*1、输出 Fibonacci 数列：1,1,2,3,5,8,.....的前 20 个数。

```
#include<stdio.h>
main()
{   int fib[20],i;
    fib[0]=fib[1]=1;
    for(i=2;i<20;i++)
        fib[i]=fib[i-1]+fib[i-2];
    for(i=0;i<20;i++)
    {   if(i!=0&&i%5==0)        //每输出 5 个数换一行
            printf("\n");
        printf("%5d",fib[i]);
    }
}
```

*2、求出一组整数中的最大值。

```
#include<stdio.h>
main()
{   int a[10],i,max;
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    max=a[0];
    for(i=1;i<10;i++)
        if(max<a[i])  max=a[i];
    printf("一组数是：");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    printf("最大值为：%d",max);
}
```

学习了指针后就可以使用动态数组实现，程序如下：

```
#include<stdio.h>
#include<malloc.h>
main()
{   int *a,i,n,max;
    printf("请输入一组数的个数：");
    scanf("%d",&n);
    a=(int*)malloc(sizeof(int)*n);//空间分配
    printf("请输入%d 个整数：",n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    max=a[0];
    for(i=1;i<n;i++)
        if(max<a[i])
            max=a[i];
    printf("一组数是：");
    for(i=0;i<n;i++)
```

```
            printf("%d ",a[i]);
        printf("\n");
        printf("最大值为：%d",max);
        free(a);//空间收回
    }
```

**3、请选择一种排序方法，将一组数按从小到大排序。请尽量优化程序。

1）选择法：首先在未排序序列中找到最小元素，存放到排序序列的起始位置，然后，再从剩余未排序元素中继续寻找最小元素，然后放到已排序序列的末尾。以此类推，直到所有元素均排序完毕。

```
#include<stdio.h>
#define N 10
main()
{   int a[N]={5,3,4,9,10,8,2,7,1,6};
    int i,j,max,t;
    for(i=0;i<N-1;i++)
    {   max=i;
        for(j=i+1;j<N;j++)
            if(a[max]<a[j])
                max=j;
        if(max!=i)
        {   t=a[max];
            a[max]=a[i];
            a[i]=t;
        }
    }
    for(i=0;i<N;i++)
        printf("%d  ",a[i]);
    printf("\n");
}
```

2）冒泡排序法：它重复地访问要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。访问数列的工作重复地进行直到没有再需要交换，也就是说该数列已经排序完成。这个算法的名字由来是因为越小的元素会经由交换慢慢“浮”到数列的顶端。

算法描述：

①比较相邻的元素。如果第一个比第二个大，就交换他们两个。

②对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。

③针对所有的元素重复以上的步骤，除了最后一个。

④持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

```
#include<stdio.h>
#define N 10
main()
{   int a[N]={5,3,4,9,10,8,2,7,1,6};
    int i,j,t;
    bool exchange=true;
    for(i=0;i<N&&exchange;i++)
```

```
    {   exchange=false;
        for(j=N-1;j>=i;j--)
            if(a[j+1]<a[j])
            {   t=a[j+1];
                a[j+1]=a[j];
                a[j]=t;
                exchange=true;
            }
    }
    for(i=0;i<N;i++)
        printf("%d  ",a[i]);
    printf("\n");
}
```

3）插入排序法：对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。

算法描述：

①从第一个元素开始，该元素可以认为已经被排序

②取出下一个元素，在已经排序的元素序列中从后向前扫描

③如果该元素（已排序）大于新元素，将该元素移到下一位置重复步骤③，直到找到已排序的元素小于或者等于新元素的位置。

④将新元素插入到该位置后。

⑤重复步骤②~⑤

```
#include<stdio.h>
#define N 10
main()
{   int a[N]={5,3,4,9,10,8,2,7,1,6};
    int i,j,t;
    for(i=1;i<N;i++)
    {   t=a[i];
        for(j=i-1;t<a[j]&&j>=0;j--)
            a[j+1]=a[j];
        a[j+1]=t;
    }
    for(i=0;i<N;i++)
        printf("%d  ",a[i]);
    printf("\n");
}
```

