

## 互换两个数字

```
#include <stdio.h>

void swap_3(int * p, int * q) //形参名字是 p 和 q，接收实参数数据的是 p 和 q，而不是*p 和*q
{
    int t;
    t = *p; *p = *q; *q = t;
}

int main(void)
{
    int a = 3;
    int b = 5;
    swap_3(&a, &b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

```
#include "stdio.h"
#include "conio.h"
main()
{
    int i, j, k;
    printf("\n");
    for(i=1; i<5; i++) /*以下为三重循环*/
        for(j=1; j<5; j++)
            for(k=1; k<5; k++)
                {
                    if (i!=k && i!=j && j!=k) /*确保 i、j、k 三位互不相同*/
                        printf("%d,%d,%d\n", i, j, k);
                }
    getch();
}
```

## 求 1 到 n 之间奇数的平均值

```
#include <stdio.h>

int main(void)
{
    int i;
    int sum = 0;
    int cnt = 0;
    float avg;
    for (i=1; i<n+1; ++i)
    {
        if (i%2 == 1)
        {
            sum += i;
            ++cnt;
        }
    }
    avg = 1.0*sum / cnt;  //1.0 默认是 double 类型

    printf("sum = %d\n", sum);
    printf("cnt = %d\n", cnt);
    printf("avg = %f\n", avg);

    return 0;
}
```

## 用\*号输出字母 C 的图案。

```
#include "stdio.h"
#include "conio.h"
main()
{
    printf("Hello C-world!\n");
    printf(" ****\n");
    printf(" *\n");
    printf(" * \n");
    printf(" ****\n");
    getch();
}
```

## 菲波拉契序列

1   2   3   5   8   13   21   34

```
# include <stdio.h>
int main(void)
{
    int n;
    int f1, f2, f3;
    int i;

    f1 = 1;
    f2 = 2;

    printf("请输入您要求的想的序列: ");
    scanf("%d", &n);

    if (1 == n)
    {
        f3 = 1;
    }
    else if (2 == n)
    {
        f3 = 2;
    }
    else
    {
        for (i=3; i<=n; ++i)
        {
            f3 = f1 + f2;
            f1 = f2;
            f2 = f3;
        }
    }

    printf("%d\n", f3);

    return 0;
}
```

## 电梯程序 switch 用法

```
# include <stdio.h>
int main(void)
{
    int val;

    printf("请输入您要进入的楼层: ");
    scanf("%d", &val);

    switch (val)
    {
        case 1:
            printf("1 层开!\n");
            break;
        case 2:
            printf("2 层开!\n");
            //break;
        case 3:
            printf("3 层开!\n");
            break;
        default:
            printf("没有盖到这一层!\n");
            break;
    }
    return 0;
}
```

## 解一元二次方程

```
# include <stdio.h>
# include <math.h>
int main(void)
{
    double a, b, c;
    double delta;
    double x1, x2;
    char ch;

    do
    {
        printf("请输入一元二次方程的三个系数:\n");
        printf("a = ");
        scanf("%lf", &a);
        printf("b = ");
        scanf("%lf", &b);

        printf("c = ");
        scanf("%lf", &c);

        delta = b*b - 4*a*c;
        if (delta > 0)
        {
            x1 = (-b + sqrt(delta)) / (2*a);
            x2 = (-b - sqrt(delta)) / (2*a);
            printf("有两个解, x1 = %lf, x2 = %lf\n", x1, x2);
        }
        else if (0 == delta)
        {
            x1 = x2 = (-b) / (2*a);
            printf("有唯一解, x1 = x2 = %lf\n", x1, x2);
        }
        else
        {
            printf("无实数解!\n");
        }
        printf("您想继续么(Y/N): ");
        scanf(" %c", &ch); // %c 前面必须得加一个空格 原因略
    } while ('y'==ch || 'Y'==ch);
    return 0;
}
```

## 把数组元素给全部倒过来

