C语言中的4种隐式类型转换：

1、算术运算式中，低类型能够转换为高类型

2、赋值表达式中，右边表达式的值自动隐式转换为左边变量的类型，并赋值给它

3、函数调用中参数传递时，系统隐式地将实参转换为形参的类型后，赋给形参

4、函数有返回值时，系统将隐式地将返回表达式类型转换为返回值类型，赋给调用函数

显式语句

if (B) S1 else S2

if B goto B.true

goto B.false

B.true: (S1的代码)

。。。

goto S.next

B.false: (S2的代码)

。。。

S.next

for (i=1;i<=N;i++) S

i:1;

again: if i<=N

{

(S的代码)

。。。

i：i+1；

goto again；

}

While (B) S

Loop: if B goto B.true

Goto S.next

B.true: (S的代码)

。。。

goto Loop；

S.next;

Do S while (S)

Loop S

if B is true

goto Loop；

else end

提高执行效率的措施

1、循环嵌套中将较长循环设为内存循环，较短循环设为外置循环，以减少 cpu 跨切循环层的次数，提高程序 的运行效率。

2、让用户可以直接操作进程虚拟地址，给用户提供了高效的内存拷贝函数，变量类型是弱类型，可以各种强 制转换。

3、API，可以调用系统 API，接近底层

4、嵌入汇编语言，更贴近极限

5、使用寄存器变量，提高存取速度。

6、使用条件编译，可减少被编译的语句，从而减少目标程序的长度，减少运行时间，当条件编译段比较多时， 目标程序长度可大大减少，这样可以提升程序执行效率(谭浩强原话)。

7、位操作:位操作可以减少除法和取模的运算。理论上可以用位运算来完成所有的运算和操作。一般的位操 作是用来控制硬件或做数据变换使用。灵活的位操作可以有效地提高程序运行的效率。

数组越界产生什么后果

数组越界，将会把数据存放到一个未知的区域，而这个未知的区域如果恰好是系统很重要的位置，可能了就会 修改系统程序导致系统出错甚至是崩溃，如果这个位置不是和系统有关，可能结果并不明显或者没有产生影响， 但为了系统安全需要进行数组下标越界检查。

结构体和联合体的区别

联合体是构造数据类型，用几个不同类型的变量共占一段内存，所占内存长度是各最长成员占的内存长度。 结构体是构造数据类型，把不同类型的数据组合成一个整体，所占内存长度是各成员占的内存长度的总和。

指针在函数中的应用和特点

指针作为函数参数: 函数之间参数的传递要对数据进行拷贝，而利用指针作为函数参数传递数据的本质，就是在主调函数和被调函 数中，通过指向同一内存地址的不同指针(函数传值时拷贝了一个指针)访问相同的内存区域，从而实现数据 的传递和交换。指针作为函数的返回值(指针型函数): 不能把一个指向局部变量的指针作为返回值，因为函数内部声明的局部变量在函数结束后其生命周期已经结 束，内存会自动释放，这时它的内存地址无意义。如果将其作为函数返回值返回给主调函数并在主调函数中访 问这个指针所指向的数据，将产生不可预料的后果。

什么是数据类型和存储类型，存储类型有哪几种，以及它们的生成时、初始化、作用域。

存储类型是数据在内存中的存储方式，有四种:auto，static，extern，register

auto和 register 用于声明内部变量；

auto 变量存储在栈中；

register 变量存储在寄存器中;

extern 用于声明外部变量；

static 用于声明内部和外部变量；

extern 和 static 变量是存储在静态存储区中；

内部变量默认是 auto 类型， 外部变量默认是 extern 类型。

C 语言中，常量存储在哪儿?static 全局变量和 static 局部变量存储在哪儿?

常量存储在常量区，该区数据可读不可写。static 全局变量和 static 局部变量都存储在全局数据区。

C 语言，除了关键字，还有哪些单词类型(好像是这么问的，百度了一下应该是问的 C 的语法符号)?C 的存 储类型关键字有哪四个?

单词类型:标识符，运算符、分隔符、常量、注释符存储类型关键字:关键字:auto、extern、register、static

数据类型的框架图

数据类型:

基本类型:

整型

字符型

浮点型:单精度浮点型，双精度浮点型

构造类型: 数组类型，结构类型，联合类型，枚举类型

指针类型

空类型

定义布尔型的方法

typedef enum{0,1}bool;

include <stdbool.h>