



人人网2015研发笔试卷E

一. 单项选择题

1. 34^{17} (34的17次方) 对6取余, 结果是多少?

- ☐ A 2
- ☐ B 3
- ☐ C 4
- ☐ D 5

正确答案: C

2. 有如下算式成立, $13*7=88$, 是采用 () 进制计算的。

- ☐ A 14
- ☐ B 13
- ☐ C 12
- ☐ D 11

正确答案: B

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

- ☐ A 希尔排序
- ☐ B 快速排序
- ☐ C 堆排序
- ☐ D 冒泡排序

正确答案: A

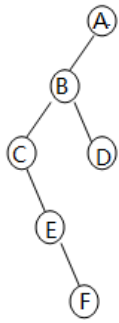
4. 二叉排序树中的最小值在二叉排序树的何处?

- ☐ A 只能在根节点
- ☐ B 只能在叶子节点
- ☐ C 可能在叶子节点, 也可能在根节点, 也可能在只有右孩子的父节点
- ☐ D 可以在任何节点

正确答案: C

5.

一棵树用左儿子右兄弟表示法呈如下结构, 请问这棵树原先结构前序遍历是?



- A ABFEDC
- B ABCEFD
- C ABDCEF
- D ADCEFB

正确答案: B

6. 一个含有 n 个顶点和 e 条边的简单无向图，在其邻接矩阵存储结构中共有（）个零元素。

- A e
- B $2e$
- C n 的2次方- e
- D n 的2次方- $2e$

正确答案: D

7. 下面程序中，输出是什么？

```
int fun(int x){
    int count = 0;
    while(x){
        count++;
        x = x &(x-1)
    }
    return count;
}
int main(){
    cout << "fun(2015)=" << fun(2015)<<endl;
}
```

- A fun(2015)=11
- B fun(2015)=10
- C fun(2015)=9
- D fun(2015)=8

正确答案: B



8. 若系统中有五台绘图仪，有多个进程均需要使用两台，规定每个进程一次仅允许申请一台，则至多允许（ ）个进程参与竞争，而不会发生死锁。

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

正确答案：C

9. 通过文件名存取文件时，文件系统内部的操作过程是通过？

- A 文件在目录中查找文件数据存取位置。
- B 文件名直接找到文件的数据，进行存取操作。
- C 文件名在目录中查找对应的i节点，通过i节点存取文件数据。
- D 文件名在目录中查找对应的超级块，在超级块查找对应i节点，通过i节点存取文件数据

正确答案：C

10. 以下哪个协议不是无状态协议？

- A TCP协议
- B HTTP协议
- C UDP协议
- D FTP协议

正确答案：A

二. 填空题

11. 12个元素的排序数组进行二分查找，每个元素被查找的概率是相等的，平均比较次数为()。

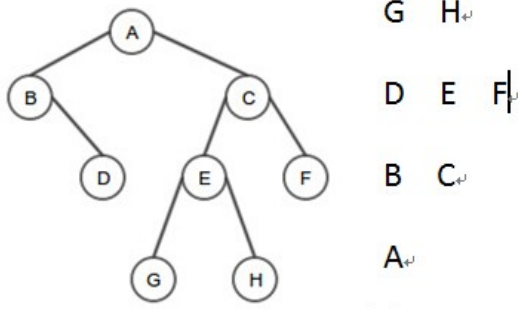
正确答案：37/12

12. $(a_1+a_2+a_3+\dots+a_n)/b$ 与 $a_1/b+a_2/b+\dots+a_n/b$ （除法为整除）最大差值为()。

正确答案：n-1

三. 问答题

13. 有如下图所示(左)的一棵二叉树，请设计一种遍历方式，使得按照如下方式（右）输出各个元素：（从下到上，从右到左输出，要求每层之间换行，同行元素之间用tab分割，写出完整代码）。



正确答案:

```
// 逆序 BFS 搜索
void reverseBFS(Tree *t)
{
    if(NULL == t)
    {
        return;
    }
    vector<vector<char>> vv;
    queue<Tree*> que;
    que.push(t);
    int length = 1;
    while(!que.empty())
    {
        vector<char> v;
        while(length != 0)
        {
            Tree *t = que.front();
            //cout<<t->data<<endl;
            que.pop();
            if(t->left)
            {
                que.push(t->left);
            }
            if(t->right)
            {
                que.push(t->right);
            }
            v.push_back(t->data);
            v.push_back(' ');
            length--;
        }
        v.push_back('\n');
        vv.push_back(v);
        length = que.size();
    }
    //cout<<vv.size()<<endl;
    for (int i = vv.size() - 1; i >= 0; i--)
    {
        vector<char> v = vv[i];
        for(int j = 0; j < v.size(); j++)
        {
            cout<<v[j];
        }
    }
}
```

14. 某星球上出现了一种怪物，这种怪物是单亲繁殖，从出生起第3个月起每个月就能繁衍一批后代共m个，但是这种怪物很短命，生存第5个月后会毙命。目前该星球有一个这样的怪物，请编写程序计算n个月后怪物的总数。(这里我们假定第5个月怪物繁衍后再毙命)

正确答案:



```
#include <iostream>
using namespace std;
int sum(int n,int m)
{
    int res=1;
    int A[100];
    A[0]=1;
    A[1]=0;
    A[2]=0;
    if(n<3)
        return res;
    for(int i=3;i<=n;i++)
    {
        A[i]=(res-A[i-1]-A[i-2])*m;
        res+=A[i];
        if(i>=5)
            res-=A[i-5];
    }
    return res;
}
int main()
{
    int n,m;
    cin>>n>>m;
    cout<<sum(n,m)<<endl;
    return 0;
}
```

15. 有一个二叉树，节点全部为整数，如何找到一个子树，它所有节点的和最大？要求编程实现。

正确答案：
使用递归方法，可以使用前序遍历，首先分别计算左右子树各自的子树和，然后记录目前最大的；再加入当前的父节点，再计算父节点开头的子树是否最大；该层递归上去即可。

```
public class MaxSumSubTree {
    class TreeNode{
        TreeNode left,right;
        int tag;
        TreeNode(int tag){
            this.tag = tag;
        }
    }

    private TreeNode maxRoot = new TreeNode(0);
    public int find(TreeNode root){
        if(root==null){
            return 0;
        } else{
            int lSum = find(root.left);
            int rSum = find(root.right);
            if(maxRoot.tag<lSum)
                maxRoot = root.left;
            if(maxRoot.tag<rSum)
                maxRoot = root.right;
            return root.tag+lSum+rSum;
        }
    }
}
```

16. 一般在大型系统中，都会为每个资源分配一个唯一的ID，在大型系统中这个并非易事，目前人人网一天产生新鲜事在千万量级，现在由你来设计一个产生新鲜事ID的模块。要求写出解题思路和伪代码。

拿分法宝：

- 1) 新鲜事ID绝对不能重复
- 2) 你可以借助DB等辅助工具，提供InsertDB, UpdateDB, QueryDB三API供你使用，假设访问DB不会有异常。
- 3) 高并发情况要考虑，提供Lock, Unlock两个API供你使用。
- 4) 要求写出解题思路和伪代码出来。

正确答案：



技术QQ群: 379386529



微博: <http://www.weibo.com/nowcoder>



微信

登录牛客网, 参与以上题目讨论, 查看更多笔试面试题