



下面线程间的通讯机制中, 关键路径上不会产生系统调用从而减少用户态到内核态的上下文切换的是\_\_\_\_\_。

正确答案: A 你的答案: 空 (错误)

pthread\_spin\_lock  
pthread\_mutex  
信号量  
pthread\_rwlock  
管道  
消息队列

有一个类 B 继承自类 A, 他们数据成员如下:

```
1  class A {  
2  ...  
3  private:  
4      int &a;  
5  };  
6  class B : public A {  
7  ...  
8  private:  
9      int a;  
10 public:  
11     const int b;  
12     A c;  
13     static const char* d;  
14     A* e;  
15 };
```

则构造函数中, 成员变量一定要通过初始化列表来初始化的是\_\_\_\_\_。

正确答案: A 你的答案: 空 (错误)

b c  
b c e  
b c d e  
c e  
b d  
b e

一人以 d 元在淘宝买入 n 套《三体》, d 为正整数, 其中两套他以成本一半送给朋友, 余下的每套高于原价 8 元卖出, 加上送给朋友的两套, 如果全部利润是 72 元, 那么 n 最小可能是\_\_\_\_\_。

正确答案: E 你的答案: 空 (错误)

18  
17





15  
13  
12  
10

进程 P1 和 P2，资源 A 和 B，进程对资源的操作定义如下：读取资源 A: rA 读取资源 B: rB 写入资源 A: wA 写入资源 B: wB 同时规定，进程读取某一资源前，必须加上共享锁，写入某一资源前必须加上排他锁；资源在被进程加共享锁时，可以被其他进程继续加共享锁，但不能加排他锁；资源在被进程加排他锁时，其他进程无法加锁。进程完成全部操作后释放锁。进程 P1 和进程 P2 并发执行，如下哪种运行序列可能会出现死锁\_\_\_\_\_。

正确答案: D 你的答案: 空 (错误)

P1 (rA->rB) , P2 (rA->wB)  
P1 (rA->rB) , P2 (wA->rB)  
P1 (wB->rA) , P2 (rA->rB)  
P1 (wB->rA) , P2 (wA->rB)  
P1 (wA->wB) , P2 (wA->wB)  
P1 (wA->rB) , P2 (wA->wB)

假设基准值为数组首元素的快速排序，要使得数组满足非降序排列，下列数据分布导致快排算法效率最低的是\_\_\_\_\_。

正确答案: E 你的答案: 空 (错误)

2-6-3-7-5-1-4  
6-2-3-5-4-1-7  
7-5-3-2-4-1-6  
1-5-7-2-4-6-3  
1-2-3-4-5-6-7  
4-1-3-7-5-6-2

二进制值 0.00110011001100110011001100110011 转换成 10 进制后最接近\_\_\_\_\_。

正确答案: C 你的答案: 空 (错误)

0.1  
0.15  
0.2  
0.22  
0.25  
0.3

将整数数组 (7-6-3-5-4-1-2) 按照堆排序的方式原地进行升序排列，请问在第一轮排序结束之后，数组的顺序是\_\_\_\_\_。

正确答案: C 你的答案: 空 (错误)

2-6-3-5-4-1-7



6-2-3-5-4-1-7  
6-5-3-2-4-1-7  
1-5-3-2-4-6-7  
5-4-3-2-1-6-7  
5-1-3-2-4-6-7

函数 func 的定义如下:

```
1 void func(const int& v1, const int& v2)
2 {
3     std::cout << v1 << ' ';
4     std::cout << v2 << ' ';
5 }
```

在 Visual Studio 开发环境下, 以下代码输出结果为\_\_\_\_\_。

```
1 int main (int argc, char* argv[])
2 {
3     int i=0;
4     func(++i, i++);
5     return 0;
6 }
```

正确答案: D 你的答案: 空 (错误)

0 1  
1 2  
2 1  
2 0  
0 2  
1 0

将森林转换为对应的二叉树, 若在二叉树结点中, 结点 m 是结点 n 的双亲结点的双亲结点, 则在原来的森林中, m 和 n 可能具有的关系是\_\_\_\_\_。

- 1.父子关系
- 2.m 的双亲结点与 n 的双亲结点是兄弟关系
- 3.兄弟关系

正确答案: B 你的答案: 空 (错误)

只有 3  
1 和 3  
1 和 2  
1、2 和 3  
只有 1  
只有 2

在 1,2,3,.....,999,1000 中, 有\_\_\_\_\_个数各位数之和模 10 等于 0。

正确答案: A 你的答案: 空 (错误)





99  
100  
101  
150  
200  
201

两人在一个  $n$  个点的无向完全图上进行游戏，每次可以选择当前图中两个端点度数奇偶性相同的边删除，谁不能操作谁输，则在  $n=1,2,3,\dots,9,10$  中，有\_\_\_\_\_个图先手有必胜策略。

正确答案: D 你的答案: 空 (错误)

2  
3  
4  
5  
6  
7

以下程序输出结果是\_\_\_\_\_。

```
1  class A
2  {
3  public:
4      virtual void func(int val = 1)
5      { std::cout<<"A->"<<val <<std::endl;}
6      virtual void test()
7      { func();}
8  };
9  class B : public A
10 {
11 public:
12     void func(int val=0)
13 {std::cout<<"B->"<<val <<std::endl;}
14 };
15 int main(int argc ,char* argv[])
16 {
17     B*p = new B;
18     p->test();
19     return 0;
20 }
```

正确答案: B 你的答案: 空 (错误)

A->0  
B->1



A->1

B->0

编译出错

以上都不对

关于线程和进程，不正确的描述是\_\_\_\_\_。

正确答案: D 你的答案: 空 (错误)

