{
    for(j=1;j<=n-i;j++) //每行上的空格数是 n-1,n-2,...,1,0这样一个递减序列，而 i每次递增 1，要将一个递增序列改为递减序列，直接在前面加上 -号
    cout<<' ';
    for( j=1;j<=2*i-1;j++ )//每行上的字符以 2递增，因此是 2*i
    {
        cout<<' *' ;
    }
}

```

```

    }

    cout<<endl;

}

```

练习：可以尝试打出如下图形：

如 n=3, 要求输出

```

    * * *

    * * *

* * *

    *

    * * *

* * * * *

    * * *

    *

    *           *

    * * *      * * *

* * * * * * * * * *

    *                   *

    * * *              * * *

* * * * * * * * * * * *

                * * *

                    *

a

bbb

```

CCCCC

bbb

a

2 逻辑推理题目

解逻辑判断与推理题这类题首先要把复杂的逻辑关系进行分析、抽象、化简，然后进行尝试排除，得到最后结果。可以用计算机来验证推断的结果，复杂的还可以直接用逻辑表达式来表示各种关系条件，用计算机穷举各种可能情况，再根据条件采取累试排除或选择需要的组合。

【例 1】四位同学不知是哪一位打碎了教室窗户的玻璃，老师问是哪一个同学打的。A 说：“不是我”，B 说：“是 C”，C 说：“是 D”，D 反驳说：“他胡说！”。四个人当中只有一个人说假话，试判断哪一位打碎了教室窗户的玻璃？

分析：用 A,B,C,D 分别表示这四位同学，用 1 表示打碎了玻璃，0 表示没有打碎。

题目中四句话可表示成：

$$A \neq 1$$

$$C = 1$$

$$D = 1$$

$$D \neq 1$$

A,B,C,D 只有一位同学打碎了玻璃，因此 $A+B+C+D$ 的值为 1。

逻辑表达式只有 1 和 0 两个值，为 1 表示真，为 0 表示假，

只有一个人说假话，则说明四个表达式的和应为 3。

程序如下：

```
for(A=0;A<2;A++)
```

```
for(B=0;B<2;B++)
```

```

for(C=0;C<2;C++)
    for(D=0;D<2;D++)
if (A+B+C+D==1&&(A!=1)+(C==1)+(D==1)+(D!=1)==3)
    { if(A==1) cout<<'A' ;
      if(B==1) cout<<'B' ;
      if(C==1) cout<<'C' ;
      if(D==1) cout<<'D' ;
    }

```

四个循环分别用于穷举 A,B,C,D四位同学打碎玻璃和没有打碎玻璃的所有情况，加上判断就可以确定到底是谁打碎了玻璃。

也可以用其他方法来做，尝试一下。

【例 2】有四个学生对我国四大淡水湖排列次序如下：

甲：洞庭湖最大，洪泽湖最小，鄱阳湖第三。

乙：洞庭湖最小，洪泽湖最大，鄱阳湖第二，太湖第三。

丙：洞庭湖第三，洪泽湖最小。

丁：洪泽湖第二，鄱阳湖最大，太湖最小。

每人都只说对了一个，试编程排出正确次序。

分析：用 d h p t 分别表示洞庭湖、洪泽湖、鄱阳湖、太湖。

四个湖泊的顺序号只能是 1 2 3 4 的不同组合，所以 $d+h+p+t=10$

则四个人的话可表示为：

甲： $d=1, h=4, p=3$

乙： $h=1, d=4, p=2, t=3$

丙： $h=4, d=3$

丁： $p=1, t=4, h=2, t=3$

由于每人都只说对了一个，所以上述表示每个人的观点的式子中只有一个真值为1，其他为0

程序如下：

```
for(d=1;d<5;d++)
    for(h=1;h<5;h++)
        for(p=1;p<5;p++)
            for(t=1;t<5;t++)
                if((d!=h)&&(d!=p)&&(d!=t)&&
                    (h!=p)&&(h!=t)&&(p!=t)&&
                    (d==1)+(h==4)+(p==3)==1&&
                    (d==4)+(h==1)+(p==2)+(t==3)==1&&
                    (d==3)+(h==4)==1&&
                    (h==2)+(p==1)+(t==4)==1)
                { cout<<"dong ting hu is:"<<d<<endl;
                  cout<<"hong ze hu is:"<<h<<endl;
                  cout<<"po yang hu is:"<<p<<endl;
                  cout<<"tai hu is:"<<t<<endl;
                }
```

程序写得稍嫌复杂，实际上可以简便一些。

