1.将流量控制用于 TCP 数据传输的原因是什么？

同步设备速度以便发送数据

同步并对序列号排序，从而以完整的数字顺序发送数据

防止传入数据耗尽接收方资源

在服务器上同步窗口大小

2. 在 CPU 与主存之间设置高速缓冲存储器 Cache，其目的是为了

扩大主存的存储容量

提高CPU对主存的访问效率

提高外存储器的速度

既扩大主存容量又提高存取速度

既扩大主存容量又提高CPU对主存的访问效率

3. 设指针变量p指向双向链表中结点A（A不是最右边节点），指针变量s指向被插入的结点X，则在结点A的右面插入结点X的操作序列为：

p-&gt;right=s； s-&gt;left=p； p-&gt;right-&gt;left=s； s-&gt;right=p-&gt;right；

s-&gt;left=p； s-&gt;right=p-&gt;right； p-&gt;right=s； p-&gt;right-&gt;left=s；

p-&gt;right=s； p-&gt;right-&gt;left=s； s-&gt;left=p； s-&gt;right=p-&gt;right；

s-&gt;left=p； s-&gt;right=p-&gt;right； p-&gt;right-&gt;left=s； p-&gt;right=s；

4. 设无向图G中的边的集合E={(a，b)，(a，e)，(a，c)，(b，e)，(e，d)，(d，f)，(f，c)}，则从顶点a出发进行深度优先遍历可以得到的一种顶点序列为

aedfcb

acfebd

aebcfd

aedfbc

5. [不定项选择题]

关于数据库索引的说法哪些是正确的

经常被查询的字段建议创建索引

很少被查询的字段建议创建索引

内容很少变动的字段不建议创建索引

内容经常变动的字段不建议创建索引

索引可以提高数据插入效率

6. [不定项选择题]

下列措施中，能缩短程序执行时间的是

提高时钟频率

优化数据通路结构

程序编译优化

7. [不定项选择题]

以下是链表的特点的是

不必预先分配较多存储空间

可随机访问任一元素

插入删除不需要移动元素

所需空间与线性表长度成正比

8. [不定项选择题]

在现代计算机上，即使是单核单CPU系统，一个程序的死循环bug，也不会导致别的程序完全得不到时间运行，这跟哪些因素有关

时钟中断

OS进程（线程）时间片划分

虚拟内存机制

OS抢占式调度

9.[填空题]

有以下代码，则find(6)的返回值为\_\_\_\_\_

int find(int n){

if (n <= 0){

return 0;

}else if(n > 0 && n <= 2){

return 1;

}

return find(n-1)+find(n-2);

}

10. [填空题]

下面C程序运行后的打印结果是\_\_\_\_\_\_\_\_

#include <stdio.h>

void func(int\* a) {

static int j = -1;

do { j +=1; a[j] = a[j] + a[j+1]; } while (j < 2);

}

main( ) {

int k, a[10] = {1, 2, 3, 4, 5};

for (k=1; k<3; k++) func(a);

for (k=0; k<5; k++) printf("%d", a[k]);

printf("\n");

}

11. [填空题]

一棵完全二叉树中有65个结点，则该完全二叉树的深度为 \_\_\_\_\_\_

12. [填空题]

用二分查找法查找一个长度为112、已排序的数组，若查找目标不存在数组中，需要比较 \_\_\_\_\_\_次

13. [填空题]

已知一个线性表{24, 19, 33, 56, 72, 68}，假定采用hash函数h(key)=key%7计算hash地址，并存储在hash表A[0…6]中，若采用线性探测方法解决冲突（即若发生冲突，则从冲突位置顺序探测hash表中的其他存储单元，直到找到空位置为止），则在该hash表上查找元素68，需要查找多少\_\_\_\_\_\_ 步

14. [填空题]

已知二叉树的中序遍历结果为MFLEDABKCGHJI，后序遍历结果为FELMDKHGJICBA，则其先序遍历结果为\_\_\_\_

15. [填空题]

一组记录的关键值为(46，79，56，38，40，84)，则利用快速排序的方法，以第一个记录的关键值46为基准得到的一次划分结果为()

16. [填空题]

