



人人网2015研发笔试卷C

一. 单项选择题

1. 若 $12 \times 25 = 311$ 成立，则用的是几进制？

- A 7
- B 8
- C 9
- D 11

2.

某32位系统下, C++程序如下所示, sizeof 的值应为？

```
char str[] = "http://www.renren.com" (长度为21)
char *p = str;
```

请计算

```
sizeof (str) = ? (1)
sizeof (p) = ? (2)
void Foo (char str[100]){
    sizeof (str) = ? (3)
}
void *p = malloc (100);
sizeof (p) = ? (4)
```

- A 22, 22, 100, 100
- B 4, 4, 4, 4
- C 22, 4, 4, 4
- D 22, 4, 100, 4

3. 有字符序列 (Q,H,C,Y,P,A,M,S,R,D,F,X), 新序列 (F,H,C,D,P,A,M,Q,R,S,Y,X) 是下列 () 排序算法一趟扫描结果。

- A 堆排序
- B 快速排序
- C 希尔排序
- D 冒泡排序

4. 关于排序算法的以下说法，正确的是？

- A 快速排序的平均时间复杂度为 $O(n \log n)$ ，最坏时间复杂度为 $O(n \log n)$
- B 堆排序的平均时间复杂度为 $O(n \log n)$ ，最坏时间复杂度为 $O(n^2)$



- ☐ C 冒泡排序的平均时间复杂度为 $O(n^2)$ ，最坏时间复杂度为 $O(n^2)$
- ☐ D 归并排序的平均时间复杂度为 $O(n\log n)$ ，最坏时间复杂度为 $O(n^2)$

5. 假设要存储一个数据集，数据维持有序，对其的操作只有插入、删除和顺序遍历，综合存储效率和运行速度，下列哪种数据结构是最适合的是？

- ☐ A 数组
- ☐ B 链表
- ☐ C 哈希表
- ☐ D 队列

6. 设有 n 个关键字具有相同的Hash函数值，则用线性探测法把这 n 个关键字映射到Hash表中需要做几次线性探测？

- ☐ A n^2
- ☐ B $n*(n+1)$
- ☐ C $n*(n+1)/2$
- ☐ D $n*(n-1)/2$

7. 数据库事务正确执行的四个基本要素不包括？

- ☐ A 隔离性
- ☐ B 持久性
- ☐ C 强制性
- ☐ D 一致性

8. 下列的进程状态变化中，哪些是不可能发生的？

- ☐ A 运行→就绪
- ☐ B 运行→等待
- ☐ C 等待→运行
- ☐ D 等待→就绪

9. 以下哪些方式/命令不可以查看某IP是否可达？

- ☐ A telnet
- ☐ B ping
- ☐ C tracert
- ☐ D top

10. 当用一台机器作为网络客户端时，该机器最多可以保持多少个到服务端的连接？

- ☐ A 1
- ☐ B 少于1024
- ☐ C 少于65535



D 无限制

二. 填空题

11. 假设网络带宽是128MB/s, 网络单向延时为100ms, 1000个客户端(单线程)同时向服务器传输64KB大小的文件, 每个请求大小为64KB, 服务器磁盘并发写入速度30MB/s, 在传输过程中, 服务器吞吐量为 ()MB/S , 单个请求响应时间为 ()ms

12. 由权值分别为3, 8, 6, 2, 5的叶子结点生成一棵哈夫曼树, 它的带权路径长度为()

三. 问答题

13. 给定一棵二叉树, 求各个路径的最大和, 路径可以以任意节点作为起点和终点。

比如给定以下二叉树:

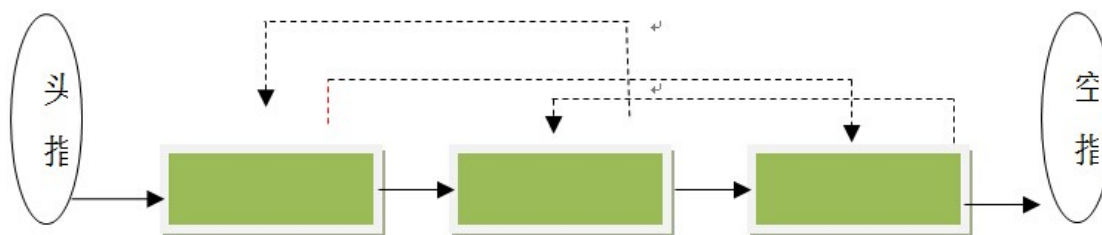
```
    2
   /\
  5 3
```

返回10。

```
/**
 * Definition for binary tree
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
int maxPathSum(TreeNode *root)
```

14.

有一个链表, 其中每个对象包含两个指针p1, p2, 其中指针p1指向下一个对象, 指针p2也指向一个对象, 沿p1可以像普通链表一样完成顺序遍历, 沿p2则可能会有重复。一种可能的例子如下, 其中实线箭头是p1, 虚线箭头是p2:



问题: 设计函数, 翻转这个链表, 并返回头指针。链表节点的数据结构如下:

```
struct Node{
    Node * p1;
    Node * p2;
    int data;
```



```
};
```

函数定义如下：

```
Node * revert(Node* head);
```

15. 编辑距离，又称Levenshtein距离，是指两个子串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。请尝试写出一个算法来计算两个字符串的编辑距离，并计算其复杂度？在某些应用场景下，替换操作的代价比较高，假设替换操作的代价是插入和删除的两倍，算法该如何调整？



技术QQ群：157594705



微信：www_nowcoder_com



微博：<http://www.weibo.com/nowcoder>

[登录牛客网](#)，参与以上题目讨论，查看更多笔试面试题