

不设中间变量，交换a和b	<pre>#include <stdio.h> int main() { int a,b; while(scanf("%d %d",&a,&b)!=EOF) {a=a+b; b=a-b; a=a-b; printf("%d %d\n",a,b);} return 0; }</pre>
最小公倍数和最大公因数 (如果有两个数字a b，他们的最小公倍数用辗转相除法，最大公因数和最小公倍数的关系是：最大公因数*最小公倍数=a*b)	<pre>#include <stdio.h> int zuidagongyueshu(int a,int b) { int temp; while(a%b!=0) { temp=a%b; a=b; b=temp; } return b; } int zuixiaogongbeishu(int a,int b) { return a*b/zuidagongyueshu(a,b); } int main() { int a,b; scanf("%d%d",&a,&b); printf("%d %d\n",zuidagongyueshu(a,b),zuixiaogongbeishu(a,b)); return 0; }</pre>
统计单词字母数字字符	<pre>#include <stdio.h> int main() { char a[100]; gets(a); int i,n,b,c,d; n=0,b=0,c=0,d=0; for(i=0;a[i];i++) { if(a[i]>='0'&&a[i]<='9') n++; else if(a[i]>='a'&&a[i]<='z' a[i]>='A'&&a[i]<='Z') b++; else if(a[i]==' ') c++; else d++; } }</pre>

	<pre> } printf("%d %d %d %d\n",b,n,c,d); return 0; } </pre>
1021 2+22+222+2222+22222	<pre> #include<stdio.h> int main() { int i,n,sum=0,sum1=0; scanf("%d",&i); for(n=1;n<=i;n++) { sum1=sum1*10+2; sum+=sum1; } printf("%d",sum); return 0; } </pre>
1022 阶乘的和	<pre> #include <stdio.h> long long jiecheng(long long n) { long long i,s; s=n; for(i=1;i<s;i++) { n=n*(s-i); } return n; } int main() { long long n,i,sum; sum=0; scanf("%lld",&n); for(i=1;i<=n;i++) sum=sum+jiecheng(i); printf("%lld\n",sum); return 0; } </pre>
1022简洁版	<pre> #include<stdio.h> int main() { long long sum=0,m=1; int i,num; scanf("%d",&num); for(i=1;i<=num;i++) { m*=i; sum+=m; } printf("%lld",sum); return 0; } </pre>
用递归实现阶乘	<pre> int diguijiecheng(int n) { return (n==1)?n:n*diguijiecheng(n-1); } </pre>

	<pre> } </pre>
1023	<pre> #include <stdio.h> #include <math.h> double sum1(double a) { int i; double sum; sum=0; for(i=1;i<=a;i++) { sum+=i; } return sum; } double sum2(double b) { int i; double sum; sum=0; for(i=1;i<=b;i++) { sum+=pow(i,2); } return sum; } double sum3(double c) { double i; double sum; sum=0; for(i=1;i<=c;i++) { sum+=(1.00/i); } return sum; } </pre>
1024 三位水仙花数	<pre> #include <stdio.h> int main() { int a,b,c,i; a=0; b=0; c=0; for(i=100;i<1000;i++) { a=i/100; b=i/10%10; c=i%10; if(i==a*a*a+b*b*b+c*c*c) printf("%d\n",i); } return 0; } </pre>
输出m和n之间的水仙花数	<pre> #include <stdio.h> </pre>

```

int main()
{
    int m,n;
    while(scanf("%d%d",&m,&n)!=EOF)
    {
        int a,b,c,i;
        a=0;
        b=0;
        c=0;
        int j=m,p=0;
        for(i=m;i<n;i++)
        {
            a=i/100;
            b=i/10%10;
            c=i%10;
            if(i==a*a*a+b*b*b+c*c*c)
            {
                printf(" %d"+!p,i);
                p++;
            }
            else
            {
                j++;
                if(j==n)
                    printf("no");
            }
        }
        printf("\n");
    }
    return 0;
}

```

冒泡排序

```

#include <stdio.h>

void BubbleSort(int number[],int len)
{
    int temp;
    int i,j;
    for (i=0;i<len-1;i++)//外循环是排序趟数，len个数字排序len-1次
    {
        for(j=0;j<len-1-i;j++)//内循环为每趟比较的次数，第i趟比较len-i次
        {
            if(number[j]>number[j+1])//两个数字比较，如果左数>右数，就交换
            {
                temp = number[j];
                number[j]=number[j+1];
                number[j+1]=temp;
            }
        }
    }
}

int main()
{
    int n,i,index;
    printf("请输入要比较的数字的个数:");
    scanf("%d",&n);
    int number[n];
    printf("请输入%d个数字\n",n);
    for (i=0; i<n; i++)
    {
        scanf("%d",&index);
    }
}

```

	<pre> number[i]=index; } printf("输入结束， 您的%d个数字是:\n",n); for (i=0; i<n; i++) printf ("%d ", number[i]); putchar ('\n'); printf("下面对这%d个数字进行升序排序\n",n); BubbleSort(number,n); printf("排序的结果是:\n"); for (i=0; i<n; i++) printf ("%d ", number[i]); putchar ('\n'); return 0; } </pre>
1025 找完数	<pre> #include <stdio.h> int judge(int n) { int j,sum; sum = 0; for(j=1;j<n;j++) { if (n%j==0) sum+=j; } if(n==sum) return 1; else return 0; } int main() { int n,i,j; scanf("%d",&n); for(i=1;i<n;i++) { if(judge(i)==1) { printf("%d its factors are ",i); for(j=1;j<i;j++) { if(i%j==0) printf("%d ",j); } printf("\n"); } } return 0; } </pre>
<u>1026</u> <u>分数求和</u> <u>(多复习)</u>	<pre> #include <stdio.h> int main() { int i,n; double sum,a,b,c; sum=1; a=1; </pre>