为了解决进程间的同步和互斥问题，通常采用一种称为信号量机制的方法。若系统中有6个进程共享若干个资源R，每个进程都需要5个资源R，那么使系统不发生死锁的资源R的最少数目是\_\_\_\_\_\_

17. [填空题]

已知一算数表达式的中缀表达式为 a\*f+(b-c/d)\*e，其后缀形式\_\_\_\_\_\_\_\_

18. [填空题]

袋子中分别一叠纸币，其中5元面值的纸币6张，10元面值的纸币5张，20元面值的纸币4张，从袋子中任意取4张纸币，则每种面值至少取到一张的概率为 1 %。（保留两位小数）

19. [填空题]

8瓶水中1瓶有毒，用动物测试。毒发症状在喝水2小时后开始出现，而你也只有2个小时的时间，则最少需要用\_\_\_\_\_\_\_\_\_只动物测试

20. [填空题]

【O】ping命令使用的协议是

21. [问答题]

题目描述

给定字符串s, 要求把s中多于一个的连续空压缩成一个空格，并将连续的非空格字符串倒序打印出来，例如，给定"abc def efg"，打印"cba fed gfe"

22. [问答题]

题目描述

围棋棋盘上有一片连续的白子，没有黑子。请写一个函数，计算返回该片白子的气数。函数输入参数为任一个白子的位置。

注：围棋规则：格子棋盘，棋子下在十字交叉点上，纵横线19\*19。一片棋子，与其中任一子相邻的空交叉点称为这片子的1口气，所有这样的交叉点数量是这片子的气数。比如中央的单独一个棋子，上下左右4口气，气数为4；棋盘左上角的单独棋子，右边加下边两口气，气数是2；中央的两个相连的白子，气数为6。等等。

23. [问答题]

题目描述

请设计一个整数容器，支持两个操作：add(x)和popMedia()两个操作。add(x)是向容器中加入一个整数；popMedia()是返回容器中当前所有数的中位数，如果中位数是容器中的数字，则返回的同时还从容器中把它删除。无需写出完全代码，描述清楚设计实现即可。另外，你能使两个操作都小于O(N)的时间吗？

注：中位数定义为：如果容器中整数的数量为奇数个，则是最中间的那个数字，如果为偶数个，则是最中间两个数的平均值。

24. [问答题]

题目描述

【测试方向优先】你用浏览器打开一个网站，却没有按预期看到应有的网页内容。请分析各种可能的原因，如果这些原因表现出来的现象不同，也请描述

25. [问答题]

题目描述

【测试方向优先】一个智能玩具，有N个不同模块，每个模块已独立测试过。每个模块有输入输出两个接口，可分别对接任意另外两个模块，（接一个的输出和另一个的输入），这样整个N个模块便可线性组装出多种不同模型（头尾不连成环形）。对任一种模型结果，有现成的通用测试过程。现在为了全覆盖测试，请你写函数生成出所有的模型作为测试用例。（每个模型都必须是用上全部N个模块来组装）

26. [问答题]

题目描述

【运维方向优先】a. 请描述TCP协议3次握手建立连接的过程。b. 为什么协议设计是3次握手连接建立而不是2次或4次，如果2次有什么问题，如果4次有什么问题？

27. [问答题]

题目描述

【运维方向优先】你用浏览器打开一个电商网站，准备浏览购物。从你输入完网站的网址敲下Enter键，到网站首页迅速呈现出来的这段时间，你的电脑系统里发生了哪些事情？请根据你所学的计算机知识，尽可能详细的解释。（提示：从软硬件的尽量多的层次考虑和描述。）

