

第三章 C语言的数据类型

3.1 变量和常量

- 常量

不占内存，在程序运行时其值不能改变的量，在程序运行时它作为操作对象直接出现在运算器的各种寄存器中
- 变量

变量是指在程序运行时可以改变的量，变量的功能就是存储数据。
- 标识符

在程序中使用的变量名，函数名，标号等统称为标识符。标识符只能由字母数字下划线组成，且第一个字符必须是字母或下划线。标识符区分大小写

3.2 数据类型和关键字

- 基本数据类型(32为操作系统数据范围)

类型名称	占字节数	其他叫法	表示的数据范围
char	1	signed char	-128 ~ 127
unsigned char	1	none	0 ~ 255
int	4	signed int	-2,147,483,648 ~ 2,147,483,647
unsigned int	4	unsigned	0 ~ 4,294,967,295
short	2	short int	-32,768 ~ 32,767
unsigned short	2	unsigned short int	0 ~ 65,535
long	4	long int	-2,147,483,648 ~ 2,147,483,647
unsigned long	4	unsigned long	0 ~ 4,294,967,295
float	4	none	3.4E +/- 38 (7 digits)
double	8	none	1.7E +/- 308 (15 digits)
long double	10	none	1.2E +/- 4932 (19 digits)

- 关键字

C语言中的32个关键字			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

3.3 数据类型和取值范围

```
/* 获取当前操作系统的字节数(bytes)和位数(bit)*/
# include<stdio.h>
int main(void){
    printf("char:bytes %d;bit %d\n",sizeof(char),sizeof(char)*8);
    printf("unsigned char:bytes %d;bit %d\n",sizeof(unsigned
char),sizeof(unsigned char)*8);
    printf("int:bytes %d;bit %d\n",sizeof(int),sizeof(int)*8);
    printf("unsigned int:bytes %d;bit %d\n",sizeof(unsigned int),sizeof(unsigned
int)*8);
    printf("short :bytes %d;bit %d\n",sizeof(short),sizeof(short)*8);
    printf("unsigned short :bytes %d;bit %d\n",sizeof(unsigned
short),sizeof(unsigned short)*8);
    printf("long :bytes %d;bit %d\n",sizeof(long),sizeof(long)*8);
    printf("long long :bytes %d;bit %d\n",sizeof(long long),sizeof(long
long)*8);
    printf("unsigned long :bytes %d;bit %d\n",sizeof(unsigned
long),sizeof(unsigned long)*8);
    printf("float :bytes %d;bit %d\n",sizeof(float),sizeof(float)*8);
    printf("double :bytes %d;bit %d\n",sizeof(double),sizeof(double)*8);
    printf("long double :bytes %d;bit %d\n",sizeof(long double),sizeof(long
double)*8);
    return 0;
}
/*
char:bytes 1;bit 8
unsigned char:bytes 1;bit 8
int:bytes 4;bit 32
unsigned int:bytes 4;bit 32
short :bytes 2;bit 16
unsigned short :bytes 2;bit 16
long :bytes 8;bit 64
long long :bytes 8;bit 64
unsigned long :bytes 8;bit 64
float :bytes 4;bit 32
double :bytes 8;bit 64
long double :bytes 16;bit 128
*/
```

```

/*获取char, int, long, long数据类型的取值范围*/
#include<stdio.h>
#include<limits.h>
int main(void)
{
    printf("char:MIN %d;MAX %d\n", CHAR_MIN, CHAR_MAX);
    printf("short:MIN %d;MAX %d\n", SHRT_MIN, SHRT_MAX);
    printf("int:MIN %d;MAX %d\n", INT_MIN, INT_MAX);
    printf("long:MIN %ld;MAX %ld\n", LONG_MIN, LONG_MAX);
    printf("long long:MIN %lld;MAX %lld\n", LLONG_MIN, LLONG_MAX);
    return 0;
}
/*
char:MIN -128;MAX 127
short:MIN -32768;MAX 32767
int:MIN -2147483648;MAX 2147483647
long:MIN -9223372036854775808;MAX 9223372036854775807
long long:MIN -9223372036854775808;MAX 9223372036854775807
*/
// 也可以按照字节 位数进行计算

```

3.4 本章小结和练习

输出8进制 16进制的方法

```

// 无前缀方式
printf("%o",num)// 无前缀o的8进制数
printf("%x",num)// 无前缀0x的小写16进制数
printf("%X",num) //无前缀0X的大写16进制数
// 有前缀的方式
printf("%#o",num) //有前缀o的8进制数

printf("%#x",num) //有前缀0x的小写16进制数

printf("%#X",num) //有前缀0X的大写16进制数

```

练习

```

/*
输入:
输入一个整数, 不超过10^9
输出:
输出这个整数的八进制、十进制和十六进制, 三个数字用空格分开, 最后一个有换行
eg:
输入:
100
输出:
0144 100 0x64
*/
# include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    printf("%#o %#d %#x\n",n,n,n);
}

```

```
    return 0;
}
```

第四章 C语言中的基本输入输出

4.1 字符输出函数 putchar

字符输出函数，其功能是在终端(显示器)输出单个字符

```
# include<stdio.h>
int main(void){
    int x=100;
    //putchar('A');// 输出大写字母A
    //putchar(x);// 输出d
    putchar('\n');
    return 0;
}
```

4.2 字符输入函数 getchar

`getchar` 函数的功能是接收用户从键盘输入的一个字符，其一般调用形式为: `getchar()`，返回一个 ASCII 码

```
# include<stdio.h>
int main(void){
    char c;
    c = getchar();
    printf("%s",c);
    return 0;
}
```

4.3 格式化输出函数 printf

`printf` 函数叫做格式化输出函数，其功能是按照用户指定的格式，把指定的数据输出到屏幕上。

格式字符	
d, i	以十进制形式输出有符号整数(正数不输出符号)
O	以八进制形式输出无符号整数(不输出前缀0)
x	以十六进制形式输出无符号整数(不输出前缀0x)
U	以十进制形式输出无符号整数
f	以小数形式输出单、双精度类型实数
e	以指数形式输出单、双精度实数
g	以%f或%e中较短输出宽度的一种格式输出单、双精度实数
C	输出单个字符
S	输出字符串

八进制 o 为小写

```
# include<stdio.h>
int main(void){
    printf("%d\n",10);// 整数
    printf("%o\n",10);// 八进制
    printf("%#x\n",10);//16进制
    printf("%u\n",10);//十进制输出无符号整数
    printf("%7.3f\n",1.0);// 宽度为7, 小数位为3
    printf("%C\n",100);//字符
    return 0;
}
/*
10
12
a
10
1.000
d
*/
```

```
/* 使用可变宽度输出字段*/
# include<stdio.h>
int main(void)
{
    unsigned width,precision;
    int number = 256;
    double weight = 25.5;
    printf("Please input number's width:\n");
    scanf("%d",&width);
    printf("The number is: %*d\n",width,number);
    printf("Then please input width and precision:\n");
    scanf("%d %d",&width,&precision);
    printf("weight = %*.*f\n",width,precision,weight);
    return 0;
}
```

```

}
/*
Please input number's width:
20
The number is:                256
Then please input width and precision:
9 5
weight = 25.50000
*/

```

具体的说，如果转换说明符为 `/*d`，那么参数列表中应该包括一个 `*` 的值和一个 `d` 的值，来控制宽度和变量的值。该技术也可以和浮点值一起使用来指定精度和字段宽度。

4.4 格式化输入函数 `scanf`

`scanf` 函数称为格式输入函数，即按照格式字符串的格式，从键盘把数据输入到指定的变量中。使用注意事项

- 格式说明符中，可以指定数据的宽度，但不能指定数据的精度

```

float a;
scanf("%10f",&a); //正确使用
scanf("10.2f",&a); //错误写法

```

- 输入 `long` 类型数据时，必须使用 `%ld`，输入 `double` 数据必须使用 `%lf` 或 `%le`
- 附加格式说明符 `*`，使对应的输入数据不附给相应的变量

```

/* 用*跳过scanf接收的数字*/
# include<stdio.h>
int main(void){
    int num;
    printf("please enter three number:\n");
    scanf("%*d %*d %d",&num);
    printf("The last number is %d\n",num);
    return 0;
}
/*
please enter three number:
10 20 30
The last number is 30
*/

```

转换说明符	
%c	把输入解释成一个字符
%d	把输入解释成一个有符号十进制整数
%e,%f,%g,%a	把输入解释成一个浮点数(%a是C99的标准)
%E,%F,%G,%A	把输入解释成一个浮点数(%A是C99的标准)
%i	把输入解释成一个有符号十进制整数
%o	把输入解释成一个有符号的八进制整数
%p	把输入解释成一个指针(一个地址)
%s	把输入解释成一个字符串: 输入的内容以第一个非空白字符作为开始, 并且包含直到下一个空白字符的全部字符
%u	把输入解释成一个无符号十进制整数
%x,%X	把输入解释成一个有符号十六进制整数

4.5 总结&练习

```
/*
要将"China"译成密码, 译码规律是: 用原来字母后面的第4个字母代替原来的字母.
例如, 字母"A"后面第4个字母是"E". "E"代替"A". 因此, "China"应译为"Glmre".
请编一程序, 用赋初值的方法使c1、c2、c3、c4、c5五个变量的值分别为, 'c'、'h'、'i'、'n'、'a',
经过运算, 使c1、c2、c3、c4、c5分别变为'g'、'l'、'm'、'r'、'e', 并输出。
*/
#include<stdio.h>
int main(void){
    char c1,c2,c3,c4,c5;
    scanf("%c%c%c%c%c",&c1,&c2,&c3,&c4,&c5);
    c1 = c1+4;
    c2 = c2+4;
    c3 = c3+4;
    c4 = c4+4;
    c5 = c5+4;
    printf("%c%c%c%c%c",c1,c2,c3,c4,c5);
    return 0;
}
/*
China
Glmre
*/
```

第五章 运算符和表达式

5.1 基本运算符

基本运算符

运算符	符号	操作	用例	结果
加法	+	使它两侧的值加到一起	1+2	3
减法	-	从它前面的数减去后面的数	5-3	2
乘法	*	将它前面的数乘以后面的数	2*3	6
除法	/	用它左边的值除以右边的值	8/3	2(整数的除法会丢掉小数部分)
取模	%	求用它左边的值除以右边的数后的余数	5%3	2(%运算符两侧的操作数必须为整数)

理解自增自减运算

```
#include<stdio.h>
int main()
{
    int a,b;
    a=b=5;
    //printf("%d",a-- + --b);// 9 a-- 后缀模式  --b 前缀模式
    printf("%d      %d\n",a--,--b);// 5 4
    printf("%d      %d\n",a--,--b);// 4 3
    printf("%d      %d\n",a--,--b);// 3 2
    printf("%d      %d\n",a--,--b);// 2 1
    printf("%d      %d\n",a--,--b);// 1 0
    return 0;
}
```

对于前缀运算符，先执行自增自减运算，在计算表达式的值，而后缀运算符，则先计算表达式的值，在执行自增或自减运算。

```
# include<stdio.h>
int main(void){
    int a=20;
    int b=5;
    int c=6;
    printf("a = %d b = %d c = %d\n",a,b,c);
    printf("a + b = %d\n",a+b);
    printf("a - c = %d\n",a-c);
    printf("a * b = %d\n",a*b);
    printf("a / c = %d\n",a/c);
    printf("a %% c = %d\n",a%c);/*两个%才会输出一个%*/
    return 0;
}
/*
a = 20 b = 5 c = 6
a + b = 25
a - c = 14
a * b = 100
a / c = 3
a % c = 2
*/
```


5.2 sizeof 运算符

sizeof 是一种单目运算符，以字节为单位返回某操作数的大小，用来求某一类型变量的长度，其运算对象可以是任何数据类型或变量。

单目运算符 指的是只需要一个操作数，自增：++ 自减操作：-- 双目运算符指的是 需要两个操作数
如 + - * / %

```
# include<stdio.h>
int main(void){
    int floatsize = sizeof(float);
    printf("float sizeof is %d bytes\n",floatsize);
    printf("long sizeof is %d bytes\n",sizeof(long));
    return 0;
}
/*
float sizeof is 4 bytes
long sizeof is 8 bytes
*/
```

5.3 三目运算符

格式

```
表达式1? 表达式2: 表达式3
2>1?10:20 // 10
int a=3,b=5;
int c =10;
c?:(a+b):(a-b)//8
```

