**C试题讲解**

**问答题部分：（面试中问到的内容）**

1. 局部变量能否和全局变量重名？ （华为）

解析：能，局部会屏蔽全局。要用全局变量，需要使用"::"

局部变量可以与全局变量同名，在函数内引用这个变量时，会用到同名的局部变量，

而不会用到全局变量。对于有些编译器而言，在同一个函数内可以定义多个同名的局部变量.

比如在两个循环体内都定义一个同名的局部变量，而那个局部变量的作用域就在那个循环体内。

1. 如何引用一个已经定义过的全局变量？ （华为）

解析：extern

可以用引用头文件的方式，也可以用extern关键字，如果用引用头文件方式来引用某个在头文件中

声明的全局变理，假定你将那个编写错了，那么在编译期间会报错，如果你用extern方式引用时，

假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错。

1. static全局变量与普通的全局变量有什么区别？

解析：1) 全局变量(外部变量)的说明之前再冠以static 就构成了静态的全局变量。

全局变量本身就是静态存储方式， 静态全局变量当然也是静态存储方式

从以上分析可以看出，

static全局变量与普通的全局变量有什么区别：

static全局变量只初使化一次，防止在其他文件单元中被引用;

static局部变量和普通局部变量有什么区别：

static局部变量只被初始化一次，下一次依据上一次结果值； static函数与普通函数有什么区别：

static函数在内存中只有一份，普通函数在每个被调用中维持一份拷贝

1. 进程和线程的差别（微软亚洲）

解析：答:线程是指进程内的一个执行单元,也是进程内的可调度实体. 与进程的区别:

(1)调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位

(2)并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可并发执行

(3)拥有资源：进程是拥有资源的独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源.

(4)系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销

1. 用宏定义写出swap（x，y）（思科）

解析：#define swap(x, y) x = x + y; y = x - y; x = x - y

1. 什么是预编译，何时需要预编译：(思科)

解析：就是指程序执行前的一些预处理工作,主要指#表示的. 何时需要预编译?

１)、总是使用不经常改动的大型代码体。

２)、程序由多个模块组成，所有模块都使用一组标准的包含文件和相同的编译选项。

在这种情况下，可以将所有包含文件预编译为一个预编译头。

1. 下述三个有什么区别？char \* const p;char const \* p，const char \*p; （慧通）

解:const char\* p和char const \*p修饰的是p所指向的变量，

不可以用\*p=’a’等这样的方式修改变量区的内容. 指针本身是可以修改的，比如说

char x,y; const \*p = &x, p = &y，修改指针本身是可以的,\*p = ‘b’就不行。

Char \*const p;中const修饰的是指针，表明指针在操作过程中不可以修改，

而指针所指向的内容是可以修改的，比如说char x ,y; char\* const p = &x; \*p = ‘b’是可以的，

但p=&y;是不可以的，const修饰的是指针，不可以修改指针的指向，

只可以修改指针所捆绑的变量

1. int (\*s[10])(int) 表示的是什么啊 ？ （慧通）

解析:这样的表达式可以遵循这样的规则:从右向左，由近及远，括号优先；

比如:从a符号开始。其右边是[10]，说明a是个数组，其中存了十个元素。  
 再看a的左边是一个\*。说明数组中存的是指针。  
 现在在看(\*a[10])的右边是(int);说明所存的指针是指向有一个int形参的函数  
 现在看(\*a[10])的左边是个int ，说明指向的函数的返回值为int 类型；

1. 进程之间通信的途径，进程间通信方法（慧通）

解析：无名管道，命名管道，消息队列，信号，共享内存，套接字。等

1. 什么是“引用”？申明和使用“引用”要注意哪些问题？

解析：引用就是某个目标变量的“别名”(alias)，对应用的操作与对变量直接操作效果完全相同。申明一个引用的时候，切记要对其进行初始化。引用声明完毕后，相当于目标变量名有两个名称，即该目标原名称和引用名，不能再把该引用名作为其他变量名的别名。声明一个引用，不是新定义了一个变量，它只表示该引用名是目标变量名的一个别名，它本身不是一种数据类型，因此引用本身不占存储单元，系统也不给引用分配存储单元。不能建立数组的引用

12.include <filename.h>和include “filename.h”的区别？

解析：<>是从标准库路径搜索，

“”是从用户当前工作目录开始，找不到，在到标准库开始

13.Heap和Stack的区别？

解析：Heap是堆，Stack是栈。

栈的空间由操作系统自动分配和回收，而堆上的空间由程序员申请和释放。

栈的空间大小较小，而堆的空间较大。

栈的地址空间往低地址方向生长，而堆向高地址方向生长。

栈的存取效率更高。

程序在编译期间对变量和函数的内存分配都在栈上，

且程序运行过程中对函数调用中参数的内存分配也是在栈上。

14.sizeof和strlen之间的区别？

解析：（1）sizeof操作符的结果类型是size\_t，它在头文件中的typedef为

unsigned int类型，该类型保证能容纳实现所建立的最大对象的字节大小。

（2）sizeof是运算符，strlen是函数

（3）sizeof可以用类型做参数，strlen只能用char \*做参数，

且必须是以’\0’结尾的。

（4）数组做sizeof的参数不退化，传递给strlen就退化为指针。

（5）大部分编译程序在编译的时候就把sizeof计算过了

是类型或是变量的长度。

（6）strlen的结果要在运行的时候才能计算出来，用来计算字符串的长度，

而不是类型占用内存的大小。

（7）sizeof后如果是类型必须加括号，如果是变量名可以不加括号。

（8）当使用了一个结构类型或变量时，sizeof返回实际的大小。

（9）数组作为参数传递给函数时传的是指针而不是数组，

传递的是数组的首地址。

（10）计算结构变量的大小就必须讨论数组对齐问题。

（11）sizeof操作符不能用于函数类型，不完全类型或位字段。

15.#define DOUBLE(x) x+x ，i = 5\*DOUBLE(5)； i 是多少?  
 解析：i 为30。

16.unsigned short hash(unsigned short key)   {   return (key>>)%256   }

   解析 请问hash(16),hash(256)的值分别是:   A.1.16;B.8.32;C.4.16;D.1.32

17.bool double int float.各种类型与0值的比较(概率较大)

解析：请写出 BOOL flag 与“零值”比较的 if 语句。if ( flag ) 或者    if ( !flag )

如下写法均属不良风格，不得分。     if (flag == TRUE)

  if (flag == 1 )          if (flag == FALSE)          if (flag == 0)

请写出 float  x 与“零值”比较的 if 语句。（4分） 标准答案示例：

const float EPSINON = 0.00001;

if ((x >= - EPSINON) && (x <= EPSINON)

不可将浮点变量用“==”或“！=”与数字比较，

应该设法转化成“>=”或“<=”此类形式

如下是错误的写法，不得分。     if (x == 0.0)        if (x != 0.0)

请写出 char \*p 与“零值”比较的 if 语句。（3分） 标准答案：

 if (p == NULL) 或者    if (p != NULL)

如下写法均属不良风格，不得分。     if (p == 0)      if (p != 0)          if (p)          if (!)

18.结构与联合有和区别？

解析：

1. 结构和联合都是由多个不同的数据类型成员组成， 但在任何同一时刻， 联合中只存放了一个被选中

的成员（所有成员共用一块地址空间）， 而结构的所有成员都存在（不同成员的存放地址不同）。

1. 对于联合的不同成员赋值， 将会对其它成员重写，  原来成员的值就不存在了，

而对于结构的不同成员赋值是互不影响的

19.简述数组与指针的区别？

    解析：数组要么在静态存储区被创建（如全局数组），要么在栈上被创建。

指针可以随时指向任意类型的内存块。

（1）修改内容上的差别

char a[] = “hello”; a[0] = ‘X’;

char \*p = “world”; // 注意p 指向常量字符串

  p[0] = ‘X’; // 编译器不能发现该错误，运行时错误

     （2） 用运算符sizeof 可以计算出数组的容量（字节数）。

sizeof（p），p 为指针得到的是一个指针变量的字节数，而不是p 所指的内存容量。

C++/C 语言没有办法知道指针所指的内存容量，除非在申请内存时记住它。

注意当数组作为函数的参数进行传递时，该数组自动退化为同类型的指针。

char a[] = "hello world"; char \*p = a; cout<< sizeof(a) << endl; // 12 字节

cout<< sizeof(p) << endl; // 4 字节     计算数组和指针的内存容量

void Func(char a[100]) { cout<< sizeof(a) << endl; // 4 字节不是100 字节 }

20.如何判断一段程序是由C 编译程序还是由C++编译程序编译的？   

解析：#ifdef \_\_cplusplus cout<<"c++"; #else cout<<"c"; #endif

21.main 主函数执行完毕后，是否可能会再执行一段代码，给出说明？

解析：可以，可以用\_onexit 注册一个函数，它会在main 之后执行

int fn1（void）， fn2（void）， fn3（void）， fn4 （void）； void main( void ) {

String str("zhanglin"); \_onexit( fn1 ); \_onexit( fn2 ); \_onexit( fn3 ); \_onexit( fn4 );

printf( "This is executed first.\n" ); }

int fn1() { printf( "next.\n" );return 0; }

int fn2(){printf( "executed " ); return 0; }

int fn3() { printf( "is " ); return 0; }

int fn4() { printf( "This " ); return 0; }

22.int id[sizeof(unsigned long)]; 这个对吗？为什么??

