

有个特殊的餐厅, 对客人的要求是如果要离开餐厅一定要看下餐厅内有没有比你迟进来的人, 一定要所有比你迟进来的人离开后你才能离开, 有一天甲, 乙, 丙, 丁四个客人先后进入了这家餐厅, 那么他们离开的顺序不可能是:

正确答案: E 你的答案: 空 (错误)

丙, 乙, 甲, 丁

甲, 乙, 丙, 丁

乙, 甲, 丙, 丁

乙, 丙, 甲, 丁

丁, 丙, 甲, 乙

丁, 丙, 乙, 甲

在关系型是数据库中, 有两个不同的事务同时操作数据库中同一表的同一行, 不会引起冲突的是:

正确答案: F 你的答案: 空 (错误)

其中一个 DELETE 操作, 一个是 SELECT 操作

其中两个都是 UPDATE

其中一个是 SELECT, 一个是 UPDATE

其中一个 SELECT

其中一个是 DELETE, 另一个是 UPDATE

两个都是 DELETE

众所周知我们所在的宇宙的质能公式是 $E=mc^2$, 其中 c 是真空中光速。和我们的宇宙平行的另一个宇宙 meta, 研究显示他们使用的质能公式是 $E=(2+\sqrt{3})m$, 当一个物体的质量很大的时候, 对应的能量 E 非常大, 数据也非常的长。但 meta 宇宙里面的智慧生物非常的懒, 他们只愿意把 E 取整, 然后记录对应的能量 E 的最后一位整数, 比如 $m=0$ 时, 他们会记录 1, $m=1$ 时, 他们会记录 3, $m=2$ 时, 他们会记录 3。现在请问当 $m=80$ 时, 他们会记录多少?

正确答案: C 你的答案: 空 (错误)

1

2

3

4

5

6

页高速缓存是 Linux kernel 使用的主要的磁盘缓存技术。它允许系统把存放在磁盘上的一些数据保留在内存中, 以便减少对磁盘的访问。进程对页高速缓存区中的数据修改之后, 数据页被标记为“脏数据”在下列哪些条件下, 脏数据不会被写入磁盘?

正确答案: B E 你的答案: 空 (错误)

页高速缓存空间不足

突然断电

变脏以来, 太久没有过更新

通过系统调用 (`sync()`, `fsync()`, `fdatasync()`) 来强行对将对快设备的更新同步到磁盘

内存足够大

磁盘足够大

设一组初始关键字记录关键字为 (12,15,1,18,2,35,30,11), 则以 12 为基准记录的一趟快速排序结束后的结果为

正确答案: D 你的答案: 空 (错误)

11, 1, 2, 12, 35, 18, 30, 15

11, 2, 1, 12, 15, 18, 35, 30

1, 2, 11, 12, 15, 18, 35, 30

11, 2, 1, 12, 18, 35, 30, 15

1, 2, 11, 12, 15, 18, 30, 35

以上都不是

多项式 $P(X)=a+bx+cx^2+dx^3$, 对于任意 x , 计算 $P(X)$ 中最少需要用到乘法操作的次数是多少?

正确答案: A 你的答案: 空 (错误)

3

4

5

6

1

2

有一个班 31 人, 女同学 15 人, 男同学 16 人, 现在要玩一个结组游戏, 每组由男女两名同学构成, 每个同学最多只能在一个组。结组的同学可以到老师那里领 100 元钱, 然后按照预先分配的协议进行分配, 钱最小可分单元为 1 元。未能结组的同学无法领到钱, 不允许在组之间传递利益。那么一下命题正确的是:

正确答案: A 你的答案: 空 (错误)

男生和女生可以得到一样多的钱

男生最多得 49 元

男生最多得 47 元

男生最多得 46 元

男生最多得 1 元

男生最多得 0 元

现代的企业是建立在大规模协作的基础上的, 员工之间, 团队之间, 部门之间, 企业之间的协作都是成功的重要因素。好的企业在协作上是高效的。以下说法中不合适的是 ()

正确答案: B 你的答案: 空 (错误)

一个项目能容纳的人员是有限的, 当增加到一定规模项目进度反而会变慢。

一个项目协作为办了信息对称, 多方参与的情况下直接召集多方在一起开会就能协调好协作建立的条件包括互补和共赢

能力结构类似的成员之间较多样互补型员工之间更容易产生竞争关系

协作中的权利和责任应当相称

如果有可能的话, 信息交互较多的事务更合适在一个团队内或有一个人完成, 相较于进行分工。

以下程序的运行结果是?

```

public class TestThread {
    public static void main(String args[]) {
        Runnable runner = new Runnable() {
            @Override
            public void run() {
                System.out.print("foo");
            }
        };
        Thread t = new Thread(runner);
        t.run();
        System.out.print("bar");
    }
}

```

正确答案: A 你的答案: 空 (错误)

foobar

barfoo

foobar 或者 barfoo 都有可能

Bar

Foo

程序无法正常运行

输入图片大小为 200×200 , 依次经过一层卷积 (kernel size 5×5 , padding 1, stride 2), pooling (kernel size 3×3 , padding 0, stride 1), 又一层卷积 (kernel size 3×3 , padding 1, stride 1) 之后, 输出特征图大小为:

正确答案: C 你的答案: 空 (错误)

95

96

97

98

99

100

一个二叉树有 100 个子节点数为 2 的节点, 100 个子节点数为 1 的节点, 那么个子节点数为 0 的节点 (叶节点) 的个数为:

正确答案: A 你的答案: 空 (错误)