5.4 表达式和语句

表达式

操作数和运算符组成

语句

一个语句是一条完整的计算机指令，在C语言中一个分号为一个语句。

5.5 总结&练习

```
/*
温度转换
输入一个华氏温度，要求输出摄氏温度。公式为 c=5(F-32)/9，取位2小数。
输入
一个华氏温度，浮点数
输出
摄氏温度，浮点两位小数
*/
# include<stdio.h>
int main(void){
    double f,c;
    scanf("%lf",&f);// double 类型个数输入函数使用 %lf
    c = 5*(f-32)/9;
    printf("c=%0.2lf",c);
```

```

        return 0;
    }
    /*
    100
    c=37.78
    */

```

```

/*
已知半径r，求一个圆的面积是多大。
*/
# include<stdio.h>
# include<math.h>
# define PI (atan(1.0)*4) // 定义宏
int main(void){
    double r,s;
    scanf("%lf",&r);
    s = r*r*PI;
    printf("%.2lf",s);
    return 0;
}
/*
84.95
*/

```

```

/*
拆分一个三位数的个位、十位、百位
*/
# include<stdio.h>
int main(void)
{
    int n,a[3];
    scanf("%d",&n);
    a[0] = n%10;//个位
    a[1] = n/10%10;// 十位
    a[2] = n/100;// 百位
    printf("%d %d %d",a[0],a[1],a[2]);
    return 0;
}
/*
321
1 2 3
*/

```

第六章 C语句和程序流

6.1 表达式和语句

6.2 选择结构

```

//形式一
if(表达式) /*若条件成立则实行花括号里的语句，反之则不执行*/
{
    //语句
}

```

```

}
//形式二
if(表达式) /*若表达式成立则执行语句1，否则执行语句2*/
{
    //语句1
}
else
{
    //语句2
}
// 形式3
if(表达式) /*如果表达式成立，执行语句1否则继续判断表达式2*/
{
    //语句1
}
else if(表达式2) /*如果表达式成立，执行语句2否则继续判断表达式3*/
{
    //语句2
}
else if(表达式3) /*如果表达式成立，则执行语句3否则继续判断下一个表达式*/
{
    //语句3;
}
//... ..
else /*如果以上表达式都不成立 则执行语句4*/
{
    //语句4
}
// switch结构
switch(表达式) /*首先计算表达式的值*/
{
    case 常量表达式1: 语句1;
    case 常量表达式2: 语句2;
    case 常量表达式3: 语句3;
    // ... ..
    case 常量表达式n: 语句n;
    default: 语句n+1;
}

```

6.3 循环结构

提供三种循环结构

- for

```

for(i=0;i<100;i++)
{
    printf("i count is %d\n",i);
}

```

- while

```

while(i++<10)
{
    printf("count %d ",i);
}

```

- do while

```
do
{
    printf("count %d", i);
}while(i<20);
```

6.4 总结&练习

```
/*
三个数找最大值
题目描述：有三个整数a b c,由键盘输入，输出其中的最大的数。
输入：一行数组，分别为a b c
输出：a b c其中最大的数
*/
# include<stdio.h>
int main(void){
    int a,b,c,t;
    scanf("%d,%d,%d",&a,&b,&c);
    if(a>b){
        t=a;
    }else{
        t = b;
    }
    if (c>t){
        t = c;
    }
    printf("%d\n",t);
    return 0;
}
/*
10,20,30
30
*/
```

```
/*
分段函数求值
题目描述：
有一个函数

$$y = \begin{cases} x & x < 1 \\ 2x-1 & 1 \leq x < 10 \\ 3x-11 & x \geq 10 \end{cases}$$

输入：一个数x
输出：一个数y
*/
# include<stdio.h>
int main(void){
    int x,y;
    scanf("%d",&x);
    if (x<1){
        y = x;
    }else if(x>=1 && x<10){
        y = 2*x -1;
    }else{
        y = 3*x-11;
    }
```

```

    }
    printf("%d",y);
    return 0;
}
/*
5
9
*/

```

```

/*
题目描述：给出一百分制成绩，要求输出成绩等级‘A’、‘B’、‘C’、‘D’、‘E’。 90分以及90分以上为A，
80-89分为B， 70-79分为C， 60-69分为D， 60分以下为E。
输入：一个整数0—100以内
输出：一个字符，表示成绩等级
*/
# include<stdio.h>
int main(void){
    char c;
    int score;
    scanf("%d",&score);
    if (score>=90){
        c = 'A';
    }else if(score>=80 && score<=89){
        c = 'B';
    }else if(score>=70 && score<=79){
        c = 'C';
    }else if(score>=60 && score<=69){
        c = 'D';
    }else{
        c = 'E';
    }
    printf("%c",c);
    return 0;
}
/*
92
A
*/

```

```

/*
题目描述：给出一个不多于5位的整数，要求 1、求出它是几位数 2、分别输出每一位数字 3、按逆序输出各位数字，例如原数为321,应输出123
输入：一个不大于5位的数字
输出： 三行 第一行 位数 第二行 用空格分开的每个数字，注意最后一个数字后没有空格 第三行 按逆序输出这个数
*/
#include<stdio.h>
int main()
{
    int i,count=0,x[5];
    for (i=0;i<5;i++){
        int temp=0;
        temp = getchar();// 返回ASCII码
        if (temp==10)// 10 为换行的ASCII码
            break;
        count++;
    }
}

```

```

        x[i] = temp-48;//字符0的ASCII码为48
    }
    printf("%d\n",count);
    for(i=0;i<=count-1;i++)
        printf("%d ",x[i]);
    printf("\n");
    for (i=count-1;i>=0;i--)
        printf("%d",x[i]);
    return 0;
}

```

```

/*
题目描述：
企业发放的奖金根据利润提成。利润低于或等于100000元的，奖金可提10%；
利润高于100000元，低于200000元（100000<I≤200000）时，低于100000元的部分按10%提成，高于
100000元的部分，可提成 7.5%；
200000<I≤400000时，低于200000元部分仍按上述办法提成，（下同），高于200000元的部分按5%提
成；
400000<I≤600000元时，高于400000元的部分按3%提成；

600000<I≤1000000时，高于600000元的部分按1.5%提成；
I>1000000时，超过1000000元的部分按1%提成。从键盘输入当月利润I,求应发奖金总数。
输入：一个整数，当月利润。
输出：一个整数，奖金。
*/
#include<stdio.h>
int main()
{
    int x,y;
    scanf("%d",&x);
    if(x<=100000)
    {
        y = x*0.1;
    }
    else if (x>100000 && x<200000){
        y = (x-100000)*0.075+100000*0.1;
    }else if(x>200000 && x<=400000){
        y = (x-200000)*0.05 + 100000*0.075 + 100000*0.1;
    }else if(x>400000 && x<=600000){
        y = 100000*0.1+100000*0.075+200000*0.05+(x-400000)*0.03;
    }else if(x>600000 && x<=1000000){
        y = 100000*0.1+100000*0.075+200000*0.05+200000*0.03+(x-600000)*0.015;
    }else{
        y=(x-
1000000)*0.01+100000*0.1+100000*0.075+200000*0.05+200000*0.03+400000*0.015;
    }
    printf("%d\n",y);
    return 0;
}
/*
1000000000
10029500
*/

```

```

/*
题目描述：输入两个正整数m和n，求其最大公约数和最小公倍数。

```

输入:两个整数

输出:最大公约数, 最小公倍数

```
*/
# include<stdio.h>
int gcd(int a,int b);//函数声明
int lcm(int a,int b);//函数声明
int main(void){
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d %d",gcd(a,b),lcm(a,b));
    return 0;
}
int gcd(int a,int b){
    if (b==0)
        return a;
    return gcd(b,a%b);//递归调用
}
int lcm(int a,int b){
    return (a*b)/gcd(a,b);// 利用 最大公约数*最小公倍数=两数值乘积
}
/*
5 7
1 35
*/
```

/*

题目描述:输入一行字符, 分别统计出其中英文字母, 数字, 空格和其他字符的个数

输入:一行字符,长度不超过200

输出:统计值

```
*/
# include<stdio.h>
int main(void){
    int a,c,d,b,e;
    a =c =d =b=0;
    while((e=getchar())!=10)// 10为换行的ASCII码
    {
        if (e>=65 && e<=90 || e>=97 && e<=122){
            a++;// 英文字母 , a自增
        }else if (e>=48 && e<=57){
            b++;// 遇到数字字符, b自增
        }else if(e==32){
            c++;//空格 c自增
        }else{
            d++;// 遇到其他字符, 自增
        }
    }
    printf("英文字母%d, 数字%d,空格%d,其他%d",a,b,c,d);
    return 0;
}
/*
qsdf fg 123$%^
英文字母6, 数字3,空格2,其他3
*/
```

/*

求 $S_n = a + aa + aaa + \dots + aa \dots aaa$ （有 n 个 a ）之值，其中 a 是一个数字，为2。例如， $n=5$ 时
 $= 2 + 22 + 222 + 2222 + 22222$ ， n 由键盘输入。

输入： n

输出： S_n 的值

*/

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

```
int main(void){
```

```
    int i,n,sum=0,a=2,sum1=0;
```

```
    scanf("%d",&n);
```

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

```
        sum = sum*10+a;
```

```
        sum1 +=sum;
```

```
    }
```

```
    printf("%d",sum1);
```

```
    return 0;
```

```
}
```

```
/*
```

```
5
```

```
24690
```

```
*/
```

```
/*
```

描述：求 $S_n = 1! + 2! + 3! + 4! + 5! + \dots + n!$ 之值，其中 n 是一个数字(n 不超过20)。

输入： n

输出： S_n

```
*/
```

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

```
long long fac(int n);// 声明函数
```

```
int main(void){
```

```
    int n,i;
```

```
    long long Sn=0;
```

```
    scanf("%d",&n);
```

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

```
        Sn = Sn+fac(i);
```

```
    }
```

```
    printf("%lld",Sn);
```

```
    return 0;
```

```
}
```

```
long long fac(int n){
```

```
    long long f;
```

```
    if (n==0)
```

```
        return 1;
```

```
    f = fac(n-1)*n;
```

```
    return f;
```

```
}
```

类型名称	字节数	取值范围
------	-----	------

signed` `char`	`1	-128~+127
----------------	----	-----------

short` `short`	`2	-32768~+32767(一个字节8位，2个字节16位，所以取值范围为2^16,考虑到有无符号，对半分)
----------------	----	---

int`	`4	-2147483648~+2147483647
------	----	-------------------------

long` `int`	`4	-2147483648~+2141483647
-------------	----	-------------------------

long` `long` `int`	`8	-9223372036854775808~+9223372036854775807
--------------------	----	---

//long int 的简写是 long 占位符是：%ld

//long long int的简写是 long long 占位符是: %lld

有符号

short %hd

int %d

long %ld

无符号

short %hd

int %u

long %lu

字符

char %c

double %lf

科学计数法输出double %e

字符串 %s

/*

题目描述: 求以下三数的和,保留2位小数 1~a之和 1~b的平方和 1~c的倒数和

输入:a b c

输出: $1+2+\dots+a + 1^2+2^2+\dots+b^2 + 1/1+1/2+\dots+1/c$

*/

include <stdio.h>

int main(void){

int a,b,c,i;

float sum=0.0;

scanf("%d %d %d",&a,&b,&c);

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

sum += i;

}

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

sum += i*i;

}

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

sum += 1.0/i;// 1.0/i 返回浮点数,往精度高的方向靠

}

printf("%.2f",sum);

return 0;

}

/*

题目描述:打印出所有"水仙花数",所谓"水仙花数"是指一个三位数,其各位数字立方和等于该本身。 例如:

153是一个水仙花数,因为 $153=1^3+5^3+3^3$ 。

输出每一个水仙花数

*/

include <stdio.h>

int main(void){

int i=100;

for (i=100;i<=999;i++){

if((i/100)*(i/100)*(i/100)+(i/10%10)*(i/10%10)*(i/10%10)+(i%10)*(i%10)*(i%10)==i){

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

}

}

return 0;