解析:对,这个 sizeof是编译时运算符，编译时就确定了可以看成和机器有关的常量。

23.如何打印出当前源文件的文件名以及源文件的当前行号？

解析： cout << \_\_FILE\_\_ ;  
 cout<<\_\_LINE\_\_ ;  
 \_\_FILE\_\_和\_\_LINE\_\_是系统预定义宏，这种宏并不是在某个文件中定义的

而是由编译器定义的

24.以下函数能求出数组的长度吗？void fun(char str[]){ int len = sizeof(str)/sizeof(str[0]);}

解析：不能，数组作为参数传递给函数时，数组名被退化为指针，

因此函数中的sizeof(str)实际是在求一个指针的sizeof，答案为4，

因此不能计算出数组的长度。

25. 输出的结果是多少，并分析过程？

unsigned short A = 10;

printf(“%u\n”, ~A);

char ch = 128;

printf(“%d\n”, ch);

答：~A=4294967285，首先将A转化为int类型，即对应的二进制数值为：

00000000 00000000 00000000 00001010，

~A=11111111 11111111 11111111 11110101，

其实这种情况最高位是1，认为是负数，但是在输出中指定以无符号数输出，

于是结果为4294967285=4294967295（四字节表示的最大数）-10.

ch = 128对应的二进制为：10000000，在输出中以整数形式输出，由于最高位是1，

于是就是负数，10000000是该负数的补码，根据求补码的反步骤计算，先-1，

得到01111111，在取反得10000000=128，由于本身是负数，即为-128.

26. New delete 与malloc free 的联系与区别?  
 解析：都是在堆(heap)上进行动态的内存操作。用malloc函数需要指定内存分配的字

节数并且不能初始化对象，new 会自动调用对象的构造函数。delete 会调用对象的

destructor，而free 不会调用对象的destructor.

27.用变量a给出下面的定义 （特别重要）

a) 一个整型数（An integer） b) 一个指向整型数的指针（A pointer to an integer）

c) 一个指向指针的的指针，它指向的指针是指向一个整型数（A pointer to a pointer to an integer）

d) 一个有10个整型数的数组（An array of 10 integers）

e) 一个有10个指针的数组，该指针是指向一个整型数的（An array of 10 pointers to integers）

f) 一个指向有10个整型数数组的指针（A pointer to an array of 10 integers）

g) 一个指向函数的指针，该函数有一个整型参数并返回一个整型数

解析：a) int a; // An integer

b) int \*a; // A pointer to an integer

c) int \*\*a; // A pointer to a pointer to an integer d) int a[10]; // An array of 10 integers

e) int \*a[10]; // An array of 10 pointers to integers

f) int (\*a)[10]; // A pointer to an array of 10 integers

g) int (\*a)(int); // A pointer to a function a that takes an integer argument and returns an integer

h) int (\*a[10])(int);

28.在不用第三方参数的情况下，交换两个参数的值  
 解析：方法1:#include <stdio.h> void main(){  
 int i=60; int j=50; i=i+j; j=i-j; i=i-j;  
 printf("i=%d/n",i); printf("j=%d/n",j);}

方法二：  
 i^=j;  
 j^=i;  
 i^=j;、

29.i最后等于多少? int i = 1; int j = i++; if((i>j++) && (i++ == j)) i+=j;

解析：:i = 5

30.线程同步的机制有哪些？他们有什么优缺点（可以说笔试必考）

解析：  
 1.Critical Section  
 A.速度快  
 B.不能用于不同进程  
 C.不能进行资源统计(每次只可以有一个线程对共享资源进行存取)

2.Mutex  
 A.速度慢  
 B.可用于不同进程  
 C.不能进行资源统计

3.Semaphore  
 A.速度慢  
 B.可用于不同进程  
 C.可进行资源统计(可以让一个或超过一个线程对共享资源进行存取)

4.Event  
 A.速度慢  
 B.可用于不同进程  
 C.可进行资源统计

31.指针和引用的区别？

解析：引用是在C++中引入的。它们之间的区别有：

（1） 非空区别：指针可以为空，而引用不能为空

（2）可修改区别：如果指针不是常指针，那么就可以修改指向，而引用不能

（3） 初始化区别：指针定义时可以不初始化,而引用在定义的时必须初始化

32.全局变量可不可以定义在可被多个.C文件包含的头文件中 为什么

  解析：可以，在不同的C文件中以static形式来声明同名全局变量。

   可以在不同的C文件中声明同名的全局变量，

前提是其中只能有一个C文件中对此变量赋初值，此时连接不会出错

33.结构struct和类class有什么异同？

解析：在c语言中struct只能对数据进行聚合，而c++的class把数据以及对数据的处理方法

也同时聚合为一体，增加了内聚性。 此外class拥有可再生性和可抽象性，实现的代码的复用。

集中体现在派生的功能和多态的功能。 同时class也比struct具备更好的封装性，

体现在三种访问权限上。

在C++ 中的struct和class的结构基本一致，只是struct的默认权限为Public而class为private。

34. 逻辑思维题：101个硬币100真、1假，真假区别在于重量。请用无砝码天平称两次给出真币重还是假币重的结论。（貌似是华为的题）

解析:101个先取出2堆,33,33  
 第一次称,如果不相等,说明有一堆重或轻那么把重的那堆拿下来,再放另外35个中的33  
 如果相等,说明假的重,如果不相等,新放上去的还是重的话,说明假的轻(不可能新放上去的轻)

第一次称,如果相等的话，这66个肯定都是真的,从这66个中取出35个来,

与剩下的没称过的35个比下面就不用说了

35.char str1[]=”abc”; char str2[] = “abc”; str1==str2正确吗：

**解析：错误，**因为str1和str2是位于堆栈上的，它们占用不同的内存空间。

Const char str3[] = “abc”; const char str4[] = “abc”；str3==str4为FALSE，

同样它们是位于堆栈上的内存空间，是不同的。

Const char \*str5=”abc”,

const char \*str6=”abc”;

char \*str7=”abc”,

char \*str8 = “abc”,

str5==str6

str7==str8为TRUE，因为”abc”是位于文字常量区的，

系统将几个“abc”进行优化，使它们位于同一块内存区，因此指针的指向也就相同了。

36.内联函数和宏的差别？

答：内联函数和普通函数相比可以加快程序运行的速度，因为不需要中断调用，在编译的时候内联函数可以直接被镶嵌到目标代码中，而宏只是一个简单的替换。内联函数要做参数类型检查，这是与宏相比的优势。

Inline是指嵌入代码，就是在调用函数的地方不是跳转，而是把代码直接写到那里去。对于短小的代码来说，inline可以带来一定效率的提升，而且和C时代的宏函数相比，inline更安全可靠。可是这是以增加空间消耗为代价的。

Inline一般只适用于：一个函数被不断地重复调用；函数只有简单的几行，且函数内不能含有for while switch语句。

37.delete与 delete []区别

delete只会调用一次析构函数，而delete[]会调用每一个成员的析构函数。在More Effective  C++中有更为详细的解释：“当delete操作符用于数组时，它为每个数组元素调用析构函数，然后调用operatordelete来释放内存。”delete与New配套，delete []与new []配套

38.求下面函数的返回值（微软）

int func(x)

{

     int countx = 0;

while(x){

       countx ++;

       x = x&(x-1);

  }

return countx;

}

假定x = 9999。

解析：返回值是8思路：将x转化为2进制，看含有的1的个数

39. 下面程序的结果是多少？

P1=(unsigned char \*)0x801000;

p2=(unsigned long \*)0x810000;

请问p1+5=  ;

p2+5=  ;

解析:0x801005(相当于加上5位) 0x810014(相当于加上20位);

1. 下面的代码输出是什么，为什么？

void foo(void)  
 {  
    unsigned int a = 6;  
    int b = -20;  
    (a+b > 6) ? puts("> 6") : puts("<= 6");  
 }  
解析：这个问题测试你是否懂得C语言中的整数自动转换原则，我发现有些开发者懂得极少这些东西。不管如何，这无符号整型问题的答案是输出是 ">6"。原因是当表达式中存在有符号类型和无符号类型时所有的操作数都自动转换为无符号类型。因此-20变成了一个非常大的正整数，所以该表达式 计算出的结果大于6。这一点对于应当频繁用到无符号数据类型的嵌入式系统来说是丰常重要的。如果你答错了这个问题，你也就到了得不到这份工作的边缘。

41 TCP/IP 建立连接的过程?(通信公司考得比较多)  
 答：在TCP/IP协议中，TCP协议提供可靠的连接服务，采用三次握手建立一个连接。  
　　第一次握手：建立连接时，客户端发送syn包(syn=j)到服务器，

并进入SYN\_SEND状 态，等待服务器确认；  
第二次握手：服务器收到syn包必须确认客户的SYN（ack=j+1）同时自己也发送一个  
 SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN\_RECV状态；  
第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=k+1)  
 此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

42.求下面函数的返回值（微软）

int func(x)

{

int countx = 0;

while(x)

{

countx ++;

x = x&(x-1);

}

return countx;

}

假定x = 9999。 答案：8

解析：将x转化为2进制，看含有的1的个数。

知道了这是统计9999的二进制数值中有多少个1的函数，且有9999＝9×1024＋512＋256＋15

9×1024中含有1的个数为2；

512中含有1的个数为1；

256中含有1的个数为1；

15中含有1的个数为4；

故共有1的个数为8，结果为8。

1000 - 1 = 0111，正好是原数取反。这就是原理。

用这种方法来求1的个数是很效率很高的。

不必去一个一个地移位。循环次数最少。

43. C语言同意一些令人震惊的结构,下面的结构是合法的吗，如果是它做些什么？

int a = 5, b = 7, c; c = a+++b;

解析：这个问题将做为这个测验的一个愉快的结尾。不管你相不相信，上面的例子是完全合乎语法的。

根据最处理原则，编译器应当能处理尽可能所有合法的用法。因此，上面的代码被处理成：

c = a++ + b;

因此, 这段代码持行后a = 6, b = 7, c = 12。

如果你不知道答案，我也不把这个当作问题。我发现这个问题的最大好处是:

这是一个关于代码编写风格，代码的可读性，代码的可修改性的好的话题

44请写出下面代码在 32 位平台上的运行结果，并说明 sizeof 的性质：

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

char a[30];

char \*b = (char \*)malloc(20 \* sizeof(char));

printf("%d\n", sizeof(a));//1

printf("%d\n", sizeof(b));//2

printf("%d\n", sizeof(a[3]));//3

printf("%d\n", sizeof(b+3));//4

printf("%d\n", sizeof(\*(b+4)));//5

return 0 ;

}

45. 在C++ 程序中调用被 C 编译器编译后的函数，为什么要加 extern “C”声明?