	<pre> b=2; c=0; scanf("%d",&n); for(i=1;i<n;i++) { sum+=a/b; c=b; b=a+b; a=c; } printf("%.2lf\n",sum+n); return 0; } </pre>
1027 小球弹起	<pre> #include <stdio.h> #include <math.h> double bounce(int M,int N) { double height; height = M/pow(2,N); return height; } double distance(int M,int N) { double sum,distance; int i; sum=0; for(i=1;i<N;i++) { sum+=bounce(M,i); } distance=M+2*sum; return distance; } int main() { int M,N; scanf("%d %d",&M,&N); printf("%.2lf %.2lf\n",bounce(M,N),distance(M,N)); return 0; } </pre>
1028 猴子吃桃	<pre> #include <stdio.h> int main() { int i,n,m; m=1; scanf("%d",&n); for(i=1;i<n;i++) { m=(m+1)*2; } printf("%d\n",m); return 0; } </pre>
1029 迭代法求平方根	<pre> #include <stdio.h> #include <math.h> </pre>

	<pre> int main() { double m,n,p,q; scanf("%lf",&n); m=n/2.00; q=m; p=0; while(fabs(q-p)>0.00001) { p=(m+n/m)*(1.00/2.00); q=m; m=p; } printf("%.3lf\n",p); return 0; } </pre>
1035 最大公约数 最小公倍数	<pre> #include <stdio.h> int zuidagongyueshu(int x,int y) { int z; while(x%y) { z=x%y; x=y; y=z; } return y; } int zuixiaogongbeishu(int x,int y) { return x*y/zuidagongyueshu(x,y); } int main() { int x,y; scanf("%d %d",&x,&y); printf("%d %d\n",zuidagongyueshu(x,y),zuixiaogongbeishu(x,y)); return 0; } </pre>
1037 判断素数	<pre> #include <stdio.h> int ifprime(int x) { int i; for(i=2;i<x;i++) { if(x%i==0) return 1; } return 0; } int main() { int x; </pre>

	<pre> scanf("%d",&x); if(ifprime(x)==0) printf("prime\n"); else printf("not prime\n"); return 0; } </pre>
1050 电报字母加密	<pre> #include <stdio.h> #include <string.h> #define LEN 100 int main() { int i,j; char a[LEN]; gets(a); j=strlen(a); for(i=0;i<j;i++) { if((a[i]>='a'&&a[i]<='y') (a[i]>='A'&&a[i]<='Y')) printf("%c",a[i]+1); else if(a[i]=='z') putchar('a'); else if(a[i]=='Z') putchar('A'); else printf("%c",a[i]); } return 0; } </pre>
1051 三个数字从小到大排序	<pre> #include <stdio.h> int main() { int a,b,c,temp; scanf("%d%d%d",&a,&b,&c); if(a>b) { temp=a; a=b; b=temp; } if(a>c) { temp=a; a=c; c=temp; } if(b>c) { temp=b; b=c; c=temp; } printf("%d %d %d ",a,b,c); printf("\n"); return 0; } </pre>
1070	<pre> #include <stdio.h> </pre>

分段函数

```
int main()
{
    int x;
    double y;
    while(scanf("%d",&x)!=EOF)
    {
        if(x>=0&&x<=100)
        {
            y=2*x-100;
            printf("%.0lf\n",y);
        }
        else if(x>100&&x<=500)
        {
            y=200;
            printf("%.0lf\n",y);
        }
        else if(x>500&&x<=1000)
        {
            y=1.5*x-650;
            if(y-(int)y!=0.00)
                printf("%.1lf\n",y);
            else
                printf("%.0lf\n",y);
        }
    }

    return 0;
}
```

1006
输入一批学生的成绩，遇到0结束

```
数组版（繁）
#include<stdio.h>
#define LEN 100

int main()
{
    int number[LEN];
    int index,i,n;
    for (i=0; i<LEN&&index>0; i++)
    {
        scanf("%d",&index);
        number[i]=index;
    }
    n=i;
    int x,y,z;
    x=0;
    y=0;
    z=0;
    for (i=0; i<n-1; i++)
    {
        if(number[i]>=85)
            x++;
        else if(number[i]>=60)
            y++;
        else
            z++;
    }
    printf(">=85:%d\n",x);
    printf("60-84:%d\n",y);
    printf(">=85:%d\n",x);
}
```

	<pre> return 0; } While循环版（荐）： #include<stdio.h> int main(void) { double scores; int x,y,z; x=0; y=0; z=0; scanf("%lf",&scores); while(scores>0) { if(scores>85) { x=x+1; } else if((scores>=60)&&(scores<=84)) { y=y+1; } else { z=z+1; } scanf("%lf",&scores); } printf(">=85:%d ",x); printf("60-84:%d ",y); printf("<60:%d ",z); return 0; } </pre>
输入一串数字，遇到回车结束数组的输入	<pre> char ch=0; i=0; while(ch!='\n') { scanf("%ld",&index); number[i]=index; ch=getchar(); i++; } </pre>
括号配对数组版	<pre> #include <stdio.h> int main() { int n,i; while(scanf("%d",&n)!=EOF) { char a[n]; gets(a); int number=0; for(i=0;a[i];i++) { if(a[i]=='(') number++; if(a[i]==')') number--; } } } </pre>

	<pre> if(number<0) break; } if(number==0) printf("Yes\n"); else printf("No\n"); } return 0; } </pre>
<p>a+b 已知行数</p>	<pre> #include <stdio.h> int main() { int i,n,a,b; scanf("%d",&n); int sum[n]; for(i=0;i<n;i++) { scanf("%d %d",&a,&b); sum[i]=a+b; } for(i=0;i<n;i++) { printf("%d\n",sum[i]); } return 0; } </pre>
<p>a+b 结束标记版(00)</p>	<pre> #include <stdio.h> int main() { int a,b,c[10]; int i=0; while(1) { scanf("%d %d",&a,&b); if(a!=0&&b!=0) { c[i]=a+b; i++; } else break; } int n=i; for(i=0;i<n;i++) printf("%d\n",c[i]); return 0; } </pre>
<p>a+b 行列标记版</p>	<pre> #include <stdio.h> int main() { int a,sum,n,i; sum=0; while(scanf("%d",&n)!=EOF) { if(n==0) continue; int number[n]; </pre>