}

/*

153

```
370
371
407

*/
```

百位:n/100 十位:n/10%10 个位:n%10

```
/*
题目描述:一个数如果恰好等于不包含它本身所有因子之和,这个数就称为"完数"。 例如,6的因子为1、2、3,而6=1+2+3,因此6是"完数"。 编程序找出N之内的所有完数,并按下面格式输出其因子
输入:N
输出 ? its factors are ? ? ?
*/
#include<stdio.h>
int main(void){
    int N;
    scanf("%d",&N);
    int i,j,k,sum=0;
    for (i=2;i<=N;i++){
        for(j=1;j<=i/2;j++){// 寻找因子,并计算所有因子的和
            if (i%j==0)
                sum +=j;
        }
        if (sum==i){// 判断是否是完数
            printf("%d its factors are ",i);
            for(k=1;k<=i/2;k++){// 寻找因子,输出因子
                if(i%k==0)
                    printf("%d ",k);
            }
            printf("\n");
        }
        sum =0;//每次循环后将sum重置为0,开始下次循环,这一步非常重要
    }
    return 0;
}
/*
10000
6 its factors are 1 2 3
28 its factors are 1 2 4 7 14
496 its factors are 1 2 4 8 16 31 62 124 248
8128 its factors are 1 2 4 8 16 32 64 127 254 508 1016 2032 4064
*/
```

```
/*
有一分数序列: 2/1 3/2 5/3 8/5 13/8 21/13..... 求出这个数列的前N项之和,保留两位小数。
输入: N
输出数列前N项和
*/
#include<stdio.h>
int main(void){
    int a=2,b=1,i,N,temp;
    double sum=0;
    scanf("%d",&N);
    for (i=1;i<=N;i++){
        sum +=a*1.0/b;// 记得1.0, 否则没有小数
    }
}
```

```

        temp = a+b;
        b =a;
        a = temp;
    }
    printf("%.21f",sum);
    return 0;
}
/*
10
16.48
*/

```

```

/*
一球从M米高度自由下落，每次落地后返回原高度的一半，再落下。 它在第N次落地时反弹多高？共经过多少米？ 保留两位小数
输入 M N
输出： 它在第N次落地时反弹多高？共经过多少米？ 保留两位小数，空格隔开，放在一行
*/
# include<stdio.h>
int main(void){
    int size,i;
    double item=0,sum=0,high;
    scanf("%1f %d",&high,&size);// size 次数 high 高度
    item = high;
    for (i=0;i<size;i++){
        if (i==0)
            sum +=item;
        else sum+=2*item;
        item = (double)item/2;
    }
    printf("%.21f %.21f",item,sum);
    return 0;
}
/*
1000 5
1000 5
*/

```

```

/*
猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个。 第二天早上又将剩下的桃子吃掉一半，又多吃一个。以后每天早上都吃了前一天剩下的一半零一个。 到第N天早上想再吃时，见只剩下一个桃子了。求第一天共摘多少桃子。
输入:N
输出:桃子总数
*/
# include<stdio.h>
int main(void){
    int i,N,s=1;// s桃子总数 N 天
    scanf("%d",&N);
    for(i=1;i<N;i++){//因为猴子是第 N 天就只剩 1 个桃子了，所以第 N 天就没有算上
        s=(s+1)*2; //从第 N 天往回推，每次都是 s 个桃子加 1 个再乘 2 最后当 i=N 时结束循环
    }
    printf("%d",s);
    return 0;
}

```

```

/*
10
1534
*/

```

```

/*
牛顿迭代法求平方根
假设a。欲求a的平方根，首先猜测一个值x1=a/2，然后根据迭代公式 $x_{n+1} = (x_n + a/x_n) / 2$ ，算出x2，
再将x2代公式的右边算出x3等等，直到连续两次算出的xn和x（n+1）的差的绝对值小于某个值，即认为找到了
精确的平方根。例算步骤如下。
*/
# include<stdio.h>
# include<math.h>
int main(void){
    double x1,x2;
    int a;
    scanf("%d",&a);
    x1 = a*1.0/2;
    x2 = (x1+a*1.0/x1)/2;
    do {
        x1 = x2;
        x2 =(x1+a*1.0/x1)/2;
    }while(fabs(x1-x2)>=1e-5);
    printf("%lf",x2);
    return 0;
}
/*
6
2.449490
*/

```

```

/*
筛选法筛选素数，思路 创建要筛选的数组，将素数的元素重置为1，最后输出元素值不为1的元素就是素数。
*/
# include<stdio.h>
int main(void){
    int N;
    scanf("%d",&N);
    int arr[N];
    for (int i=0;i<N;i++){
        arr[i] = i+1;
    }
    for(int i=1;i<N;i++){
        if(arr[i]!=1){
            for(int j=i+1;j<N;j++){
                if (arr[j]!=1){
                    if (arr[j]!=1){
                        if(arr[j]%arr[i]==0){
                            arr[j]=1;
                        }
                    }
                }
            }
        }
        printf("%d\n",arr[i]);
    }
}

```

```
    return 0;
}
```

第七章 函数

7.1 函数的定义

函数是C源程序的基本模块，程序的很多功能都是通过对函数模块的调用来实现的。

```
返回值类型 函数名(形参表说明) /*函数首部*/
{
    说明语句 /*函数体*/
    执行语句
}
```

7.2 函数的调用

在主调函数中调用某函数之前应对该被调函数进行声明。在主调函数中对被调函数进行说明的目的是使编译系统知道被调函数返回值的类型,以便在主调函数中按此种类型对返回值进行相应的处理。其一般形式为:

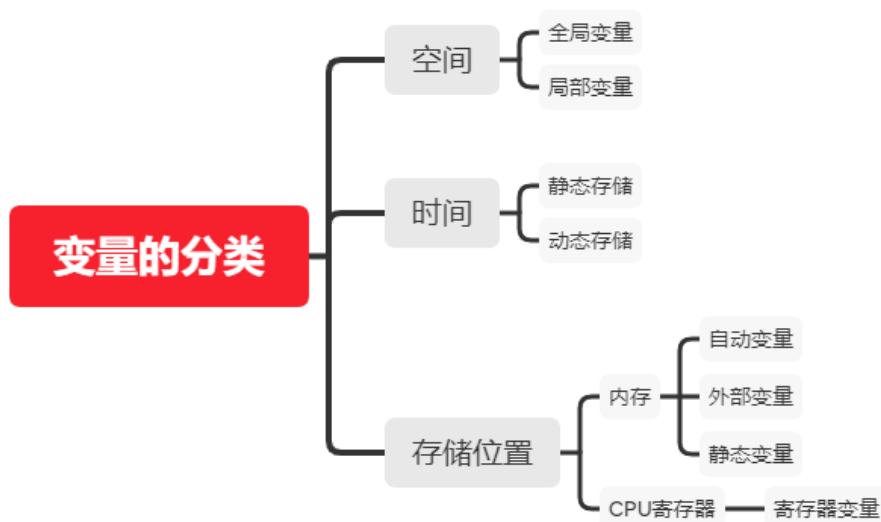
类型说明符 被调函数名(类型 形参,类型 形参...);

需要注意的是,函数的声明和函数的定义有本质上的不同。主要区别在以下两个方面:

- (1)函数的定义是编写一段程序,应有函数的具体功能语句——函数体;而函数的声明仅是向编译系统的一个说明,不含具体的执行动作。
- (2)在程序中,函数的定义只能有一次,而函数的声明可以有多次。

7.3 变量的存储类型

变量的分类



详细解释

- 全局变量

全局变量是在函数之外定义的变量,其作用范围为从定义处开始到本文件结束,编译时,编译系统为其分配固定的内存单元,在程序运行的自始至终都占用固定单元。

- 局部变量

局部变量是在一个函数或复合语句内定义的变量,它仅在函数或复合语句内有效,编译时,编译系统不为局部变量分配内存单元,而是在程序运行过程中,当局部变量所在的函数被调用时,编译系统根据需要,临时分配内存,调用结束,空间释放。

- 自动变量

函数中的**局部变量**,如不专门声明为 static 存储类别,都是动态地分配存储空间的,数据存储在动态存储区中。函数中的形参和在函数中定义的变量(包括在复合语句中定义的变量)都属此类,在调用该函数时系统会给它们分配存储空间,在函数调用结束时就自动释放这些存储空间。这类局部变量称为自动变量。自动变量用关键字 auto 进行存储类别的声明,例如声明一个自动变量:

- 外部变量

外部变量(即全局变量)是在函数的外部定义的,它的作用域为从变量定义处开始,到本程序文件的末尾。如果外部变量不在文件的开头定义,其有效的作用范围只限于定义处到文件末尾。如果在定义点之前的函数想引用该外部变量,则应该在引用之前用关键字 extern 对该变量进行“外部变量声明”。表示该变量是一个已经定义的外部变量。有了此声明,就可以从“声明”处起,合法地使用该外部变量。用 extern 声明外部变量,扩展程序文件中的作用域

- 静态变量

有时希望函数中的局部变量的值在函数调用结束后不消失而保留原值,这时就应该指定局部变量为静态局部变量,用关键字 static 进行声明。

- 寄存器变量

为提高效率,C 语言允许将局部变量的值存放在 CPU 的寄存器中,这种变量叫做寄存器变量,用关键字 register 声明。使用寄存器变量需要

注意 只有局部自动变量和形式参数可以作为寄存器变量;一个计算机系统上的寄存器数目有限,不能定义任意多个寄存器变量;不能使用取地址运算符 & 求寄存器变量的地址。

7.4 总结&练习

```
/*
题目描述:输入10个数字,然后逆序输出。
输入:10个整数
输出:逆序输出,空格分开
*/
#include<stdio.h>
int fun();// 声明函数
int main(void){
    fun();
    return 0;
}
// 定义函数
int fun(){
    int a[10],i,j;
    for (i=0;i<10;i++)
        scanf("%d",&a[i]);
    for (j=9;j>=0;j--)
        printf("%d ",a[j]);
}
```

```
/*
题目描述:写两个函数,分别求两个整数的最大公约数和最小公倍数,用主函数调用这两个函数,并输出结果两个整数由键盘输入。
输入:两个数
```

输出：最大公约数，最小公倍数

```
*/
#include<stdio.h>
int gcd(int a,int b);// 定义最大公约数
int lcm(int a,int b);//定义最小公倍数
int main(void){
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d %d",gcd(a,b),lcm(a,b));
    return 0;
}
int gcd(int a,int b){
    if (b==0)
        return a;
    else
        return gcd(b,a%b);
}
int lcm(int a,int b){
    return (a*b)/gcd(a,b);// 利用公式 最大公约数*最小公倍数=两数乘积
}
/*
6 15
3 30
*/
```

```
/*
题目描述:求方程 的根，用三个函数分别求当 $b^2-4ac$ 大于0、等于0、和小于0时的根，并输出结果。从主函数输入a、b、c的值。
输入:a b c
输出:x1= ,x2=
*/
#include<stdio.h>
#include<math.h>
void tow_num(int a, int b, double num);
void imag_num(int a, int b, double num);
void one_num(int a, int b);
//用三个函数分别求当 $b^2-4ac$ 大于0、等于0、和小于0时的根，并输出结果。从主函数输入a、b、c的值。
/*
**求值公式 $(-b + \text{pow}((b*b - 4ac), 0.5))/ 2 * a$ 
*/
int main(int argc, char* argv[])
{
    int a, b, c;
    double num = 0;
    scanf("%d %d %d", &a, &b, &c);
    num = (double)b * b - 4 * a * c;
    /*
    **判断 $b^2-4ac$ 的情况
    */
    if (num > 0)
        tow_num(a, b, num);
    else if (num < 0)
        imag_num(a, b, num);
    else
        one_num(a, b);
    return 0;
}
```

```

/*
**大于等于0时的情况
*/
void tow_num(int a, int b, double num)
{
    printf("x1=%0.31f x2=%0.31f", (double)(-b + pow(num, 0.5)) / (2 * a),
    (double)(-b - pow(num, 0.5)) / (2 * a));
}
/*
**等于0时的情况
*/
void one_num(int a, int b)
{
    printf("x1=%0.31f x2=%0.31f", (double)-b / (2 * a), (double)-b / (2 * a));
}
/*
**小于0时的情况
*/
void imag_num(int a, int b, double num)
{
    double real_num, unreal_num;
    real_num = (double)-b / (2 * a);
    unreal_num = (double)pow(-num, 0.5) / (2 * a);
    printf("x1=%0.31f+%0.31fi x2=%0.31f-%0.31fi", real_num, unreal_num,
    real_num, unreal_num);
}
/*
5 10 5
x1=-1.000 x2=-1.000
*/

```

```

b^2-4ac>0 x=(-b+-`sqrt` (b^2-4ac))/2a b^2-4ac<0` (-b+-`sqrt` (4ac-b^2)) i/2a
b^2-4ac=0 -b/2a

