# 2024-03-28

1. 字符串隐含 \0 的结束标志，定义字符串时需加上；strlen函数返回值是字符串 \0 之前的字符个数。
2. 转义字符：\0（\ddd）：表示3位八进制数。 \xdd：表示2位十六进制数。\0：null值，ASCII码为0，相当于数字0。’0’：字符0，ASCII码为48。
3. sizeof关键字或运算符，优先级高于算术运算符，计算类型的长度，表达式不真实计算，根据类型就能计算出所占空间的大小。
4. signed与unsigned：int、char默认有符号，有符合打印%d，无符号打印%u。
5. 浮点数：默认double，float类型需后面加f。
6. 局部变量在栈区，需初始化；全局变量在静态区，默认为0。
7. 负数求模，符号由第一个操作数决定。
8. scanf：除%c外，都会跳过前面的空白字符。%s：直到空白字符结束，末尾加\0字符串结束标识符，一般指定字符串最大长度。%\*：忽略输入时参数之间的间隔符。返回值：成功读取的参数个数，读取错误或文本末尾时返回-1（EOF）。
9. 常量建议放在==的左边。
10. 逻辑运算符：左值满足条件就结束，后续表达式不计算。
11. 数组的类型：把数组名去掉剩余就是数组的类型。
12. 四舍五入：(int)((b\*100)+0.5)/100.0;
13. static修饰局部变量，
14. 栈区分配内存地址，vsX86环境下，按照定义顺序先使用高地址，再使用低地址。
15. 野指针：未初始化，越界访问，指向释放空间。要避免返回局部变量的地址。
16. assert(P != NULL)，为真：程序正常运行。为假：程序中断。#define NDEBUG：不需要做断言时，写在#include <assert.h>之前。
17. 指针比较大小，允许跟数组最大长度的后一个地址比较，不允许跟首地址的前一个地址比较，原因是前一个地址表示的字段可能用来维护数组的长度。
18. 结构体变量初始化：{.成员变量名= }
19. 调试：数组传参，看形参的数组元素， 数组名,10 显示数组元素
20. 小端字节序储存：内存由低到高编址，数据的低位字节存到低地址开始，到高位字节存到高地址结束。大端字节序储存：数据的高位字节存到低地址开始，到低位字节存到高地址结束。
21. 整形提升：32位原码>>>反码>>>补码>>>截取>>>整形提升>>>根据类型+最高位符号位填补
22. 整形提升：对于short与char类型。有符号（变量的类型，char一般默认有符号），前面补符号位；无符号，前面补0
23. 浮点数在内存中的存储：(-1)S\*M\*2E  s(1bit)E(8bit)M(23bit) E：指数+127存储 M：不存1. 只存小数部分，实际上前面这个1放进去了
24. 字符串的值是首字符的地址。
25. sizeof(arr):计算整个数组的大小；&arr+1：跳过整个数组。
26. 数组传参：形参的本质是首元素指针，即便是数组形式。
27. 数组指针：二维数组传参，形参要么是数组指针，要么是二维数组。