1. 内核的裁剪怎么裁剪
2. 深入理解uboot

3.TCP/IP TCP/UDP原理是什么？

4.项目里你做了些什么？怎么做的？为什么要那样做？

5.指针与数组的用法及计算

6.单链表和双链表的用法

7.虚函数，构造函数，析构函数，复制和赋值函数

8.串口，驱动

9.冒泡排序

10.二极管和三极管的基本应用

11.C基础知识的加深

12.面试前深入了解该公司的产品及技术

13.原来做的工作怎么与以后的工作联系到一块

14.我们为什么要录用你，你有哪些能力，你是否合适这个岗位

15.你为什么第一份工作选择广达，你为什么要离开它，你通过培训你有什么收获，你对我们公司有什么了解，你有什么问题要问的

16.sizeof,#define,设置和清零，static作用,strcpy,中断，GPIO，volatile,链表的注意事项，大小端

17.编写一个函数输入一个整数并输出其2/8/16/进制

1不按顺序  
（1）编写一个c语言程序，判断大端模式和小端模式。  
（2）嵌入式系统的任务状态。  
（3）编写一个宏，实现判断数组a元素的个数。  
（4）堆和栈的区别  
（5）static的作用  
（6）conse的作用  
（7）实时性linux系统和非实时性系统的区别  
（8）传值调用和传址调用  
（9）sizeof和strlen的用法  
（10）在arm9模式下，对数值的某一位进行置1和清0，  
（11）在arm9模式下，对一个地址为0×80000000的寄存器，写入数值0×78.。  
（12）char a[5][4]，a的地址为0，问printf(“%d”，a[4]+3）值  
（13）全局变量、局部变量、局部静态变量  
(14)char \* =”abcdef”；  
char a[]=”abcdef”； 编译之前的区别  
（15）union的长度  
（16）信号的种类，

（17）进程

七、linux开发  
1、linux内核有几种锁,它们有什么 不同。  
2、字符设备驱动和块设备驱动的区 别。  
3、画出你所理解的linux系统整体 架构。（从硬件到应用层）

1、 int a=2,b=11,c=a+b++/a++; 则c值为多少？  
【考点】编码规范。  
表面上考察你对运算符优先级的掌握程度，但实际上优先级这些玩意很难死记硬背得住？大家的疑惑不就是运算符的结合顺序么？那么如何去避免呢？c=a+((b++)/(a++))不就行了么，其实问题背后考察的是你的编码规范，如何写清晰易懂的代码，如何在一个团队中让自己的代码狠容易毫无偏差的让新人看懂～

2、 Static用在全局变量定义和函数内部局部变量定义有何区别？  
【考点】编码规范及模块化设计。  
Static变量的两个属性涉及存储位置及可见域。在大型的程序设计中static全局变量可以有效的避免名字空间的重复及无意的变量引用，这样可以有效的提高程序设计的安全性，同时也利用模块化作业

3、 在程序设计中哪些地方需要使用define？如何用define来定义一年内有多少小时的常量？  
【考点】可移植性。  
嵌入式程序设计的特点就是软硬件平台可变性，有效利用define常量可以提高程序的可移植性，这样改动方便，不易出错

4、 Define宏语句和inline函数有什么区别？  
【考点】时空效率及宏的副作用。  
嵌入式系统平台通常存储资源有限，但同时又对实时性有一定的要求，二者如何权衡需要考虑。Define宏语句相对于函数调用能提高运行时间性能，但消耗了空间，并且不标准的宏语句定义在不标准的编码中更容易出现副作用，因此inline函数则是define语句的完美替代品。小资金干大事正是嵌入式的终极目标！

5、 哪些地方可以用到const？const变量和函数输入参数用const修饰有哪些作用？  
【考点】程序设计可靠性。  
Const只读变量可以有效克服define常量的一些弊端，如无类型安全检查。Const形参可以有效避免程序内部更改无意的更改。人非圣贤孰能无误，所以我们要将这些可能的失误统统交给智能的编译器，这样就可以将错误扼杀在摇篮中