```

```

/*
题目描述:写一个判断素数的函数，在主函数输入一个整数，输出是否是素数的消息。
输入一个数
输出:如果是素数输出prime 如果不是输出not prime
*/
# include<stdio.h>
int is_Prime();
int main(void){
    is_Prime();
    return 0;
}
int is_Prime(){
    int i,a;
    scanf("%d",&a);
    for (i=2;i<=a/2;i++){
        if (a%i==0){
            printf("not prime");
            break;
        }
        else{
            printf("prime");
        }
    }
}

```



```
}
```

```
/*
写一个函数，使给定的一个二维数组（3×3）转置，即行列互换。
输入：一个3x3的矩阵
输出：
*/
# include<stdio.h>
int main(void){
    int a[3][3],i,j;
    for (i=0;i<3;i++){
        for(j=0;j<3;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            printf("%d ",a[j][i]); //打印的时候直接行列转换
        }
        printf("\n");
    }
    return 0;
}
```

```
/*
写一函数，使输入的一个字符串按反序存放，在主函数中输入并输出反序后的字符串（不包含空格）。
输入：一行字符
输出：逆序后的字符串
*/
# include<stdio.h>
int main()
{
    char arr[1000];
    char *p = arr;
    scanf("%s",&arr);
    while (*p!='\0'){
        p++;
    }
    while(p!=arr){
        p--;
        printf("%c",*p);
    }
    return 0;
}
/*
123adfd678@##
##@876dfda321
*/
```

```
/*
题目描述：写一函数，将两个字符串连接
输入：两行字符串
输出：链接后的字符串
*/
# include<stdio.h>
```

```
# include<string.h>
int main(void){
    char st1[100],st2[100];
    gets(st1);// 获取整行字符输入
    gets(st2);
    strcat(st1,st2);//将st2追加到st1后
    puts(st1);// 输出字符换行
    return 0;
}
```

补充 gets() 和 scanf()区别 puts() 和 scanf() 区别

```
#include<stdio.h>
int main()
{
    char a[10],b[10];
    int c,d;
    scanf("%s",a);
    printf("a:%s\n",a);
    c = getchar();
    printf("c:%d",c);
    gets(b);
    printf("b:%s\n",b);
    d = getchar();
    printf("d:%c",d);
    return 0;
}
/*
hello 123
world
// 输出:
a:hello
c:32b:123
d:w
*/
```

scanf() 当遇到回车，空格和TAB键会自动在字符串后面天加 \0,但是回车，空格和tab键仍会留在输入的缓冲区中，所以 a 的输出 是hello\0 \0代表的是字符串的结尾，因为空格保留在缓冲区中，所以c的值为32，空格的ASCII码值，gets()可接受回车键之前输入的所有字符，并用'\0'替代 '\n'.回车键不会留在输入缓冲区中，所以 b 输出123，空格不在缓冲去所以 d输出 w;puts () 在输出字符串时会将'\0'自动转换成'\n'进行输出，也就是说，puts方法输出完字符串后会自动换行。

```
/*
题目描述：
写一函数，将一个字符串中的元音字母复制到另一个字符串，然后输出。
输入：一行字符串
输出：顺序输出其中的元音字母（aeiou）
*/
#include<stdio.h>
int main()
{
    char a[5]={'a','e','i','o','u'};// 定义元音字母数组
    char b[100];
    scanf("%s",b);
    for(int j=0;j<100;j++){// 双循环遍历
        for (int i=0;i<5;i++){
```

```

        if (a[i]==b[j]){
            printf("%c",b[j]);
        }

    }

}

return 0;
}

```

```

/*
题目描述：写一函数，输入一个四位数字，要求输出这四个数字字符，但每两个数字间空格。如输入1990，应
输出"1 9 9 0"。
输入:1990
输出:1 9 9 0
*/
#include<stdio.h>
int main()
{
    char a[4];
    scanf("%s",a);
    char *p =a;// 定义指针
    for (int i=0;a[i]!='\0';i++){// 使用指针进行遍历
        printf("%c ",*(p+i));
    }
    return 0;
}

```

```

/*
题目描述：写一函数，统计字母 数字 空格 其他字符的个数
输入：!@#$%^QWERT 1234567
输出：5 7 4 6
*/
// 方法一 使用ASCII码 有可能超时
#include<stdio.h>
int filter();
int main(void)
{
    filter();
    return 0;
}
int filter(){

    int number=0,string=0,space=0,others=0,e;
    while((e=getchar())!=10){// getchar() 一个个读取输入的字符 10 为换行的ASCII码
        if(e==32){// 空格
            space++;
        }else if(e>=48 && e<=57){// 数字
            number++;
        }else if(e>=65 && e<=90 || e>=97 && e<=122){ // 字母
            string++;
        }else{// 其他
            others++;
        }
    }
    printf("%d %d %d %d",string,number,space,others);
}

```

```

/*
题目描述：判断一个数是否为"水仙花数"，所谓"水仙花数"是指这样的数：首先是一个三位数，其次，其各位数字的立方和等于该数本身。例如：371是一个"水仙花数"， $371=3^3+7^3+1^3$ 。
输入：一个三位数
输出：1或者0(1代表此数为水仙花数,0代表此数不是水仙花数)
*/
#include<stdio.h>
int main(void){
    int a,b,c,d;
    scanf("%d",&a);
    b = a/100;// 百位
    c = a/10%10;// 十位
    d = a%10;// 个位
    if (a== b*b*b+c*c*c+d*d*d){
        printf("%d",1);
    }else{
        printf("%d",0);
    }
    return 0;
}

```

```

/*
题目描述：输出所有的"水仙花数"。所谓"水仙花数"是指这样的三位数：其各位数字的立方和等于该数本身。例如：371是一个"水仙花数"， $371=3^3+7^3+1^3$ 。
输出所有的"水仙花数"(从小到大的顺序输出，一行一个)
*/
#include<stdio.h>
int main(void){
    for(int i=100;i<=999;i++){
        int b = i/100;
        int c = i/10%10;
        int d = i%10;
        if (i== b*b*b+c*c*c+d*d*d){
            printf("%d\n",i);
        }
    }
    return 0;
}
/*
153
370
371
407
*/

```

```

/*
题目描述：
一个自然数被8除余1,所得的商被8除也余1,
再将第二次的商被8除后余7,最后得到一个商为a.
又知这个自然数被17除余4.所得的商被17除余15,
最后得到一个商是a的2倍.

求这个自然数.
*/
// 直接暴力求解
#include<stdio.h>

```

```

int main()
{
    int i;
    for(i=21;i<2000;i++)
    {
        if(i%8==1&& i/8%8==1&& i/8/8%8==7&& i%17==4&& i/17%17==15&& i/17/17==i/8/8/8*2)
        printf("%d\n",i);
    }
    return 0;
}
/*
1993
*/

```

```

/*
题目描述：两个不同的自然数A和B，如果整数A的全部因子(包括1，不包括A本身)之和等于B；且整数B的全部因子(包括1，不包括B本身)之和等于A，则将整数A和B称为亲密数。求3000以内的全部亲密数。
3000以内的全部亲密数(输出格式:(A,B)，不加换行，不加分隔符号)
一对亲密数只输出一次，小的在前
(220,284)(1184,1210)(2620,2924)
*/
#include <stdio.h>
int wan(int a) {
    int sum = 1, i;
    for (i = 2; i < a; i++) {
        if (a % i == 0)
            sum += i; // a的全部因子求和
    }
    return sum;
}
int main() {
    int i, j;
    int sum, sam;
    for (i = 1; i <= 3000; i++) {
        sum = wan(i);
        if (sum == 1) continue; //素数
        for (j = i + 1; j <= 3000; j++) {
            if (sum != j) continue; //先判断 i因数和与j 是否相等(省时)
            sam = wan(j);
            if (sam == 1) continue; //素数
            if (sum == j && sam == i) {
                printf("(%d,%d)", i, j);
            }
        }
    }
    return 0;
}

```

```

/*
题目描述：按递增顺序依次列出所有分母为40，分子小于40的最简分数。
输出：
1/40,3/40,7/40,9/40,11/40,13/40,17/40,19/40,21/40,23/40,27/40,29/40,31/40,33/40,
37/40,39/40,
*/
# include<stdio.h>

```

```

int yinzi();
int main(void){
    yiniz();
    return 0;
}
int yiniz(){
    int i;
    for(i=1;i<40;i++){
        if(i%2!=0 && i%5!=0){
            printf("%d/40,",i); // 格式化输出, 注意,
        }
    }
}
/*
1/40,3/40,7/40,9/40,11/40,13/40,17/40,19/40,21/40,23/40,27/40,29/40,31/40,33/40,
37/40,39/40,
*/

```

```

/*
题目描述:输入一串字符,将其中的大写变成小写,若不为大写则原样输出
*/
// 思路使用ASCII码 有可能超时
#include<stdio.h>
int lower();
int main(void){
    lower();
    return 0;
}
int lower(){
    int e=0;
    while((e=getchar())!=10){ // 10 为换行
        if (e>=65 && e<=90){ // [65,90] A-Z 加32 将大写字母转为小写
            e = e+32;
        }
        printf("%c",e);
    }
}
// 方法二:
#include<stdio.h>
#include<string.h>
int main(){
    char a[100];
    gets(a);
    for(int i=0; i<strlen(a); i++)
    {
        if(a[i]>='A'&&a[i]<='Z')
            a[i]+=32;
    }
    puts(a);
    return 0;
}

```

```

/*
题目描述:
某侦察队接到一项紧急任务,要求在A、B、C、D、E、F六个队员中尽可能多地挑若干人,但有以下限制条件:
1)A和B两人中至少去一人; a+b>=1

```

2)A和D不能一起去; $a+d \leq 1$

3)A、E和F三人中要派两人去; $a+e+f=2$

4)B和C都去或都不去; $b+c=2$ or $b+c=0$

5)C和D两人中去一个; $c+d=1$

6)若D不去,则E也不去。 $d+e=0$ or $d=1$

问应当让哪几个人去?

输出: A,B,C,F,

*/

// 去设置为1 不去设置为0

include<stdio.h>

int main(void){

int a,b,c,d,e,f;

for(a=0;a<=1;a++){

for(b=0;b<=1;b++){

for(c=0;c<=1;c++){

for(d=0;d<=1;d++){

for(e=0;e<=1;e++){

for(f=0;f<=1;f++){

// a 和 b 至少去一个 $a+b \geq 1$ a和d不能一起去 $a+d \leq 1$,要么a去要

么 d去, 要么都不去,

if(a+b>=1&& a+d<=1&& a+e+f==2&&

(b+c==2 || b+c==0)&& c+d==1&& (d+e==0 || d==1))

{

if(a)

printf("A,");

if(b)

printf("B,");

if(c)

printf("C,");

if(d)

printf("D,");

if(e)

printf("E,");

if(f)

printf("F,");

}

}

}

}

}

}

}

return 0;

}

/*

A,B,C,F,

*/

/*

题目描述: 所给字符串正序和反序连接, 形成新串并输出

输入:123abc

输出: 123abccba321

*/

include<stdio.h>

include<string.h>

int main(void){

char a[50];

```

    gets(a);
    for (int i=0;i<strlen(a);i++){
        printf("%c",a[i]);
    }
    for(int i=strlen(a)-1;i>=0;i--){
        printf("%c",a[i]);
    }
    return 0;
}

```

```

/*
题目描述:验证尼科彻斯定理,即:任何一个整数m的立方都可以写成m个连续奇数之和。
样例输入:13
样例输出:13*13*13=2197=157+159+161+163+165+167+169+171+173+175+177+179+181
*/
/*
通过演绎推理 可以得出规律
1*1*1=1=1
2*2*2=8=3+5
3*3*3=27=7+9+11
4*4*4=64=13+15+17+19
5*5*5=125=21+23+25+27+29
可以看到我们只需要找到第一项,后续就是一个等差数列求和,第一项等于 a*a-(a-1)
*/
#include<stdio.h>
int main(void){
    int a;
    scanf("%d",&a);
    int i = a*a-(a-1);// 获取第一个奇数的值
    printf("%d*d*d=%d=%d=",a,a,a,a*a*a);
    for (int j=1;j<=a;j++){
        if(j!=a){
            printf("%d+",i);
        }else{
            printf("%d",i);
        }
        i +=2;
    }
    return 0;
}