	<pre> for(i=0;i<n;i++) { scanf("%d",&a); number[i]=a; sum+=number[i]; } printf("%d\n",sum); sum = 0; } return 0; } </pre>
<p>a+b 行列已知行数版</p>	<pre> #include <stdio.h> int qiuhe(int n) { int i,number[n],sum,a; sum=0; for(i=0;i<n;i++) { scanf("%d",&a); number[i]=a; sum+=number[i]; } return sum; } int main() { int i,n,a; scanf("%d",&n); int sum[n]; for(i=0;i<n;i++) { scanf("%d",&a); sum[i]=qiuhe(a); } for(i=0;i<n;i++) { printf("%d\n",sum[i]); } return 0; } </pre>
<p>a+b 行列版</p>	<pre> #include <stdio.h> int main() { int i,n,number,a; int sum=0; while(scanf("%d",&n)!=EOF) { int number[n]; for(i=0;i<n;i++) { scanf("%d",&a); number[i]=a; sum+=number[i]; } printf("%d\n",sum); sum=0; } return 0; } </pre>

a+b 空行输出版	<pre> #include <stdio.h> int main() { int a,b,i; i=0; while(scanf("%d%d",&a,&b)!=EOF) { if(i==0) printf("%d\n",a+b); else { printf("\n"); printf("%d\n",a+b); } i++; } return 0; } </pre>
a+b 行数已知空行输出版	<pre> #include <stdio.h> int qiuhe(int n) { int i,number[n],sum,a; sum=0; for(i=0;i<n;i++) { scanf("%d",&a); number[i]=a; sum+=number[i]; } return sum; } int main() { int i,n,a; scanf("%d",&n); int sum[n]; for(i=0;i<n;i++) { scanf("%d",&a); sum[i]=qiuhe(a); } for(i=0;i<n;i++) { if(i==0) printf("%d\n",sum[i]); else { printf("\n"); printf("%d\n",sum[i]); } } return 0; } </pre>
递归实现斐波那契数列	<pre> int Fibon1(int n) { return (n==1 n==2)?1:Fibon1(n - 1) + Fibon1(n - 2); } </pre>
逆序数	<pre> #include <stdio.h> </pre>

	<pre> int power(int x,int n) { if(n==0) return 1; if(n==1) return x; if(n%2==0) return power(x*x,n/2); else return power(x*x,n/2)*x; } int main() { int x,k=0,temp; scanf("%d",&x); temp=x; while(temp) { temp/=10; k++; } int i,j,sum=0,number; for(i=0;i<k;i++) { number=x%10; sum+=number*power(10,k-i-1); x/=10; } printf("%d\n",sum); return 0; } </pre>
1-n有多少个平方数	<p>最快捷方法： 直接取(int)sqrt(n)</p>
二维数组打印一个10行的杨辉三角	<pre> #include <stdio.h> #define N 10 void Print(int a[N][N]) { for(int i=0;i<N;i++) { for(int j=0;j<=i;j++) printf("%5d",a[i][j]); printf("\n"); } } int main() { int a[N][N] = {0}; for(int i=0;i<N;i++) a[i][0]=1; for(int i=1;i<N;i++) for(int j = 1;j<=i;j++) a[i][j]=a[i-1][j-1]+a[i-1][j]; Print(a); return 0; } </pre>
递归实现一个数的n次方	<pre> int power(int x,int n) { if(n==0) </pre>

	<pre> return 1; if(n==1) return x; if(n%2==0) return power(x*x,n/2); else return power(x*x,n/2)*x; } </pre>
100位以内大整数加法	<pre> #include <stdio.h> #include <string.h> #define N 101 int main() { char a[N],b[N],c[N]; while(scanf("%s",a)!=EOF&&scanf("%s",b)!=EOF) { int n1 = strlen(a); int n2 = strlen(b); int num1[N]={0}; int num2[N]={0}; int i; for(i=0;i<n1;i++) { num1[i]=a[n1-i-1]-'0'; } for(i=0;i<n2;i++) { num2[i]=b[n2-i-1]-'0'; } int n=n1>=n2?n1:n2; for(i=0;i<n;i++) { num1[i]+=num2[i]; if(num1[i]>=10) { num1[i]-=10; num1[i+1]++; } } if(num1[n]!=0) { c[0]=num1[n]+'0'; for(i=1;i<=n;i++) { c[i]=num1[n-i]+'0'; } c[n+1]='\0'; } else { for(i=0;i<n;i++) { c[i]=num1[n-i-1]+'0'; } c[n]='\0'; } printf("%s\n",c); } return 0; } </pre>
100位以内大整数减法	<pre> #include <stdio.h> #include <string.h> </pre>

```

#include <string.h>
#define N 101

void swap(int a[], int b[], int len1, int len2)
{
    int max = len1>len2?len1:len2;
    for(int i=0; i<max; i++)
    {
        a[i]=a[i]+b[i];
        b[i]=a[i]-b[i];
        a[i]=a[i]-b[i];
    }
}

int main()
{
    char a[N],b[N],c[N];
    while(scanf("%s",a)!=EOF&&scanf("%s",b)!=EOF)
    {
        int len1=strlen(a);
        int len2=strlen(b);
        int numa[N]={0};
        int numb[N]={0};
        for(int i=0;i<len1;i++)
        {
            numa[i]=a[len1-i-1]-'0';
        }
        for(int i=0;i<len2;i++)
        {
            numb[i]=b[len2-i-1]-'0';
        }
        int flag = 0;
        if(len1<len2)
        {
            flag=1;
            swap(numa,numb,len1,len2);
            len1=len1+len2;
            len2=len1-len2;
            len1=len1-len2;
        }
        else if(len1==len2)
        {
            for(int i=len1-1; i>=0; i--)
            {
                if(numa[i]==numb[i])
                    continue;
                else if(numa[i]>numb[i])
                    break;
                else
                {
                    flag=1;
                    swap(numa,numb,len1,len2);
                    len1=len1+len2;
                    len2=len1-len2;
                    len1=len1-len2;
                    break;
                }
            }
        }
        int down=0;
        int max = len1>len2?len1:len2;
        for(int i=0; i<max; i++)
        {