6、 Int，short，char在32位平台下各占几个字节？一个结构体含有int、short、char变量各一个，结构体占据的总内存空间多少？  
【考点】跨平台移植。  
嵌入式并非x86，其硬件平台具备很大的差异性，同为int在不同的架构体系下可能占据的字节数是不一样的。不同的平台也有不同的编译器，其在变量对齐方面可能有不同的特性。因此在结构体定义时如何设计成员变量的顺序以有效减少占据的内存空间，以及如何填充特定字段保证访问对齐问题，都是嵌入式系统结构体设计时应该考虑的跨平台可移植性问题

7、 如何将unsigned int明确定义为一个32位类型？define和typedef有何区别？  
【考点】跨平台移植。  
define和typedef都可以实现变量类型重定义，但是typedef类型的指针变量可以有效去除变量定义的二义性。在嵌入式的工程项目中，通常为了保证数据类型的统一且便于跨平台移植时修改数据类型方便，通常都会单独定义一个数据类型的头文件，把所有用到的数据类型typedef成所熟悉的形式

8、 有没有用过volatile？有哪些典型的适用场合？  
【考点】编译优化的副作用。  
通常为了提高程序运行的效率，编译器会自动进行一些优化，如将变量放在寄存器中，以减少存储器访问次数，在数据长期未变时不重新读取内存等等。但是优化有时会带来问题，如硬件寄存器、多任务共享变量、中断和主程序共享变量，优化有可能带来数据访问不一致性的问题，因此对于这些个别变量，我们需要用volatile声明告诉编译器取消优化

9、 参数传递有哪些形式？寄存器和堆栈传递各有什么优缺点？  
【考点】编译优化、调用性能、接口设计。  
每种体系结构及对应的编译器对参数传递都有自己的规定。参数传递并非总是通过堆栈进行的，参数入栈出栈是需要耗费时间的，编译器总是尽量优化利用寄存器来传递参数，因为寄存器的访问效率要高，但当参数过多时，将放弃优化从而用栈传递参数。因此为了提高调用性能，应尽量减少参数个数，太多时可以将所有参数重新定义为一个结构体，利用结构体指针来传递参数。在函数接口设计时应考虑硬件平台和编译器的特性，以灵活定义参数形式

10、 中断服务程序设计应注意的问题？中断触发方式的选择？中断处理太长怎么办？中断处理程序如何与主应用程序交互？  
【考点】中断服务程序设计。  
嵌入式程序最大的特点是经常需要和硬件打交道，中断是接收外界输入的典型方式，通常都决定了系统运行流程，因此如何高效不丢失的处理中断是每个嵌入式程序员应考虑的问题

11、 TCP/IP和OSI七层模型是如何划分的？各层的功能，这么设计有什么好处？  
【考点】层次化软件设计保证可移植性。  
分层模型最大的好处就是某一层变化了，只要其提供上层的接口未变，那么上层就无需做任何改动。因此只需要更改本层的实现即可。嵌入式平台因为软硬件多变性，为了最大限度的利用先前的成果，软件设计一定要遵循这种层次化模型，这样才能保证其可移植性

12、 TCP和UDP的区别？各种网络互连设备都用在哪一层？如何建立链接？如何进行拥塞控制？  
【考点】协议设计可靠性及缓冲区设计管理。  
TCP和UDP的最大区别在于可靠性，TCP通过三次握手协议及超时机制安全可靠的建立或者释放连接。协议设计最大的问题就是如何保证效率合可靠性，TCP的设计为我们提供了一个参考。而滑动窗口机制可以有效的进行拥塞控制，但窗口大小的设计则关系到内存利用率及缓冲效率可靠性等问题。在嵌入式的驱动程序设计中，经常会开辟缓冲区来进行流量控制及防止数据覆盖，缓冲区的大小设计则需要更加具体的应用情况设计才能保证可靠性合灵活性

