1. [单选题]

下面关于B+树的叙述中，错误的是\_\_\_\_\_\_\_\_。

是多路平衡树

可用于文件的索引结构

可进行顺序查找

关键字指向实际数据块

2. [单选题]

以下关于贝叶斯网络和马尔科夫网络说法正确的是

贝叶斯网络是有向图， 而马尔科夫网络是无向图

贝叶斯网络和马尔科夫网络都是无向图， 只是网络结构差别

贝叶斯网络和马尔科夫网络都是有向图， 只是网络结构差别

贝叶斯网络为无向图， 而马尔科夫网络是有向图

3. [单选题]

在设计模式中，应该优先使用()关系从而实现复用。

委派

继承

创建

都不对

4. [单选题]

在一个C类地址段内，需要将网络划分为 7个子网，每个子网有15个主机，则将使用哪个子网掩码 （）

255.255.255.224

255.255.224

255.255.255.240

都不是

5. [单选题]

以下哪个方法不是用于模型选择的

交叉验证

AIC

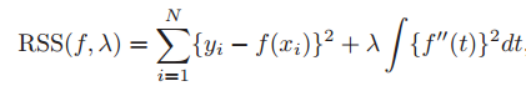
BIC

维特比算法

6.

[单选题]

关于以下目标函数说法错误的是：



当λ为无穷大的时候， f(x)为线性函数

当λ为0， 则f(x)为任意能完全拟合样本点的函数

对于一般的λ而言，存在最优解，最优解为自然三次样条曲线（natural cubic spline）

对于一般的λ而言， 存在最优解， f(x)为线性函数

7. [单选题]

Linux下哪个命令可以用于判断host1主机是否能够访问host2主机的端口

ping

ifconfig

telnet

netstat

8[单选题]

以下关于操作系统，说法错误的是（）

用管程实现进程同步时，管程中的过程是不可中断的

多道程序的执行失去了封闭性和再现性，因此多道程序系统不需要封闭性和再现性

使用SPOOLING技术可以实现虚拟设备

当 CPU 处于管态时，它可以执行计算机系统中的全部指令

9. [单选题]

有一个算法的递推关系式为：T(n) = 9 T(n / 3) + n，则该算法的时间复杂度为（）(^符号是幂的意思)

O(n^3)

O(nlogn)

O(n)

O(n^2)

10. [单选题]

由下面5个点:1,1,2,3,5构成的哈夫曼树的带权路径长度为（）

23

24

25

26

11. [编程题] 青草游戏

时间限制：1秒

空间限制：32768K

牛牛和羊羊都很喜欢青草。今天他们决定玩青草游戏。  
最初有一个装有n份青草的箱子,牛牛和羊羊依次进行,牛牛先开始。在每个回合中,每个玩家必须吃一些箱子中的青草,所吃的青草份数必须是4的x次幂,比如1,4,16,64等等。不能在箱子中吃到有效份数青草的玩家落败。假定牛牛和羊羊都是按照最佳方法进行游戏,请输出胜利者的名字。

**输入描述:**

输入包括t+1行。

第一行包括一个整数t(1 ≤ t ≤ 100),表示情况数.

接下来t行每行一个n(1 ≤ n ≤ 10^9),表示青草份数

**输出描述:**

对于每一个n,如果牛牛胜利输出"niu",如果羊羊胜利输出"yang"。

**输入例子1:**

3

1

2

3

**输出例子1:**

niu

yang

niu

12. [编程题] 无聊的牛牛和羊羊

时间限制：1秒

空间限制：32768K

牛牛和羊羊非常无聊.他们有n + m个共同朋友,他们中有n个是无聊的,m个是不无聊的。每个小时牛牛和羊羊随机选择两个不同的朋友A和B.(如果存在多种可能的pair(A, B),任意一个被选到的概率相同。),然后牛牛会和朋友A进行交谈,羊羊会和朋友B进行交谈。在交谈之后,如果被选择的朋友之前不是无聊会变得无聊。现在你需要计算让所有朋友变得无聊所需要的时间的期望值。

输入描述:

输入包括两个整数n 和 m(1 ≤ n, m ≤ 50)

输出描述:

输出一个实数,表示需要时间的期望值,四舍五入保留一位小数。

输入例子1:

2 1

输出例子1:

1.5

13. [编程题] 幸运子序列

时间限制：1秒

空间限制：32768K

牛牛得到一个长度为n的整数序列V,牛牛定义一段连续子序列的幸运值为这段子序列中最大值和次大值的异或值(次大值是严格的次大)。牛牛现在需要求出序列V的所有连续子序列中幸运值最大是多少。请你帮帮牛牛吧。

输入描述:

第一行一个整数n,即序列的长度。(2<= n <= 100000)

第二行n个数，依次表示这个序列每个数值V[i], (1 ≤ V[i] ≤ 10^8)且保证V[1]到V[n]中至少存在不同的两个值.

输出描述:

输出一个整数,即最大的幸运值

输入例子1:

5

5 2 1 4 3

输出例子1:

7

答案：DAAAD

DCBDC

11.#include<stdio.h>

int dp[10],n,x,i,j;

int main(){

for(i=1;i<10;i++){

int flag=0;

for(j=1;j<=i;j\*=4)

if(dp[i-j]==0) flag=1;

dp[i]=flag;

}

for(scanf("%d",&n),i=0;i<n;i++)

scanf("%d",&x),printf("%s\n",dp[x%10]?"niu":"yang");

}

12.#include<stdio.h>

double f[100];

int main()

{

int yes,no,all;

scanf("%d%d",&yes,&no);

all=yes+no;

double pos=(double)all\*(all-1)/2;

for(int i=1;i<=no;i++)

{

int n=i,y=all-i;

double c1=((double)y\*(y-1)/2)/pos;

double c2=(double)y\*n/pos;

double c3=((double)n\*(n-1)/2)/pos;

f[i]+=c2\*(f[i-1]+1);

if(i>=2) f[i]+=c3\*(f[i-2]+1);

f[i]+=c1;

f[i]=f[i]/(1-c1);

}

printf("%.1lf",f[no]);

return 0;

}

//列出公式后计算就好了

13.#include<stdio.h>

#include<stack>

#include<algorithm>

using namespace std;

int main(){

int n,i,res=0,x;

stack<int> s;

for(scanf("%d",&n),i=0;i<n;i++){

scanf("%d",&x);

while(s.size()&&s.top()<=x)

res=max(res,x^s.top()),s.pop();

if(s.size()) res=max(res,x^s.top());

s.push(x);

}

printf("%d\n",res);

}