一、常量

不能被修改的

1.1整型常量

1，10，2，3，····　　%d

1.2浮点型常量

3.14····　　%f

1.3字符型常量

'c','a','pw'　　%c 只能输出单个字符

%d 输出字符的ascii码

1.4字符串常量

"pw"····　　%s

1.5宏定义常量

```c
#include <stdio.h>
#define N 3
int main(int argc, const char *argv[])
{
    printf("%d\n",'p');
    printf("%d\n",N);
    printf("%d\n",N);
    return 0;
}
```

二、基本数据类型

基本数据类型，构造数据类型

2.1整形

数据类型	所占空间大小	占位符
int	4Byte = 32bit	%d
long int	32位系统下占4Byte 64位系统下占8Byte	%ld
short int	2Byte	%d
longlong int	8Byte	%ld

2.2浮点型

数据类型	所占空间大小	占位符
float	4Byte 小数点后6位	%f
double	8Byte 小数点后15位	%lf

2.3字符型

数据类型	所占空间大小	占位符
char	1Byte	%c输出字符本身 %d输出字符的ascii码

三、变量

概念：在程序运行中可以被改变的值。

3.1定义变量

格式：

存储类型 数据类型 变量名;

```
int a;  //定义了一个变量a
```

(1) 存储类型：

auto 自动类型 不写默认是auto

static 静态存储类型

register 寄存器类型

extern 从其他文件中找

volatile 从内存中找

(2)基本数据类型

(3)变量名 是一个标识符

1、由字母、数字和下划线构成

2、不能以数字开头

3、不能与关键字同名

32个关键字

```
auto    break   case    char    const
continue    default     do  double  else
enum    extern      float   for     goto
if      int     long    register    return
short   signed      sizeof  static      struct
switch  typedef     union   unsighed    void
```

3.2变量的初始化和赋值

```c
#include <stdio.h>
int main(int argc, const char *argv[])
{
    int a=3; //定义一个变量a初始化值为3
    int b = a;//定义一个变量b，用变量a给b初始化
    int c; //没有进行初始化操作，系统分配随机值
    c = 6;
    //c=a; //用变量a给变量c赋值
    printf("%d\n",c);
    printf("%d\n",b);
    return 0;
}
```

3.3强制类型转换

3.3.1显式的强制类型转换

格式： （强转之后的类型）变量名;

小的数据类型向大的数据类型强转是安全的，

大的数据类型向小的数据类型强转是不安全的（会发生数据的丢失，这里大小指的是数据所占的内存空间）

3.3.2隐式的强制类型转换

```c
#include <stdio.h>
int w()
{
    return 1.23;
}
int main(int argc, const char *argv[])
{
    int a = 65; //定义了一个int类型的变量a
    printf("%f\n",(float)a);//把int类型的变量a强制转换成float类型
    printf("%d\n",(char)a); //把int类型的变量a强制转换成char类型，这时a会被强转成数值对应的ASCII的字符，这里用%d输出，输出的是字符的ASCII码值，所以这里输出还是65。不建议这样强转，因为当整形数据表示的数在ASCII码范围内时，会转换成对应的ASCII码值，当整形数据超出0-127，会发生数据的丢失，因为int类型占4Byte，char类型占1Byte。
    printf("%d\n",(short int)a);//把int类型的变量a强制转换成shaort int类型
    int ret = w();  //接收了一下函数w的返回值，不理解也没关系，后面将函数的时候会讲。
    printf("%d\n",ret); //return 1.23但是输出是1，因为发生了隐式的强制类型转换。
    return 1.23;
}
```

3.4原码、反码和补码

有符号数 signed 默认都是有符号数 **最高位0为正1为负**

无符号数 unsigned

23 原、反、补：00010111

正数 原反补是一样的，

负数，最高位为1，反码是原码除符号位之外其他位取反，补码是反码+1。

-0 原1000 0000

反1111 1111

补10000 0000 高位截取 ---> 0000 0000　　1000 0000　-128　-2^7

有符号数的取值范围：[-128,127]

快速计算出 0111 1111+1 --->1000 0000

练习：-23　　原10010111

反11101000

补11101001

-32　　原10100000

反11011111

1

补1110000

用补码反推1011 0000　　-48

反 1100 1111

原1011 0000

-75　10110101

-96　1010 0000

-34　1101 1110

-123　10000101

-0　　1000 00000