进程的隔离性要好于线程

线程在资源消耗上通常要比进程轻量

不同进程间不会共享逻辑地址空间

同一个进程的线程之间共享内存，包括堆和栈

进程间有途径共享大量内存中的数据

线程间通讯可以通过直接访问全局变量，或者使用进程间通讯的机制 (IPC)

如果我国希望在 30 年后实现经济翻番，并且假设我国经济增长速度每年都保持相同，那么至少需要每年保持\_\_\_\_\_增长速度才可实现目标。

正确答案: D 你的答案: 空 (错误)

1.5%

1.8%

2.0%

2.4%

3.3%

7.0%

下面是折半查找的实现，**data** 是按升序排列的数据，**x** 是查找下标，**y** 是查找的上标，**v** 是查找的数值，返回 **v** 在 **data** 的索引，若没找到返回-1。代码不正确是\_\_\_\_\_。

```
1 public int bsearch(int[] data, int x, int y, int v) {
2     int m;
3     while(x<y) { //1
4         m = x + (y-x)/2; //2
5         if(data[m] == v) return m; //3
6         else if(data[m] > v) y = m; //4
7         else x = m; //5
8     }
9     return -1; //6
10 }
```

正确答案: A D E 你的答案: 空 (错误)

1

2

3

4



5

6

如果下列的公式成立： $84 \times 148 = B6A8$ 。则采用的是\_\_\_\_进制表示的

正确答案: C 你的答案: 空 (错误)

15

11

12

14

16

以上都不对

有一个程序中有 A,B,C 三个线程同时对一个文件进行读写操作，其中的 A,B 是写进程只负责往里面写数据，C 是读线程，同时把读取的数据从文件中删除，A 线程单独写满文件需要 10 个小时，B 单独写程序需要 6 小时，C 线程需要 15 小时才能读取完整个文件，不考虑三个线程之间的相互影响的情况下现在\_\_\_\_小时才能写满文件

正确答案: A 你的答案: 空 (错误)

5

6

5.5

4.5

4.8

5.3

有甲、乙、丙三位员工分别负责前端、后端、数据、算法、测试、运维。每人负责两项。已知：数据和前端住在一起 甲是三人中最年轻的 前端和丙有空一起下棋 后端比算法年长，又比乙年轻 三人中最年长的住得最远 那么，三人分别负责\_\_\_\_\_。

正确答案: F 你的答案: 空 (错误)

甲-前端&测试；乙-算法&运维；丙-后端&数据

甲-后端&算法；乙-前端&运维；丙-测试&数据

甲-前端&运维；乙-测试&算法；丙-后端&数据

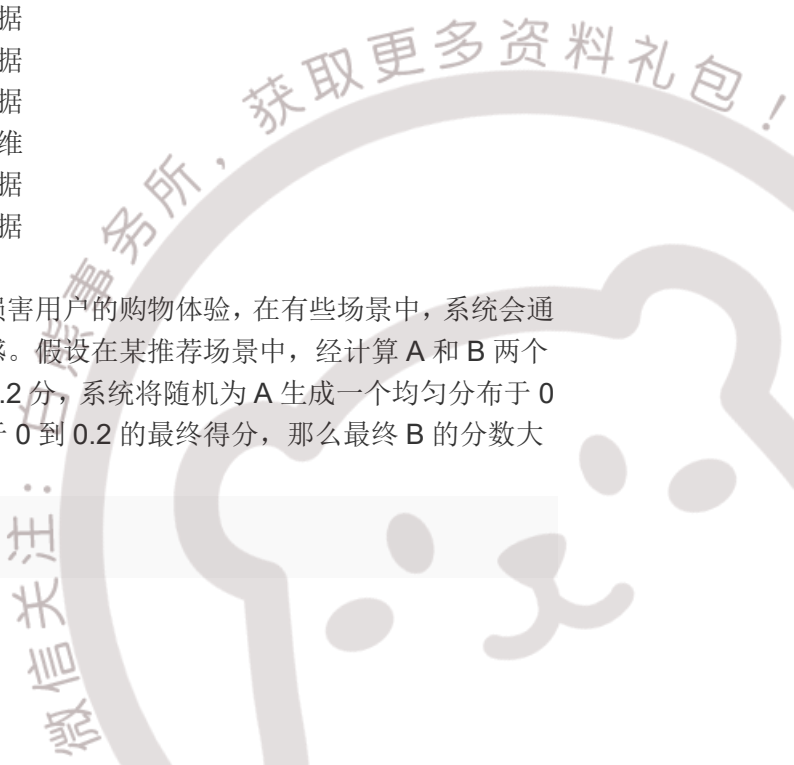
甲-算法&数据；乙-测试&前端；丙-后端&运维

甲-前端&算法；乙-后端&运维；丙-测试&数据

甲-前端&算法；乙-测试&运维；丙-后端&数据

商品推荐场景中过于聚焦的商品推荐往往会损害用户的购物体验，在有些场景中，系统会通过一定程度的随机性给用户带来发现的惊喜感。假设在某推荐场景中，经计算 A 和 B 两个商品与当前访问用户的匹配度分别为 0.8 和 0.2 分，系统将随机为 A 生成一个均匀分布于 0 到 0.8 的最终得分，为 B 生成一个均匀分布于 0 到 0.2 的最终得分，那么最终 B 的分数大于 A 的分数的概率为\_\_\_\_\_。

正确答案: B 你的答案: 空 (错误)





1/16

1/8

3/16

3/8

1/4

1/3

从 1,2,3,4...8,9 里任意选择一部分数(至少 1 个),能得到多少种不同的乘积

正确答案: C 你的答案: 空 (错误)

144

148.

152

156

160

164