```
for(d=1;d<5;d++)
    for(h=1;h<5;h++)
        if(d!=h)
```

```

for(p=1;p<5;p++)
if(p!=h&&h!=d)
{ t=10-d-h-p;
  if((d==1)+(h==4)+(p==3)==1)
    if((d==4)+(h==1)+(p==2)+(t==3)==1)
      if((d==3)+(h==4)==1)
        if((h==2)+(p==1)+(t==4)==1)
          { cout<<"dong ting hu is:"<<d<<endl;
            cout<<"hong ze hu is:"<<h<<endl;
            cout<<"po yang hu is:"<<p<<endl;
            cout<<"tai hu is:"<<t<<endl;
          }
        }
      }
    }
  }
}

```

循环减少一个，表达式分开写，看起来要清晰些。

练习：

(1) 某年级数学竞赛中，A、B、C、D、E五位同学获得前五名。A说：“D是第一，C是第四”；B说：“A是第一，E是第二”；C说：“B是第二，D是第三”；D说：“E是第三，A是第五”；E说：“E是第二，C是第四”。每人都只说对了-一个，试编程排出正确次序。

(2) 甲、乙、丙三位同学中有一人趁大家不在教室的时候把教室打扫干净了。事后老师问是谁做的好事。甲说：“是乙做的”，乙说：“不是我”，丙说：“不是我”。三个人当中只有一个人说真话，试判断哪一位做了好事？

(3) 有红、黄、蓝、白四色球各一个，放置在编号为 1 2 3 4 四个格子的盒中，每个格子中只能放一个小球，他们的顺序不知。甲、乙、丙三人猜测的顺序如下：甲说：蓝色球编号为 1, 黄色球编号为 2; 乙说：蓝色球编号为 2, 白色球编号为 3; 丙说：红色球编号为 2, 白色球编号为 4。结果证明三个人各猜中一半。问四个格子的盒中放置小球情况。

一维数组讲解与练习

一、为什么要使用数组

学到现在，我们用分支结构解决了计算机的智能判断问题，用循环充分利用了计算机处理速度快的特点，也可以用循环对大量数据进行处理，但这些数据却无法大量保存，在这点上还没有利用上计算机存储容量大的特点。在程序语言中有一种数据结构可以解决这个问题，那就是数组。

例 1 输入 1000 个学生的某门课程的成绩，打印出低于平均分的同学号数与成绩。

分析：在解决这个问题时，虽然可以通过读入一个数就累加一个数的办法来求学生的总分，进而求出平均分。但因为只有读入最后一个学生的分数以后才能求得平均分，且要打印出低于平均分的同学，故必须把 1000 个学生的成绩都保留下来，然后逐个和平均分比较，把高于平均分的成绩打印出来。如果，用简单变量 $a_1, a_2, \dots, a_{1000}$ 存放这些数据，可想而知程序要很长且繁。

要想如数学中使用下标变量 a_i 形式表示这 1000 个数，则可以引入下标变量 $a[i]$ 。这样问题的程序可写为：

```
tot=0; //tot表示总分
```

```
for(i=1;i<=1000;i++) //读入每一个学生的成绩，并累加它到总分 }
```

```
{
```

```

    cin>>a[i];

    tot:=tot+a[i];

}

ave=tot/1000; //计算平均分

for( i=1;i<=1000;i++)

    if (a[i]<ave) cout<<" No." <<i<<': '<<a[i]; //如果第 i个同学成绩小于平均分，则将其输出

```

而要在程序中使用下标变量，则必须先说明这些下标变量的整体——数组，即数组是若干个同名（如上面的下标变量的名字都为 a）下标变量的集合。

二、一维数组

1、一维数组的定义和引用

当数组中每个元素只带有一个下标时，我们称这样的数组为一维数组。

数组是同一类型的一组值（10个 char、15个 int..），在内存中顺序存放。

整个数组共用一个名字，而其中的每一项又称为一个元素。

1)定义方式：

类型说明符 数组名 [常量表达式];