```

```

/*
题目描述:将四个整数进行从小到大的顺序排列
输入: 5 4 3 2 1
输出: 1 2 3 4 5
*/
#include<stdio.h>
int main(void){
    int i,j,t,a[5];
    for(i=0;i<=4;i++){
        scanf("%d",&a[i]);
    }
    for(i=0;i<5;i++){// 外循环为排序的趟数
        for(j=i+1;j<=4;j++){// 内循环为每趟比较的次数

```



```

        if(a[i]>a[j]){
            t =a[i];
            a[i]=a[j];
            a[j]=t;
        }
    }
}
for (i=0;i<5;i++){
    printf("%d ",a[i]);
}
return 0;
}

```

```

/*
题目描述:将十个数进行从大到小的顺序进行排列
输入:1 2 3 4 5 6 7 8 9 10
输出:10 9 8 7 6 5 4 3 2 1
*/
#include<stdio.h>
int main(void){
    int i,j,t,a[10];
    for(i=0;i<=9;i++){
        scanf("%d",&a[i]);
    }
    for(i=0;i<10;i++){// 外循环为排序的趟数
        for(j=i+1;j<=9;j++){// 内循环为每趟比较的次数
            if(a[i]<a[j]){
                t =a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
    for (i=0;i<10;i++){
        printf("%d ",a[i]);
    }
    return 0;
}

```

```

/*
题目描述:输入一个字符串,数出其中的字母的个数.
样例输入:1241fdk54AIEJ92854&%%$GJ
*/
# include<stdio.h>
# include<string.h>
int main(void){
    char a[100];
    gets(a);
    int sum=0;
    for(int i=0;i<strlen(a);i++){
        if((a[i]>='a'&& a[i]<='z')|| (a[i]>='A'&& a[i]<='Z')){
            sum++;
        }
    }
    printf("%d",sum);
    return 0;
}

```

```
}
```

```
/*
```

题目描述：斐波纳契数列

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.....这个数列则称为“斐波纳契数列”，其中每个数字都是“斐波纳契数”。

输入：一个整数N(N不能大于40)

输出：由N个“斐波纳契数”组成的“斐波纳契数列”。

```
*/
```

```
# include<stdio.h>
int fibo(int N);
int main(void){
    int N;
    scanf("%d",&N);
    fibo(N);
    return 0;
}
int fibo(int N){
    int a[N];
    for(int i=0;i<N;i++){
        if (i<2){
            a[i]=1;
        }else{
            a[i]=a[i-1]+a[i-2];
        }
    }
    for (int i=0;i<N;i++){
        printf("%d ",a[i]);
    }
}
```

```
/*
```

题目描述：输入若干个整数,以-1标记输入结束。输出其中的最大数

输入：若干个整数。（以-1标记输入结束）

输出：其中的最大数

```
*/
```

```
# include<stdio.h>
int main(void){
    int max,n;
    scanf("%d",&n);
    max = n;
    while(n!=-1){
        scanf("%d",&n);
        if(n!=-1){
            if(max<=n){
                max = n;
            }
        }
    }
    printf("%d\n",max);
    return 0;
}
```

```
/*
```

题目描述：求 $1+2!+3!+\dots+N!$ 的和

输入:正整数N (N <=20)

输出:1+2!+3!+...+N!的和 (结果为整数形式)

*/

// 思路使用递归求解

include<stdio.h>

long long fibo(int n);// 设置返回值为 long long int 类型, long long 为long long int 简写

int main(void){

long long sum=0;// 防止溢出, 设置long long 类型

int n;

scanf("%d",&n);

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

sum +=fibo(i);

}

printf("%lld",sum);// long long 的格式化输出 %lld

return 0;

}

long long fibo(int n){

long long s1;

if (n==1){

return 1;

}else{

s1 = fibo(n-1)*n;

}

return s1;

}

/*

题目描述: 利用 $\pi/4=1-1/3+1/5-1/7\ldots$ 公式求pi的近似值, 当某一项的绝对值小于 10^{-6} 为止

输出: PI的近似值

*/

#include<stdio.h>

int main(void){

double sum;

double i=1;

int j=1;

// 不知道循环次数使用while 循环

while(1.0/i>=1e-6){

if(j%2==0){

sum -=1/i;

}else{

sum +=1/i;

}

i +=2;

j++;

};

printf("%.61f",4*sum);

return 0;

}

/*

题目描述:求 $S=a+aa+aaa+aaaa+aa\ldots a$ 的值, 其中a是一个一位的整数。

例如2+22+222+2222+22222(此时共有5个数相加)

输入:整数a和n (n个数相加, $1\leq n, a\leq 9$)

输出:s的值

*/

```
# include<stdio.h>
int main(void){
    int a,n;
    scanf("%d %d",&a,&n);
    int sum=0,sum1=0;
    for(int i=1;i<=n;i++){
        sum = sum*10+a;
        sum1 +=sum;
    }
    printf("%d",sum1);
    return 0;
}
```

```
/*
题目描述：3025这个数具有一种独特的性质：将它平分为二段，即30和25，使之相加后求平方，即
(30+25)2，恰好等于3025本身。请求出具有这样性质的全部四位数
*/
# include<stdio.h>
int main(void){
    int s1,s2;
    for (int a=1000;a<=9999;a++){
        // 将数据进行拆分 注意 除法/ 和Python 不一样 Python // 代表整除，
        s1 = a/100;
        s2 = a- (a/100)*100;
        if ((s1+s2)*(s1+s2)==a){
            printf("%d ",a);
        }
    }
    return 0;
}
```

```
/*
题目描述：按如下递归公式求函数值。
x=1时 f(x)=10; x>1时 f(x)=f(x-1)+2
输入：整型变量x
输出：f(x)
*/
# include<stdio.h>
int f(int x);
int main(void){
    int x;
    scanf("%d",&x);
    int sum=0;
    sum = f(x);
    printf("%d",sum);
    return 0;
}
int f(int x){
    if(x==1){
        return 10;
    }else{
        return f(x-1)+2;
    }
}
```

```

/*
题目描述:求矩阵的两对角线上的元素之和
输入:矩阵的行数N
和一个N*N的整数矩阵a[N][N](N<=10)
输出:
所输矩阵的两对角线上的元素之和
*/
/*
解题思路:
1. 设元素下标为i, j , 矩阵阶数为Length, 和为sum;

2. 主对角线元素下标满足: i==j;

3. 副对角线元素下标满足: i+j-1==Length;

4. 输入一个元素Matrix, 判断下标是否满足主副对角线元素下标条件, 满足, sum加上这个数;
*/
#include <stdio.h>

int main()
{
    int sum = 0, Length, Matrix;
    scanf( "%d", &Length );      //阶数

    for ( int i = 1; i <= Length; i++ )    //行
        for ( int j = 1; j <= Length; j++ )    //列
        {
            scanf( "%d", &Matrix );
            if ( (i == j) || (i + j - 1) == Length )    //判断
                sum += Matrix;          //求和
        }
    printf( "%d", sum );    //输出
    return(0);
}

```

```

/*
题目描述:求出1-N中的所有素数
输入: 大于1的正整数N
输出: 1-N中的所有素数,(以从小到大的格式输出)
*/
#include<stdio.h>
void out_put(int i);
int main(void){
    int N;
    scanf("%d",&N);
    for (int i=2;i<=N;i++){
        out_put(i);
    }
    return 0;
}
void out_put(int i){
    int j;
    for (j=2;j<i;j++){
        if(i%j==0){
            break;
        }
    }
}

```

```

        if(j==i){
            printf("%d ",i);
        }
    }
}

```

/*

题目描述：一辆以固定速度行驶的汽车，司机在上午10点看到里程表上的读数是一个对称数(即这个数从左向右读和从右向左读是完全一样的)，为95859。两小时后里程表上出现了一个新的对称数。问该车的速度是多少？新的对称数是多少？

*/

```

#include<stdio.h>
int main()
{
    int a,b,c,d,e;
    int f;
    for(a=95860;a<99999;a++)//因为已经是95859了,所以直接从59860开始
    {
        b=a/10000;// 万位
        c=a/1000%10;//千位
        d=a/10%10;//十位
        e=a%10;//个位
        if(b==e && c==d)
        {
            printf("%d\n",a);
            break;
        }
    }
    return 0;
}

```

/*

题目描述：中国古代数学家张丘建在他的《算经》中提出了著名的“百钱买百鸡问题”：鸡翁一，值钱五，鸡母一，值钱三，鸡雏三，值钱一，百钱买百鸡，问翁、母、雏各几何？

*/

// 直接暴力求解

```

#include<stdio.h>
int main(void){
    for (int x=0;x<=100;x++){
        for(int y=0;y<=100;y++){
            for(int z=0;z<=100;z++){
                if (5*x+3*y+(z/3.0)==100&& x+y+z==100){
                    printf("cock=%d,hen=%d,chicken=%d\n",x,y,z);
                }
            }
        }
    }
    return 0;
}

```

/*

题目描述：试求满足下述立方和不等式的m的整数解。

$$1^3+2^3+\dots+m^3 \leq n$$

本题算法如下：

对指定的n，设置求和循环，从i=1开始，i递增1取值，把i³(或i*i*i)累加到s，直至s>=n，脱离循环作相应的打印输出。

```

输入:n
输出:m
*/
#include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    int sum=0;
    int i=0;
    while(sum<=n){
        i++;
        sum +=i*i*i;
    }
    printf("%d",i-1);// 跳出循环时 i已经自增了，需要往后移一位

    return 0;
}

```

```

/*
题目描述:编写一个程序判断一个数是否为素数
输入:整数
输出: 1或0(其中1表示此数为素数,0为表示为不是素数)
*/
// 不考虑 1 2 的情况，默认输入的n >2
# include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    int flag=0;// 设置标志位 flag=0 代表不是素数
    for (int i=2;i<n;i++){
        if(n%i==0){
            flag=0;// [2,n-1]之间有能够整除的数，说明不是素数
            break;
        }else{
            flag=1;
        }
    }
    if (flag==1){
        printf("%d",1);
    }else{
        printf("%d",0);
    }
    return 0;
}

```

```

/*
题目描述：自守数是指一个数的平方的尾数等于该数自身的自然数。
例如：
25^2=625
76^2=5776
9376^2=87909376
请求出200000以内的自守数？
*/
/* 思路
(625-25)%100==0
(5776-76)%100==0

```

```

(87909376-9376)%10000==0
(平方-自然数)%pow(10,n)==0 n 为自然数位数
*/
#include<stdio.h>
#include<math.h>
long Figure(long n);// 定义获取数字位数的函数
int main(void){
    for (long i=0;i<=200000;i++){
        long sum =0;
        sum = i*i;
        long n = 0;
        n =Figure(i);// 得到自然数位数
        long n1 = pow(10,n);// 获取位数的平方
        long n2 = sum-i;// 平方-自然数
        if (fmod(n2,n1)==0){
            printf("%ld ",i);
        }
    }
    return 0;
}
// 定义求自然数位数的方法
long Figure(long n)
{
    long i = 0;

    while(n!=0)
    {
        n = n / 10;//每次除以10
        i++;//统计循环次数
    }

    return i;
}

```

```

/*
题目描述:一个球从100m高度自由落下,每次落地后反弹回原来高度的一半,再落下,再反弹.求它在第N次落地
时共经过多少米?
*/
#include<stdio.h>
int main(void){
    double h=100;
    double s=0;
    int n;
    scanf("%d",&n);
    for (int i=1;i<=n;i++){
        s += h;
        h/=2;
        s+=h;
    }
    printf("%.4lf",s-h);// 注意最后一次落地不计算反弹了,所以这里要减去
    return 0;
}

```

```

/*

```


题目描述:相传国际象棋是古印度舍罕王的宰相达依尔发明的。舍罕王十分喜爱象棋,决定让宰相自己选择何种赏赐。这位聪明的宰相指着 8×8 共64格的象棋说:陛下,请您赏给我一些麦子吧。就在棋盘的第1格放1粒,第2格放2粒,第三格放4粒,以后每一格都比前一格增加一倍,依此放完棋盘—64格,我就感激不尽了。舍罕王让人扛了一袋麦子,他要兑现他的许诺。

请问,国王要兑现他的许诺共要多少粒麦子赏赐他的宰相?

```
*/  
/*  
1 2^0  
2 2^1  
3 2^2  
4 2^3  
累加求和  
*/  
# include<stdio.h>  
# include<math.h>  
int main(void){  
    double sum;  
    for (int i=0;i<64;i++){  
        sum +=pow(2,i);  
    }  
    printf("%lf",sum);// 用double存储超大的数  
    return 0;  
}
```

```
/*
```

