**1.题目描述**

下面代码片段中有哪些隐患或者错误

void \*get1(){

char buf[12] = {"abc"};

return buf;

}

void get2(int \*p){

p = new int[10];

}

int main(){

int \*m;

get2(m);

m = get1();

memcpy(m, "aaa", 3);

return 0;

}

参考答案

* int \*m；//指针最好初始化，野指针容易出现非法访问
* p=new int [10];//new分配空间在堆，不会自动释放，需要手动释放内存
* return buf//指定返回类型不匹配，返回临时变量会出问题
* memcpy(m, "aaa", 3);//非法拷贝
* 首先调用get2(m)的时候，因为是值传递，无法修改m的实际值，即无法讲new申请的内存地址传出、
* 再者调用get1(m)的时候，函数内定义的数组是局部变量，对应的内存在这个函数结束的时候就会被释放，因此返回的地址是无效的，调用这样的地址造成严重错误。
* 最后调用memcpy(m,"aaa",3)往一个没有权限的无效内存空间拷贝数据是非法的。

**2题目描述**

用变量a给出以下定义：

一个指向指针的指针，它指向的指针是指向一个整型数

一个有10个指针的数组，该指针是指向一个整型数的

一个指向有10个整型数数组的指针

参考答案

int \*\*p;

int\*p[10];

int(\*p)[10];

**3题目描述**

简述SPI IIC UART接口的区别和各自收发数据的方法

参考答案

1. UART是全双工方式，主机和从机至少要接三根线，RX（接收）、TX（发送）和GND；

2. SPI是全双工串行总线，由一个主设备和多个从设备组成，由四跟信号线组成：SDI（串行数据输入）、SDO（串行数据输出）、SCK（串行移位时钟）、CS（从使能），以CS选择从机；

3. IIC是半双工通信，由两根信号线组成：SCL（时钟线）和SDA（数据线），以地址选择从机。

**4题目描述**

从以下几种无线通信方式中选取一种了解的尽可能详细的描述一下：BLE,WiFi ,NFC,NB-IoT,eMTC

参考答案

BLE：蓝牙技术(bluetooth)，它是一种无线数据与语音通信的开放性全球规范，它以低成本的短距离无线连接为基础，可为固定的或移动的终端设备提供廉价的接入服务。其传输频段为全球公众通用的2.4GHzISM频段，提供1Mbps的传输速率和10m的传输距离。蓝牙协议的标准版本为802.15.1，由蓝牙小组（SIG）负责开发。

WiFi：Wi-Fi(Wireless Fidelity,无线高保真)也是一种无线通信协议，正式名称是IEEE802.11b，与蓝牙一样，同属于短距离无线通信技术。Wi-Fi速率最高可达11Mb/s。Wi-Fi工作在2.4GHz频段。Wi-Fi是以太网的一种无线扩展，理论上只要用户位于一个接入点四周的一定区域内，就能以最高约11Mb/s的速度接入Web。

NFC： NFC（Near Field Communication，近距离无线传输）是由Philips、NOKIA和Sony主推的一种类似于RFID(非接触式射频识别)的短距离无线通信技术标准。NFC采用了双向的识别和连接。在20cm距离内工作于13.56MHz频率范围。NFC通过在单一设备上组合所有的身份识别应用和服务，帮助解决记忆多个密码的麻烦，同时也保证了数据的安全保护。

NB-IOT：窄带物联网（Narrow Band Internet of Things, NB-IoT）成为万物互联网络的一个重要分支。NB-IoT是IoT领域一个新兴的技术，支持低功耗设备在广域网的蜂窝数据连接，也被叫作低功耗广域网(LPWAN)。NB-IoT支持待机时间长、对网络连接要求较高设备的高效连接。据说NB-IoT设备电池寿命可以提高至少10年，同时还能提供非常全面的室内蜂窝数据连接覆盖。NB-IoT构建于蜂窝网络，只消耗大约180kHz的带宽，可直接部署于GSM网络、UMTS网络或LTE网络，以降低部署成本、实现平滑升级。

eMTC：全称是 LTE enhanced MTO，是基于LTE演进的物联网技术。为了更加适合物与物之间的通信，也为了更低的成本，对LTE协议进行了裁剪和优化。eMTC基于蜂窝网络进行部署，其用户设备通过支持1.4MHz的射频和基带带宽，可以直接接入现有的LTE网络。eMTC支持上下行最大1Mbps的峰值速率,可以支持丰富、创新的物联应用。

