C语言

1. C语言有多少关键字?

32个，注意sizeof是关键字，不是函数。

1. C程序的一般结构那 些？

顺序，分支，循环

1. 简单列举几种常用的循环结构和分支结构。

循环：For,while do,if goto(慎用)。分支：if else, switch case(注意default 的使用，和break 的添加)。

1. while 和dowhile 有何区别？

前者先判断再执行，后者先执行再判断；后者比前者多执行一次。

1. continue 和 break 有何需别？

前者结束本次循环继续下次循环，后者直接推出循环。

1. register 有何作用?

声明一个变量常驻寄存器，以达到提高程序运行效率的目的，在嵌入式开发中尤其常用，但它是一个建议性的声明，编译器会根据程序情况决定是否申请成功。

1. static 有何作用？

两方面：1）限定存储域；2）限定作用域。

1. volatile关键字有什么用途？

声明为一个异变的变量，一般的变量会暂存在处理器的缓存区中，当内存中变量发生改变时处理器可能无法及时更新其变量的值导致灾难，在嵌入式开发中尤其重要，也可达到防止编译器优化的目的。

1. 指针函数和函数指针有和区别？

概念不同，前者是一个函数返回值为指针类型；后者为一个指针，指向一个函数入口地址。

1. 简述0,'0',"0",'\0'的区别。

分别为数值0，字符0（ascii码数值为48），字符串0（占两个字节，有‘\0’作为结束符），字符串结束标志（其数值等于0是绝对的0，但意义不同）

1. 使用gets时应注意哪些事项？

防止越界，gets不检查目的地址的空间大小。

1. 简述堆和栈的区别：

前者为程序动态分配，创建和销毁都需要手动完成（使用完一定要销毁，防止内存泄露）；后者为系统自动分配，在函数返回时自动销毁（所以不能在函数中返回变量）。

1. 函数传参有哪些方法？各有何特色？

值传递和地址传递，前者在传递时函数形参接受实参的拷贝，所以原形参不会被改变（但是大数据的拷贝会降低效率）；后者传递变量的地址，不需拷贝，但是会改变参数的值。

1. 全局变量和局部变量有何区别？

前者在静态区，作用域是整个程序；后者在栈区，作用域仅限于函数内部。

1. 局部变量能否和全局变量重名？

可以，局部变量会屏蔽全局变量。

1. 静态全局变量和普通全局变量有何区别？

存储域方面：都相同，都在静态区；作用域方面：不同，前者是本文件，后者是全局可见。

1. 静态局部变量和普通局部变量有何区别？

存储域：不相同，一个在静态去，一个在栈区；作用域：相同，都在函数内部；

但是声明周期不同，前者声明周期为整个程序的声明周期知道程序推出自动消亡；后者声明周期为一次函数调用，函数返回即销毁。

18.地址对齐方式有哪几种？

字节对齐，半字对齐，字对齐

19.

Struct a{

Char a;

Int b;

Short c;

}

Sizeof(a) = ?

结构体a的大小是4\*3=12个字节，而不是1+2+4=7个字节；这里主要考察内存对齐（地址对齐）的特点，应为结构体中最大数据宽度是整型（4字节），所以按字对齐。

思考一下如何改良结构使得存储空间变小。

20. 简述程序的局部变量存在于（栈区）中，全局变量存在于（静态区 ）中，动态申请数据存在于（ 堆区）中。

21.结构体和联合体的区别？

22. 宏函数与内联函数和普通函数的执行效率的比：

宏函数效率高，因为是编译器在预处理阶段的替换；

23.C语言中注释有哪些方法？

/\*\*/和 #if 0 #endif 。后者是预处理命令，可以方便的对代码区进行选择编译。

解释一下语句的含义：

Const int a;

int const a;

const int \*a;

int \* const a;

int const \*const a ;

前两个的作用是一样，a是一个常整型数。第三个意味着a是一个指向常整型数的指针（也就是，整型数是不可修改的，但指针可以）。第四个意思a是一个指向整型数的常指针（也就是说，指针指向的整型数是可以修改的，但指针是不可修改的）。最后一个意味着a是一个指向常整型数的常指针（也就是说，指针指向的整型数是不可修改的，同时指针也是不可修改的）。如果应试者能正确回答这些问题，那么他就给我留下了一个好印象。顺带提一句，也许你可能会问，即使不用关键字const，也还是能很容易写出功能正确的程序，那么我为什么还要如此看重关键字const呢？我也如下的几下理由：

1). 关键字const的作用是为给读你代码的人传达非常有用的信息，实际上，声明一个参数为常量是为了告诉了用户这个参数的应用目的。如果你曾花很多时间清理其它人留下的垃圾，你就会很快学会感谢这点多余的信息。（当然，懂得用const的程序员很少会留下的垃圾让别人来清理的。）

2). 通过给优化器一些附加的信息，使用关键字const也许能产生更紧凑的代码。

3). 合理地使用关键字const可以使编译器很自然地保护那些不希望被改变的参数，防止其被无意的代码修改。简而言之，这样可以减少bug的出现。

24.

int \* a=null；

\*a = 2；

上式有错吗，错到哪里？（指针没有初始化）

25、头文件中的 ifndef/define/endif 干什么用？（5分）

答：防止该头文件被重复引用。

26、#include <filename.h> 和 #include “filename.h” 有什么区别？（5分）

答：对于#include <filename.h> ，编译器从标准库路径开始搜索 filename.h

对于#include “filename.h” ，编译器从用户的工作路径开始搜索 filename.h