解析：函数和变量被C++编译后在符号库中的名字与C语言的不同，被extern "C"修饰的变量和

函数是按照C语言方式编译和连接的。由于编译后的名字不同，C++程序不能直接调用C 函数。

C++提供了一个C 连接交换指定符号extern“C”来解决这个问题。

46.下面函数的运行结果是什么？

main()

{

int a[5]={1,2,3,4,5};

int \*ptr=(int \*)(&a+1);

printf("%d,%d",\*(a+1),\*(ptr-1));

}

解析：2,5

\*(a+1)就是a[1]，\*(ptr-1)就是a[4], 执行结果是2, 5。&a+1不是首地址+1，系统会认为加一个a数组

的偏移，是偏移了一个数组的大小(本例是5个int)。int \*ptr=(int \*)(&a+1); 则ptr实际是&(a[5]),也就

是a+5

47、设有以下说明和定义：

typedef union {long i; int k[5]; char c;} DATE;

struct data { int cat; DATE cow; double dog;} too;

DATE max;

则语句 printf("%d",sizeof(too)+sizeof(max));的执行结果是：\_\_\_\_\_\_

解析：DATE是一个union, 变量公用空间. 里面最大的变量类型是int[5], 占用20个字节.

所以它的大小是20，data是一个struct, 每个变量分开占用空间.

依次为int4 + DATE20 + double8 = 32.所以结果是 20 + 32 = 52.

48.下面的代码有什么作用？

#ifndef \_\_INCvxWorksh

#define \_\_INCvxWorksh

#ifdef \_\_cplusplus

extern "C" {#endif /\* \*/ #ifdef \_\_cplusplus }

#endif

#endif /\* \_\_INCvxWorksh \*/

解析：头文件中的编译宏

#ifndef　\_\_INCvxWorksh

#define　\_\_INCvxWorksh

#endif 的作用是防止被重复引用。

1. 把11011011000101转换成10进制，8进制和16进制

解析8进制的结果是33305 10进制的结果是14021 16进制的结果是36C5

50:下列代码运行后得到的结果是多少(耐为数控)?

#define SQLARE(a) ((a)\*(a))

Int a = 5;

Int b;

B = SQLARE(a++);

解析：代码预编译后的 b = (a++)\*(a++); 后++理解为遇到分好才++,结果为25

51：下列程序运行后a的值是多少? 为什么?

Unsigned int a = 100;

Unsigned int i;

For(i=100;i >=0.;i--)

a++;

解析： 由于i是无符号型的int类型,值永远不可能小于0,所以循环i>=0永远成立.

52：下列程序的运行结果是什么? 为什么?

void foo(void)

{

Unsigned int a = 6;

Int b = -20;

(a+b>6)?puts(“>6”):puts(“<=6”);

}

解析：由于a是无符号型的,有符号的和无符号的相加会都变成无符号的.结果是>6

53：要求设置一绝对地址为0x67a9的整形变量的值为0xaa66..编译器是一个纯粹的asc编辑器,

解析: Int \* p; p =(int\*) 0x67a9; \*p = 0xaa66;

54:逻辑思维题：有一种小虫，每隔2秒分裂一次，分裂后的2只小虫经过2秒后又会分裂一次，如果最初命中只有一只小虫，那么2秒后变2只，再经过2秒后变成4只，。。。两分钟后正好满满一瓶小虫，假设这个瓶中最初放两只小虫请问，经过多少时间后正巧也是满满的一瓶？为什么？

解析：其实很简单，就是前面两秒的时候本来从一个分裂两个要两秒，

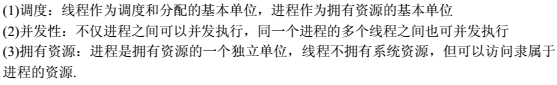
而第二种情况直接两个等于前面的两秒省略了，所以时间是一份五十八秒

2.0次方=1（这个是第一个起点）  
 2.1次方=2（这个是第二个起点）  
 2.3次方=4  
 第二次比第一个少分裂一次.  
 同样2个小虫就是分裂1分58秒.

1. If(n==8)和if(8 == n) 的区别，哪一个好？为什么(大唐)

解析：第一种是把把常量放比较符后面，后者相反，第二种好处是if(8=n)编译器会报错

56 线程和进程的区别：





57：测试分为哪几类？

解析： 针对系统功能的测试叫做黑盒测试 测试函数功能函数的接口等叫做白盒测试

58;sizeof和strlen的区别：

解析：1.sizeof的结果类型是size\_t，它在头文件中typedef为unsigned int类型。  
 该类型保证能容纳实现所建立的最大对象的字节大小。   
 2.sizeof是算符，strlen是函数。   
 3.sizeof可以用类型做参数，strlen只能用char\*做参数， 。  
 4.数组做sizeof的参数不退化，传递给strlen就退化为指针了。   
 5在编译的时候就把sizeof计算过了 是类型或是变量的长度这就是0;  
 6.strlen的结果要在运行的时候才能计算出来，不是类型占内存的大小。   
 7.sizeof后如果是类型必须加括弧，如果是变量名可以不加括弧。

8.当适用了于一个结构类型时或变量， sizeof 返回实际的大小，

1. const和#define都是代表不可改变的，他们有什么不一样呢？

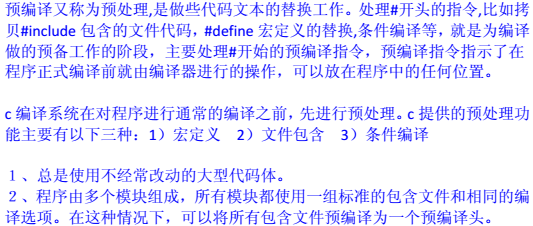
解析：① const 常量有数据类型，而宏常量没有数据类型。编译器可以对const修饰的数据进行类型安全检查。而对宏常量只进行字符替换，没有类型安全检查，并且在字符替换时可能会产生意料不到的错误。

 ② 有些集成化的调试工具可以对const 常量进行调试，但是不能对宏常量进行调试。

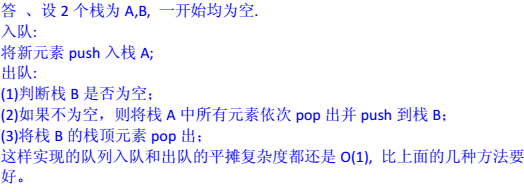
1. 堆栈溢出一般是什么原因导致的？

解析：没有回收垃圾资源，比如:malloc的内存没有回收，递归调用层次太深，没有退出处理

61什么是预编译，什么时候需要预编译。



1. 用两个栈实现一个队列的功能，怎实现，说出具体的思路

解析：

63在桌面上有一堆硬币,其中有十三枚正面向上,其他的硬币都是反面向下,硬币大小形状都一样,你闭上眼睛(无其他人,工具帮助),用手将硬币分成两堆,使两堆正面朝上的数目相等.请问你有什么办法?(华为)

解析：方法一:把硬币分成两堆，把两堆都竖起来，没有正面朝上的 哈哈哈。。还有其他办法。。

64 小明一家过一座桥，过桥时是黑夜，所以必须有灯。现在小明过桥要1秒， 小明的弟弟要3秒，小明的爸爸要6秒，小明的妈妈要8秒，小明的爷爷要12秒。每次此桥最多可过两人，而过桥的速度依过桥最慢者而定，而且灯在点燃后30秒就会 熄灭。问：小明一家如何过桥？(中科院)

解析:第一步,小明与弟弟过桥,小明回来,耗时４秒；第二步,小明与爸爸过河,弟弟回来,

耗时９秒；第三步,妈妈与爷爷过河,小明回来,耗时１３秒；最后,小明与弟弟过河,

耗时４秒,总共耗时３０秒,多么惊险!