答案

1.C

2.B

3.D

4.A

5.AD

6.ABC

7.ACD

8.ABD

9.8

10.35745

11.7

12.7

13.4

14.ADMLFEBCKIJGH

15.40,38,46,56,79,84

16.25

17.af\*bcd/-e\*+

18.52.75

19.3

20.ICMP

21.参考要点：代码应一趟扫描处理完，分出每个词并逆序。

22.参考要点：白子是一个连通图（每个节点最多4条边），函数实现图的遍历，并计算每个节点（白子）的气数相加。

代码容易出现的错误：

遍历时未考虑回到原点形成无限循环的状况，遍历时未考虑棋盘边界情况

计数未考虑多个白子可以有公共的交叉点作为气，从而重复相加导致气数多算的情况

23.参考要点：

使用一个最大堆加一个最小堆组成结构。最大堆的堆顶值小于等于最小堆的堆顶值。也就是说把容器内的数据分两半，小的一半放在一个最大堆里，大的一半放在一个最小堆里。

add(x)操作时，根据x的大小，加入其中一个堆。如果两个堆元素数量相差到2，则调整堆使其数量相同。

popMedia()操作时，返回其中一个堆的堆顶，或两个堆顶的平均值。返回值是堆顶的话，还要删除该堆顶并调整堆，删除后如果两个堆元素数量相差到2，则调整堆使其数量相同。

add()/popMedia()操作时间复杂度均为O(lgN)。

24.参考要点：

人的原因：弄错网址；拼写错误；未打开网络；。。。

本机原因：浏览器版本低不兼容网页；本机防火墙阻止访问；中病毒导致浏览器工作不正常；。。。

服务端原因：服务器宕机；服务器出错；服务器高负载无法及时回应；服务器超负载主动拒绝回应；本机被服务端加入了黑名单；

网络原因：网络不通；网络拥塞；DNS解析失败；DNS解析到错误的IP；。。。

其他环境原因：域名被劫持；访问被黑客攻击；。。。

25.参考要点：实质是写一个函数，生成打印N个不同元素（模块）的所有N!个全排列（模型）。

示意伪代码：递归函数，两个参数，一个是排列前缀，一个是子排列的所有元素，元素使用整数类型。

// 初始调用传入left为空，childrenToPermute为N个元素的list。

function printPermutation(List<int> left, List<int> childrenToPermute)

{

if(childrenToPermute.size() == 0){

printList(left); // 将left中元素依次打印，即为一个排列，也即模型。

return; //递归结束

}

childrenToPermute.foreach( x => {

List<int> newLeft = left.copy().add(x);

List<int> newChildren = childrenToPermute.copy().remove(x);

printPermutation(newLeft, newChildren); // 递归调用

}

);

}

26.a 略

b. 握手达到3次使得两边都确认了通道的两个方向都是连通的，因为自己发出的包都得到了对方的回应，并且交换了初始信息（各自的包序号，窗口大小等），从而认为连接建立是合适的。

2次握手的话不足以保证通道双向正常，导致双方认知不一致，服务端会消耗不必要的资源（服务端认为连接建立，创建并维持连接状态数据，但因为单方向不通，确认包未被客户端收到，客户端认为连接不成功），同时还会存在安全问题被利用做攻击（不断发SYNC让服务端不断增加连接资源）。

4次握手的话，不比3次能提供更多的信息，多一次round-trip增大了连接建立的时间开销。

27.参考要点：回答是否从硬件（键盘网卡）、OS、TCP协议栈、运行库、浏览器、HTTP、HTML/JS等多层面描述。

参考回答：键盘硬件中断；OS处理中断，转换为特定消息放入浏览器程序的事件队列；浏览器的消息循环处理该消息，请求网址；OS请求本地域名缓存或域名服务器解析网址中的域名，得到IP；浏览器向该IP建立TCP连接（默认80端口）；浏览器发送GET请求，包含网站的路径，TCP协议栈组装为TCP包，通过网卡发送；浏览器等待网站回复，进程被OS切换为等待状态；网站返回的数据到来，网卡产生中断；OS处理中断，TCP协议栈将数据读入buffer；浏览器获得数据，处理HTTP头，显示HTML网页

更多：OS发ARP包获得网关MAC地址，所有DNS请求、网站TCP等数据包均发向该网关；浏览器IO等待期间OS切换运行系统中其他进程；浏览器将HTTP头中解析出的cookie保存到文件系统；根据网页内容发起更多的HTTP请求获取图片、运行内嵌的javascript脚本等；将网页按照HTTP 头的指示缓存；将网址加入浏览历史保存到文件系统；浏览器整个处理过程中，运行库和OS对内存做相应分配释放，磁盘做相应的IO操作；。。。