题目描述:角谷猜想:

日本一位中学生发现一个奇妙的“定理”,请角谷教授证明,而教授无能为力,于是产生角谷猜想。猜想的内容是:任给一个自然数,若为偶数除以2,若为奇数则乘3加1,得到一个新的自然数后按照上面的法则继续演算,若干次后得到的结果必然为1。请编程验证

输入:任意整数

输出:演算过程

```
*/  
# include<stdio.h>  
int main(void){  
    int n;  
    scanf("%d",&n);  
    while(n!=1){  
        if (n%2==0){  
            printf("%d/%d=%d\n",n,2,n/2);  
            n = n/2;  
        }else{  
            printf("%d*3+1=%d\n",n,(n*3)+1);  
            n = n*3+1;  
        }  
    }  
    return 0;  
}  
/*  
输入: 10  
演算过程:  
10/2=5  
5*3+1=16  
16/2=8  
8/2=4  
4/2=2  
2/2=1  
*/
```

```

/*
题目描述:编写一个程序,计算1977! 的值
*/
// 大数阶乘使用数组
#include<stdio.h>
int main()
{
    int num[10000],i,j=1,rem=0,n,len=1;
    for(i=0;i<10000;i++)
    {
        num[i]=0;    //初始值为0
    }
    num[0]=1;    //令其第一位为1
    while(j<=1977)    //设置变量当小于等于1977时执行循环
    {
        for(i=0;i<=len;i++)    //len为位数,就是num[i]所存的位数
        {
            rem=rem+num[i]*j;
            num[i]=rem%10;
            rem=rem/10;
            if(rem>0&&i==len)
            {
                len++;
            }
        }
        j++;
    }
    for(i=len;i>=0;i--)    //最后将所以位数倒置,就是所想要输出的数据了
        printf("%d",num[i]);
}

```

```

/*
计算1~N之间所有奇数之和
输入:正整数N
输出:所有奇数的和
*/
# include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    int sum=0;
    for (int i=1;i<=n;i+=2){// 步长为2取奇数,避免判断
        sum +=i;
    }
    printf("%d",sum);
    return 0;
}

```

```

/*
计算t=1+1/2+1/3+...+1/n
输入:n
输出: t (保留六位小数)
*/
# include<stdio.h>
int main(void){
    int n;

```

```

scanf("%d",&n);
double sum=0;
for (int i=1;i<=n;i++){
    sum += 1.0/i;
}
printf("%.6lf",sum);
return 0;
}

```

```

/*
题目描述： 计算一个整数N的阶乘
输入：一个整数N，(0 <=N <=12)
输出：整数N的阶乘。
*/
#include <stdio.h>
int main()
{
    int i = 1,n = 0;
    int sum = 1;
    scanf("%d",&n);
    if((n>=0) && (n<=12))
    {
        while(i<=n)
        {
            sum *= i;
            i++;
        }
    }
    printf("%d\n",sum);
    return 0;
}

```

```

/*
题目描述： 计算：  $t=1-1/(2*2)-1/(3*3)-\dots-1/(m*m)$ 
输入：整型变量m
输出：t(保留六位小数)
*/
#include<stdio.h>
int main(void){
    int m;
    scanf("%d",&m);
    double sum=1;// 使用double 的原因是，float精度只有6~7 位，double精度有15位
    for(int i=2;i<=m;i++){
        sum -=1.0/(i*i);
    }
    printf("%.6lf",sum);
    return 0;
}

```

float 精度 6~7位，能保证的只有6位，double的精度为15~16位,能保证的有15位

```

/*
题目描述：张王李三家各有三个小孩。一天，三家的九个孩子在一起比赛短跑，规定不分年龄大小，跑第一得9分，跑第2得8分，依此类推。比赛结果各家的总分相同，且这些孩子没有同时到达终点的，也没有一家的两个或三个孩子获得相连的名次。已知获第一名的是李家的孩子，获得第二的是王家的孩子。问获得最后一名的是谁家的孩子？
*/

```

```

*/
#include<stdio.h>
int main()
{
// a b 为张家, c d 为李家
int a,b,c,d;//a+b=6    c+d=7
for(a=6;a>0;a--)
{
    for(b=6;b>0;b--)
    {
        for(c=7;c>0;c--)
        {
            for(d=7;d>0;d--)
            {
                add(a,b,c,d);
            }
        }
    }
}
}
int add(int a,int b,int c,int d)
{
    if(a+b==6&& c+d==7&&a!=b&&a!=c&&a!=d&&b!=c&&b!=d&&c!=d&&a-b!=1&&a>b&&c-
d!=1&&c>d)//判断, 避免重复, 相连
    {
        if(a==1||b==1)printf("L");
        else if(c==1||d==1)printf("W");
        else printf("Z");
    }
}
}
/*
W
*/

```

```

/*
题目描述: 某人有四张3分的邮票和三张5分的邮票, 用这些邮票中的一张或若干张可以得到多少种不同的邮
资?
*/
#include<stdio.h>
int main(void){
    int count=0;
    for(int i=0;i<=4;i++){
        for (int j=0;j<=3;j++){
            if(3*i+5*j>0){
                count++;
            }
        }
    }
    printf("%d\n",count);
    return 0;
}

```

```

/*
题目描述: 一个正整数如果等于组成它的各位数字的阶乘之和, 该整数称为阶乘和数。
例如, 145=1!+4!+5!, 则145是一个三位阶乘和数。

```

请问:共有多少个阶乘和数? (不会超过十万)

输入:无

输出:所有的阶乘和数(按字典序,即1打头的在前,2打头的次之,..., 空格分隔)

1 145 2 40585

*/

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

```
#include<math.h>
```

```
int jiechen(int n);
```

```
int main()
```

```
{
```

```
    int i,j,k;
```

```
    for(i=1;i<=9;i++)
```

```
        for(j=0;j<=4;j++)
```

```
            for(k=0;k<pow(10,j);k++)// 这个循环用来排序
```

```
            {
```

```
                int s,temp,sum=0;;
```

```
                s=i*pow(10,j)+k;
```

```
                temp=s;
```

```
                while(temp)
```

```
                {
```

```
                    sum+=jiechen(temp%10);// 分别求每一位的阶乘 例如 121    121%10==1
```

```
12%10==2 1%10==1
```

```
                    temp/=10;//减少位数 , 循环求阶乘
```

```
121/10=12
```

```
12/10==1 1/10=0 退出循环
```

```
                }
```

```
                if(sum==s)
```

```
                    printf("%d\n ",sum);
```

```
            }
```

```
    }
```

```
int jiechen(int n)
```

```
{
```

```
    int i,sum=1;
```

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

```
        sum*=i;
```

```
    return sum;
```

```
}
```

/*

题目描述:如果一个正整数等于其各个数字的立方和,则称该数为阿姆斯特朗数(亦称为自恋性数)。

如 $407=4^3+0^3+7^3$ 就是一个阿姆斯特朗数。试编程求大于1小于1000的所有阿姆斯特朗数。

输入:无

输出:从小到大输出,数之间用两个空格分开

*/

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

```
int main(void){
```

```
    for(int i=2;i<1000;i++){
```

```
        int a = i/100;// 百位
```

```
        int b = i/10%10;// 十位
```

```
        int c = i%10;// 个位
```

```
        if (a*a*a+b*b*b+c*c*c==i){
```

```
            printf("%d ",i);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

/*

153 370 371 407 */

第八章 数组

8.1 一维数组的定义和使用

数组是具有相同数据类型的一组数据。

```
# include<stdio.h>
int main(void){
    int a[10]={1,2,3,4,5};
    float b[10]={1.1,2.2,3.3,4.4,5.5};
    char c[256]='c','1';
    for (int i=0;i<10;i++){
        printf("%d\n",a[i]);
    }
    for (int i=0;i<10;i++){
        printf("%f\n",b[i]);
    }
    for(int i=0;i<10;i++){
        printf("%c\n",c[i]);
    }
    return 0;
}
/*
1
2
3
4
5
0
0
0
0
0
1.100000
2.200000
3.300000
4.400000
5.500000
0.000000
0.000000
0.000000
0.000000
0.000000
c
1
*/
```

当在函数中只定义数组时，数组里的值和函数里定义一个变量的值一样，都是未初始化过的，我们也可以定义的时候并初始化赋值，并且，当给部分元素赋初值的时候，未被赋值的元素将自动赋值为0，更细一些，int类型未被赋值的元素为0，浮点型为小数类型，而字符类型则为'\0'。

8.2 二维数组的定义和使用

```
// 定义一个三行四列的二维数组按行进行赋值
int a[3][4]={1,2,3,4},{5,6,7,8},{9,10,11,12}};
// 定义一个三行四列的二维数组
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

8.3 字符数组和字符串

- 字符数组

用来存放字符的数组称为字符数组，字符数组的各个元素依次存放字符串的各个字符。

```
# include<stdio.h>
int main(void){
    char c[6]={'c','h','i','n','a','\0'}; // \0 代表字符串的结束符，如果不加，系统会自动加上
    char a[]={'china'};
    for (int i=0;i<6;i++){
        printf("%c",c[i]);
    }
    for(int i=0;i<6;i++){
        printf("%c",a[i]);
    }
    return 0;
}
```

8.4 总结&练习

```
/*
题目描述:求一个3*3矩阵对角线元素之和
输入:矩阵
输出:主对角线副对角线元素和
*/
// 思路矩阵使用二维数组a[x][y] 来存储，主对角线数据特征 x=y 副对角线数据特征 x+y=2
# include<stdio.h>
int main(){
    int x,y,sum1=0,sum2=0,a[3][3];
    for(x=0;x<3;x++){
        for(y=0;y<3;y++){
            scanf("%d",&a[x][y]);
            if(x==y){
                sum1 +=a[x][y];
            }
            if (x+y==2){
                sum2 +=a[x][y];
            }
        }
    }
    printf("%d %d",sum1,sum2);
    return 0;
}
/*
```

输入:

1 2 3

4 5 6

7 8 9

输出:

15 15

*/

/*

题目描述: 已有一个已正序排好的9个元素的数组, 今输入一个数要求按原来排序的规律将它插入数组中。

输入: 第一行, 原始数列。 第二行, 需要插入的数字。

输出: 排序后的数列

*/

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

```
int main(){
```

```
    int nums[10];
```

```
    for(int x=0;x<9;x++){
```

```
        scanf("%d",&nums[x]);
```

```
    }
```

```
    scanf("%d",&nums[9]);
```

```
    int i, j, temp;
```

```
    //冒泡排序算法: 进行 n-1 轮比较
```

```
    for(i=0; i<10-1; i++){
```

```
        //每一轮比较前 n-1-i 个, 也就是说, 已经排序好的最后 i 个不用比较
```

```
        for(j=0; j<10-1-i; j++){
```

```
            if(nums[j] > nums[j+1]){
```

```
                temp = nums[j];
```

```
                nums[j] = nums[j+1];
```

```
                nums[j+1] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    //输出排序后的数组
```

```
    for(i=0; i<10; i++){
```

```
        printf("%d\n", nums[i]);
```

```
    }
```

```
    //printf("\n");
```

```
    return 0;
```

```
}
```

/*

题目描述: 输入10个数字, 然后逆序输出。

输入: 十个整数

输出: 降序输出

*/

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

```
int main(void){
```

```
    int nums[10];
```

```
    for (int x=0;x<10;x++){
```

```
        scanf("%d",&nums[x]);
```

```
    }
```

```
    for(int i=9;i>=0;i--){
```

```
        printf("%d ",nums[i]);
```



```

    }
    return 0;
}

```

```

/*
题目描述:编程，输入一个10进制正整数，然后输出它所对应的八进制数。
*/
# include<stdio.h>
int main(void){
    int num;
    scanf("%d",&num);
    printf("%o",num);
    return 0;
}

```

```

/*
题目描述:输入一个华氏温度，要求输出摄氏温度。公式为
 $C = 5/9(F-32)$ 
保留两位小数
*/
# include<stdio.h>
int main(void){
    int F;
    scanf("%d",&F);
    printf("%.2f",5.0*(F-32)/9);
    return 0;
}

```

```

/*
有一个函数如下，写一程序，输入x，输出y值。
 $y = x \quad x < 1$ 
 $y = 2x - 1 \quad 1 \leq x < 10$ 
 $y = 3x - 11 \quad x \geq 10$ 
保留两位小数
*/
# include<stdio.h>
int main(void){
    int x;
    scanf("%d",&x);
    float y;
    if(x<1){
        y=x;
    }else if(x>=1 && x<10){
        y = 2*x-1;
    }else{
        y = 3*x-11;
    }
    printf("%.2f",y);
    return 0;
}