```
int    a[4];           // int是数组元素的类型说明，4表示元素个数，a表示数组名称，整个说明表明 a数组由 4个 int型元素组成，这四个元素分别是：a[0]，a[1]，a[2]，a[3]
```

要注意的两个地方：

其序号从 0开始，而不是从 1开始。所以 int a[4];实际上不包含 a[4]这个元素

C++不允许对数组的大小作动态的定义，即数组的大小不能是变量，必须是常量。即不能定义为 int a[n];只能是 int a[1000];等，即元素个数必须是确定的。

2) 一维数组元素的引用

数组必须先定义，具体引用时（赋值、运算、输出）其元素等同于变量。

数组元素的赋值与输出通常采用的是循环。

例 2 输入 100个数，将其倒序输出。

```
int main( )
{
    int i, a[100];
    for ( i=0; i<100; i++)
        cin>>a[i];
    for ( i=99; i>=0 ; i--)
        cout<<a[i]<<' ' ;
}
```

3) 一维数组的初始化

除了用循环读入处理外，一维数组还可以直接进行赋值操作。

```
int a, b = 0, c[5] = { 1, 2, 3, 4, 5 };
```

此例也可以写成，int c[5]; c[0]=1; c[1]=2; c[2]=3; c[3]=4; c[4]=5;

注意：不能给数组整体赋值，只能一个一个地赋值

比如上例写成：int a, b = 0, c[5] = { 1, 2, ..., 5 }就是错误的。

2 一位数组的应用实例

例 3 求 Fibonacci数列：1,1,2,3,5,8,...的前 20个数，将它们 5个数一行的输出，每两个数之间隔一个制表符的距离。

分析：Fibonacci数列可表示为以下数学递推公式：

$$F_1=1 \quad (n=1)$$

$$F_2=1 \quad (n=2)$$

$$F_n=F_{n-1}+F_{n-2} \quad (n \geq 3)$$

```
int main( )
```

```

{      int    i;

        int    f [21];

f[1]=1;

f[2]=1;

        for (i=2 ; i<21 ; i++ )

                f [i]=f [i-1]+f [i-2];

for ( i=1; i<21; i++)

{

        cout<<f [i]<<" \t" ;

if (i%5== 0)    cout<<" \n" ;

}

}

```

例 4 输入十个正整数 把这十个数按由大到小的顺序排列。

将数据按一定顺序排列称为排序，排序的算法有很多，其中选择排序是一种较简单的方法。

分析：要把十个数按从大到小顺序排列，则排完后，第一个数最大，第二个数次大，……。因此，我们第一步可将第一个数与其后的各个数依次比较，若发现比它大的，则与之交换，比较结束后，则第一个数就是最大的数。同理，第二步，将第二个数与其后各个数再依次比较，又可得出次大的数。如此方法进行比较，最后一次，将第九个数与第十个数比较，以决定次小的数。于是十个数的顺序排列结束。

例如下面对 5个数进行排序，这 5个数分别为 8 2 9 10 5 按选择排序方法，过程如下：

初始数据 ： 8 2 9 10 5

第一次排序： 8 2 9 10 5

9 2 8 10 5

10 2 8 9 5

10 2 8 9 5

第二次排序：10 8 2 9 5

10 9 2 8 5

10 9 2 8 5

第三次排序：10 9 8 2 5

10 9 8 2 5

第四次排序：10 9 8 5 2

对于十个数，则排序要进行 9次。源程序如下：

```
int i,j,a[10];  
for (i=0;i<10;i++) cin>>a[i]; //读入 10个初始数据  
for(i=0;i<9;i++) //进行 9次排序  
    for(j=i+1;j<10;j++) //将第 i个数与其后所有数比较  
        if(a[i]<a[j]) //若有比 a[i]大 则与之交换  
            {t=a[i];a[i]=a[j];a[j]=t;}  
for (i=0;i<10;i++) cout<<a[i]<<" \t" ;
```

练习：

看程序写结果：

```
#include<iostream>  
using namespace std;  
int main(){  
    int a[7],i,k=0;  
    for(i=0;i<=6;i++)
```

```

        if(i%2) a[i]=k+1;

        else a[i]=k+2;

        cout<<a[6]-a[0];

    }

```

output:

```

#include<iostream>

using namespace std;

int main(){

    int a[30],n,i;

    cin>>n;

    a[0]=0;a[1]=1;

    for(i=2;i<=n;i++) a[i]=a[i-1]+a[i-2];

    for(i=0;i<=n;i++) cout<<a[i]<<" \t" ;

    cout

}

```

输入：10

输出：

```

#include<iostream>

using namespace std;

int main(){

    int i,s,max,a[10];

    for(i=0;i<10;i++) cin>>a[i];

    max=a[0];s=a[0];

    for(i=1;i<10;i++){

```



```

        if(s<0) s=0;

        s=s+a[i];

        if(s>max) max=s;

    }

    cout<<"max="<<max;

}

```

输入 : -2 13 -1 4 7 8 -1 -18 24 6

输出 : max=

```

#include<iostream>

using namespace std;

int main(){

    int a['z'];

    char b;

    for(b='a';b<='z';b++) a[b]=0;

    for(cin>>b;b!='?'){

        if(b>='a'&&b<='z') a[b]++;

        cin>>b;

    }

    for(b='a';b<='z';b++)

        if(a[b]>0) cout<<b<<': '<<a[b]<<endl;

}

```

输入 : how many apples are there on the table?

输出 :

编程练习 :

1. 输入一串小写字母（以 "."为结束标志），统计出每个字母在该字符串中出现的次数（若某字母不出现，则不要输出）。

例：

输入：aaaabbbccc.

输出：a:4

b:3

c:3

提示：凡有序类型的数据都可以做数组的下标，如 `int a['z']`；可以表示以小写字母为下标的 26 个从 `a['a']` 到 `a['z']` 的整数变量。这样设定后就可以很方便的进行计数处理。

2. 输入一个不大于 1000000000 的正整数 N，将它转换成一个二进制数。

例如：

输入：100

输出：1100100

提示：十进制转换成二进制的过程是用十进制数除以 2，然后用数组将余数保存起来，再用商除以 2，如此循环直到商为 0 为止，最后将余数倒序输出。

3. 输入一个由 10 个整数组成的序列，其中序列中任意连续三个整数都互不相同，求该序列中所有递增或递减子序列的个数。

例如：

输入：1 10 8 5 9 3 2 6 7 4

输出：6

对应的递增或递减子序列为：

1 10

10 8 5

5 9

9 3 2

2 6 7

7 4

提示：这个就自己看了。

逻辑推理类：

例 1 有标号为 A B C D和 1 2 3 4的 8个球，每两个球装一盒，分装 4盒。标号为字母的球与标号为数字的球有着某种一一对应的关系（称为匹配）并已知如下条件：

匹配的两个球不能在一个盒子内；

2号匹配的球与 1号球在一个盒子里；

A号和 2号球在一个盒子里；

B匹配的球和 C号球在一个盒子里；

3号匹配的球与 A号匹配的球在一个盒子里；

4号是 A或 B号球的匹配球；

D号与 1号或 2号球匹配。

请写出这四对球匹配的情况。

【答案】这四对球匹配的情况如下：

A B C D

4 3 1 2

本题是一个推理题，考查学生对问题分析判断和逻辑推理能力。

1) 由 可知 2号球与 A球放在一个盒中，所以 A球只能与 1 3 4球匹配，如果 A与 1号匹配，则由 、 可知 1号球要放在两个盒中，所以 A球不能与 1匹配；

若 A球与 3 号球匹配，则由 得 A号球与 3号球放在一个盒中，所以 A球只能与 4 号球匹配；

2) B球仅能与 1 2 3号球匹配，若与 1号球匹配，则由 可知 1号球需和 C球在一个盒子中与条件 矛盾；若与 2号球匹配也与条件 矛盾，所以 B球仅能与 3 号球匹配；