65(北京赞同科技) 现有1~100共一百个自然数，已随机放入一个有98个元素的数组a[98]。要求写出一个尽量简单的方案，找出没有被放入数组的那2个数，并在屏幕上打印这2个数。注意：程序不用实现自然数随机放入数组的过程。

 解析： //a[98]中存放着原数组；  
 int i=0;  
 unsigned char b[100];  
 for(i=0;i<100;i++)  
  b[i]=0;  
  
 //依次填入b[100]，b[0]=1,b[1]=2,b[2]=3,.......  
 for(i=0;i<98;i++)  
  b[a[i]-1]=a[i];

//没有被填入数字的应该还是0，b[n]=0  
 for(i=0;i<100;i++)  
  if(0==b[i]) printf("num:%d \n",(i+1));

1. 简单说明TCP与UDP的区别

解析：1。TCP是基于连接的，UDP是基于无连接

2。对系统资源的要求（TCP较多，UDP少）

3。UDP程序结构较简单

4。流模式与数据报模式

5。TCP保证数据正确性，UDP可能丢包，TCP保证数据顺序，UDP不保证

67. 键字volatile有什么含意?并给出三个不同的例子。

解析：一个定义为volatile的变量是说这变量可能会被意想不到地改变，这样，编译器就不会去假设这个变量的值了。精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。下面是volatile变量的几个例子：

1) 并行设备的硬件寄存器（如：状态寄存器）

2) 一个中断服务子程序中会访问到的非自动变量

3) 多线程应用中被几个任务共享的变量

补充问题：

1. 一个参数既可以是const还可以是volatile吗？解释为什么。

可以。

一个例子是只读的状态寄存器。它是volatile因为它可能被意想不到地改变。它是const因为程序不应该试图去修改它。

1. 一个指针可以是volatile 吗？解释为什么。

可以。

尽管这并不很常见。一个例子是当一个中服务子程序修该一个指向一个buffer的指针时。

68. C和C++有什么不同?

解析：① 从机制上：c是面向过程的（但c也可以编写面向对象的程序）；c++是面向对象的，提供了类。但是，c++编写面向对象的程序比c容易

② 从适用的方向：c适合要求代码体积小的，效率高的场合，如嵌入式；c++适合更上层的，复杂的；linux核心大部分是c写的，因为它是系统软件，效率要求极高。

③ C语言是结构化编程语言，C++是面向对象编程语言。C++侧重于对象而不是过程，侧重于类的设计而不是逻辑的设计。

69 main 主函数执行完毕后，是否可能会再执行一段代码，给出说明？

解析：可以，可以用\_onexit 注册一个函数，它会在main 之后执行int fn1(void),

Int main( void ){

String str("zhanglin");

\_onexit( fn1 );

printf( "This is executed first.\n" );

}

int fn1(){printf( "next.\n" );return 0;}

70 101个硬币100真、1假，真假区别在于重量。请用无砝码天平称两次给出真币重还是假币重的结论。

解析： 第一步：把硬币分为三堆儿（50，50，1），先拿两个50 放天平比较，

如果相等，说明单个的就是假的，再随便拿一个真硬币和这个假硬币放天平比较一下就可得到结果。

第二步：如果第一步的两个50 比较后不相等，此时一端50重，一端50轻。

如果拿轻的一端50，分成两堆（25，25） 放到天平上，如果这两个25相等，

则假硬币（重）在重的一端50里，否 则（不相等）假硬币（轻）在这两个（25，25）里，

也就是轻的一端50里。

1. 变量的声明和定义有什么区别？

解析：  
 2. 一个变量的定义会为变量分配存储空间，也有可能同时为其提供初始值；  
 3. 声明则仅给出变量的数据类型和变量名称；  
 4. 变量的定义肯定也是变量的声明，反之则未必；  
 5. 在作用域内，变量的定义有且仅有一个，声明则可以有多个；  
 6. 有extern的时候，两者的情况区分：

72下列代码:char \* GetStr(){char \*tmp;tmp = "123"return tmp;}

void main(){printf("%s", GetStr());}

会输出123吗？123创建在堆上还是栈上呢？123的空间是什么时候释放的？

解析：可以输出123，首先123是一个字符串常量，存储在只读常量区，

在函数调用完成之后不会被释放，他的生命周期是程序开始到程序结束，

他既不是在堆上也不是在栈上。

73 关键字volatile有什么含意 并给出三个不同的例子。  
 解析：一个定义为volatile的变量是说这变量可能会被意想不到地改变，这样，

编译器就不会去假设这个变量的值了。精确地说就是，优化器在用到这个变量时必

须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。

下面是volatile变量的几个例子：   
 1). 并行设备的硬件寄存器（如：状态寄存器）   
 2). 一个中断服务子程序中会访问到的非自动变量(Non-automatic variables)   
 3). 多线程应用中被几个任务共享的变量

74 C语言同意一些令人震惊的结构,下面的结构是合法的吗，如果是它做些什么？   
 int a = 5, b = 7, c;

c = a+++b;  
 解析:上面的例子是完全合乎语法的。问题是编译器如何处理它？

水平不高的编译作者实际上会争论这个问题，根据最处理原则，

编译器应当能处理尽可能所有合法的用法。因此，上面的代码被处理成：   
 c = a++ + b;   
 因此, 这段代码持行后a = 6, b = 7, c = 12。

75.在8086 汇编下，逻辑地址和物理地址是怎样转换的？（Intel）

答案：通用寄存器给出的地址，是段内偏移地址，相应段寄存器地址\*10H+通用寄存器内地址，就得到了真正要访问的地址。

76 Itearator和指针的区别是什么？

解析：游标和指针，游标是指针，但不仅仅是指针。游标和指针很像，功能很像指针，但是实际上，游标是通过重载一元的”\*”和”->”来从容器中间接地返回一个值。将这些值存储在容器中并不是一个好主意，因为每当一个新值添加到容器中或者有一个值从容器中删除，这些值就会失效。在某种程度上，游标可以看作是句柄（handle）。通常情况下游标（iterator）的类型可以有所变化，这样容器也会有几种不同方式的转变：  
iterator——对于除了vector以外的其他任何容器，你可以通过这种游标在一次操作中在容器中朝向前的方向走一步。这意味着对于这种游标你只能使用“++”操作符。而不能使用“--”或“+=”操作符。而对于vector这一种容器，你可以使用“+=”、“—”、“++”、“-=”中的任何一种操作符和“<”、“<=”、“>”、“>=”、“==”、“!=”等比较运算符。

77参数传递有几种方式；实现多态参数传递采用什么方式，

解析：传值，传指针或者引用

78操作系统中进程调度策略有哪几种？

　　 解析：这个是操作系统的知识,FCFS(先来先服务)，优先级，时间片轮转，多级反馈

79.short i = 0; i = i + 1L；这两句有错吗

解析:代码一是错的，代码二是正确的。 在数据安全的情况下大类型的数据向小类型的数据转换一定要显示的强制类型转换。

80 .IP地址的编码分为哪俩部分？

解析:IP地址由两部分组成，网络号和主机号。不过是要和“子网掩码”按位与上之后才能区分哪些是网络位哪些是主机位。

81system（"pause");作用?

答:系统的暂停程序，按任意键继续，屏幕会打印，"按任意键继续。。。。。"

省去了使用getchar（）；

82知道一个有序二叉树的中序遍历和前序遍历，求后续遍历的值(待出题)

83

**选择题填空题部分：**

1:在一个带有附加表头单链表HL中,若向表头插入一个由指针p所指向的节点，则执行(南方城墙)( )

A HL = p; p->next=HL; B: p->next = HL->next; HL->next = p;

C:p->next = HL; p=HL; D:P->next = HL; HL=p;

2:(赞同科技)2<<3的结果是多少\_\_\_\_

解析：这个是左移 10向左移动三位 结果是10000

3:sizeof的运算结果在哪个时期确定的\_\_\_\_\_\_\_

解析： 编译时，C/C++是静态强类型语言，所有数据的尺寸在编译时确定

4:(赞同科技)已知int a = 0x122则a/2的值是\_\_\_\_\_\_

解析：先把a转换成二进制，然后再转换成10进制，然后再除以得到的结果是十进制145

5:(赞同科技)无符号二进制11001000&&10001的结果是多少？

解析：首先&&是逻辑与，真&&真=真，所以得到的答案是真，结果是1

6:下列结构体的sizeof结果是多少？

Struct data{char a; double b; char c;}；sizeof（struct data）= \_\_\_\_\_\_\_\_\_

解析：这个考察的是结构体的对齐补齐，首先按照double对其，所以结果是8+8+8=24

7: 要求交换x和y的值，且通过正确调用返回交换之后的结果，则以下符合要求的函数是：

A: void guns(int \*x,int \*y){int \*p; \*p = \*x; \*x = \*y;\*y = \*p;}

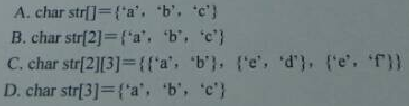
B:void funb(int x,int y){int t; t = x; x = y; y = t;}

C: void fund(int \*x,int \*y){int p; p = \*x; \*x = \*y; \*y = p;}

D:void fund(int \*x ,int \*y){\*x= \*y; \*y = \*x;}

解析：要交换两个变量，肯定不可以使用直接传参的方法，那就使用指针，选择c

8：（亚信）下列数组初始化正确的是()



解析：考察的是数组，选择c，因为c中a[2][3]是两行列，需要{{‘a’,’b’,’c’},}这种方式进行初始化

9:(亚信)设 int a = 10; b = 11;c = 12;表达式(a+b)<c&&b==c的值是（）

A 0 B -2 C 1 D 2  
 解析：选B，本题考查的是运算符的优先级，应该先计算a+b的值为21，

然后与c比较大小的值为 0，根据“&&”的运算规则，整个表达式的值为0。

10(亚信)执行下列程序后，a的值为()

Void main(){int a,b; for(a =1,b=1;a<=100;a++){if(b >=20)break; if(b%3==1){b+=3； continue;}b -=5;}}

解析：1、首先注意for循环的控制条件，当b>=20或者a>100则跳出for循环，也即b<20且a<=100时执行for循环，明白这点很重要。  
2、其次，第一次进入循环，a=1,b=1均满足循环条件，但b%3==1条件满足，故执行b=b+3操作，得到b=4,注意有continue，所以后面语句不执行，直接跳转到a++这个语句，所以第一次循环完之后：a=2,b=4  
3、再次，进入第二轮循环，b%3==1也是满足的，故再次b=b+3，此轮同上轮一样，执行之后:a=3,b=7,进入下一轮。  
4、此后，和2、3两步同理，都是b%3==1满足，因为每次都是加3，而之后又去模3，且都跳过for后面语句直接执行a++，所以，一直循环到b=22跳出for循环。此时a为8.  
  