```


	<pre> numa[i]=numa[i]-numb[i]-down; down=0; if(numa[i]<0) { down=1; numa[i]+=10; } } if(flag==1) printf("-"); if(numa[max]!=0) { c[0]=numa[max]+'0'; for(int i=1;i<=max;i++) { c[i]=numa[max-i]+'0'; } c[max+1]='\0'; } else { for(int i=0;i<max;i++) { c[i]=numa[max-i-1]+'0'; } c[max]='\0'; } int i; int m = 0; for(int j=0; j<max; j++) { if(c[j]=='0') continue; else { m=1; break; } } if(m==0) printf("0"); else { for(i=0;c[i]!='0';i++); for(;i<max;i++) printf("%c",c[i]); } putchar('\n'); } return 0; } </pre>
通过两点的直线	<pre> #include <stdio.h> int main() { double x1,x2,y1,y2,k,b; scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2); if (y1==y2) printf("y=%.5lf\n",y1); else if(x1==x2) printf("x=%.5lf\n",x1); else { </pre>

	<pre> k=(y1-y2)/(x1-x2); b=y1-k*x1; if(b==0.00) printf("y=%.5lfx",k); else { printf("y=%.5lfx",k); printf("%+.5lf\n",b); } } printf("\n"); return 0; } </pre>
序列号验证	<pre> #include <stdio.h> #include <string.h> int main() { char number[10]; scanf("%s",number); int sum=0; for(int i=0;i<10;i+=2) { sum+=(number[i]-'0')+((number[i+1]-'0')*10); } if(sum>=100) { int a=sum%100; int b=(a%10)*10; int c=a/10; int d=sum/100; sum=d*10+b+c; } if(sum==99) printf("Yes\n"); else printf("No\n"); return 0; } </pre>
计算位移	<pre> #include <stdio.h> #include <math.h> int main() { double a,b,c; double s; scanf("%lf %lf %lf",&a,&b,&c); s=a*c+(1.00/2.00)*b*pow(c,2); printf("%.4lf\n",s); return 0; } </pre>
超市热敏纸	<pre> #include <stdio.h> int main() { int a; double b; scanf("%d %lf",&a,&b); if(a==1001) { printf("Product Price\n"); printf("1001/Apple 16.00 Yuan/kg\n"); if(b*16.00>10.00) </pre>

	<pre> printf(" %.2lf Yuan\n",b*16.00); else printf(" %.2lf Yuan\n",b*16.00); printf(" Weight\n"); if(b>=10.00) printf(" %.3lf kg\n",b); else printf(" %.3lf kg\n",b); } else if(a==1002) { printf("Product Price\n"); printf("1002/Grape 10.00 Yuan/kg\n"); if(b*10.00>10.00) printf(" %.2lf Yuan\n",b*10.00); else printf(" %.2lf Yuan\n",b*10.00); printf(" Weight\n"); if(b>=10.00) printf(" %.3lf kg\n",b); else printf(" %.3lf kg\n",b); } else if(a==1003) { printf("Product Price\n"); printf("1003/Pear 9.98 Yuan/kg\n"); if(b*9.98>10.00) printf(" %.2lf Yuan\n",b*9.98); else printf(" %.2lf Yuan\n",b*9.98); printf(" Weight\n"); if(b>=10.00) printf(" %.3lf kg\n",b); else printf(" %.3lf kg\n",b); } return 0; } </pre>
金币兑换	<pre> #include <stdio.h> int main() { int n,d,a; while(scanf("%d",&n)!=EOF) { if(n<5) printf("0\n"); else { a=n-4; if(a%4==0) { if(a%8==0) d=a/4; else d=(a/4)+1; } else d=(a/4)+1; printf("%d\n",d); } } } </pre>

	<pre> } return 0; } </pre>
网页配色代码表示法	<pre> #include <stdio.h> int main() { int r,g,b; scanf("%d%d%d",&r,&g,&b); printf("#%02X%02X%02X\n",r,g,b); return 0; } </pre>
硬盘容量表示	<pre> #include <stdio.h> #include <math.h> int main() { int b; while(scanf("%d",&b)!=EOF) { getchar(); char d=getchar(); double c; if(d=='T') { c=b*pow(1000,4)*pow(1024,-4); printf("%.1lf T\n",c); } else if(d=='G') { c=b*pow(1000,3)*pow(1024,-3); printf("%.1lf G\n",c); } } return 0; } </pre>
scanf忽略空格	scanf(" %c",&a) (在%c前加空格)
简单加密	<pre> #include <stdio.h> int main() { char a; while(scanf("%c",&a)!=EOF) { if(a>=97&&a<=122) printf("%c\n",a-32); else if(a>=65&&a<=106) printf("%c\n",a+32); else if(a>=48&&a<=57) { if(a==57) printf("0\n"); else printf("%c\n",a+1); } else { printf("%d\n",a); } } return 0; } </pre>

二维数组的输入，并且找最大最小值	<pre> #include <stdio.h> int main() { int a[3][4]; int i,j; for(i=0;i<3;i++) { for(j=0;j<4;j++) scanf("%d",&a[i][j]); } int max,min; max=a[0][0]; min=a[0][0]; int maxi=0,maxj=0; int mini=0,minj=0; for(int i=0;i<3;i++) { for(int j=0;j<4;j++) { if (max<a[i][j]) { max=a[i][j]; maxi=i; maxj=j; } if (min>a[i][j]) { min=a[i][j]; mini=i; minj=j; } } } printf("max is %d. Location:<%d,%d>\n",max,maxi+1,maxj+1); printf("min is %d. Location:<%d,%d>\n",min,mini+1,minj+1); return 0; } </pre>
不用strcat，把后一个字符串拼到第一个字符串后面	<pre> #include <stdio.h> int main() { char a[100]; char b[100]; gets(a); gets(b); int enda=0; int j=0; for(int i = 0;a[i];i++) enda++; while(b[j]) { a[enda+j]=b[j]; j++; } a[enda+j]='\0'; printf("%s\n",a); return 0; } </pre>
等差数列与素数	<pre> #include <stdio.h> int ifprime(int number) { </pre>

```

int i;
for(i=2;i<number;i++)
{
    if(number%i==0)
        return 0;
}
return (number==1)?0:1;
}
int main()
{
    int a,d,n;
    while(scanf("%d%d%d",&a,&d,&n)!=EOF)
    {
        int count=0;
        while(1)
        {
            if(ifprime(a))
                count++;
            if(count==n)
                break;
            a+=d;
        }
        printf("%d\n",a);
    }
    return 0;
}

