

## C语言编程常见错误

1. 忘记定义变量：程序中使用的每一个变量都必须定义其类型，否则无法为变量分配存储空间。
2. 数据类型与指定的输入输出格式不符：输入输出的数据的类型与用户指定的输入输出格式声明不一致。
3. 未注意 `int` 和 `short int` 数据的取值范围。
4. 使用函数 `scanf` 时，未使用变量的地址符 `&`。需要指明“向哪个地址所标识的单元送值”。
5. 输入数据的形式与要求不符：`scanf` 函数中格式字符串中，除了格式说明符以外的字符必须原样输入。
6. 在使用 `scanf` 函数向字符数组输入数据时，在数组名前多加了地址符 `&`。数组名本身就是代表数组的首地址。
7. 在使用 `scanf` 函数向数值型数组输入数据时，误用数值型数组名。因为输入的是多个数据，必须分别通过指定的数组元素输入。（`for` 循环 结合 `&a[i]`）
8. 语句后面漏写了分号：不是所有的编程语言的语句末尾都有分号，但是 C 语言的语句末尾必须有分号，因为分号是 C 语句必不可少的一部分。
9. 把预处理指令当做 C 语句，而错误地在末尾加了分号。预处理指令不是 C 语句，在指令后面不应该加分号。
10. 在控制语句（`if`、`for`、`while` 等）中也不要乱加分号。
11. 应该有花括号的复合语句，没有加花括号。
12. 括号不配对：当一个语句中使用了多层括号时应保证括号正确配对。
13. 标识符应区分大小写。
14. 区分关系运算符“`==`”和赋值运算符“`=`”。
15. 对数组的定义或引用数组元素都要使用方括号。
16. 防止数组下标越界，注意数组下标是从 0 开始的。所以最大下标是：`数组长度-1`。
17. 定义和引用二维数组和多维数组时，必须将每一维的数据分别用方括号括起来。
18. 数组名并不代表数组中全部元素，仅代表数组的首元素地址，因此，要把一个数组赋值给另一个数组，需要使用循环将原数组的每一个元素都复制到新数组中去。
19. 区分字符数组和字符指针。字符数组名仅代表数组首地址，而字符指针是指向字符数据的指针变量。
20. 在引用指针之前没有给它赋予确定的值。
21. `switch` 语句的各分支中不应该漏写 `break` 语句。
22. 区分字符和字符串：字符单撇号，字符串双撇号。
23. 使用自增（`++`）和自减（`--`）运算时应避免歧义性，拿不准就加括号。`i++` 是“后自加”，即先取变量 `i` 的值再进行自增。
24. 忘记对所调用的函数进行函数原型声明。（编译出错）
25. 函数声明与函数定义不匹配：函数类型不能漏写且应匹配，参数类型、数目和顺序都要匹配。
26. 在需要加头文件时没有用 `#include` 指令包含相关的头文件。
27. 误认为函数形参值的改变会影响实参的值。（值传递是实参向形参传递数据，单向传递）
28. 函数的实参和形参类型不一致。（编译警告，运行结果不符合预期）
29. 混用了不同类型的指针。（编译警告，指向不同类型的指针间的赋值必须进行类型转换，比如给一个 `struct` 变量分配存储单元时）
30. 应注意系统对函数参数的求值顺序的处理方法。函数求值的顺序到底是从左到右还是从右往左，与编译环境有关，C 标准并没有规定。多数情况下从左到右求解和从右到左求解结果是相同的。但是，仍旧会有会引起二义性的可能。因此应该保证程序在不同的编译环境下具有相同的运行结果，以

使程序更通用。

31. 区分数组名和指针变量。
32. 区分结构类型和结构变量，只能对一个结构变量赋值，而不能对结构类型（派生的数据类型）赋值，类型不占存储单元，需要分配存储单元的是变量。
33. 使用文件时忘记打开，或打开方式与使用情况不匹配。
34. 在打开文件时，指定的文件名找不到。
35. 忘记关闭文件：虽然系统会自动关闭所用文件，但是可能导致数据丢失。因此，用完文件后必须关闭。