**5题目描述**

试描述MQTT、CoAP、HTTP三种协议的区别

参考答案

MQTT：MQTT(消息队列遥测传输)是ISO 标准(ISO/IEC PRF 20922)下基于发布/订阅范式的消息协议。它工作在 TCP/IP协议族上，是为硬件性能低下的远程设备以及网络状况糟糕的情况下而设计的发布/订阅型消息协议。MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT协议是轻量、简单、开放和易于实现的，这些特点使它适用范围非常广泛。在很多情况下，包括受限的环境中，如：机器与机器（M2M）通信和物联网（IoT）。其在，通过卫星链路通信传感器、偶尔拨号的医疗设备、智能家居、及一些小型化设备中已广泛使用。

CoAP：CoAP是一种计算机协议，应用于物联网，基于REST架构。由于物联网中的很多设备都是资源受限型的，即只有少量的内存空间和有限的计算能力，所以传统的HTTP协议应用在物联网上就显得过于庞大而不适用。 IETF的CoRE工作组提出了一种基于REST架构的CoAP协议。CoAP是6LowPAN协议栈中的应用层协议。

HTTP：http是一个简单的请求-响应协议，它通常运行在TCP之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。请求和响应消息的头以ASCII码形式给出；而消息内容则具有一个类似MIME的格式。

**协议栈：**

MQTT： TCP 长链接， CoAP： UDP 低功耗短连接， HTTP： TCP

**请求方式**

MQTT： 发布订阅， CoAP： request/ response， HTTP： request/ response

**使用特点与使用场景**

MQTT是多个客户端通过一个\*\*\*\*\*\*传递消息的多对多协议。它通过让客户端发布消息、\*\*\*决定消息路由和复制来解耦生产者和消费者。虽然MQTT持久性有一些支持，但它是最好的实时通讯总线。

CoAP基本上是一个在Client和Server之间传递状态信息的单对单协议。虽然它支持观察资源，但是CoAP最适合状态转移模型，而不是单纯的基于事件。

HTTP是适合使用在性能好一些的终端上，相对以上一些比较重，对设备要求相对高一些。不适合M2M的场景。

**6题目描述**

试描述TCP建立和断开连接时的三次握手和四次挥手

参考答案

三次握手：

1.客户端先发送标志位SYN=1，seq=x请求与服务器建立连接

2.服务器收到客户端的TCP报文后，返回标志位SYN=1，ACK=1，seq=y，ack=x+1的报文应答客户端并同意建议连接

3.客户端收到服务器的TCP报文后，返回标志位ACK=1，seq=x+1，ack=y+1的报文表示接收到服务器的消息并建立连接

四次挥手：

1.客户端向服务器发送标志位FIN=1，seq=x的报文请求断开连接

2.服务器收到后，返回标志位ACK=1，seq=y，ack=x+1的报文告诉客户端收到报文，并准备断开连接

3.服务器做好断开连接的准备后，给客户端发送标志位FIN=1，ACK=1，seq=u，ack=x+1的报文告诉客户端已做好准备断开连接

4.客户端收到报文后，返回标志位ACK=1，seq=x+1，ack=u+1的报文表示收到消息并断开连接

**7题目描述**

一个无序的长度为n的数组，找到最小的值，请用C语言编码实现以下函数

int get\_min(int\* array, int n) {

}

参考答案

#include <iostream>

#include <stdio.h>

using namespace std;

int get\_min(int\* array, int n)

{

 int temp = array[0];

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

 {

 if(temp > array[i])

 {

 temp = array[i];

 }

 }

 return temp;

}

int main()

{

 int Arr[10];

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

 {

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

 }

 cout << "the min data is: " << get\_min(Arr,sizeof(Arr)/sizeof(int));

 return 0;

}

**8题目描述**

工厂生产摩拜单车包含两道工序，工序A每5min生产一辆摩拜单车，生产完成后送到工序B检测，每1min检测一辆，检测失败的需要返回工序A重新生产；试用多线程（多任务）的机制实现上述的生产工序，实现产能的最大化。

参考答案

1. 创建两个线程ThreadA,ThreadB,分别用于A和B两个工序,创建消息队列QueueA用于A发送给B两个线程之前做消息通讯，创建一个链表List，用于存放B返回给A的单车，同事链表有一个互斥锁，Lock，用于保护链表，避免A B两个线程同时进行操作；

2. ThreadA,生产完一辆单车，通过消息队列发送给ThreadB，然后检测链表list，是否有需要重新生产的单车，若有则进行重新生产；

3. ThreadB,处于监听消息队列，若监听到A发过来的消息队列在，进行检测处理，若有不合格的单车，则把单车加入到链表List中；