```

把一个数组分成奇数和偶数分别排序

```

#include <stdio.h>

void BubbleSort(int number[],int len)
{
    int temp;
    int i,j;
    for (i=0;i<len-1;i++)
    {
        for(j=0;j<len-1-i;j++)
        {
            if(number[j]>number[j+1])
            {
                temp = number[j];
                number[j]=number[j+1];
                number[j+1]=temp;
            }
        }
    }
}

int main()
{
    int n,i,index;
    while(scanf("%d",&n)!=EOF)
    {
        int number[n];
        for (i=0; i<n; i++)
        {
            scanf("%d",&index);
            number[i]=index;
        }
        BubbleSort(number,n);
        int k[n],j=0;
        for (i = 0; i < n; i++)
            if (number[i]%2!=0)
            {
                k[j] = number[i];

```

	<pre> j++; } for (i = 0; i < n; i++) if (number[i]%2==0) { k[j] = number[i]; j++; } for (i=0; i<n-1; i++) printf ("%d ",k[i]); printf("%d\n",k[n-1]); } return 0; } </pre>
数行数	<pre> #include <stdio.h> int ifempty(char c[]) { int count=0; for(int i=0;c[i];i++) { if(c[i]!='\0'&& c[i]!=' ') count++; } return count; } int main() { int ch; while(scanf("%d",&ch)!=EOF) { char c[100]; int count =0; if(getchar()=='#') { while(1) { gets(c); if(c[0]!='\0') ; else if(c[0]=='#') break; else if(ifempty(c)) count++; } } printf("%d\n",count); } return 0; } </pre>
三个字符串从小到大排序	<pre> #include <stdio.h> #include <string.h> int main() { char a[3][100]; char temp[100]; for(int i=0;i<3;i++) scanf("%s",a[i]); for(int i=0;i<2;i++) { for(int j=i+1;j<3;j++) { </pre>

	<pre> if(strcmp(a[i],a[j])>=0) { strcpy(temp,a[i]); strcpy(a[i],a[j]); strcpy(a[j],temp); } } } for(int i=0;i<3;i++) printf("%s\n",a[i]); return 0; } </pre>
输出字符串中的元音字母	<pre> #include <stdio.h> #include <string.h> int main() { char a[100]; char b[100]; char c[6]={'a','e','i','o','u'}; while(gets(a)!=NULL) { int i=0; int k=0; int m=strlen(a); while(i<m) { for(int j=0;j<6;j++) { if(a[i]==c[j]) { b[k]=a[i]; k++; break; } } i++; } for(int i=0;i<k;i++) printf("%c",b[i]); printf("\n"); } return 0; } </pre>
围圈报数	<pre> #include <stdio.h> int main() { int a[100]={0}; int n; scanf("%d",&n); for(int i = 0, count =0, sum = 0; sum < n-1; i++) { if(a[i%n]==3) continue; a[i%n]=count%3+1; count++; if(a[i%n]==3) sum++; } int pos=0; for(int i=0;i<n;i++) </pre>

	<pre> if(a[i]!=3) { pos=i; break; } printf("%d\n",pos+1); return 0; } </pre>
格式化输出与打印	<pre> #include <stdio.h> #include <string.h> int main() { char a[10][100]; int i=0; char name[100]; char tel[100]; char title[100]; while(gets(a[i])!=NULL) { if(strstr(a[i],"Name: ")) { strcpy(name,a[i]+6); i++; } if(strstr(a[i],"TEL: ")) { strcpy(tel,a[i]+5); i++; } if(strstr(a[i],"Title: ")) { strcpy(title,a[i]+7); i++; } } printf("%s %s's telephone number is %s.\n",title,name,tel); return 0; } </pre>
分解质因数/自然数因式分解	<pre> #include <stdio.h> int ifprime(int x) { int i; for(i=2;i<x;i++) { if(x%i==0) return 1; } return 0; } int main() { int x; while(scanf("%d",&x)!=EOF) { if(!ifprime(x)) printf("%d=%d\n",x,x); else { printf("%d=",x); </pre>

	<pre> for(int i = 2 ; i <= x ;i++) { while(x%i==0) { printf("%d",i); x/=i; if(x!=1) printf("*"); } } printf("\n"); } } return 0; } </pre>
任意多字符串排序	<pre> #include <stdio.h> #include <string.h> int main() { char a[100][100]; char temp[100]; int j=0; while(gets(a[j])!=NULL) j++; for(int i=0;i<j-1;i++) { for(int m=i+1;m<j;m++) { if(strcmp(a[i],a[m])>=0) { strcpy(temp,a[i]); strcpy(a[i],a[m]); strcpy(a[m],temp); } } } for(int i=0;i<j;i++) printf("%s\n",a[i]); return 0; } </pre>
ISBN码验证	<pre> #include <stdio.h> int main() { char a[100][13]; int i=0; int number[9]; while(gets(a[i])!=NULL) { int index=0; number[0]=a[i][0]-'0'; number[1]=a[i][2]-'0'; number[2]=a[i][3]-'0'; number[3]=a[i][4]-'0'; number[4]=a[i][6]-'0'; number[5]=a[i][7]-'0'; number[6]=a[i][8]-'0'; number[7]=a[i][9]-'0'; number[8]=a[i][10]-'0'; if(a[i][12]!='X') </pre>