101

100

200
300
99
1

某种类型的双核 CPU 的性能提升了 $1/3$ ，假定该提升是通过对每条指令缩短执行时间实现的，那么它每条指令执行时间缩短了 ()。

正确答案: A 你的答案: 空 (错误)

$1/4$
 $1/8$
 $1/3$
 $1/6$
 $1/5$
 $1/2$

一个 map-reduce 任务由 m 个 mapper 和 r 个 reducer 构成，计算的效率可以认为正比于 mr 的乘积（数据管道的个数），在限定任务的 mr 乘积约等于 10000 的情况下，假定每个 mapper 和每个 reducer 的成本分别为 1 和 7，那么最佳的资源分配最接近于以下哪个方案？

正确答案: C 你的答案: 空 (错误)

mapper100 个，reducer100 个
Mapper200 个，reducer50 个
Mapper264 个，reducer38 个
Mapper316 个，reducer32 个
Mapper500 个，reducer20 个
Mapper1000 个，reducer10 个

如果你有相关经验，很多景点的餐馆商铺经营方式很有趣。以下描述错误的是：

正确答案: F 你的答案: 空 (错误)

由于景点的大部分顾客是一次性的，因此商铺的信用在其他条件相同时可能更低

景点的餐馆为了招揽顾客使用托儿会比居民区的餐馆使用托儿效果好

景点常常卖一些当地特产，比如青岛的海边会有卖贝壳的，这些贝壳产自附近的海域

店铺使用托儿的有效原因是，人们常常做出多数人做出的选择，而忽视自己自然状态下的决策

一些景点在出口位置上安排一个商店，商店内的通道曲折，为的是顾客多花些时间看东西

景点内一些玩射箭的场所经营状况会比在居民区附近设置的类似场所好，原因是景点内的游人玩起来更在状态

一个机器人玩抛硬币的游戏，一直不停的抛一枚不均匀的硬币，硬币有 A,B 两面，A 面的概率为 $3/4$ ，B 面的概率为 $1/4$ 。问第一次出现连续的两个 A 年的时候，机器人抛硬币的次数的期望是多少？

正确答案: F 你的答案: 空 (错误)

$9/4$
 $11/4$
 $15/4$
4

5

28/9

小 a 和小 b 一起玩一个游戏，两个人一起抛掷一枚硬币，正面为 H，反面为 T。两个人把抛到的结果写成一个序列。如果出现 HHT 则小 a 获胜，游戏结束。如果 HTT 出现则小 b 获胜。小 a 想问一下他获胜的概率是多少？

正确答案: C 你的答案: 空 (错误)

3/4

1/2

2/3

5/9

1/3

1/4

假定某同学使用 Naive Bayesian (NB) 分类模型时，不小心将训练数据的两个维度搞重复了，那么关于 NB 的说法中正确的是：

正确答案: B D 你的答案: 空 (错误)

这个被重复的特征在模型中的决定作用会被加强

模型效果相比无重复特征的情况下精确度会降低

如果所有特征都被重复一遍，得到的模型预测结果相对于不重复的情况下的模型预测结果一样。

当两列特征高度相关时，无法用两列特征相同时所得到的结论来分析问题

NB 可以用来做最小二乘回归

以上说法都不正确

以下哪个行为，不会明显加剧客户端运行过程中的卡顿：

正确答案: C 你的答案: 空 (错误)

在主线程集中处理耗时的操作

在子线程集中处理耗时的操作

在其它进程集中处理耗时的操作

提高后台线程的优先级

降低主线程的优先级

页面存在多个重叠显示的控制件

以下程序的输出是：

```

#include <iostream>

using namespace std;

unsigned int f(unsigned int n) {
    if (n == 0 || n == 1) {
        return 1;
    }
    return f(n-1) + f(n-2);
}

void count(int n) {
    unsigned int tmp = n - ((n >>1) &033333333333) - ((n >>2) &011111111111);
    std::cout << ((tmp + (tmp >>3)) &030707070707) %63 << std::endl;
}

int main()
{
    count(f(7));
    count(f(9));
    return 0;
}

```

学校图书馆共有 300 万册图书，想统计其中 Computer， Science， 计算机， 科学这几个词出现的次数， 并按照自然年度分类， 如 2016 年出版的书籍中这几个词各自出现的次数， 2015 年—依次类推。

```

HashMap<String, Integer> map = new HashMap<>();
map.put("Computer",0);
map.put("Science",0);
map.put("计算机",0);
map.put("科学",0);

```

```

public HashMap<String, Integer> calculateTimes(HashMap<String, Integer> map,String
bookName){
    if(bookName.length() <= 0){

```

```

        return map;
    }

    if(bookName.contains("Computer")){
        map.put("Computer", map.get("Computer")+1);
    }else if(bookName.contains("Science")){
        map.put("Computer", map.get("Science")+1);    }else if(bookName.contains("计算机
")){
        map.put("计算机", map.get("计算机")+1);
    }else if(bookName.contains("科学")){
        map.put("科学", map.get("科学")+1);
    }
    return map;
}

}设计接口并且实现一个多线程安全的堆，并且要求可以删除堆中任意元素（非堆顶元素），
要求尽量高效，假设已有标准的 mutex 实现可以使用。
deleteElement(int[] heap, int index, int len) {    mutex.lock();
    heap[index] = heap[len-1];
    len--;
    while(2*index + 1 < len && (heap[index] > heap[2*index+1]|| heap[index] >
heap[2*index +2])) {
        int swapi = heap[2*index+1] < heap[2*index +2]) ? 2*index+1 : 2*index+2;
        swap(heap,index,swapi);
        index = swapi;
    }
    mutex.unlock();
}
}

```