

Week4

a.为什么过度使用 `if` 语句会减慢程序的速度？

`if` 语句的执行过程包括对条件进行评估，然后根据评估结果进行分支 (branch)。分支意味着程序需要跳转到内存中的不同部分来执行相应的指令。这种在内存中来回跳转的操作会带来一些开销 (overhead)。因此，在编写需要高速运行的代码时，程序员有时会特意避免使用分支和 `if` 语句，因为执行比较和分支操作会减慢程序的速度。

b.在汇编语言中 `jne` 是什么意思？

在汇编语言中，`jne` 的意思是“jump not equal”（如果不相等则跳转）。

c.编译时和运行时有何区别？

编译时 (compile time) 指的是编译器处理源代码的阶段。在这个阶段，编译器可以进行优化，例如，如果它知道一个比较的结果（比如比较6和5是否相等），它可以在编译时就完成这个计算，这个过程被称为常量折叠 (constant folding)。运行时 (run time) 指的是编译好的程序实际执行的阶段。如果在运行时进行条件检查，会比在编译时就确定结果花费额外的时间。

d.比较表达式的返回类型是什么？

比较表达式的返回类型是布尔类型 (Boolean)。换句话说，它是一个 `true` 或 `false` 的语句。这个结果可以被存储在一个 `bool` 类型的变量中。

e. `else if` 是 C++ 中的一个关键字组合吗？

不，`else if` 并不是 C++ 中的一个关键字组合。C++ 中没有

`else if` 这个关键字，只有 `else` 关键字和 `if` 关键字。

`else if` 实际上是两个独立的语句：一个 `else` 语句，后面紧跟着一个 `if` 语句。

f.在C++ 中 `nullptr` 的数值是什么？

`nullptr`（或 `null`）的数值是0。在代码中，检查一个指针是否为 `null`，可以检查它是否等于0或 `nullptr`。

g.如果你把 `for` 循环的条件部分留空会发生什么？

如果你把 `for` 循环的条件部分留空，这和在条件部分写 `true` 是一样的。这将导致一个永不 `false` 的条件，从而形成一个无限循环 (endless loop)。

h. `while` 循环和 `do-while` 循环之间有什么区别？

`while` 循环和 `do-while` 循环唯一的区别在于，`do-while` 循环的循环体无论如何都会被至少执行一次。

`while` 循环会先检查条件，如果条件一开始就是 `false`，循环体内的代码就永远不会执行。而

`do-while` 循环会先执行一次循环体内的代码，然后再检查条件决定是否继续循环。

i. `continue` 和 `break` 的使用有什么区别？

`continue` 和 `break` 的主要区别如下：

- `continue`：只能在循环内部使用，它的作用是结束当前这一次的循环迭代，并立即开始下一次迭代。当程序执行到 `continue` 时，会跳过当前迭代中余下的代码，直接回到循环的开始处进行下一次迭代的判断和执行。
- `break`：主要用于循环（也用于 `switch` 语句），它的作用是完全跳出并结束整个循环。当程序执行到 `break` 时，循环会立即终止，程序将跳转到循环结构之后的第一行代码继续执行。