	<pre> index=a[i][12]-'0'; int sum=0; for(int j=0;j<9;j++) sum+=number[j]*(j+1); if((sum%11)==10&& a[i][12]=='X') printf("Right\n"); else if(index==(sum%11)) printf("Right\n"); else if((sum%11)==10&& a[i][12]!='X') { a[i][12]='X'; printf("%s\n",a[i]); } else { a[i][12]=sum%11+'0'; printf("%s\n",a[i]); } i++; } return 0; } </pre>
<p>约瑟夫环最后留存的人的标号</p> <p>n是人数， m是报号上限</p>	<pre> int cir(int n,int m) { int p=0; for(int i=2;i<=n;i++) { p=(p+m)%i; } return p+1; } </pre>
<p>链表实现谁先拿饮料</p>	<pre> #include <stdio.h> #include <stdlib.h> typedef struct Node { int data; struct Node* next; }N, *P; P createList(int n) { P phead = (P)malloc(sizeof(N)); phead->data=1; phead->next=NULL; P p = phead; for(int i=2; i<=n; i++) { P newNode = (P)malloc(sizeof(N)); newNode->data=i; newNode->next=NULL; p->next=newNode; p=p->next; } p->next=phead; return phead; } void chulie(P phead, int m) { P p = phead; while(p->next!=phead) </pre>

	<pre> p=p->next; P begin = phead; while(begin->next!=begin) { for(int i=1; i<m; i++) { p = begin; begin=begin->next; } p->next=begin->next; printf("%d ",begin->data); free(begin); begin=p->next; } printf("%d\n",begin->data); free(begin); } int main() { int n,m; while(~scanf("%d%d",&n,&m)) { P phead=createList(n); chulie(phead,m); } return 0; } </pre>
四个数字中第二大的数字	<pre> #include <stdio.h> int main() { int number[4]; while(scanf("%d",&number[0])!=EOF) { for(int i=1; i<4; i++) scanf("%d",&number[i]); int max=number[0]; int secondmax=0; for(int i=1; i<4; i++) { if(max<number[i]) { secondmax=max; max=number[i]; } else { if(secondmax<number[i]&&number[i]!=max) secondmax=number[i]; } } if(number[0]==number[1]&&number[1]==number[2]&&number[2]==number[3]) secondmax=number[0]; printf("%d\n",secondmax); } return 0; } </pre>
移动若干个数字	<pre> #include <stdio.h> int main() { </pre>

从(1,1)到(m,n)有几种走法

求最大子序列和以及长度

	<pre> length=j-i; } } } printf("%d %d\n",maxSum,length+1); } return 0; } </pre>
把10个数字中最小的数与第一个数对换，把最大的数与最后一个数对换	<pre> #include <stdio.h> void get(int a[10]) { for(int i=0; i<10; i++) scanf("%d",&a[i]); } void change(int a[10]) { int max=a[0]; int min=a[0]; int temp; int t=0; for(int i=0; i<10; i++) { if(min>=a[i]) { min=a[i]; t=i; } } temp=a[t]; a[t]=a[0]; a[0]=temp; for(int i=0; i<10; i++) { if(max<=a[i]) { max=a[i]; t=i; } } temp=a[t]; a[t]=a[9]; a[9]=temp; } void print(int a[10]) { for(int i=0; i<10; i++) printf(" %d"+!i, a[i]); putchar("\n"); } int main() { int a[10]={0}; get(a); change(a); print(a); return 0; } </pre>
汉诺塔	<pre> #include <stdio.h> </pre>

	<pre> void move(char a, char b) { printf("Move a plate from column %c to %c.\n",a ,b); } void hanoi(int i, char a, char b, char c) { if(i == 1) move(a, c); else if(i == 2) { move(a, b); move(a, c); move(b, c); } else if(i > 2) { hanoi(i-1, a, c, b);//把a上的n-个盘子用c做辅助，移动到b上 move(a, c);//再把a上的一个盘子移动到c上 hanoi(i-1, b, a, c);//再把b上的i-1个盘子用a做辅助，移动到c上 } } int main() { int n; while(~scanf("%d",&n)) hanoi(n, 'A', 'B', 'C'); return 0; } </pre>
求一个串的next数组	<pre> #include <stdio.h> void getnext(char *T, int *next) { int i, j; i = 1; j = 0; next[1]=0; while(i<T[0]) { if(j==0 T[i]==T[j]) { ++i; ++j; next[i]=j; } else j=next[j]; } } int main() { char T[10]={9,'a','b','a','b','a','a','a','b','a'}; int next[9]; getnext(T,next); for(int i=1; i<=9; i++) printf("%d", next[i]); putchar('\n'); return 0; } </pre>

<p>任意多的数字，求第二大 如果全相同输出-1</p>	<pre>#include <stdio.h> int main() { int n; while(scanf("%d",&n)!=EOF) { int number[n]; for(int i=0; i<n; i++) scanf("%d",&number[i]); int max = number[0]; int secondmax = -1; for(int i=0; i<n; i++) { if(max<number[i]) { secondmax=max; max=number[i]; } else { if(secondmax<number[i]&&number[i]!=max) secondmax=number[i]; } } printf("%d\n",secondmax); } return 0; }</pre>
<p>正常成绩有多少</p>	<pre>#include <stdio.h> #include <math.h> int main() { int n; while(scanf("%d",&n)!=EOF) { int number[n][3]; int sum[n]; double ave[n]; double fangcha[n]; int m[n]; for(int i=0; i<n; i++) sum[i]=0; for(int i=0; i<n; i++) ave[i]=0; for(int i=0; i<n; i++) fangcha[i]=0; for(int i=0; i<n; i++) m[i]=0; for(int i=0; i<n; i++) for(int j=0; j<3; j++) scanf("%d",&number[i][j]); for(int i=0; i<n; i++) for(int j=0; j<3; j++) sum[i]+=number[i][j]; for(int i = 0; i<n; i++) ave[i]=sum[i]/3.00; for(int i = 0; i<n; i++) for(int j=0; j<3; j++) fangcha[i]+=(pow(number[i][j]-ave[i],2))/3.00; for(int i=0; i<n; i++)</pre>

