

1. 关键字 `volatile` 用于修饰变量，表示该变量在程序运行过程中可能会在完全不可预知的情况下被修改（如中断等），而为了保证数据的实时性与有效性，处理器在读取该变量的值时必须从该变量的内存地址读取而不是像通常情况下从寄存器中获取该数据的备份。
2. 关键字 `static` 用于修饰本地变量。用 `static` 修饰的本地变量将成为静态本地变量，当程序运行离开函数时，静态本地变量会继续存在并保存其值。由于静态本地变量只会在被定义时初始化，故以后每次进入该函数时静态本地变量仍然会保留上一次离开函数时的值

```
int example[5] = {1,2,3,2,1}
```

```
#define SECONDS_OF_A_YEAR 60*60*24*365
```

5. 就是指将变量的值按照指定格式输出的函数，如 `%d`、`%f` 等等，规定数据（尤其是小数）的输出位数、精度等等性质，一般用 `printf()`、`sprintf()`、`fprintf()` 等等。
6. `&`和变量放在一起是取地址；
a**&**b：（按位与）把a和b都转换成二进制数，再进行与的运算；
a**&&**b：（逻辑与）两个操作数均为 true（非0）时，其结果才为 true；只要有一个为零（false），a**&&**b就为零。
7. `j = i++` 等价于 `j=i;i=i++`；`j = ++i` 等价于 `i=i++;j=i`
8. 以 `char` 类型的数组形式存储，并在末尾添加 `\0`。
9. 形参即形式参数，指的是函数的参数列表中的参数。实参即实际参数，指的是在调用函数时传递给形参的值。
10. 该变量前加上 `static`
11. 存在
12. 指针

```
for(;;)
{
    // code
}
while(1)
{
    // code
}
```

14. 该代码是标准头文件结构，运用条件编译与宏，保证该头文件在一个编译单元中只会被 `#include`（包含）一次从而避免重复声明的冲突。

二、简单应用

```
#define MIN(a,b) ((a)>(b)?(b):(a))
```

```

int WriteOne (int a)
{
    return a|0x10;          //与00010000按位或运算
}

Int WriteZero(int a)
{
    return a&0xef;          //与11101111按位与运算
}

```

3. 计算一个整数在用二进制表示时1的个数

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,b,c,d;
    printf("头的数量=");
    scanf("%d",&a);
    printf("脚的数量=");
    scanf("%d",&b);
    c=(b-2*a)/2;
    d=a-c;
    printf("兔的数量=%d,鸡的数量=%d",c,d);
    system("pause");
    return 0;
}

```

5. 两种修改方式

```

char *getString()
{
    char *str = "Hello world";
    return str;
}

```

```

char *getString()
{
    static char str = "Hello world";
    return str;
}

```

6. 在调用scanf传递值时，`*p`对`&a`进行了解引用，此时`*p`表示为整型常量10，但scanf要求输入的是存储变量的地址。

三、高阶应用

能实现功能就行。