```
#include <stdio.h>
int main(void)
{
    int a[8] = {1, 2, 3, 4, 5, 6, 7, 8};
    int i, j;
    int t;

    i = 0;
    j = 7;
    while (i < j)
    {
        t = a[i];
        a[i] = a[j];
        a[j] = t;

        i++;
        --j;
    }

    for (i=0; i<8; ++i)
        printf("%d\n", a[i]);

    return 0;
}
```

/\*完善程序，实现将输入的字符串反序输出，

如输入 windows 输出 swodniw。\*/

```
#include <string.h>
main()
{   char   c[200],c1;
    int i,j,k;
    printf("Enter a string: ");
    scanf("%s",c);
    k=strlen(c);
    for (i=0,j=k-1;i<k/2;i++,j--)
        {   c1=c[i];c[i]=c[j];c[j]=c1;   }
    printf("%s\n",c);
}
```

指针法:

```
void invert(char *s)
{int i,j,k;
  char t;
  k=strlen(s);
  for(i=0,j=k-1;i<k/2;i++,j--)
      {   t=*(s+i);   *(s+i)=*(s+j);   *(s+j)=t; }
}

main()
{FILE*fp;
charstr[200],*p,i,j;
if((fp=fopen("p9_2.out","w"))==NULL)
    {printf("cannot open the file\n");
    exit(0);
    }

    printf("input str:\n");
    gets(str);
    printf( "\n%s" ,str);
    fprintf(fp, "%s" ,str);
    invert(str);
    printf( "\n%s" ,str);
    fprintf(fp, "\n%s" ,str);
    fclose(fp);
}
```

## 从字符数组 s 中删除存放在 c 中的字符

```
#include <stdio.h>
main()
{   char   s[80],c;
    int    j,k;
    printf("\nEnter a string: ");
    gets(s);
    printf("\nEnter a character: ");
    c=getchar();
    for(j=k=0;s[j]!='\0';j++)
        if(s[j]!=c)
            s[k++]=s[j];
    s[k]='\0';
    printf("\n%s",s);
```



已知数组 **a** 中的元素已按由小到大顺序排列，以下程序的功能是将输入的一个数插入数组 **a** 中，插入后，数组 **a** 中的元素仍然由小到大顺序排列\*/

```
main()
{   int a[10]={0,12,17,20,25,28,30};           /*a[0]为工作单元，从 a[1]开始存放数据*/
    int x, i, j=6;                             /*j 为元素个数*/
    printf("Enter a number: ");
    scanf("%d",&x);
    a[0]=x;
    i=j;                                       /*从最后一个单元开始*/
    while(a[i]>x)
    {   a[i+1]=a[i]; i--;   }   /*将比 x 大的数往后移动一个位置*/
    a[++i]=x;
    j++;                                     /*插入 x 后元素总个数增加*/
    for(i=1;i<=j;i++) printf("%8d",a[i]);
    printf("\n");
}
```

## 判断一个数字是否是素数

```
# include <stdio.h>
bool IsPrime(int val)
{
    int i;

    for (i=2; i<val; ++i)
    {
        if (0 == val%i)
            break;
    }
    if (i == val)
        return true;
    else
        return false;
}

int main(void)
{
    int val;
    int i;

    scanf("%d", &val);
    if ( IsPrime(val) )
        printf("Yes!\n");
    else
        printf("No!\n");

    return 0;
}
```

## 求 1 到某个数字之间所有的素数

```
#include <stdio.h>

bool IsPrime(int m)//本函数的功能是: 判断 m 是否是素数, 是返回 true, 不是返回 false
{
    int i;
    for (i=2; i<m; ++i)
    {
        if (0 == m%i)
            break;
    }
    if (i == m)
        return true;
    else
        return false;
}

void TraverseVal(int n)//本函数的功能是把 1 到 n 之间所有的素数在显示器上输出
{
    int i;

    for (i=2; i<=n; ++i)
    {
        if ( IsPrime(i) )
            printf("%d\n", i);
    }
}

int main(void)
{
    int val;

    scanf("%d", &val);
    TraverseVal(val);