3) 由 可知，D球仅能与 1 2号球匹配，若与 1号球匹配，则由 得到 1号球与 2号球的匹配球在一个盒子，则 C与 1号球在一个盒子内，这样与 条件矛盾，所以 D球仅能 2号球匹配，因而得 C则和 1球匹配。

例 2 有 100张的一摞卡片，玲玲拿着它们，从最上面的一张开始按如下的顺序进行操作：把最上面的第一张卡片舍去，把下一张卡片放在这一摞卡片的最下面。再把原来的第三张卡片舍去，把下一张卡片放在最下面。反复这样做，直到手中只剩下一张卡片，那么剩下的这张卡片是原来那一摞卡片的第几张？

分析与解：可以从简单的不失题目性质的问题入手，寻找规律。列表如下：

卡片总																	
数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17 ...
剩下第几																	
张	1	2	2	4	2	4	6	8	2	4	6	8	10	12	14	16	
	2	...															

(1) 当 $N=2a$ ($a=0, 1, 2, 3, \dots$) 时，剩下的这张卡片是原来那一摞卡片的最后一张，即第 $2a$ 张；

(2) 当 $N=2a+m$ ($m< 2a$) 时，剩下的这张卡片是原来那一摞卡片的第 $2m$ 张。

取 $N=100$, 因为 $100=26+36$, $2\times 36=72$, 所以剩下这张卡片是原来那一摞卡片的第 72张。

练习：

在 a,b,c,d,e,f 六件物品中，按下面的条件能选出的物品是：

- (1)a,b 两样至少有一样
- (2)a,d 不能同时取
- (3)a,e,f 中必须有 2 样
- (4)b,c 要么都选，要么都不选
- (5)c,d 两样中选一样
- (6)若 d 不选，则 e 也不选

推数：已知 $D=5$ ，各字母都代表不同的数字，请推算出其他字母代表的数字。

$$\begin{array}{r} \text{DONALD} \\ + \text{GERALD} \\ \hline \text{ROBERT} \end{array}$$

请写出结果并简述推理过程。

排列和组合类

在我们解决实际问题中，有许多问题需要将结果一一列举出来。如，四个人照相，问他们有多少种排列方法，并输出排列的形式，又如 10 个朋友见面，每两个人握一次手，共有多少种方法，输出所有握手情况等，诸如此类问题很多，可以用数学中的排列和组合问题解决。

在学习排列组合以前，我们先学习加法原理和乘法原理。

加法原理与乘法原理

（一）加法原理

引例 1 从甲地去乙地，可以乘火车，可以乘汽车，还可以乘轮船。一天中，火车有 4 班，汽车有 2 班，轮船有 2 班，轮船有 3 班，那一天中乘坐这些交通工具从甲到乙地有多少种不同的选择

在一天中，从甲地到乙地乘火车有 4 种选择。乘汽车有 2 种选择，乘轮船有 3 种选择，以上无论选择了哪一种方法，都有可以从甲地到达乙地。因此，一天当中乘坐这些交通工具从甲地到乙地的不同选择共有：

$$4+2+3=9(\text{种})$$

把“从甲地到乙地”看成为“完成一件事”，完成它有三类方法（火车、汽车、轮船），每类方法都可以独立完成。

第一类有 4 种方法（火车有 4 班）

第二类有 2 种方法（汽车有 2 班）

第三类有 3 种方法（轮船有 3 班）

因此完成一件事（从甲地到乙地）共有： $4+2+3=9$ 种不同的方法。

加法原理

加法原理 做一件事，完成它有 n 类方法，第一类有 m_1 种，第二类有 m_2 种，……，第 n 类有 m_n 种，那么样完成这件事共有：

$$N=m_1+m_2+\dots+m_n$$

加法原理的特点是：分类独立完成。

例 1 书架上层有不同的数学书 15 本，中层有不同的语文书 18 本，下层有不同的物理 7 本。现从其中任取一本书，问有多少种不同的取法？

解：从书架上任取一本书，有三类取法：第一类取法是从书架上层取出一本数学系书，可以从 15 本中任取一种，有 15 种取法；第二类取法是从书架的第二层取出一本语文书，可以从 18 本中任取一种，有 18 种取法；第三类取法是从书架的下层取出一本物理书，可从 7 本中任取一种，有 7 种取法。只在书架上任意取出一本书，任务即完成，根据加法原理，不同的取法一共有：

$$N=m_1+m_2+m_3=15+18+7=40(\text{种})$$

