# 计算机的秘密

### 1.1 什么内联函数不能实现复杂的逻辑？

内联函数的调用实际上会将函数体copy到调用的地方，减少函数调用的开销，从而提高效率。那为什么内联函数不能实现太复杂的内容呢？如果内联函数实现太复杂的内容，则意味着该函数体会被copy到项目的多个地方，整体来看会增加代码量，反而会增加内存的开销，也会降低执行的效率。所以不建议在内联函数中实现复杂的逻辑。

### 1.2 什么是编译器？解释器？

编译器和解释器都是将高级语言转化为计算机可以理解和执行的机器语言的工具，但它们的工作方式有所不同。

编译器是一种将源代码翻译为可执行文件的程序。编译器将源代码作为输入，并将其转换为机器语言，这种转换是一次性的。编译器在将代码转换为可执行文件之前，会对其进行词法分析、语法分析、语义分析、中间代码生成、目标代码生成等多个阶段的处理。由于编译器将源代码转换为可执行文件，因此这种方式的代码运行速度通常很快。

解释器是一种将源代码逐行翻译为计算机可执行的机器语言的程序。解释器不需要将源代码全部翻译为机器语言，它会一行一行地读取源代码，并且每次执行一行代码。解释器会将代码转换为计算机可以理解的中间代码，然后执行这些中间代码。这种方式的代码运行速度通常比编译器慢。

总的来说，编译器将整个程序编译为可执行文件，解释器则是逐行解释程序。编译器可以提高程序的运行速度，但需要额外的编译时间，而解释器不需要额外的编译时间，但程序的运行速度通常比编译器慢。

### 1.3 c++中引用C文件，为什么需要使用extern？(可以从编译过程进行回答)

在C++中引用C文件时，需要使用 extern 关键字，是因为 C++ 和 C 在函数调用、变量的存储方式和链接方式等方面的实现有所不同。

C语言中，函数和变量都默认是使用 extern 进行声明和定义的，这意味着它们是在全局作用域中定义的，并且可以在程序的任何位置进行访问和使用。但是，在 C++ 中，函数和变量默认情况下是使用 static 进行声明和定义的，这意味着它们只能在当前文件中使用。

因此，在 C++ 中引用 C 文件中的函数和变量时，需要使用 extern "C" 来告诉编译器这些函数和变量是 C 样式的，并且使用 C 语言的命名约定，以便正确地链接到 C 文件中定义的函数和变量。如果不使用 extern "C"，则 C++ 编译器将使用 C++ 样式的名称重载机制来解析函数名和变量名，这会导致链接错误。