```

```

/*
题目描述:编制程序，输入n个整数（n从键盘输入，n>0），输出它们的偶数和。
*/

```

```
# include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int sum=0;
    for(int j=0;j<n;j++){
        if(a[j]%2==0){
            sum +=a[j];
        }else{
            continue;
        }
    }
    printf("%d",sum);
    return 0;
}
```

```
/*
题目描述:sum=2+5+8+11+14+..., 输入正整数n, 求sum的前n项和。
*/
# include<stdio.h>
int main(void){
    int n;
    scanf("%d",&n);
    int sum=0;
    int count =1;
    int a =2;
    while(count<=n){
        sum +=a;
        a = a+3;
        count++;
    }
    printf("%d",sum);
    return 0;
}
```

```
/*
求出10至1000之内能同时被2、3、7整除的数, 并输出。

每行一个。
*/
# include<stdio.h>
int main(void){
    for (int i=10;i<=1000;i++){
        if (i%2==0 && i%3==0 && i%7==0){
            printf("%d\n",i);
        }
    }
}
```

%42 也可以

```
/*
```

从键盘输入任意20个整型数，统计其中的负数个数并求所有正数的平均值。

保留两位小数

```
*/
# include<stdio.h>
int main(void){
    int a[20];
    int count=0,sum=0;
    for (int i=0;i<20;i++){
        scanf("%d",&a[i]);
    }
    for (int i=0;i<20;i++){
        if (a[i]<0){
            count++;
        }else{
            sum +=a[i];
        }
    }
    printf("%d %.2f",count,sum*1.0/(20-count));
    return 0;
}
```

```
/*
输入两个正整数m和n，求其最大公约数和最小公倍数。
*/
# include<stdio.h>
int gcd(int a,int b); //函数声明
int lcm(int a,int b); //函数声明
int main(void){
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d\n%d",gcd(a,b),lcm(a,b));
    return 0;
}
int gcd(int a,int b){
    if (b==0)
        return a;
    return gcd(b,a%b); //递归调用
}
int lcm(int a,int b){
    return (a*b)/gcd(a,b); // 利用 最大公约数*最小公倍数=两数值乘积
}
```

```
/*
输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。
*/
# include<stdio.h>
int main(void){
    int a,c,d,b,e;
    a =c =d =b=0;
    while((e=getchar())!=10) // 10为换行的ASCII码
    {
        if (e>=65 && e<=90 || e>=97 && e<=122){
            a++; // 英文字母，a自增
        }else if (e>=48 && e<=57){
            b++; // 遇到数字字符，b自增
        }
    }
}
```

```

    }else if(e==32){
        c++; //空格 c自增
    }else{
        d++; // 遇到其他字符，自增
    }
}
printf("英文字母%d, 数字%d, 空格%d, 其他%d", a, b, c, d);
return 0;
}

```

```

/*
求1+2!+3!+4!+...+30!。

科学计数法，保留两位小数。
*/
# include<stdio.h>
double fibo(n);
double sum=0;
int main(void){
    for (int i=1;i<=30;i++){
        sum +=fibo(i);
    }
    printf("%.2e\n",sum);
    return 0;
}
double fibo(n){
    if (n==1){
        return 1;
    }else{
        return n*fibo(n-1);
    }
}
}

```

科学计数法的使用 %.2e

第九章 指针

9.1 地址

地址是唯一可以表示某一点的一个编号。在计算机中我们常常使用16进制来表示地址。在32为操作系统下，地址范围 0~4,294,967,295 之间，在C语言中 & 代表取地址符

```

#include<stdio.h>
int main(void){
    int i;
    int a[9]={1,2,3,4,5,6,7,8,9};
    char b[7]='a','b','c','d','e','f','g';
    for (i=0;i<9;i++){
        printf("int Address:0x%x,value:%d\n",&a[i],a[i]);
    }
    printf("\n");
    for (i=0;i<10;i++){
        printf("char Address:0x%x,value:%d\n",&a[i],b[i]);
    }
    printf("int arr address is 0x%x",&a);
    return 0;
}

```

```

}
/*
int Address:0xc1937500,value:1
int Address:0xc1937504,value:2
int Address:0xc1937508,value:3
int Address:0xc193750c,value:4
int Address:0xc1937510,value:5
int Address:0xc1937514,value:6
int Address:0xc1937518,value:7
int Address:0xc193751c,value:8
int Address:0xc1937520,value:9

char Address:0xc1937500,value:97
char Address:0xc1937504,value:98
char Address:0xc1937508,value:99
char Address:0xc193750c,value:100
char Address:0xc1937510,value:101
char Address:0xc1937514,value:102
char Address:0xc1937518,value:103
char Address:0xc193751c,value:0
char Address:0xc1937520,value:-98
char Address:0xc1937524,value:53
int arr address is 0xc1937500
注意数组的地址取值方式 &a,数组地址等于首位元素的地址*
/

```

9.2 指针

9.2.1 指针的定义

地址是逻辑内存上的编号，而指针虽然也表示一个编号，也是一个地址，但是两种性质确不相同。地址代表的是常量，而指针代表的是变量。就好比内存是一把尺子，而指针就是尺子上的游标，可以左右移动，他某一个时刻是指向一个地方的，这就是指针变量。

9.2.2 指针的定义格式

对指针变量定义的一般形式为: 类型说明符 变量名;

其中,这里的与前面的类型说明符共同说明这是一个指针变量,类型说明符表示该指针变量所指向的变量为何种数据类型,变量名即为定义的指针变量名。除此之外, C还提供*运算符获取地址上对应的值。

```

# include<stdio.h>
int main(void){
    int num=2021;
    int *p = &num;
    printf("num Address = 0x%x,num=%d\n",&num,num);
    printf("p = 0x%x,*p=%d\n",p,*p);
    printf("%d\n",&num);
    printf("*p=%d\n",*p);
    printf("&num =0x%x\n",&num);
    printf("*&num=%d\n",&num);// &获取地址值,*获取地址上对应的值
    return 0;
}
/*
num Address = 0xf39c5bf4,num=2021
p = 0xf39c5bf4,*p=2021
2021

```

```

*p=2021
&num =0xf39c5bf4
*&num=2021
*/

```

在64位系统中，指针占8个字节

```

#include<stdio.h>
struct INFO
{
    int a;
    char b;
    double c;
};
int main()
{
    int *p;
    char *p1;
    float *p2;
    double *p3;
    struct INFO *p4;    //struct INFO类型为结构体类型 我们将会在后面的章节中讲解
    void *p5;
    printf("int point size is :%d\n",sizeof(p));
    printf("char point size is :%d\n",sizeof(p1));
    printf("float point size is :%d\n",sizeof(p2));
    printf("double point size is :%d\n",sizeof(p3));
    printf("struct point size is :%d\n",sizeof(p4));
    printf("void point size is :%d\n",sizeof(p5));
    return 0;
}
/*
int point size is :8
char point size is :8
float point size is :8
double point size is :8
struct point size is :8
void point size is :8
*/

```

9.3 数组和指针

数组元素，我们可以使用数组下标进行访问，也可以用指针进行访问。在C语言中规定，数组名代表数组的收地址，也就是数组的地址等于第0号元素的地址。

```

#include<stdio.h>
int main(void){
    int *p;//定义 p 为指向整型变量的指针
    int a[10];// 定义a 为包含10个整数数据的数组
    p = &a[0];// 把a[0]元素的地址赋值给指针变量 p
    p = a;// 等价于 p=&a[0]
    //int *p = a;//等价于 int *p=&a[0] 或 p=a
    printf("p = 0x%x\n",p);
    printf("*p =%d\n",*p);
    return 0;
}
/*

```

```
p = 0x87d18b30
*p = -2016310136
*/
```

访问数组元素的两种方式

```
# include<stdio.h>
int main(void){
    int i;
    int a[10]={1,2,3,4,5,6,7,8,9,0};
    int *p = a;
    for(i=0;i<10;i++){
        printf("p value:%d a value :%d\n",*(p++),*(a+i));
    }
    printf("\n");
    return 0;
}
/*
p value:1 a value :1
p value:2 a value :2
p value:3 a value :3
p value:4 a value :4
p value:5 a value :5
p value:6 a value :6
p value:7 a value :7
p value:8 a value :8
p value:9 a value :9
p value:0 a value :0
*/
```

9.4 字符串与指针

字符串指针与字符串数组的区别

```
# include<stdio.h>
int main(void){
    char *str = "www.baidu.com";
    char string[] = "www.baidu.com";
    string[0]='h';
    printf("string = %s",string);
    //str[1]='h';
    //printf("string = %s",string);
    return 0;
}
/*
string = hww.baidu.com
字符串指针str是个变量，可以改变str 使它指向不同的字符串，但不能改变str 所指向的字符串常量的值，而
string 是一个数组，可以改变数组中保存的内容。
*/
```

第十章 结构体

10.1 结构体的定义和使用

结构体和数组类似，都是由若干元素组成，与数组不同的是，结构体的成员可以是不同类型，可以通过成员名来访问结构体的元素。结构体的定义说明了它的组成成员，以及每个成员的数据类型，定义的一般形式如下：

```
struct 结构类型名
{
    数据类型 成员名1;
    数据类型 成员名2;
    ....
}
```

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

struct _INFO{
    int num;
    char str[256];
};
int main(void){
    struct _INFO A;
    A.num = 2014;
    strcpy(A.str,"www.baidu.com");// strcpy 字符串复制
    printf("This year is %d %s",A.num,A.str);
    return 0;
}

/*
This year is 2014 www.baidu.com
*/
```

10.2 结构体的高级使用

结构体数组，结构体数组是一个数组，其数组的每一个元素都是结构体类型，在实际应用中，经常使用结构体数组来，表示具有相同数据结构的一个群体。如一个班的学生。

```
/* 定义一个结构体数组student,包含3个元素：student[0],student[1],student[2],每个数组元素都具有 struct address 的结构形式*/
# include<stdio.h>
struct address
{
    char name[30];
    char street[40];
    unsigned long tel;
    unsigned long zip;
}student[3]={
{"Zhang","Road NO.1",111111,4444},
{"Wang"," Road NO.2",222222,5555},
{"Li"," Road NO.3",333333,6666}
};
}
```


指向结构体的指针，当一个指针用来指向一个结构体变量时，称之为结构体指针变量。结构体指针变量中的值是所指向的结构体变量的首地址，通过结构指针即可访问该结构体变量。这与数组指针和函数指针的情况相同。结构体指针变量定义的一般形式为

```
struct 结构类型名 *结构指针变量名
```

10.3 共用体的定义和使用

在C语言中，允许几种不同类型的变量存放到同一内存单元，也就是使用覆盖技术，几个变量互相覆盖。这种几个不同的变量共同占有一段内存的结构，被称为共用体类型结构，简称共用体。一般定义形式为

```
union 共用体名{
    数据类型 成员名1;
    数据类型 成员名2;
    ....
}
```

共用体和结构体的区别

1. 变量长度不同

结构体变量所占内存长度是各成员所占内存长度之和，每个成员分别占有其自己的内存单元

共用体变量所占的内存长度等于其最长的成员的长度。

2. 占用空间不同

结构体是同时存在的，并一次占用一段连续的内存空间

共用体则是多个共用成员占用同一个开始的内存地址，同时他们只能存在一个，所以空间大小就是最大的那个所需的内存，如果单从一个共用体来讲，我们不知道里面存的是什么内容，需要根据程序上下文才能确定

3. 分配储存空间不同

结构体是由一系列具有相同类型或不同类型的数据构成的数据集合，简称结构。在C语言中，可以定义结构体类型，将多个相关的变量包装成为一个整体来使用。在结构体中的变量，可以是相同、部分相同，或完全不同的数据类型。

结构体类型的定义只是由用户构造了一个结构体，但定义结构体类型时系统并不为其分配存储空间。

4. 共用体所占的内存要比结构体小，C中的结构体有点像类

共用体的使用

```
/* 只有先定义了共用体变量，才能在后续的过程中引用它，不能直接引用公用体变量，而只能引用共用体变量中的成员*/
# include<stdio.h>
union INFO{
    int a;//不能 int a=1; 在共用体里赋值
    int b;
    int c;
    int d;
};
int main(void){
    union INFO A;
    A.a=1;
    A.b=2;
    A.c=3;
    printf("a:%d\n",A.a);
    printf("b:%d\n",A.b);
    printf("c:%d\n",A.c);
    //printf("d:%d\n",A.d);
```