综合起来就是，每次循环b增加3，a增加1，且当b>22时跳出循环，结束程序。所以b从1增加到22，有（22-1）/3=7，所以a=1+7=8.

11：(网鸿科技)假设堆栈的容量是3，入站的顺序是1 2 3 4 5，则出栈的循序不可能是()

A 4 5 3 2 1 b 4 5 2 3 1 C2 1 5 3 4 D2 1 3 5 4

解析：A B 首先容器最大为3 所以第一次不可能出栈4

12：32位系统中，int (\*a)[4][5]的大小是多少()

A 4 b 14 C 20 D 80

解析：这题考察的是数组指针，这里定义的是一个指针，所以大小是4

13：设定义int x;float y; 则10+x+y的值数据类型是多少()

A int B double C float D 不确定

解析：类型提升，int和float相加后类型会提升到float 所以选择c

14 IP地址的编码分为哪俩部分？

解析：IP地址由两部分组成，网络号和主机号。

不过是要和“子网掩码”按位与上之后才能区分哪些是网络位哪些是主机位。

15：什么是平衡二叉树(华为)\_\_\_\_\_\_

解析：左右子树都是平衡二叉树 且左右子树的深度差值的绝对值不大于1

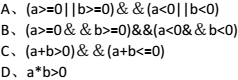
16 在main函数之前会执行哪些代码\_\_\_\_\_\_\_

解析：全局变量的初始化，宏定义等预处理

17: Internet采用哪种网络协议？该协议的主要层次结构

解析：tcp/ip 应用层/传输层/网络层/数据链路层/物理层

18 能正确表示a和b同事为正或同时为负数的逻辑表达式是()



解析：这个主要是运算符的优先级问题，选择D，赋值运算符<逻辑<关系运算符<算数运算符

1. 七个人七小时挖七米井，请问50个人挖50米时需要多少人？

解析：是7人  
 总人数（7）\*总时间（7）\*每小时一人挖的米数（设为x）=总共挖的米数（7）  
 解得未知数每人每小时挖1/7米  
 同理  
 总人数（设为y）\*总时间（50）\*每小时一人挖的米数（1/7）=总共挖的米数（50）  
 解得总人数为7人

20.C++语言的跳转语句中，对于break和continue说法正确的是（）

* 1. break语句只应用与循环体中
  2. continue语句只应用与循环体中
  3. break是无条件跳转语句，continue不是
  4. break和continue的跳转范围不够明确，容易产生问题

解析：break也用于条件语句，switch中，所以选择B

21 下列变量的计算结果是多少 int expr = 20 / 8 \* 8;

解析：16 优先级先算20/8 得到的结果是2,因为是整除,然后2\*8得到16

int expr = 4 << 2 + 1;

解析：32 考的也是优先级问题，+的优先级大于<<操作符，所以4<<3得到的结果是32

22 .设int a=1，b=2，则(a++)+b；这个表达式的值为（ ）

A.2 B.3 C.4 D.5

解析：答案是4，本来a++的优先级小于+，但加上小括号之后，提升了他的优先级，先算a++

23下列不属于对称加密算法的有（ ）

1. MD5 B.DES C.3DES D.SHA

解析：了解各种加密算法，答案选择D，不保证正确

24.下列代码int id[sizeof(unsigned long)]; 这个对吗？为什么??

解析：对

这个 sizeof是编译时运算符，编译时就确定了

可以看成和机器有关的常量。

25 用预处理指令#define 声明一个常数，用以表明1年中有多少秒（忽略闰年问题）

解析：#define SECONDS\_PER\_YEAR (60 \* 60 \* 24 \* 365)UL

如果把后面的UL忽略就没什么难题了

26 如下代码：

int main (void) {

int a = 10, b = 9, c;

c = (--a == b++) ? --a : b++;

printf ("%d, %d, %d", a, b, c);

return 0;}的输出为

A.9，11，11

B.9，11，10

C.8，11，8

D.8，10，8

解析：正确答案:D

27.以下不能对二维数组a进行正确初始化的语句是

A.int a[2][3] = {0};

B.int a[][3] = {{1,2},{0}};

C.int a[2][3] = {{1,2},{3,4},{5,6}};

D.int a[][3] = {1,2,3,4,5,6}

正确答案:C

28 若要用fopen函数打开一个新的二进制文件，该文件要既能读也能写，则文件打开方式字符串应是

A."ab+" B."wb+" C."rb+" D."ab"

解析：正确答案:B 需要打开新的 则需要w表示创建功能

29请问下面程序的运行结果

#include "stdio.h"

#define SQR(X) X\*X

int main(int argc, char\* argv[]){

int a = 10;int k = 2;int m = 1;

a /= SQR(k+m)/SQR(k+m);

printf("%d\n",a);

return 0；}

解析：答案：1

define 只是定义而已，在编择时只是简单代换X\*X而已，并不经过算术法则的

a /= k+m\*k+m/k+m\*k+m;

=>a /= (k+m)\*1\*(k+m);

=>a = a/9;

=>a = 1;

30i最后等于多少?

int i = 1;

int j = i++;

if((i>j++) && (i++ == j)) i+=j;

解析:i = 5 //主要考短路运算

31试编写函数判断计算机是大端存储还是小端存储（大端是低地址存高位，小端是低地址存地位）

bool IsBigendian(){

unsigned short usData = 0x1122;

unsigned char \*pucData = (unsigned char\*)&usData;

return (\*pucData == 0x22);}

32输出程序结果

Int main(){ char \*p1=“name”;

char \*p2;

p2=(char\*)malloc(20);

memset (p2, 0, 20);

while(\*p2++ = \*p1++);

printf(“%s\n”,p2);}

答案：:empty string.

33已知int a[3][2]={3,2,1}; 则表达式“a[0][0]/a[0][1]/a[0][2]”的值是（ B ）

* 1. 0.166667

B 1

C 0

错误的表达式。

解析：相除是整除，所以最后结果是1

34字符串：”[\\0211\”xab](\\\\0211\\xab)” 的长度为( B )

A 1 B 9 C 0 D 11

解析：如果把第一个字符\去掉的话，得到的结果是0，字符串的长度是0，如果说第一个是

转义字符的话\\代表一个\ 所以最后的结果是9

35 int \*\*a[3][4] 这个数组占用多大的空间?

A. 64 B. 12 C. 48 D. 128

解析：首先这是一个指针数组 指针放的是一个二级指针，指针大小是4

所以3\*4\*4 = 48

36请写出这样一个函数指针，它指向一种这样的函数（该函数参数为int，int，并返回值为一个指针），返回的这个指针指向的是另外一个函数（参数类型为int，返回值为int类型的函数）。

解析： 这个是一个函数指针返回函数的例子 int ( \*( \* F )( int, int ) ) (int)

37 写出判断ABCD四个表达式的是否正确, 若正确, 写出经过表达式中 a的值()

int a = 4;

(A)a += (a++); (B) a += (++a) ;(C) (a++) += a;(D) (++a) += (a++); a = ?

解析：C错误，左侧不是一个

38 int i=10, j=10, k=3; k\*=i+j; k最后的值是？

答：60，此题考察优先级，实际写成： k\*=(i+j);，赋值运算符优先级最低

39 智力题，偶尔考到.-1,2,7,28,,126 请问28 和126 中间那个数是什么？为什么？

第一题的答案应该是4^3-1=63

规律是n^3-1(当n 为偶数0，2，4)

n^3+1(当n 为奇数1，3，5)

答案：63

40下列代码执行完后val的值是多少？  
 unsigned long val;  
 char a=0x96;  
 char b=0x81;  
 val= b<<8 | a;  
 问val=\_\_\_？  
 0x8196或0xffffff96。  
 解析：取决于编译器把char默认为无符号还是有符号。  
 字节作移位或算术一般定义为unsigned char

**程序题部分：（笔试最后几个题）**

1.如何判断一个单链表是有环的？（注意不能用标志位，最多只能用两个额外指针）

解析： int  isLoop(List   L){     
        if (!L) return

-  1;     
      List s = L.next;     
      while   (s && s!=L){     
        s= s.next;     
      }       
      if(! s)

return   - 1   ;     
      else  reutrn  1   ;     
    }       
  -----------     
    1 int   isLoop(List   l){     
    2         if(!l)   return   0;     
    3         p=l.next;     
    4         wihle(p!=l&&p!=null)   {     
    5                 l.next=l;     
    6                 l=p;p=p.next;     
    7         }     
    8         if(p=l)   return   1;     
    9         return   0;   }     
  实际上，在我的面试过程中，还问到了不破坏结构的其他算法。     
  我的答案是从链表头开始遍历，如果节点next指针指向自身，则循环存在；否则将next指针指向自身，遍历下一个节点。直至next指针为空，此时链表无循环。  

2. 写程序，将一个字符串倒序？

答：直接在main函数中实现的

void main()