一.选择题（共15题，每题3分）  
（1）下面关于算法说法错误的是\_\_\_\_\_\_\_。  
a. 算法最终必须由计算机程序实现  
b. 为解决某问题的算法同为该问题编写的程序含义是相同的  
c. 算法的可行性是指指令不能有二义性  
d. 以上几个都是错误的  
（2）下面说法错误的是\_\_\_\_\_\_.  
a. 算法原地工作的含义是指不需要任何额外的辅助空间  
b. 在相同的规模n下，复杂度O(n)的算法在时间上总是优于复杂度O(2n)的算法  
c. 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界  
d. 同一个算法，实现语言的级别越高，执行效率就越低  
（3）在下面的程序段中，对x的赋值语句的频度为\_\_\_\_\_。  
for (int i; i<n; i++)  
{ for (int j=o; j<n; j++)  
{ x:=x+1;  
} }  
a. 0(2n)            b. 0(n)        c. 0(n2)     d. O(log2n)  
（4）下面说法正确的是\_\_\_\_\_\_。  
a. 数据元素是数据的最小单位；  
b. 数据元素是数据的最小单位；  
c. 数据的物理结构是指数据在计算机内的实际存储形式  
d. 数据结构的抽象操作的定义与具体实现有关  
（5）下面说法正确的是\_\_\_\_\_\_\_。  
a. 在顺序存储结构中，有时也存储数据结构中元素之间的关系  
b. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高  
c. 数据结构的基本操作的设置的最重要的准则是，实现应用程序与存储结构的独立  
d. 数据的逻辑结构说明数据元素之间的顺序关系,它依赖于计算机的储存结构  
（6） 下述\_\_\_\_\_是顺序存储结构的优点。  
a. 存储密度大  
b. 插入运算方便  
c. 删除运算方便  
d. 可方便地用于各种逻辑结构的存储表示  
（7）下面关于线性表的叙述中，错误的是\_\_\_\_\_。  
a. 线性表采用顺序存储，必须占用一片连续的存储单元  
b. 线性表采用顺序存储，便于进行插入和删除操作  
c. 线性表采用链接存储，不必占用一片连续的存储单元  
d. 线性表采用链接存储，便于插入和删除操作  
（8） 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用\_\_\_\_\_\_\_存储方式最节省时间。  
a. 顺序表      b. 双链表        c. 带头结点的双循环链表     d. 单循环链表  
（9）静态链表中指针表示的是\_\_\_\_\_\_。  
a. 内存地址     b. 数组下标     c. 下一元素地址     d. 左、右孩子地址  
（10）下面的叙述不正确的是\_\_\_\_\_\_\_。  
a. 线性表在链式存储时，查找第i个元素的时间同i的值成正比  
b. 线性表在链式存储时，查找第i个元素的时间同i的值无关  
c. 线性表在顺序存储时，查找第i个元素的时间同i 的值成正比  
d. 线性表在顺序存储时，查找第i个元素的时间同i的值无关  
（11）下面说法错误的是\_\_\_\_\_。  
a. 静态链表既有顺序存储的优点，又有动态链表的优点。所以，它存取表中第i个元素的时间与i无关。  
b. 静态链表中能容纳的元素个数的最大数在表定义时就确定了，以后不能增加。  
c. 静态链表与动态链表在元素的插入、删除上类似，不需做元素的移动。  
d. 静态链表就是一直不发生变化的链表。  
（12）在双向链表指针p的结点前插入一个指针q的结点操作是\_\_\_\_\_\_。  
a. p->Llink=q; q->Rlink=p; p->Llink->Rlink=q; q->Llink=q；  
b. p->Llink=q; p->Llink->Rlink=q; q->Rlink=p; q->Llink=p->Llink;  
c. q->Rlink=p; q->Llink=p->Llink; p->Llink->Rlink=q; p->Llink=q;  
d. q->Llink=p->Llink; q->Rlink=q; p->Llink=q; p->Llink=q;  
（13）下面说法正确的是\_\_\_\_\_\_。  
a. 顺序存储结构的主要缺点是不利于插入或删除操作；  
b. 线性表采用链表存储时，结点和结点内部的存储空间可以是不连续的；  
c. 顺序存储方式插入和删除时效率太低，因此它不如链式存储方式好；  
d. 顺序存储方式只能用于存储线性结构。  
（14）下面说法正确的是\_\_\_\_\_\_。  
a. 线性表只能用顺序存储结构实现。  
b. 为了很方便的插入和删除数据，可以使用双向链表存放数据。  
c. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。  
d. 链表是采用链式存储结构的线性表,进行插入、删除操作时，在链表中比在顺序存储结构中效率高。  
（15）下面说法正确的是\_\_\_\_\_\_\_\_\_。  
a. 数据元素是数据的最小单位。  
b. 队列逻辑上是一个下端口和上端能增加又能减少的线性表。  
c. 任何一个递归过程都可以转换成非递归过程。  
d. 只有那种使用了局部变量的递归过程在转换成非递归过程时才必须使用栈。