```

    return 0;
}
/*
a:3
b:3
c:3
*/

```

不能对共用体变量名赋值,也不能企图引用变量名来得到一个值,并且,不能在定义共用体变量时对它进行初始化。

不能把共用体变量作为函数参数,也不能是函数返回共用体变量,但可以使用指向共用体变量的指针。共用体类型可以出现在结构体类型的定义中,也可以定义共用体数组。反之,结构体也可以出现在共用体类型的定义中,数组也可以作为共用体的成员。

10.4 使用 typedef 定义类型

在C语言中,除系统定义的标准类型和用户自定义的结构体,共用体等类型之外,还可以使用类型说明语句,typedef 定义新的类型来代替已有的类型,typedef 语句的一般形式是

`typedef` 已定义的类型 新的类型

```

typedef int INTEGER;// 指定使用INTEGER 代表 int 类型
typedef float REAL;//指定使用REAL 代表float 类型
// 在具有上述typedef 语句的程序中,下列语句等价
int i,j;// 等价INTEGER i,j;
float pi;// 等价与REAL pi

```

typedef 的使用

```

# include<stdio.h>
# include<string.h>
typedef struct _INFO{
    int num;
    char str[256];
}INFO;
int main(void){
    struct _INFO A;
    INFO B;
    A.num = 2014;
    printf("This year is %d %s\n",A.num,A.str);
    printf("This year is %d %s\n",B.num,B.str);
    return 0;
}
/*
This year is 2014
This year is 899383696 ?•
*/

```

可以看到typedef 可以为关键字改名,使改名之后的INFO类型等价与struct_INFO类型,让我们在定义这种结构类型时更方便,省事。

事实上,许多windows开发中的许多我们未见过的数据类型,看起来很难懂,但绝大部分都是通过typedef定义后的基本数据类型,大家可以通过追溯变量的定义来了解。

10.5 总结

```
/*有一字符串，包含n个字符。写一函数，将此字符串中从第m个字符开始的全部字符复制成为另一个字符串。*/
#include<stdio.h>
void mycopy(char a[],char b[],int n,int m)
{
    int i,j=0;
    for(i=m-1;i<n;++i)
        b[j++]=a[i];
}
int main()
{
    char a[100]={0},b[100]={0};
    int n=0,m=0,i;
    scanf("%d",&n);
    getchar();//获取空格或回车，防止下次scanf受影响
    for(i=0;i<n;++i)
        scanf("%c",&a[i]);
    scanf("%d",&m);
    mycopy(a,b,n,m);
    printf("%s\n",b);
    return 0;
}
```

```
#include<stdio.h>
//定义学生结构体数组
struct student
{
    char num [30];    //注意一下这里学号的类型有字母，所以应该字符类型
    char name[30];    //姓名用字符数组的方式存储
    int grade1;        //三科分数，分别定义为整型
    int grade2;
    int grade3;
    int score;        //后面要每个学生的总分,这里定义一下比较方便
};
//输入的函数 需要传入学生的个数和结构体名
void input(int n,struct student stu[n])
{
    for(int i=0;i<n;i++) {
        scanf("%s %s %d %d %d",&stu[i].num,&stu[i].name,&stu[i].grade1,&stu[i].grade2,&stu[i].grade3); //分别输入数据，注意数组不需要&符号。
        stu[i].score=stu[i].grade1+stu[i].grade2+stu[i].grade3; //在输入学生的数据的时候同时求和
    }
}
//求每门课的平均值
void average(int n,struct student stu[n])
{
    int a=0,b=0,c=0; //赋初值
    for(int i=0;i<n;i++) //传入学生个数，通过循环，找出每门课的平均成绩
    {
        a=stu[i].grade1+a; //给每门课求和
        b=stu[i].grade2+b;
        c=stu[i].grade3+c;
    }
}
```

```

    }
    printf("%d %d %d\n",a/n,b/n,c/n);           //输出每门课的平均成绩
}
//求分数最高的那位学生
void max(int n,struct student stu[n])
{
    int m=0;
    int max=stu[0].score;                       //打擂台算法,先指定第一个为最大值,然后后面
    的再来比较,如果比第一个大,就交换
    for(int i=0;i<n;i++)                         //利用循环,遍历数据
    {
        if(max<stu[i].score)
        {
            max=stu[i].score;                   //如果后一个比前一个大,就交换
            m=i;                                //记下现在的下标
        }
    }
    printf("%s %s %d %d
%d",stu[m].num,stu[m].name,stu[m].grade1,stu[m].grade2,stu[m].grade3); //输出分数
    最高那位学生的信息
}
int main()
{
    int n;
    scanf("%d",&n);                             //    输入学生个数
    struct student S[n];                         //定义结构体数组
    input(n,S);                                  //调用函数 输入数据
    average(n,S);                                //求平均值
    max(n,S);                                    //求最大值
    return 0;
}

```

第十一章 文件操作

文件操作的三个步骤

1. 打开文件

使用 `fopen` 函数来实现，这一步主要是建立程序和文件的关系，获取文件在内存中的文件指针

2. 读写文件

分为 `fprint`, `fscanf`, `fwrite`, `fread`, `fputs`, `getss` 等多组函数实现。

3. 关闭文件

使用 `fclose` 函数实现，这一步切断文件指针和文件的关联，避免误操作。如果未关闭文件就对文件进行读写删除等操作，就是出现类似“正在被使用，无法修改”的提示。

`fopen` 函数的使用

写文件 `fprintf` 函数的使用

读文件 `fscanf` 函数的用法

写文件 `fwrite` 函数的用法

读文件 `fread` 函数的用法

第十二章 预处理

C语言中提供的预处理功能有三种：

1. 宏 定义
2. 文件包含
3. 条件编译

`#` 代表是一条预处理命令 `define` 为宏定义命令，宏命令和全局变量的区别

<http://c.biancheng.net/view/347.html>

```
// 宏定义
# include<stdio.h>
# define MAX(a,b) (a>b)?a:b
/* 带参数的宏定义*/
int main(){
    int a=12,b=15;
    printf("max=%d",MAX(a,b));
    return 0;
}
```

```
// 文件包含
# include "stdio.h"
# include <stdio.h>
```

`#include "文件名"` 或者 `#include <文件名>`但是这两种形式是有区别的:使用尖括号表示在包含文件目录中去查找(包含目录是由系统的环境变量进行设置的,一般为系统头文件的默认存放目录,比如 Linux 系统在/usr/include 目录下),而不在源文件的存放目录中查找; 使用双引号则表示首先在当前的源文件目录中查找, 若未找到才到包含目录中去查找。

```
// 条件编译
#ifdef 标识符
程序段 1
#else
程序段 2
#endif
/*
```

它的功能是如果标识符已被 `#define` 命令定义过则对程序段 1 进行编译;否则对程序段 2 进行编译。如果没有程序段 2(为空)

```
*/
```

```
// 其他处理命令
1. #error error-message
强制编译程序停止编译，主要用于程序调试
2. #line
3. #pragma
#pragma 是便宜程序实现是定义的指令,它允许由此向编译程序传入各种指令。
```

12.1 宏定义和函数的区别

 image-20210706113652767

12.2 总结

/*定义一个带参的宏，使两个参数的值互换，并写出程序，输入两个数作为使用宏时的实参。输出已交换后的两个值。*/

```
#include <stdio.h>
#include <math.h>
#define chang(a,b) t=a,a=b,b=t;
int main()
{
    int a,b,t;
    scanf("%d %d",&a,&b);
    chang(a,b);
    printf("%d %d",a,b);
    return 0;
}
```

/*

输入两个整数，求他们相除的余数。用带参的宏来实现，编程序。

输入:a b两个数

输出:a/b的余数

*/

```
# include<stdio.h>
# define mod(a,b) a%b;
int main(void){
    int a,b;
    scanf("%d %d",&a,&b);
    int c = mod(a,b);
    printf("%d",c);
    return 0;
}
```

/*

三角形面积=SQRT(S*(S-a)*(S-b)*(S-c)) 其中S=(a+b+c)/2，a、b、c为三角形的三边。 定义两个带参的宏，一个用来求area， 另一个宏用来求S。 写程序，在程序中用带实参的宏名来求面积area。

*/

```
# include<stdio.h>
# include<math.h>
# define s(a,b,c) (a+b+c)/2;
# define area(s,a,b,c) sqrt(s*(s-a)*(s-b)*(s-c));
int main(void){
    float a,b,c,S,area,n,m;
    scanf("%f%f%f",&a,&b,&c);
    S = (a+b+c)/2;
    area = area(S,a,b,c);
    printf("%.3f",area);
    return 0;
}
```

/*

输入一个年份判断是否是闰年，闰年判断标准能被400整除，或 能被4整除但是不能被100整除。

*/

```
#include<stdio.h>
#define LEAP_YEAR(y) ((y%4==0&& y%100!=0) || (y%400==0))
int main()
{
    int y;
    scanf("%d",&y);
    if(LEAP_YEAR(y)==0){
        printf("N");
    }else{
        printf("L");
    }
    return 0;
}
```

```
/*
分别用函数和带参的宏，从三个数中找出最大的数。
*/
# include<stdio.h>
# define max(a,b,c) (a>b?a:b)>c?(a>b?a:b):c;
int Max(a,b,c);
int main(void){
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    float max_value = max(a,b,c);
    float Max_value = Max(a,b,c);
    printf("%.3f\n%.3f",max_value,Max_value);
    return 0;
}
int Max(a,b,c){
    return (a>b?a:b)>c?(a>b?a:b):c;
}
```

第十三章 位运算

C语言中的位运算，是以数值的二进制位单位进行操作

1. << 左移 向左（即高位）移位，右侧补0
2. >> 右移 向右（即低位）移位，左侧补0
3. ~ 按位取反 如名，即0变1,1变0
4. & 按位与 相对应的两个位都为1则为1，反之为0
5. | 按位或 相对应的两个位至少有一个为1即为1，反之为0
6. ^ 按位异或 相对应的两个位相同为0，相异（不同）为1

```
// 左移 右移
# include<stdio.h>
int main(void){
    int a,b;
    a=13<<2;
    b=25>>3;
    int d=10;
    printf("10<<3=%d\n",d<<3);
    printf("a=%d,b=%d\n",a,b);
    return 0;
}
/*
```

```
10<<3=80
a=52,b=3
*/
```

左移N位的被指是乘以2的N次方，右移N位的本质是除以2的N次方

```
//& 按位与运算符
#include<stdio.h>
int main(void){
    int a;
    a = 3&5;
    printf("a=%d\n",a);
    return 0;
}
```

按位与运算符的作用 1.清零，我们可以对某一个数与0进行按位与运算，由于两个位都为1才为1，因此最终全部位都变为0，起到清零的作用。2.取指定位 如某些存储场景下，“第1~3位表示xxxx”，我们需要取出1~3位，则可以让原数值与数字7进行按位与运算，得到的结果即是原数值的1~3位的值。3.判断奇偶 可以发现，数字的奇偶取决于二进制位的最低一位是1还是0，因此只需要与1按位与运算，判断是1是0即可得知奇偶。

```
// 按位或运算符
# include<stdio.h>
int main(void){
    int a;
    a = 8|7;
    print("a=%d\n", a);
    return 0;
}
```

按位或运算符的作用， 对一个数字的指定位置为1 如“某个数字的第七位”表示开关，原先是0，需要改为1的状态，即可以将这个数字与64按位或，即可得到第七位变为1，其余位的值依旧不变。

```
// 按位异或运算符 不同则为1，相同为0
// 异或操作 交换数字
# include<stdio.h>
int sqap(int *a,int *b)
{
    if (*a!=*b){
        *a = *a^*b;
        *b = *b^*a;
        *a = *a^*b;
    }
    return 0;
}
```

异或运算符的作用 1.指定位数的翻转 如想对某个数字的低4位进行翻转，则可以将这个数字与15（二进制为00001111）进行按位异或运算，既可以将原数字的低四位进行翻转，即高四位不变，低四位0变1,1变0 2.与0异或还是原值 3.交换两个数字


```
// ~取反运算符
# include<stdio.h>
int main(void){
    unsigned int a=1;
    printf("~a=%u\n",~a);
    return 0;
}
/*
~a=4294967294
*/
```