{

char \*source = "hello";

char \*des;

int len = strlen(source);

des = (char \*)malloc(len+1); //申请空间必须是len+1，加1是为了放结束符

if (!des) { exit(1); }

char \*s = &source[len-1];

char \*d = des;

while (len--!=0) {

\*d++ = \*s--;

}

\*d = '\0'; //必须要

cout<<source<<endl;

cout<<des<<endl;

}

3.编写strlen函数，编写strcpy函数

解析：int Strlen(const char \*str){

int len = 0;

assert(str != NULL);

while (\*str++ != '\0') {

len++;

}

return len;

}

非空判断是必须进行的操作，可以使用断言的方式assert(str) != NULL才会继续

char \*StrCpy(char \*strDes, const char \*strSrc){

assert((strDes != NULL) && (strSrc != NULL));

char \*address = strDes;

while ((\*strDes++ = \*strSrc++) != '\0') ;

return address;

}

首先必须判断两个指针是否为空，由于复制后的指针需要返回

因此需要一个指针来记录地址的初始值，最后将复制的结果返回是为了进行链式操作。

4.下面关于“联合”的题目的输出？（非常重要）

      a) ＃i nclude <stdio.h>

union {  int i; char x[2]; }a;

void main() {

a.x[0] = 10;

a.x[1] = 1;

a.printf("%d",a.i); }

     解析：266 （低位低地址，高位高地址，内存占用情况是Ox010A）

b）      main()      {

           union{                   /\*定义一个联合\*/

                int i;

               struct{             /\*在联合中定义一个结构\*/

                     char first;

char second;

}half;

}number;

            number.i=0x4241;         /\*联合成员赋值\*/

            printf("%c%c\n", number.half.first, mumber.half.second);

           number.half.first='a';   /\*联合中结构成员赋值\*/

            number.half.second='b';

  getch();

      }

答案： AB   （0x41对应'A'，是低位；Ox42对应'B'，是高位）

6261 （number.i和number.half共用一块地址空间）

5.一个数组，下标从0到n，元素为从0到n的整数。判断其中是否有重复元素。     
   解析： int   hasDuplicate(int[]   a,   int   n){     
            for(int   i=0;i<n;++i){     
              while(a[i]!=i && a[i]!=-1){     
                  if(a[a[i]]==-1)

return   1;     
                  a[i]=a[a[i]];     
                  a[a[i]]=-1;     
                    }     
          if(a[i]==i) {a[i]=-1;}     
 }     
  return   0;  }

6.反转一个链表：   
    解析：  
    1     List   reverse(List   l)   {     
    2     if(!l)   return   l;     
    3     list   cur   =   l.next;     
    4     list   pre   =   l;     
    5     list   tmp;     
    6     pre.next   =   null;     
    7     while   (   cur   )   {     
    8         tmp   =   cur;     
    9         cur   =   cur.next;     
  10         tmp.next   =   pre;     
  11         pre   =   tmp;     
  12     }     
  13     return   tmp;     
  14   }  

7.输出一个字符串所有排列。注意有重复字符。     
      
    1char[]   p;     
    2void   perm(char   s[],   int   i,   int   n){     
    3   int   j;     
    4   char   temp;     
    5   for(j=0;j<n;++j){     
    6     if(j!=0   &&   s[j]==s[j-1]);     
    7     elseif(s[j]!='@'){     
    8       p[i]=s[j];     
    9       s[j]='@';     
  10       if(i==n-1){     
  11         p[n]='\0';     
  12         printf("%s",   p);     
  13       }else{     
  14         perm(s,i+1,n);     
  15       }     
  16       s[j]=p[i];     
  17     }     
  18   }     
  19}     
  --------------------------     
  1void   main()   {     
  2     char   s[N];     
  3     sort(s);     
  4     perm(s,0,strlen(s));     
  5}  

8：编写一个函数，作用是把一个char组成的字符串循环右移n个。比如原来是“abcdefghi”如果n=2，移位后应该是“hiabcdefgh”，函数头是这样的：

//pStr是指向以'\0'结尾的字符串的指针

//steps是要求移动的n

void LoopMove ( char \* pStr, int steps )

{

　//请填充...

}

解析：第一种方法

void LoopMove ( char \*pStr, int steps )

{

　 int n = strlen( pStr ) - steps;

　 char tmp[MAX\_LEN];

　 strcpy ( tmp, pStr + n );

　 strcpy ( tmp + steps, pStr);

　 \*( tmp + strlen ( pStr ) ) = '\0';

　 strcpy( pStr, tmp );

}

第二种方法：

void LoopMove ( char \*pStr, int steps )

{

　 int n = strlen( pStr ) - steps;

　 char tmp[MAX\_LEN];

　 memcpy( tmp, pStr + n, steps );

　 memcpy(pStr + steps, pStr, n );

　 memcpy(pStr, tmp, steps );

}

9：请写代码实现100！注意溢出问题

解析：用数组的方式实现

#include <stdio.h>

int main()

{

int n = 1;

scanf("%d",&n);

int a[20000];

int carry;

int i;

int digit = 1;

a[0] = 1;

int temp;

for(i = 2; i <= n; ++i)

{

carry = 0;

for(int j = 1; j <= digit; ++j) //digit 大数的位数

{

temp = a[j-1] \* i + carry;

a[j-1] = temp % 10;

carry = temp / 10; //进位

}

while(carry)

{

a[++digit-1] = carry % 10;

carry /= 10;

}

}

for(i = digit; i >=1; --i)

{

printf("%d",a[i-1]);

}

printf("\n");

n++;

return 0;

}

1. 编写一个数组冒泡排序的算法实现

void func() {

int i,j,temp;

int a[10];

for(i=0;i<10;i++)

scanf ("%d,",&a[i]);

for(j=0;j<=9;j++) {

for (i=0;i<10-j;i++)

｛

if (a[i]>a[i+1]) {

temp=a[i];

a[i]=a[i+1];

a[i+1]=temp;

}

｝

}

for(i=1;i<11;i++)

printf("%d,",a[i] );

printf("\n");

}

11:约瑟夫出圈问题 ：一个船在大海中丢失了，为了解决生计，必须杀一个人吃，从第一个人开始，遇到基数为4的就拖出去杀了吃，一直循环，到最后剩下的人是谁？写代码实现

解析：

#include <stdio.h>

int main()

{

char a[5]={'a','b','c','d','e'};

int i,j;

int total =5,num=4,start=0,out=0;

for(i=0;i<5;i++)

{

if(total==1)

{

printf("%c",a[total-1]);

}else{

out=(start+num-1)%total;

printf("%c",a[out]);

for(j=out;j<total-1;j++)

{

a[j]=a[j+1];

}

total--;

start = out;

}

}

printf("\n");

} return 0;

12：自己实现atoi，函数原型如下 int atoi(const char \*str); 把字符串 str转换成 int类型。

int atoi(const char \* str)  
{  
 assert(str!=NULL);

int result = 0;  
 int sign = 1;  
 if( ('0'<= str[0] && str[0] <= '9') ||(str[0]=='-') || str[0]=='+')  
 {  
 if(str[0] == '+' || str[0] == '-')  
 {  
 if(str[0]=='-')  
 {  
 sign = -1;  
 }  
 str++;  
 }  
 }  
   
 while('0'<= \*str && \*str <= '9' )   
 {  
 result = result\*10 + (\*str++ - '0'); //'0'ASCII码48  
 }  
  
 return result \* sign;  
}

13.一个链表的结构体如下：请按要求写出如下。已知两个链表head1 和head2 各自有序，请把它们合并成一个链表依然有序。(保留所有结点，即便大小相同）

struct Node  
{  
int data ;  
Node \*next ;  
};  
typedef struct Node Node ;

解析：  
Node \* Merge(Node \*head1 , Node \*head2)  
{  
 if ( head1 == NULL)  
 return head2 ;  
 if ( head2 == NULL)  
 return head1 ;  
 Node \*head = NULL ;  
 Node \*p1 = NULL;  
 Node \*p2 = NULL;  
 if ( head1->data < head2->data )  
 {  
 head = head1 ;  
 p1 = head1->next;  
 p2 = head2 ;  
 } else {  
 head = head2 ;  
 p2 = head2->next ;  
 p1 = head1 ;  
 }  
 Node \*pcurrent = head ;  
 while ( p1 != NULL && p2 != NULL)  
 {  
 if ( p1->data <= p2->data )  
 {  
 pcurrent->next = p1 ;  
 pcurrent = p1 ;  
 p1 = p1->next ;  
 }  
 else  
 {  
 pcurrent->next = p2 ;  
 pcurrent = p2 ;  
 p2 = p2->next ;  
 }  
 }  
 if ( p1 != NULL )  
 pcurrent->next = p1 ;  
 if ( p2 != NULL )   
 pcurrent->next = p2 ;  
 return head ;  
}

14:(网鸿科技)文件中有一组数组，排序后请输出到另外一个文件中(文件操作考核)

· void Sort(vector<int> &data)  · {

·     for(int i=1; i < data.size(); i++)  ·     {

·         int tmp = data[i];

·         for(int j=i-1; j>=0 && tmp < data[j]; j--)

·             data[j+1] = data[j];

·         data[j+1] = tmp;

·     }

· }

·

· int main()  · {

·     void Sort(vector<int> &data);  ·

·     ifstream in("data.txt");

·     if(!in)  ·     {  ·         cout << "file error" << endl;  ·         exit(1);  ·     }

·      int tmp;

·     vector<int> data;