	<pre> { if(ave[i]>=80.00&&fangcha[i]<=25.00) m[i]=1; } int p=0; int max=0; for(int i=0; i<n; i++) { if(m[i]==1) { p++; if(max<p) max=p; } else { if(max<p) max=p; p=0; } } printf("%d\n",max); } return 0; } </pre>
立方根的实现	<pre> double pow13(int a) { return (double)exp(log(a)/3); } </pre>
m到n中间有多少个立方根	<pre> #include <stdio.h> int main() { long long m, n; int k; while(scanf("%lld%lld",&m, &n)!=EOF) { k=0; for(long long i = -1626; i < 1626; i++) { if(i*i*i>=m&&i*i*i<=n) k++; } printf("%d\n",k); } return 0; } </pre>
逆波兰式的实现	<pre> #include <stdio.h> #include <stdlib.h> double exp() { char a[10]; scanf("%s",a); char m = a[0]; switch(m) { case '+': return exp()+exp(); case '-': return exp()-exp(); case '*': </pre>

	<pre> return exp()*exp(); case '/': return exp()/exp(); default: return atof(a); } } int main() { double rep; rep = exp(); printf("%lf\n", rep); return 0; } </pre>
选择排序	<pre> void sort(int a[], int n) { int i, j, k, temp; for(i=0; i<n-1; i++) { k=i; for(j=i+1; j<n; j++) { if(a[j]<a[k]) k=j; } temp=a[k]; a[k]=a[i]; a[i]=temp; } } </pre>
qsort快速排序函数	<pre> #include <stdio.h> #include <stdlib.h> int mycomp(const void * p1, const void *p2) { const int * a1 = (const int *)p1; const int * a2 = (const int *)p2; if(*a1<*a2) return -1; else if(*a1 == *a2) return 0; else return 1; } int main() { int number[10]; for(int i=0; i<10; i++) scanf("%d", &number[i]); printf("当前数组如下: \n"); for(int i=0; i<10; i++) printf(" %d"+!i, number[i]); printf("\n"); qsort(number, 10, sizeof(int), mycomp); printf("排序后数组如下: \n"); for(int i=0; i<10; i++) printf(" %d"+!i, number[i]); printf("\n"); return 0; } </pre>

<p>子字符串统计</p>	<pre>#include<stdio.h> int find(char* str,char* sub) { int n; char* p,* r; n = 0; while (*str) { p = str; r = sub; while (*r) { if (*p == *r) { p++; r++; } else break; } if (*r == '\0') n++; str++; } return n; } int main() { char a[1000],b[1000]; while(~scanf("%s %s",a,b)) { int sum = find(a,b); printf("%d time(s)\n",sum); } return 0; }</pre>
<p>鸣哥出差</p>	<pre>#include <stdio.h> struct time { int minutes; int hours; }; int main() { int a,b; while(~scanf("%d %d",&a,&b)) { struct time t; int hour1,hour2; int min1,min2; min1=a%100; min2=b%100; hour1=a/100; hour2=b/100; t.hours=hour2-hour1; t.minutes=min2-min1; if(t.minutes<0) {</pre>

	<pre> t.minutes=60+t.minutes; t.hours--; } printf("The train journey time is %d hours %d minutes.\n",t.hours,t.minutes); } return 0; } </pre>
名字排序	<pre> #include <stdio.h> #include <string.h> int main() { char a[1000]; while(gets(a)!=NULL) { int len = strlen(a); int m=0; for(int i=0; i<len; i++) { if(a[i]==' ') m++; } int n=m+1; char b[n][100]; char *p; p= strtok(a, " "); int i=0; while(p) { strcpy(b[i],p); i++; p=strtok(NULL, " "); } char temp[100]; for(int i=0;i<n-1;i++) { for(int m=i+1;m<n;m++) { if(strcmp(b[i],b[m])>=0) { strcpy(temp,b[i]); strcpy(b[i],b[m]); strcpy(b[m],temp); } } } for(int i=0;i<n;i++) printf(" %s"+!i,b[i]); printf("\n"); } return 0; } </pre>
检测程序1	<pre> #include <stdio.h> #include <string.h> #include <stdlib.h> int main() { char c[1000]; char d[1000]; while(gets(c)!=NULL) { </pre>

	<pre> int i=0; char a[100][100]; char *token1; token1 = strtok(c, " "); while(token1) { strcpy(a[i],token1); i++; token1=strtok(NULL," "); } gets(d); int m=0; char b[100][100]; char *token2; token2=strtok(d, " "); while(token2) { strcpy(b[m],token2); m++; token2=strtok(NULL," "); } for(int p = 0; p<m;p++) { for(int q=0; q<i; q++) { if(strcmp(b[p],a[q])==0) printf("(%d,%d)",p,q); } } putchar('\n'); memset(a,'\0',sizeof(a)); memset(b,'\0',sizeof(b)); } return 0; } </pre>
检测程序2	<pre> #include <stdio.h> #include <string.h> #include <stdlib.h> int main() { int n; char a[100]; while(~scanf("%d",&n)) { getchar(); gets(a); int m = strlen(a); for(int i=0;i<m;i++) { if(a[i]>='a'&&a[i]<='z') { a[i]+=n; if(a[i]<0) a[i]+=230; else if(a[i]>'z') a[i]-=26; } } } } </pre>