**4、查找：在一组数中查找某给定的数 x，如果找到则输出该数所在下标位置，如果找不到则输出“not found”。请选择一种查找方法。

1）顺序查找

将查找值与数组中的元素逐个进行比较，相等即为查找成功，否则查找失败。

```
#include <stdio.h>
main()
{   int a[10]={16,20,4,8,2,14,10,6,12,18};
```

```
    int x,i,find=0;
    printf("please input data: ");
    scanf("%d",&x);
    for(i=0;i<10;i++)
      if(x==a[i])
       { find=1;
          break;
       }
    if(find==1)
      printf("%d in a[%d].\n",x,i);
    else
     printf("don't find %d\n",x);
}
```

2）二分查找

在**有序**数组（已排好序的数组）中查找某一特定元素。查找过程从数组的中间元素开始，如果中间元素正好是要查找的元素，则查找过程结束；如果某一特定元素大于或者小于中间元素，则在数组大于或小于中间元素的那一半中查找，而且跟开始一样从中间元素开始比较。如果在某一步骤数组为空，则代表找不到。这种查找算法每一次比较都使查找范围缩小一半。

```
#include<stdio.h>
main()
{   int a[10]={1,3,4,6,7,8,9,11,12,13};
    int left=0,right=9,mid,x,found=0;
    printf("请输入要查找的数：");
    scanf("%d",&x);
    while (left<=right)
    {   mid = (left+right)/2;
        if(x==a[mid])
        {   printf("a[%d]=%d\n",mid,a[mid]);
            found=1;
            break;
        }
        if(x<a[mid])
            right=mid-1;
        if(x>a[mid])
            left=mid+1;
    }
    if(!found){printf("not found!\n"); }
}
```

**5、输出杨辉三角形

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

```
   .......
#include<stdio.h>
main()
{  int i,j,n=0,a[17][17]={0};
   while(n<1 || n>16)
   { printf("请输入杨辉三角形的行数:");
     scanf("%d",&n);
   }
   for(i=0;i<=n;i++)
      a[i][0]=1;
   for(i=1;i<=n;i++)
     for(j=1;j<=i;j++)
       a[i][j]=a[i-1][j-1]+a[i-1][j];
   for(i=0;i<=n;i++)
   { for(j=0;j<=i;j++)
       printf("%5d",a[i][j]);
     printf("\n");
   }
}
```

**6、如果一个班级有 40 位同学，共有 4 门考试课程，从键盘输入各位学生的考试成绩，1）计算每位学生的平均分。2）计算每门课程的平均分。

```
#include<stdio.h>
#define N 3
main()
{
    int score[N][4],i,j;
    double avg[N]={0},savg[4]={0};
   //每个人的平均分及每门课程的平均分要存入，可以方便后面的使用
    for(i=0;i<N;i++)
    {
        printf("请输入第%d 个学生的四门课成绩",i+1);
        for(j=0;j<4;j++)
            scanf("%d",&score[i][j]);
    }
    for(i=0;i<N;i++)
    {
        for(j=0;j<4;j++)
        {
            printf("%7d",score[i][j]);
            avg[i]+=score[i][j];
            savg[j]+=score[i][j];
        }
        avg[i]/=4;
        printf("%7.1lf\n",avg[i]);
    }
```

```
    for(j=0;j<4;j++)
        printf("%7.1lf",savg[j]/N);
}
```

***7、给定一个 n*n 矩阵 A。矩阵 A 的鞍点是一个位置（i,j），在该位置上的元素是第 i 行上的最大数，第 j 列上的最小数。一个矩阵 A 也可能没有鞍点。请找出 A 的鞍点。

```
#include<stdio.h>
#define N 4
main()
{    int a[N][N]={{1,7,4,1},{4,8,3,6},{1,6,1,2},{0,7,8,9}};
    int i,j,row,colum,max,min;
    int b;
    for(i=0;i<4;i++)
    {   max=a[i][0];
        colum=0;
        for(j=0;j<4;j++)
        {   if(a[i][j]>max)
            {     max=a[i][j];  //找出每一行中的最大值
                  colum=j;      //记下最大值的列号
            }
        }
        min=a[i][colum];
        b=1;
        for(j=0;j<4;j++)
          if(a[j][colum]<min)
              b=0;   //如果列中有比 min 更小的数，就不是鞍点,b 设置为 0
        if(b==1)
        printf("%d %d %d\n",i,colum,a[i][colum]);
    }
}
```