    return 0;
}
```

## 求回文数

```
#include <stdio.h>
int main(void)
{
    int val; //存放待判断的数字
    int m;
    int sum = 0;
    printf("请输入您需要判断的数字: ");
    scanf("%d", &val);
    m = val;
    while (m)
    {
        sum = sum * 10 + m%10;
        m /= 10;
    }

    if (sum == val)
        printf("Yes!\n");
    else
        printf("No!\n");

    return 0;
}
```

**/\*输出 9\*9 口诀。\*/**

```
#include "stdio.h"
#include "conio.h"
main()
{
    int i,j,result;
    printf("\n");
    for (i=1;i<10;i++)
    {
        for(j=1;j<10;j++)
        {
            result=i*j;
            printf("%d*%d=%-3d",i,j,result); /*-3d 表示左对齐， 占 3 位*/
        }
        printf("\n"); /*每一行后换行*/
    }
    getch();
}
```

# 冒泡排序

```
# include <stdio.h>
void sort(int * a, int len)
{
    int i, j, t;

    for (i=0; i<len-1; ++i)
    {
        for (j=0; j<len-1-i; ++j)
        {
            if (a[j] > a[j+1])    // >表示升序 <表示降序
            {
                t = a[j];
                a[j] = a[j+1];
                a[j+1] = t;
            }
        }
    }
}

int main(void)
{
    int a[6] = {10, 2, 8, -8, 11, 0};
    int i = 0;

    sort(a, 6);

    for (i=0; i<6; ++i)
    {
        printf("%d ", a[i]);
    }
    printf("\n");

    return 0;
}
```

## 输出所有的水仙花数：

一个三位数各位的立方和是他本身如 153

```
#include "stdio.h"
#include "conio.h"
main()
{
    int i,j,k,n;
    printf("water flower'number is:");
    for(n=100;n<1000;n++)
    {
        i=n/100;/*分解出百位*/
        j=n/10%10;/*分解出十位*/
        k=n%10;/*分解出个位*/
        if(i*100+j*10+k==i*i*i+j*j*j+k*k*k)
            printf("%-5d",n);
    }
    getch();
}
```

## 求 1 到 1000 内的完数

该数等于其因子之和如  $6=1+2+3$

```
#include "stdio.h"
#include "conio.h"
main()
{
    static int k[10];
    int i,j,n,s;
    for(j=2;j<1000;j++)
    {
        n=-1;
        s=j;
        for(i=1;i<j;i++)
        {
            if((j%i)==0)
            {
                n++;
                s=s-i;
                k[n]=i;
            }
        }
        if(s==0)
        {
            printf("%d is a wanshu",j);
            for(i=0;i<n;i++)
            printf("%d,",k[i]);
            printf("%d\n",k[n]);
        }
    }
    getch();
}
```

## 求 $1+2!+3!+\dots+20!$ 的和

```
#include "stdio.h"
#include "conio.h"
main()
{
    float n,s=0,t=1;
    for(n=1;n<=20;n++)
    {
        t*=n;
        s+=t;
    }
    printf("1+2!+3!...+20!=%e\n",s);
    getch();
}
```

**$2/1, 3/2, 5/3, 8/5, 13/8, 21/13$ ... 求出这个数列的前 20 项之和。**

```
#include "stdio.h"
#include "conio.h"
main()
{
    int n,t,number=20;
    float a=2,b=1,s=0;
    for(n=1;n<=number;n++)
    {
        s=s+a/b;
        t=a;a=a+b;b=t; /*这部分是程序的关键，请读者猜猜 t 的作用*/
    }
    printf("sum is %9.6f\n",s);
    getch();
}
```



## 求一个 3\*3 矩阵对角线元素之和

利用双重 for 循环控制输入二维数组，再将 a[i][i]累加后输出。

```
#include "stdio.h"
#include "conio.h"

/* 如果使用的是 TC 系列编译器则可能需要添加下句 */
static void dummyfloat(float *x){ float y; dummyfloat(&y);}

main()
{
    float a[3][3],sum=0;
    int i,j;
    printf("please input rectangle element:\n");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%f",&a[i][j]);
    for(i=0;i<3;i++)
        sum=sum+a[i][i];
    printf("duijiaoxian he is %6.2f",sum);
    getch();
}
```

从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个#为止。