二.  填空题(共5题，每题5分)  
（1）    下列程序的功能是创建单向链表，请补充完整。  
#include <stdio.h>  
#include <alloc.h>  
struct link  
{ char  name[10];  
int    mark;  
struct link  \* next;  
};  
void insert(char \* name,  int mark);  
struct link \* head = NULL;  
main()  
{ char  name[10];  
int    mark;  
struct  link \*t;  
while (1)  
{  scanf(“%s %d”,  name,  &mark);  
if (strcmp(name, “#”) == 0 )  
{ break; }  
\_\_\_\_\_\_(1)\_\_\_\_\_\_\_;  
}  
for (t=head; \_\_\_\_\_\_(2)\_\_\_\_\_\_\_)  
{  
printf(“<%s>: %d\n”,  t->name,  t->mark);  
}  
}  
void insert(char \* name,  int mark)  
{  
struct link \* p;  
p = \_\_\_\_\_\_(3)\_\_\_\_\_\_\_ ;  
strcpy(p->name,  name);  
p->mark = mark;  
\_\_\_\_\_\_(4)\_\_\_\_\_\_\_;  
if ( head != NULL )  
{  
\_\_\_\_\_\_(5)\_\_\_\_\_\_\_;  
}  
head = p;  
}  
（2）    用循环链表表示的队列长度为n， 若只设头指针,则出队和入队的时间复杂度分别是\_\_\_\_\_\_和\_\_\_\_\_； 若只设尾指针,则出队和入队的时间复杂度分别是\_\_\_\_\_和\_\_\_\_\_。  
（3）    在n个记录的有序顺序表中进行折半查找,最大的比较次数是\_\_\_\_\_\_。  
（4）    仔细阅读下列程序，在空白处填入适当的语句。  
函数match(s,t)完成在字符串s中寻找与t匹配的字符，若存在一个匹配，则返回t在字符串s中的下标；否则，返回-1。其中，字符指针\*b始终指向s的第一元素。  
Match(s,t)  
Char s,t;  
｛ char \*b=s;  
char \*p, \*r;  
for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
｛  
for (p=s, r=t; \*r!=`\0` && \*p= =\*r; p++, r++);  
if\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
return(s-b);  
｝  
return(-1);  
｝  
（5）    补充下列程序：设一棵二叉序列树b，下列算法函数是实现在b中插入一个结点s。  
函数：  
void insert（btree \*b，btree \*s）  
{ if（b == NULL） b = s；  
else  
if（s->data == b->data） return();  
else  
if(s->data < b->data)  
；  
else  
；  }  
三.简答题（共3题，每题10分）  
（1）    在一个包含 n 个元素的数组 M 中查找一个元素 x。 算法假设 M 已经按升序排列了，请写出二分搜索算法的步骤。

（2）    试将一个无序的线性表A=(11,16,8,5,14,10,38,23)转换成一个按升序排列的有序线性表（用链表实现）。

（3）    何为栈和队列？简述两者的区别和联系。

一. 选择题（共15题，每题3分。）  
（ 1）在C语言中,  char型数据在内存中的存储形式是\_\_\_\_\_。  
a. 补码   b. 反码   c. 原码    d. ASCII码

（2）设C语言中,一个int型数据在内存中占2个字节,则unsigned  int型数据的取值范围为\_\_\_\_\_\_。  
a. 0～255   b. 0～32767     c. 0～65535   d. 0～2147483647

（3）已知ch是字符型变量,下面不正确的赋值语句是\_\_\_\_。  
a. ch=’a+b’;   b. ch=’\0′;   c. ch=’7′+’9′;    d. ch=5+9;

（4）已知ch是字符型变量,下面正确的赋值语句是\_\_\_\_。  
a. ch= ’123′;   b. ch= ‘\xff’;  c. ch= ‘\08′;   d. ch=”\”;