·     while(!in.eof())  ·     {  ·         in>>tmp;  ·         data.push\_back(tmp);  ·     }

·

·     Sort(data);

·

·     ofstream out("result.txt");

·     if(!out)  ·     {  ·         cout << "file error!" << endl;  ·         exit(1);  ·     }

·

·     for(vector<int>::iterator it=data.begin(); it != data.end(); it++)

·         out << \*it << " ";

·

·     out.close();

·     return 0;

· }

15：有一组数据， 请使用快速排序的方法对他们进行排序(排序算法的考核)

· void swap(int \*x,int \*y)· {

·    int temp;

·    temp = \*x;

·    \*x = \*y;

·    \*y = temp;· }

·

· int choose\_pivot(int i,int j )· {·    return((i+j) /2);· }

· void quicksort(int list[],int m,int n)· {

·    int key,i,j,k;

·    if( m < n)·    {

·       k = choose\_pivot(m,n);

·       swap(&list[m],&list[k]);

·       key = list[m];

·       i = m+1;

·       j = n;

·       while(i <= j)·       {

·          while((i <= n) && (list[i] <= key))

·                 i++;

·          while((j >= m) && (list[j] > key))

·                 j--;

·          if( i < j)

·                 swap(&list[i],&list[j]);

·       }

·      // 交换两个元素的位置

·       swap(&list[m],&list[j]);

·      // 递归地对较小的数据序列进行排序

·       quicksort(list,m,j-1);

·       quicksort(list,j+1,n);

·    }

· }

16:遍历目录下的所有文件(输入一个文件夹路径,然后查找所有文件和子文件夹的文件打印出来)

解析：void printdir(char \*dir, int depth){  
    DIR \*dp;  
    struct dirent \*entry;  
    struct stat statbuf;  
    if((dp = opendir(dir)) == NULL) {  
        fprintf(stderr,"cannot open directory: %s\n", dir);  
        return;  
    }  
    chdir(dir);  
    while((entry = readdir(dp)) != NULL) {  
        lstat(entry->d\_name,&statbuf);  
        if(S\_ISDIR(statbuf.st\_mode)) {             
            if(strcmp(".",entry->d\_name) == 0 ||  
                strcmp("..",entry->d\_name) == 0)  
                continue;  
            printf("%\*s%s/\n",depth,"",entry->d\_name);  
             
            printdir(entry->d\_name,depth+4);  
        }  
        else printf("%\*s%s\n",depth,"",entry->d\_name);  
    }  
    chdir("..");  
    closedir(dp);  
}

17 写一个程序, 要求功能：求出用1，2，5这三个数不同个数组合的和为100的组合个数。

如：100个1是一个组合，5个1加19个5是一个组合

解析：最容易想到的算法是：

设x是1的个数，y是2的个数，z是5的个数，number是组合数

注意到0<=x<=100，0<=y<=50，0<=z=20，所以可以编程为：

number=0;

for (x=0; x<=100; x++)

for (y=0; y<=50; y++)

for (z=0; z<=20; z++)

if ((x+2\*y+5\*z)==100)

number++;

cout<<number<<endl;

18实现一个函数，把一个字符串中的字符从小写转为大写。

void uppers(char \*s,char \*us)

{

for(;\*s!='\0';s++,us++)

{

if(\*s>='a'&&\*s<='z')

\*us = \*s-32;

else

\*us = \*s;

}

\*us = '\0';

}

int main()

{

char \*s,\*us;

char ss[20];

printf("Please input a string:\n");

scanf("%s",ss);

s = ss;

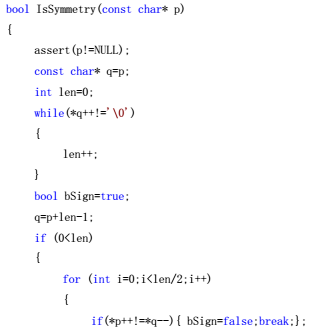
uppers(s,us);

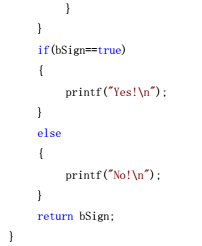
printf("The result is:\n%s\n",us);

getch();

}

19：写一个函数 int fun（char\* p）判断一个函数是否为回文，所谓的回文是（11234321）





20实现二分搜索算法。

解析：int FindBySrch(int \* ListData,int ListLength,int KeyData){

int low = 1;

int hight = ListLength；

while(low <= hight){

mid = (low+hight)/2;

if(KeyData > ListData[mid-1])

low = mid;

else if(KeyData < ListData[mid-1])

hight = mid;

else

return mid;}

return 0;

}

**21**试编写函数判断计算机的字节存储顺序是开序(little endian)还是降序(bigendian)

解析：

bool IsBigendian()

{

unsigned short usData = 0x1122;

unsigned char \*pucData = (unsigned char\*)&usData;

return (\*pucData == 0x22);

}

22 求两个数的最大公约数 函数原型如下  
 DWORD Divisor( DWORD dwFirst, DWORD dwSecond );返回最大公约数  
 解析：int hcf(int x,int y) {int t; if(x<y) {t=x; x=y; y=t;}

while((t=x%y)!=0) {x=y; y=t;}

return y;}

int lcf(int x,int y,int m) {

  return x\*y/m;   }

23用C 写一个输入的整数,倒着输出整数的函数,要求用递归方法 ;

答:

void fun( int a )

{

printf( "%d", a%10 );

a /= 10;

if( a <=0 )return;

fun( a );

}

24请写出二分查找：

int search(char\* cs, int n, char c){

int L=0;

int R=n-1;

while(L<=R){

int M = (L+R)/2;

if(cs[M]==c) return M;

if(c>cs[M]) L = M+1;

else R = M-1;

}

return -1;

}

int search(char\* cs, int from ,int to, char c){

if(from > to) return -1;

int m = (from + to)/2;

if(cs[m] == c) return m;

if(c>cs[m]) return search(cs, m+1, to, c);

return search(cs, from, m-1, c);

}

25 冒泡排序 时间复杂度: O(N\*N)

void sort(int\*a, int n){

for(int i=0; i<n-1; ++i){

int flag = true;//排好序了

for(int j=0; j<n-i-1; ++j){

if(a[j]>a[j+1]){

int t = a[j];

a[j] = a[j+1];

a[j+1] = t;

flag = false;

} }

if(flag) break;

cout << "第" << i << "次:";

print(a, n);

}

}

27插入排序 时间复杂度: O(N\*N)

void sort(int\*a, int n){

for(int i=1; i<n; i++){

int x = a[i];//要插入的值

int j;//要插入的位置

for(j=i; j>0; j--){

if(a[j-1]>x)

a[j] = a[j-1];

else break;

}

a[j] = x;

cout << "第" << i << "次:";

print(a, n);

}

}

28快速排序 时间复杂度: O(N\*log N)

void sort1(int\*a, int n){

//当一个组中的数据小于1时，分组结束

//有可能分组中是0个元素

if(n<1) return;

int L = 0;

int R = n-1;

while(L<R){

while(L<R&&a[L]<=a[R]) R--;

swap(a[L], a[R]);

while(L<R&&a[L]<=a[R]) L++;

swap(a[L], a[R]);

}

sort(a, L);//对左边分组

sort(a+L+1, n-L-1);//对右边分组

}

void sort2(int\*a, int n){

//当一个组中的数据小于1时，分组结束

//有可能分组中是0个元素

if(n<1) return;

int L = 0;

int R = n-1;

int m = a[L];//保存中间值

while(L<R){

while(L<R&&m<=a[R]) R--;

a[L] = a[R];

while(L<R&&a[L]<=m) L++;

a[R] = a[L];

}

a[L] = m;//把中间值放到正确位置

sort(a, L);//对左边分组

sort(a+L+1, n-L-1);//对右边分组

}

29选择排序 时间复杂度: O(N\*N)

void sort(int\*a, int n){

for(int i=0; i<n-1; i++){

int k = i;//最小值的下标

for(int j=i; j<n; j++){

if(a[j]<a[k]) k = j;

}

if(k!=i)swap(a[k], a[i]);

}

}

30最大公约数，最小公倍数，因式分解

1、最大公约数，最小公倍数

int m ,n , r;

int s=m\*n;

while (n!=0)

{

r=m%n;

m=n;

n=r;

}

最大公约数：m；

最小公倍数：s/m；

2、因式分解：

int n；

printf(“%d=”,n);

for(int i=2;i\*i<=n;)

{

if(n%2==0)

{

n=n/2;

printf(“%d\*”,i);

}

else

i++;

}

printf(“%d\n”,n);

31用递归算法判断数组a[N]是否为一个递增数组。

递归的方法，记录当前最大的，并且判断当前的是否比这个还大，大则继续，否则返回false结束：

bool fun( int a[], int n )

{

if( n= =1 )

return true;

if( n= =2 )

return a[n-1] >= a[n-2];

return fun( a,n-1) && ( a[n-1] >= a[n-2] );

