x=1, y=1

!x || y-- = 0 || 1 = 1 , y=0

*p++ ： 取 *p. 移动指针

计算机可以进行自动处理的基础：存储程序

进行数值运算的高精度取决于：基本字长

switch 注意有没有 break. 不能使用 continue

char (*p)[5] 指向长度为4的字符数组的指针

char *p[5] 长度为5的字符指针数组

char a[5]; *p = a

赋值. p="abcd"

⇒ *p = 'a'

1. C 的数据类型使编译器能够确定在内存中如何存储一个特定的值，以及对该数值可以执行的运算
2. 变量名必须以字母或是下划线开始，并由字母、数字和下划线组成，用户标识符不能与 保留字 同名。

字符串不是数据类型

x*=y+z (=> x= x*(y+z)

++(n*m) 不合法 ++要求操作数是一个可修改的左值（可存储的变量）

一条预处理命令独占一行，可位于程序任意位置

B. 预处理命令属于一类特殊的 C 语言语句 不属于

C. 优先级高的运算符优先计算 例如 || 等都叫计算运算

D. C 语言的输入和输出功能只能通过函数调用才能实现

- **b (binary)**: 二进制模式。数据按原样存储，不会对换行符等特殊字符做转换。
- **t (text)**: 文本模式（默认）。在某些系统（如 Windows）下，会将换行符 \n 自动转换为 \r\n。

组合示例：

- "rb"：只读、二进制文件。
- "wb+"：读写、二进制文件（不存在则创建，存在则清空）。

4. 总结对比表

为了方便记忆，你可以参考下表：

标识	读	写	追	不存在时	存在时
r	✅			返回 NULL	正常打开
r+	✅	✅		返回 NULL	正常打开
w		✅		创建新文件	清空内容
w+	✅	✅		创建新文件	清空内容
a		✅	✅	创建新文件	在末尾追加
a+	✅	✅	✅	创建新文件	在末尾

```
char *ps="0123456789";

int i=sizeof(ps);  8 指针 8字节
int j=sizeof(*ps); 1 指向首地址'0' sizeof(char)=1
int k=strlen(ps);  10
```

hello ← abc

r+ : abclo

w : abc

w+ : abc

3.以下选项中，当指针 p 为空指针时，其值为真的表达式有 BCD

A. p B. !p C. p==NULL D. p=='\0'

指针若赋值 NULL 则指向 0x0 这个地址

\0 ASCII码为 0

p==0 为真

外部变量必须定义在所有函数之外，且只能定义一次，可以多次声明 ✓

c = 2>1 ? (a=3) : (b=2) 执行后 a=3, c=3

a=1, b=1, !(a+b) = 0.

! > 算术 > 关系 > and > ||

执行后 a 四舍五入保留两位小数（如 a 的值原本是 12.666666，执行完后变为 12.670000）

a = (int)(a*100+0.5)/100.0

若希望一个函数返回多个不同类型的数值，可以将返回值定义为 结构体类型

函数定义时变量类型写全 void f(int a, int b)

定义变量时应赋初值 int n=0;

char *p2 = p1. 将指针 p1 的值赋给 p2，p1、p2 指向相同的地址

2. E8, -28, 2e-8 为合法常量. -028不合法，0开头被视为八进制数

(\x+1~2位，\+1~3位八进制) φ1 → φx16+1=65

\x41 : 十六进制转义 代表一个字符 char c='\x41' c中为合法字符

注意 \后不超过8

sizeof 统计 \0

注意 = / ==

while( )①?

/* 0 */