（5）以下选项中不正确的整型常量是\_\_\_\_\_。  
a.12L        b. -10       c. 1,900    d. 123U

（6）假设int型变量占两个字节的存储单元，若有定义：  
int x[10]={0,2,4};  
则数组x在内存中所占字节数为\_\_\_\_\_\_。  
a. 3           b. 6           c. 10             d. 20

（7）C语言中能用八进制表示的数据类型为\_\_\_\_\_。  
a. 字符型、整型                    b. 整形、实型  
c. 字符型、实型、双精度型      d. 字符型、整型、实型、双精度型

（8）以下只有在使用时才为该类型变量分配内存的存储类说明是\_\_\_\_。  
a. auto和static              b. auto和register  
c. register和static         d. extern和register

（9）下面四个选项中,均是合法整型常量的选项是\_\_\_\_。  
a. 160          b. 0xcdf    c. 01        d. 0x48a  
0xffff           01a       986,012        2e5  
011              0xe        0668           0x

（10）下面四个选项中,均是不合法的整型常量的选项是\_\_\_\_。  
a. 0f1     b. 0Xcdf c. 018    d. 0x48eg  
0xffff        017          999        068  
0011          12,456       5e2        03f

（11）下面四个选项中,均是不合法的浮点数的选项是\_\_\_\_。  
a. 160.      b. 123      c. 18      d. e3  
0.12          2e4.2       123e4         .234  
e3           .e5          0.0            1e3

（12）下面四个选项中,均是合法浮点数的选项是\_\_\_\_。  
a.  +le+1      b. 60.       c. 123e      d. e3  
5e-9.4  12e-4      1.2e-.4      .8e4  
03e2          8e5         +2e           5.e

（13）下面四个选项中,均是合法转义字符的选项是\_\_\_\_。  
a. ‘\”        b. ‘\’     c. ‘\018′   d. ‘\\0′  
‘\\’            ‘\017′     ‘\f’         ’101′  
‘\n’            ‘\”‘        ‘xab’       ‘xlf’

（14）下面四个选项中,均是不合法的转义字符的选项是\_\_\_\_。  
a. ‘\”     b. ‘\1011′    c. ‘\011′    d. ‘\abc’  
‘\\’          ‘\’           ‘\f’          ‘\1011′  
‘\xf’          ‘\a’          ‘\}’         ‘xlf’

（15）下面正确的字符常量是\_\_\_\_。  
a. “a”     b. ‘\\”    c. ‘W’    d. ”

二．填空题（每题6分）。

（1）说明下面变量的定义：  
char (\*p[])();          \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
char \*p()[];    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
char (\*p)[10][10];     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
char(\*(\*p())[])[];     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

（2）下面共同体的存储空间是\_\_\_\_\_  
union BOOKNUMBER  
{  
T\_S16 age;  
T\_S8 title[10];  
T\_S8 post[32];  
}

（3）若整数值范围是 -32768 至 32767。在执行下面的表达式之后, i 的结果是什么?  
int i=32766;  
i=i+1;          i =  
i=i+1;          i =

（4）对于联合类型：  
union x  
{  
char ch[7];  
int i[2];  
float f;  
}tom;  
那么, sizeof(tom) =

（5）有以下程序，程序的输出结果是\_\_\_\_\_\_（注意: ch[0]在低字节,ch[1]在高字节）。  
#include<stdio.h>  
union pw  
{  
int i;  
char ch[2];  
}a;

main()  
{  
a.ch[0] = 13;  
a.ch[1] = 0;  
printf(“%d\n”, a.i);  
}

三.简单题（共3题，每题10分）

（1）已知a为整数，比较下面代码的效率并说明原因：  
a++;  
a = a+1;

（2）已知a为整数，比较下面代码的效率并说明原因：  
a = a\*4;  
a << 2;

（3）说明一个变量和定义一个变量有什么区别?