```
#include "stdio.h"
main()
{ FILE *fp;
  char ch,filename[10];
  scanf("%s",filename);
  if((fp=fopen(filename,"w"))==NULL)
  {printf("cannot open file\n");
   exit(0);}
  ch=getchar();
  ch=getchar();
  while(ch!='#')
  {fputc(ch,fp);putchar(ch);
   ch=getchar();
  }
  fclose(fp);
}
```

从键盘输入一个字符串，将小写字母全部转换成大写字母，  
然后输出到一个磁盘文件“test”中保存。

输入的字符串以！结束。

```
#include "stdio.h"
main()
{
    FILE *fp;
    char str[100],filename[10];
    int i=0;
    if((fp=fopen("test","w"))==NULL)
    {
        printf("cannot open the file\n");
        exit(0);
    }
    printf("please input a string:\n");
    gets(str);
    while(str[i]!='!')
    {
        If(str[i]>='a'&&str[i]<='z')
            str[i]=str[i]-32;
        fputc(str[i],fp);
        i++;
    }
    fclose(fp);
    fp=fopen("test","r");
    fgets(str,strlen(str)+1,fp);
    printf("%s\n",str);
    fclose(fp);
}
```

有五个学生，每个学生有 3 门课的成绩，输入以上数据（包括学生号，姓名，三门课成绩），计算出平均成绩，将原有的数据和计算出的平均分数存放在磁盘文件"stud"中。

```
#include "stdio.h"
struct student
{
    char num[6];
    char name[8];
    int score[3];
    float avr;
} stu[5];
main()
{
    int i,j,sum;
    FILE *fp;
    for(i=0;i<5;i++)
    {
        printf("\n please input No. %d score:\n",i);
        printf("stuNo:");
        scanf("%s",stu[i].num);
        printf("name:");
        scanf("%s",stu[i].name);
        sum=0;
        for(j=0;j<3;j++)
        {
            printf("score %d.",j+1);
            scanf("%d",&stu[i].score[j]);
            sum+=stu[i].score[j];
        }
        stu[i].avr=sum/3.0;
    }
    fp=fopen("stud","w");
    for(i=0;i<5;i++)
    if(fwrite(&stu[i],sizeof(struct student),1,fp)!=1)
    printf("file write error\n");
    fclose(fp);
}
```

# 学生管理系统

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

struct Student
{
    int age;
    float score;
    char name[100];
};

int main(void)
{
    int len;
    struct Student * pArr;
    int i, j;
    struct Student t;

    //动态的构造一维数组
    printf("请输入学生的个数:\n");
    printf("len = ");
    scanf("%d", &len);
    pArr = (struct Student *)malloc(len * sizeof(struct Student));

    //输入
    for (i=0; i<len; ++i)
    {
        printf("请输入第%d 个学生的信息:\n", i+1);
        printf("age = ");
        scanf("%d", &pArr[i].age);

        printf("name = ");
        scanf("%s", pArr[i].name); //name 是数组名，本身就已经是数组首元素的地址，所以 pArr[i].name 不能改成 &pArr[i].name

        printf("score = ");
        scanf("%f", &pArr[i].score);
    }

    //按学生成绩升序排序 冒泡算法
    for (i=0; i<len-1; ++i)
    {
```

```

    for (j=0; j<len-1-i; ++j)
    {
        if (pArr[j].score > pArr[j+1].score) //>升序 <降序
        {
            t = pArr[j];
            pArr[j] = pArr[j+1];
            pArr[j+1] = t;
        }
    }
}

printf("\n\n 学生的信息是:\n");
//输出
for (i=0; i<len; ++i)
{
    printf("第%d 个学生的信息是:\n", i+1);
    printf("age = %d\n", pArr[i].age);
    printf("name = %s\n", pArr[i].name); //name 是数组名, 本身就已经是数组首元素的
地址, 所以 pArr[i].name 不能改成 &pArr[i].name
    printf("score = %f\n", pArr[i].score);

    printf("\n");
}

return 0;
}

```