	<pre> for(int i=0;i<m;i++) { if(a[i]>='A'&&a[i]<='Z') { a[i]+=n; if(a[i]>'Z') a[i]-=26; } } printf("%s\n",a); } return 0; } </pre>
检测程序3	<pre> #include <stdio.h> #include <string.h> int myCompare(char *b, char *a, int n) { char c[11]; memset(c,'\0',sizeof(c)); strcpy(c,a); for(int i=0; c[i]; i++) { if(c[i]>='a'&&c[i]<='z') { c[i]+=n; if(c[i]<0) c[i]+=230; else if(c[i]>'z') c[i]-=26; } } for(int i=0; c[i]; i++) { if(c[i]>='A'&&c[i]<='Z') { c[i]+=n; if(c[i]>'Z') c[i]-=26; } } return strcmp(b,c); } int main() { char c[1000]; char d[1000]; while(gets(c)!=NULL) { int i=0; char a[100][11]; int blank_count_a =0; int blank_count_b =0; for(int u = 0; u<(int)strlen(c);u++) { if(c[u]==' ') blank_count_a++; } if(blank_count_a>0) </pre>

```

{
    char *token1;
    token1 = strtok(c, " ");
    while(token1)
    {
        strcpy(a[i],token1);
        i++;
        token1=strtok(NULL," ");
    }
}
else
{
    strcpy(a[0],c);
    i=1;
}

gets(d);
int m=0;
char b[100][11];

for(int u = 0; u<(int)strlen(d);u++)
{
    if(d[u]==' ')
        blank_count_b++;
}

if(blank_count_b>0)
{
    char *token2;
    token2=strtok(d, " ");
    while(token2)
    {
        strcpy(b[m],token2);
        m++;
        token2=strtok(NULL," ");
    }
}
else
{
    strcpy(b[0],d);
    m=1;
}

```

//以上是a, b数组的读入与存取

```

int sum = 0;
int loc;

for(int flag = 0; flag <26; flag++)
{
    loc = 0;
    for(int p =0; p<m; p++)
    {
        for(int q=0; q<i; q++)
        {
            if(myCompare(b[p],a[q],flag)==0)
            {
                if(loc == 0)
                    printf("+%d ",flag);
                printf("(%d,%d)",p,q);
                loc++;
                sum++;
            }
        }
    }
}

```

	<pre> } } if(loc!=0) printf("\n"); } if(sum==0) printf("No\n"); memset(a,'\0',sizeof(a)); memset(b,'\0',sizeof(b)); } return 0; } </pre>
浮点数排序	<pre> #include <stdio.h> #include <string.h> #include <stdlib.h> int get_neg_dig(char *a)//计算负数的整数部分位数 { int dig = 0; for(int j = 1; a[j]!='.'; j++) { dig++; } return dig; } int get_pos_dig(char *a)//计算整数的整数部分位数 { int dig = 0; for(int j = 0; a[j]!='.'; j++) { dig++; } return dig; } int main() { char c[30000]; char temp[52]; while(gets(c)!=NULL) { //a数组的初始化 int a_len = 0; char a[1500][52]; memset(a,'\0',sizeof(a)); //a数组的存储 char *token; token = strtok(c, " "); while(token) { strcpy(a[a_len],token); a_len++; token=strtok(NULL, " "); } } } </pre>

//把a分为正负分别存储

```
char neg_a[1500][52];
char pos_a[1500][52];
memset(neg_a,'\0',sizeof(neg_a));
memset(pos_a,'\0',sizeof(pos_a));
int neg_a_len = 0;
int pos_a_len = 0;

for(int p=0; p<a_len; p++)
{
    if(a[p][0]=='-')
    {
        strcpy(neg_a[neg_a_len],a[p]);
        neg_a_len++;
    }
    else
    {
        strcpy(pos_a[pos_a_len],a[p]);
        pos_a_len++;
    }
}
```

//把负数和正数先按照位数分别排序

```
for(int i=0;i<neg_a_len-1; i++)
{
    for(int m=i+1; m<neg_a_len; m++)
    {
        if(get_neg_dig(neg_a[i])<=get_neg_dig(neg_a[m]))
        {
            strcpy(temp,neg_a[i]);
            strcpy(neg_a[i],neg_a[m]);
            strcpy(neg_a[m],temp);
        }
    }
}

for(int i=0; i<pos_a_len-1; i++)
{
    for(int m=i+1; m<pos_a_len; m++)
    {
        if(get_pos_dig(pos_a[i])>=get_pos_dig(pos_a[m]))
        {
            strcpy(temp,pos_a[i]);
            strcpy(pos_a[i],pos_a[m]);
            strcpy(pos_a[m],temp);
        }
    }
}
```

//整数位数相同的负数进行排序

```
for(int i=0;i<neg_a_len-1; i++)
{
    for(int m=i+1; m<neg_a_len; m++)
    {
        if(get_neg_dig(neg_a[i])==get_neg_dig(neg_a[m]))
        {
            if(strcmp(neg_a[i],neg_a[m])<=0)
            {

```

	<pre> strcpy(temp,neg_a[i]); strcpy(neg_a[i],neg_a[m]); strcpy(neg_a[m],temp); } } } //整数位数相同的正数进行排序 for(int i=0; i<pos_a_len-1; i++) { for(int m=i+1; m<pos_a_len; m++) { if(get_pos_dig(pos_a[i])==get_pos_dig(pos_a[m])) { if(strcmp(pos_a[i],pos_a[m])>=0) { strcpy(temp,pos_a[i]); strcpy(pos_a[i],pos_a[m]); strcpy(pos_a[m],temp); } } } } //打印测试 for(int i = 0; i<neg_a_len; i++) printf("%s ",neg_a[i]); for(int i = 0; i<pos_a_len; i++) printf(" %s"+li, pos_a[i]); putchar('\n'); //清空数组所有数组 memset(a,'\0',sizeof(a)); memset(neg_a,'\0',sizeof(neg_a)); memset(pos_a,'\0',sizeof(pos_a)); } return 0; }</pre>
逆序作文	<pre>#include <stdio.h> #include <string.h> void ReverseWord(char *p, char *q) { while(p<q) { char t = *p; *p++ = *q; *q-- = t; } } char * ReverSentence(char *s) { char* p = s ; char* q = s ; while(*q != '\0') {</pre>

```

        if (*q == ' ')
        {
            ReverseWord(p, q - 1);
            q++;
            p = q;
        }
        else
            q++;
    }

    ReverseWord(p, q - 1);
    return s;
}

int main()
{
    int ch;
    while(~scanf("%d",&ch))
    {
        char s[100][50];
        int i=0;
        if(getchar()=='#')
        {
            getchar();
            while(1)
            {
                gets(s[i]);
                if(strcmp(s[i],"#")==0)
                    break;
                i++;
            }
        }
        for(int j=0; j<i; j++)
        {
            printf("%s\n",ReverSentence(s[j]));
        }
    }
    return 0;
}

```