例 2 某班同学分成甲、乙、丙、丁 4 个小组，甲组 9 人，乙组 11 人，丙组 10 人，丁组 9 人。现要求该班选派一人去参加某项活动，问有多少种不同的选法？

解：该班同学分成甲、乙、丙、丁四个小组，从任何一个小组中选出一名同学去参加活动，则任务完成。在甲组有 9 种选法，乙组有 11 种选法，丙组有 10 种选法，丁组有 9 种选法，所以一共有：

$$N=9+11+10+9=39$$

巩固性练习：

1 一件工作可以用 2 种方法完成，有 5 人会用第一种方法，另外有 4 人会用第二种方法完成，要选出 1 个人来完成这件工作，共有多少种选法？

2 一个学生要从 2 本科技书，2 本政治书，3 本文艺本中任取一本，共有多少种不同的取法？

（二）乘法原理：

由 A 地去 C 地，中间必须经过 B 地，且已知由 A 地到 B 地有 3 条路可走，再由 B 地到 C 地有 2 条路可走，那么由 A 地经 B 地到 C 地有多少种不同的走法？

这里，从 A 地到 C 地不能由一个步骤直接到达，必须经过 B 地这一步骤，从 A 地到 B 地有 3 种不同的走法，分别用 a_1 a_2 a_3 表示，而从 B 地到 C 地有 2 种不同的走法，分别用 b_1 b_2 表示。所以从 A 地经 B 地到 C 地的全部走法有：

$a_1b_1; a_1b_2; a_2b_1; a_2b_2; a_3b_1; a_3b_2,$

共计 6 种。就是从 A 地到 C 地的 3 种走法与从 C 地到 B 地的 2 种走法的乘积，即： $3 \times 2 = 6$ (种)。

把“从 A 地到 C 地”看成完成一件事，那么完成这件必须分二个步骤，二个步骤都完成了，才能完成这个事。

第一个步骤有 3 种方法 (从 A 地到 B 地)

第二个步骤有 2 种方法 (从 B 地到 C 地)

因此“完成一件事”(从 A 地到 C 地)共有 $3 \times 2 = 6$ (种)不同的方法

乘法原理

乘法原理 做一件事，完成它需要 n 个先后步骤，做第一步有 m_1 种不同的方法，做第二步有 m_2 种不同的方法，……，做第 n 步有 m_n 种不同的方法，那么完成这件事共有

$$N = m_1 \times m_2 \times \dots \times m_n$$

种不同的方法。

乘法原理的特点是：分步依次完成

例 3 书架上层有不同的数学书 15 本，中层有不同的语文书 18 本，下层有不同的物理 7 本，从中取出数学、语文、物理书各一本，问有多少种不同的取法？

解：从书上取数学、语文、物理书各 1 本，可以分成 3 个步骤完成：第一步取数学书 1 本，有 15 种不同的取法；第二步取语文书 1 本，有 18 种不同的取法；第三步取物理书 1 本，有 7 种不同的取法，符合乘法原理的条件。利用乘法原理，得到： $N = 15 \times 18 \times 7 = 1890$ (种)不同的取法。

例 4 某农场要在 4 种不同类型的土地上，引种试验 A B C D 等 4 种不同品种的小麦，问有多少种不同的试验方案？

解：第一步 先考虑 A 种小麦，可在 4 种不同类型的土地中任选一种，有 4 种选法；

第二步 考虑 B 种小麦，可在剩下的 3 种不同类型的土地中任选一种，有 3 种选法；

第三步 考虑 C 种小麦，再在剩下的 2 种不同类型的土地中任选一种，有 2 种选法；

第四步 最后考虑 D 种小麦，再在剩下的 1 种不同类型的土地中任选一种，有 1 种选法。

以上四步依次完成，才算完成，依据乘法原理，可知有 $4 \times 3 \times 2 \times 1 = 24$ 不同的试验方案。

巩固性练习：

3 在一个红色口袋中，装有 20 张分别标有 1, 2, ..., 20 的红色数字卡片，在另一个黄色口袋中，装有 10 张分别标有 1, 2, ..., 10 的黄色数字卡片。在红色口袋中摸了一张数字卡片做被加数，在黄色口袋中摸出一张数字卡片做加数，列成加法式子，一共可以列成多少个符合要求的加法式子？

4 从 A 地到 B 地有 2 条路可通，从 B 地到 C 地有 3 条路可通，从 A 地到 C 地共有多少种不同的走法？

5 一个口袋内有 5 个小球，另一个口袋内有 4 个小球，所有这些小球的颜色互不相同。

(1) 从两个口袋内任取一个小球，有多少种不同的取法？

(2) 从两个口袋内各取一个小球，有多少种不同的取法？

6 从 2, 3, 5, 7 这四个数中，取两个数出来做假分数，这些假分数有多少个？