字符串中的转义符

\n 换行符

\t 水平制表符 tab 注意不是空格

\r 回车 光标移动到行首

\' 显示单引号

\" 显示双引号

\a 警告音频

\\ \字符

如果不需要识别转义符 在字符串前加上@

数学运算符

```
Console.WriteLine(4 + 2);//加
Console.WriteLine(4 - 1);//减
Console.WriteLine(4 * 2);//乘
Console.WriteLine(4 / 2);//除
Console.WriteLine(5 % 2);//模  求余数
```

自加自减运算 ++ --

++在前 先加后用

++在后 先用后加

--同理

```
int a = 1;
a++;//自身+1
Console.WriteLine(a);
Console.WriteLine(a++);//先打印 后计算
Console.WriteLine(++a);//先计算 后打印
```

手动输入三个值 求得平均数

```
Console.WriteLine("请输入三个值");
double a = double.Parse(Console.ReadLine());
double b = double.Parse(Console.ReadLine());
double c = double.Parse(Console.ReadLine());
double x = (a + b + c) / 3;
Console.WriteLine(x);
```

练习：

根据天数计算周数和剩余天数

```
int days = 14;
int week = days / 7;
int leftDay = days % 7 ;
Console.WriteLine("共"+week+"周");
Console.WriteLine("剩余" + leftDay + "天");
```

把一个4位数的每一位拆分输出

例 4096

```
int data = 5487;
Console.WriteLine(data/1000);
Console.WriteLine(data%1000/100);
Console.WriteLine(data % 100 / 10);
Console.WriteLine(data % 10);
```

二进制和十六进制

常用转换：十进制转二进制 二进制转十六进制 十六进制转二进制

进制表示：

十进制	二进制	十六进制
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

十进制：逢十进位

二进制：逢二进位

十六进制：逢十六进位

int=4byte 1个int 4个字节

1byte=8bit 1个字节 8位二进制

所以1个int值的二进制 要用32位来表示

1的二进制：

00000000 00000000 00000000 00000001

十进制和二进制的转换规则

规律 找最近的2的N次幂 然后逐个转化为二进制

例 100

64=2（6）=1000000

32=2（5）=100000

4=2（2）=100

1000000

100000

100

1100100

例 176

176=128+48=128+32+16=2（7）+2（5）+2（4）

10000000 2（7）

100000 2（5）

10000 2（4）

10110000

二进制转十进制

1001001110

=2（9）+2（6）+2（3）+2（2）+2（1）

=512+64+8+4+2=590

十六进制转二进制

二进制的1111 是十六进制的F

所以满足规律

二进制和十六进制 互转 遵循8421法则

二进制转十六进制 4转1

十六进制转二进制 1转4

例：

1001011011101111 转十六进制 4转1

1001 0110 1110 1111

=0x96EF

例：

0x1FA5 转二进制 1转4

0001 1111 1010 0101

负数的二进制问题

在计算机的二进制存储过程中 把最高位(最左边)的0或1 代表正负

当最高位是0 表示正数

当最高位是1 表示负数

负数的二进制计算方式：正数二进制取反加一

-1的二进制：

00000000 00000000 00000000 00000001 正1二进制

111111111 111111111 111111111 111111110 取反

111111111 111111111 111111111 111111111 加一

int的最大值写法

01111111 11111111 11111111 11111111 2的31次方-1

1000000 0000000 00000000 0000000

int的最小值写法

1000000 0000000 00000000 0000000 -2的31次方

```
int a = 2147483647;
```

```
 a++;
Console.WriteLine(int.MaxValue);//最大值 2147483647
Console.WriteLine(int.MinValue);//最小值 -2147483648
```

位运算

位运算是计算机最基础的运算 一般都是对二进制进行操作

位与运算 & 运算规则 都为1则为1

例

10&5=0

1010

0101

0000

3&10=2

0011

1010

0010

位或运算 | 运算规则：有1则为1

例

9|5=13

1001

0101

1101

异或运算 ^ 运算规则 01为1 10为1

9^5=12

1001

0101

1100

按位取反 ~ 1为0 0为1 单目运算符

~9 结果是-10

00000000 00000000 00000000 00001001

11111111 11111111 11111111 11110110

按位左右移

左移 << 左移一位 相当于乘以2的1次方，左移N位相当于乘以2的N次方

9<<1 结果是18

0000 1001

0001 0010

高位丢弃 低位补0

右移 >> 右移一位 相当于除2 右移N位相当于除以2的N次方

9>>1 结果是4

0000 1001

0000 0100

低位丢弃 高位补0

位运算的实际运用

1 判断1个数的奇偶 要求不能使用算数运算符

运算规律 判断最低位是不是0

使用a&1 结果是1 则是奇数 结果是0 则是偶数

```
Console.WriteLine(3 & 1);
Console.WriteLine(4 & 1);
```

2 不使用第三方变量 交换两个变量的值

两个数的异或结果 异或其中一个数得另一个数

1110

0101

1011

0101

1110

```
int a = 3;
int b = 4;
a = a ^ b;//a=7 b=4
b = a ^ b;//a=7 b=3
a = a ^ b;//a=4 b=3
Console.WriteLine(a);
Console.WriteLine(b);
```

3 使用左移运算得int最大值

```
Console.WriteLine(int.MaxValue);
int a = 1 << 31;//最小值 10000000 00000000 00000000 00000000
-2147483648
a--;//011111111 11111111 11111111 11111111 2147483647
Console.WriteLine(a);
int b = ~(1 << 31);
Console.WriteLine(b);
```

练习：

求输出结果

15&7= ?

9|25= ?

15&7&9|25= ?

int a=8;

int b=3;

Console.WriteLine(a<<(a>>b))