一. 选择题（共15题，每题3分。）  
（1）设有以下语句:  
char a=3, b=6, c;  
c = a^b<<2;  
则c的二进制值是\_\_\_\_\_。  
a. 00011011  
b. 00010100  
c. 00011100  
d. 00011000

（2）表达式0×13&0×17的值是\_\_\_。  
a. 0×17  
b. 0×13  
c. 0xf8  
d. 0xec

（3）表达式0xl3|0xl7的值是\_\_\_\_。  
a. 0×13  
b. 0×17  
c. 0xE8  
d. 0xc8

（4） 以下叙述中正确的是\_\_\_\_\_。  
a. 表达式a&＝b等价于a＝a&b  
b. 表达式a|＝b等价于a＝a|b  
c. 表达式a!＝b等价于a＝a!b  
d. 表达式a^＝b等价于a＝a^b

（5） 运作对象必须是整型数的运算符是\_\_\_\_\_。  
a. %    b. .    c. /    d. \*\*

（6） 根据以下定义，错误的表达式是\_\_\_\_\_。  
struct  
{  
int a;  
char b;  
} Q, \*p = &Q;

a. Q.a  
b. (\*p).b  
c. p->a  
d. \*p.b

（7） 若x是单精度实型变量，表达式 x = 10/4 的值是\_\_\_\_\_ 。  
a. 2.5         b. 2.0        c. 3         d. 2

（8） 表达式 !x 等价于\_\_\_\_\_。  
a. x == 0        b. x == 1       c. x != 0     d. x != 1

（9） \_\_\_\_\_\_是合法的用户自定义标识符。  
a. b-b        b. float         c. <fr>          d. \_isw

（10） 若变量已正确定义并赋值，表达式\_\_\_\_\_\_不符合C语言语法。  
a. a\*b/c;     b. 3.14%2       c. 2, b         d. a/b/c

（11） \_\_\_\_\_是不正确的字符常量。  
a. ‘\n’        b. ’1′           c. “a”          d. ‘\101′

（12） 在 C 程序中，用\_\_\_\_\_表示逻辑值”真”。  
a. 1           b. 非 0 的数      c. 非 1 的数     d. 大于 0 的数

（13） \_\_\_\_\_\_把x、y定义成float类型变量，并赋同一初值3.14。  
a. float x, y=3.14;              b. float x, y=2\*3.14;  
c. float x=3.14, y=x=3.14;     d. float x=y=3.14;

（14） 假设变量已正确定义，表达式\_\_\_\_\_\_的值不是2。  
a. 2&3       b. 1<<1         c. a==2         d. 1^3

（15） 若变量已正确定义，语句if(a>b) k=0; else k=1;和\_\_\_\_\_\_等价。  
a. k=(a>b)?1:0;              b. k=a>b;  
c. k=a<=b;                    d. a<=b ? 0 : 1;

二．填空题（每题5分）。

（1） &运算符作为单目运算符时表示的是\_\_\_\_运算；作为双目运算符时表示的是\_\_\_\_\_运算。  
（2） 请计算sizeof的值：  
char str[] = “Hello” ;  
char \*p = str ;  
int n = 10;  
请计算  
sizeof (str ) = \_\_\_\_  
sizeof ( p ) = \_\_\_\_\_  
sizeof ( n ) = \_\_\_\_\_\_

（3） 请计算sizeof的值：  
void Func ( char str[100])  
{  
// 请计算  
sizeof( str ) = \_\_\_\_\_  
}

（4） 请计算sizeof的值：  
void \*p = malloc( 100 );  
// 请计算  
sizeof ( p ) = \_\_\_\_\_\_

（5） 下列程序段的输出是\_\_\_\_\_\_\_。  
#define f(a, b, x)  a\*x+b  
printf(“%d, %d\n”, f(3,2,1), f(6, 5, f(3, 2, 1)));

（6） 下列程序段的输出是\_\_\_\_\_。  
main()  
{  
int k;  
for(k=1; k<5; k++)  
{  
if(k % 2)  printf(“\*”);  
else continue;  
printf(“#”);  
}}

三.简单题（共3题，每题10分）

（1） 指出下面程序的错误并改正：  
void main(void)  
{int a,b;  
a = 1;  
b = a/++a;  
printf(“b equals %d\n”, b);}

（2） int a=391,b=1971; 写出 a^b^a的值。

（3） 语句：x++;?++x; x=x+1;?x=l+x; ，执行后都使变量x中的值增1，请写出一条同一功能的赋值语句。