}

32 、一个学生的信息是：姓名，学号，性别，年龄等信息，用一个链表，把这些学生信息连在一起， 给出一个age, 在些链表中删除学生年龄等于age的学生信息。  
 struct stu{  
 char name[20];  
 char sex;  
 int no;  
 int age;  
 struct stu \* next;  
 }\*linklist;  
struct stu \*creatlist(int n)  
{  
 int i;  
//h为头结点，p为前一结点，s为当前结点  
 struct stu \*h,\*p,\*s;  
 h = (struct stu \*)malloc(sizeof(struct stu));  
 h->next = NULL;  
 p=h;  
 for(i=0;i<n;i++) {   
 s = (struct stu \*)malloc(sizeof(struct stu));  
 p->next = s;  
 printf("Please input the information of the student: name sex no age \n");  
 scanf("%s %c %d %d",s->name,&s->sex,&s->no,&s->age);  
 s->next = NULL;  
 p = s;  
}  
 printf("Create successful!");  
 return(h);  
}  
void deletelist(struct stu \*s,int a)  
{  
 struct stu \*p;  
 while(s->age!=a)  
 {  
 p = s;  
 s = s->next;  
 }  
 if(s==NULL)  
 printf("The record is not exist.");  
 else{  
 p->next = s->next;  
 printf("Delete successful!");  
 }  
}  
void display(struct stu \*s){  
 s = s->next;  
 while(s!=NULL){  
 printf("%s %c %d %d\n",s->name,s->sex,s->no,s->age);  
 s = s->next;  
 }  
}  
int main(){  
 struct stu \*s;  
 int n,age;  
 printf("Please input the length of seqlist:\n");  
 scanf("%d",&n);  
 s = creatlist(n);  
display(s);  
printf("Please input the age:\n");  
scanf("%d",&age);  
deletelist(s,age);  
display(s);  
return 0;  
}

33. 请在小于99999的正整数中找符合下列条件的数，它既是完全平方数，又有两位数字相同，如：144，676。用c语言编写（不能用数字转换成字符串）(华为)。

int havesamenum(int num){  
 int i=0,j;  
 char a[10] = {0};  
   
 while(num>0) {  
  j=num%10;  
  a[j]+=1;  
  num=num/10;  
 }  
 while(a[i]<=1&&i<10)  
  i++;  
 if (i<10)   return 1;  
 else   return 0;  
}  
void main(void){  
   int i,j,m;   
 m=(int)sqrt(99999);  
 for(i=1;i<m;i++) {  
     j=i\*i;  
     if (1==havesamenum(j))      printf("%6d\t",j);  
 }  
}

34.用C++写个程序，如何判断一个操作系统是16位还是32位的？不能用sizeof()函数

解析:A1:

16位的系统下，

int i = 65536;

cout << i; // 输出0；

int i = 65535;

cout << i; // 输出-1；

32位的系统下，

int i = 65536;

cout << i; // 输出65536；

int i = 65535;

cout << i; // 输出65535；

A2:

int a = ~0;

if( a>65536 ) { cout<<"32 bit"<<endl;}

else{ cout<<"16 bit"<<endl;}

**改错题部分：**

1.请问下面程序有什么错误?

解析： int a[60][250][1000],i,j,k;

for(k=0;k<=1000;k++)

for(j=0;j<250;j++)

  for(i=0;i<60;i++)

a[j][k]=0; 把循环语句内外换一下

2.下面程序的打印结果是什么？请分析

#define Max\_CB 500

void LmiQueryCSmd(Struct MSgCB \* pmsg)  {

unsigned char ucCmdNum;

......

for(ucCmdNum=0;ucCmdNum<Max\_CB;ucCmdNum++)  {  ......;  }

解析：死循环,unsigned int值范围是0~255

3下面的代码有什么问题?  
 class A{  
 public:  
 A() { p=this; }  
 ~A() { if(p!=NULL) { delete p; p=NULL; } }

A\* p;  
 };

解析：:会引起无限递归

4.观察下面的程序，找出错误，主要是字符串的操作

试题1：void test1(){  
　 char string[10];  
　 char\* str1 ="0123456789";  
　 strcpy( string, str1 );  
 }

试题2：Void test2(){  
　 char string[10],str1[10];  
　 int i;  
　 for(i=0; i<10; i++)　{

str1 ='a';

}  
strcpy( string, str1 );

}

试题3：void test3(char\*str1){  
　 char string[10];  
　 if( strlen( str1 ) <=10)　{

strcpy( string, str1 );　}  
}

解析：试题1字符,串str1需要11个字节才能存放下（包括末尾的’\0’），

而string只有10个字节的空间，strcpy会导致数组越界；

　　 试题2，如果面试者指出字符数组str1不能在数组内结束可以给3分；

如果面试者指出strcpy(string,str1)调用使得从 str1内存起复制

到string内存起所复制的字节数具有不确定性可以给7分，

在此基础上指出库函数strcpy工作方式的给10分；

　　 试题3，if(strlen(str1)<= 10)应改为if(strlen(str1) < 10)，

因为strlen的结果未统计’\0’所占用的1个字节。剖析：

试题4：

void GetMemory( char\*p ){　p = (char\*) malloc( 100);}

void Test( void ) {  
　 char\*str = NULL;  
　 GetMemory( str );   
　 strcpy( str, "hello world" );  
　 printf( str );  
 }

试题5：char\*GetMemory( void){   
　 char p[] ="hello world";   
　 return p;   
 }

void Test( void ){   
　 char\*str = NULL;   
　 str = GetMemory();   
　 printf( str );   
 }

试题6：void GetMemory( char\*\*p, int num ){

\*p = (char\*) malloc( num );

}

void Test( void){

　 char\*str =NULL;  
　 GetMemory( &str, 100 );  
　 strcpy( str, "hello" );   
　 printf( str );

}

试题7：void Test( void){  
　 char\*str = (char\*) malloc( 100 );  
　 strcpy( str, "hello" );  
　 free( str );   
　 ... //省略的其它语句}

解析：  
　　试题4传入中GetMemory(char \*p )函数的形参为字符串指针，

在函数内部修改形参并不能真正的改变传入形参的值，执行完

char \*str = NULL;  
 GetMemory( str );后的str仍然为NULL；

试题5  
 char p[] = "hello world";  
 return p;的p[]数组为函数内的局部自动变量，在函数返回后，内存已经被释放。

这是许多程序员常犯的错误，其根源在于不理解变量的生存期。

试题6 GetMemory避免了试题4的问题，传入GetMemory的参数为字符串指针的指针，

但是在GetMemory中执行申请内存及赋值语句

\*p = (char \*) malloc( num );  
 后未判断内存是否申请成功，应加上：

if ( \*p == NULL ){　...//进行申请内存失败处理}

试题7存在与试题6同样的问题，在执行char \*str = (char \*) malloc(100);  
　　后未进行内存是否申请成功的判断；另外，在free(str)后未置str为空，导致可能变成一个“野”指针， 应加上：str = NULL;试题6的Test函数中也未对malloc的内存进行释放

5:对下面的程序改错

＃i nclude <stdio.h>

int main(void) {

int \*\*p;

int arr[100];

p = &arr;

return 0;

}

解析：

int \*\*p; //二级指针

&arr; //得到的是指向第一维为100的数组的指针

应该这样写＃i nclude <stdio.h>

int main(void) {

int \*\*p, \*q;

int arr[100];

q = arr;

p = &q;

return 0;

}

6.请找出下面代码中的所有错误 (题目不错,值得一看)

说明：以下代码是把一个字符串倒序，如“abcd”倒序后变为“dcba”

  #include"string.h"   main()   {

char\*src="hello,world";

char\* dest=NULL;

int len=strlen(src);

dest=(char\*)malloc(len);

char\* d=dest;

char\* s=src[len];

while(len--!=0)

\*d++=\*s--;

printf("%s",dest);

return 0;    }

解析:方法1：一共有4个错误;

int main() {

  char\* src = "hello,world";

int len = strlen(src);

  char\* dest = (char\*)malloc(len+1);//要为分配一个空间

char\* d = dest;

 char\* s = &src[len-1];         //指向最后一个字符

while( len-- != 0 )      \*d++=\*s--;

  \*d = 0;           //尾部要加’\0’    printf("%sn",dest);

free(dest);        // 使用完，应当释放空间，以免造成内存汇泄露

dest = NULL;   //防止产生野指针  return 0;  }

方法2： (方法一需要额外的存储空间,效率不高.) 不错的想法

#include <stdio.h>

 #include <string.h>

main() {

charstr[]="hello,world";

intlen=strlen(str);

char t;

for(int i=0;i<len/2; i++) {

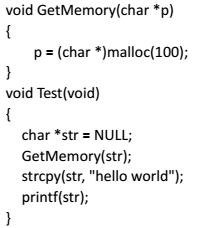
t=str[i];

str[i]=str[len-i-1];//小心一点

str[len-i-1]=t; }

printf("%s r); return 0; }

1. 下列程序是否有错误，请指出来



解析:传入GetMemory的是一个字符串指针，在函数内部改变形参的指针指向，

并不可以改变实参的指针指向，调用GetMemory后str还是NULL，

对一个NULL指针进行拷贝操作会导致段错误

1. 以下代码有什么问题？[C难]

void char2Hex( char c ) // 将字符以16进制表示

{

char ch = c/0x10 + '0'; if( ch > '9' ) ch += ('A'-'9'-1);

char cl = c%0x10 + '0'; if( cl > '9' ) cl += ('A'-'9'-1);

cout << ch << cl << ' ';

}

char str[] = "I love 中国";

for( size\_t i=0; i<strlen(str); ++I )

char2Hex( str[i] );

cout << endl;

}

9.下面的程序有什么问题？如果有问题请改正

#include <stdio.h>

#include <stdlib.h>

void getmemory(char \*p){

p=(char \*) malloc(100);

strcpy(p,"hello world");

}

int main( ){

char \*str=NULL;

getmemory(str);

printf("%s/n",str);

free(str);

return 0;}

解析：程序崩溃，getmemory中的malloc 不能返回动态内存， free（）对str操作很危险