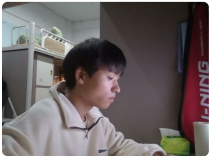


比特就业课105期Java方向笔试强训48天day16_10月27日-王世国-测评结果

考生信息  存在作弊行为

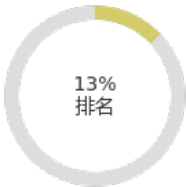


王世国
投递编号：91 | 学校：武汉轻工大学 | 邮箱：1477649017@qq.com | 职位：比特就业课105期Java2班
参考区域: 湖北省武汉市 (183.94.121.252) | 做题用时：01:39:59(2022-10-26 19:53:11开始答题，21:33:12交卷) |
作答设备：PC | 已同意诚信声明和隐私协议

89.44分 / 100分

在本次考试中，考生总成绩为89.44分/100分，评级为**A（排名前13%）**，编程能力**良好（1题通过，1题部分通过，分数排名前40%）**，编程思路**基本一致**，编程规范性**高**。该考生在本次考试中**存在**作弊行为，视频监控图片数目过少，有遮挡或关闭摄像头的嫌疑。

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	45.0	9	3	00:11:20	已阅
编程	44.44	1	40	01:25:27	已阅

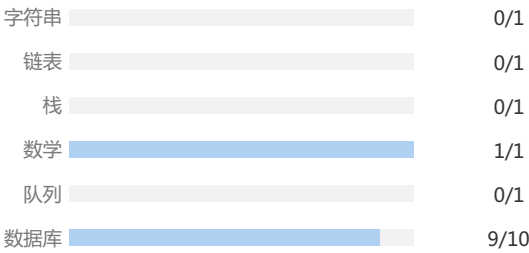
作弊风险

高风险

未开启摄像头

考生可能关闭摄像头，导致视频截图较少

知识点技能图谱



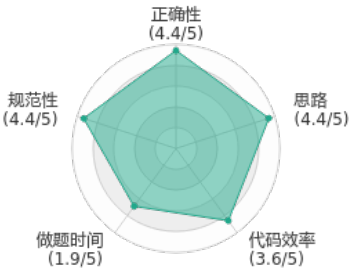
知识点	得分	正确题数
字符串	19.44	0
链表	19.44	0
栈	19.44	0
数学	25.0	1
队列	19.44	0
数据库	45.0	9

历史笔记记录

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
1	比特就业课105期+2022寒假班C1考试	20.0%	48.8/60	单选:30.0分 编程:18.75分	否	2022-03-29 11:16:18	2022-03-31 18:51:27
2	比特就业课105期+2022寒假班C2考试	66.0%	24.0/60	单选:24.0分 编程:0.0分	否	2022-04-11 14:12:23	2022-04-11 20:12:16
3	比特就业课 105期JavaSE考试	11.0%	56.0/60	单选:26.0分 编程:30.0分	否	2022-07-12 16:00:16	2022-07-13 15:48:42
4	比特就业课 105期java方向 数据结构考试	47.0%	50.0/60	单选:20.0分 编程:30.0分	否	2022-07-23 12:49:22	2022-07-25 09:56:28
5	比特就业课105期Java方向笔试强训48天day01_10月10日	36.0%	80.0/100	单选:40.0分 编程:40.0分	是，摄像头监控异常	2022-10-09 17:29:16	2022-10-09 21:57:25
6	比特就业课105期Java方向笔试强训48天day02_10月11日	13.0%	90.0/100	单选:40.0分 编程:50.0分	是，代码抄袭	2022-10-10 10:43:48	2022-10-10 21:13:15
7	比特就业课105期Java方向笔试强训48天day03_10月12日	1.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-11 10:40:53	2022-10-12 10:03:09
8	比特就业课105期Java方向笔试强训48天day04_10月13日	2.0%	95.0/100	单选:30.0分 不定项选择:15.0分 编程:50.0分	是，摄像头监控异常	2022-10-12 10:31:10	2022-10-12 21:14:05
9	比特就业课105期Java方向笔试强训48天day05_10月14日	12.0%	85.0/100	单选:35.0分 不定项选择:0.0分 编程:50.0分	是，摄像头监控异常 代码抄袭	2022-10-13 11:41:43	2022-10-14 11:33:24
10	比特就业课105期Java方向笔试强训48天day06_10月15日	13.0%	86.7/100	单选:25.0分 不定项选择:11.67分 编程:50.0分	是，摄像头监控异常	2022-10-14 10:59:38	2022-10-14 21:18:11
11	比特就业课105期Java方向笔试强训48天day07_10月17日	1.0%	100.0/100	单选:50.0分 编程:50.0分	是，摄像头监控异常	2022-10-16 16:46:53	2022-10-16 20:38:39
12	比特就业课105期Java方向笔试强训48天day08_10月18日	33.0%	87.5/100	单选:40.0分 编程:47.5分	是，摄像头监控异常	2022-10-17 16:18:42	2022-10-17 20:16:45
13	比特就业课105期Java方向笔试强训48天day09_10月19日	3.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-18 17:07:17	2022-10-19 11:02:55
14	比特就业课105期Java方向笔试强训48天day10_10月20日	58.0%	65.0/100	单选:40.0分 编程:25.0分	是，摄像头监控异常	2022-10-19 15:29:54	2022-10-19 21:44:34
15	比特就业课105期Java方向笔试强训48天day11_10月21日	2.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-20 16:46:09	2022-10-20 23:36:15
16	比特就业课105期Java方向笔试强训48天day12_10月22日	15.000001%	90.0/100	单选:40.0分 编程:50.0分	是，摄像头监控异常	2022-10-21 16:44:18	2022-10-21 22:49:24
17	比特就业课105期Java方向笔试强训48天day13_10月24日	31.0%	65.0/100	单选:40.0分 编程:25.0分	是，摄像头监控异常 代码抄袭	2022-10-22 16:07:25	2022-10-23 23:15:22
18	比特就业课105期Java方向笔试强训48天day14_10月25日	58.999996%	55.0/100	单选:30.0分 编程:25.0分	是，摄像头监控异常	2022-10-24 14:08:32	2022-10-24 23:42:01
19	比特就业课105期Java方向笔试强训48天day15_10月26日	42.0%	55.0/100	单选:30.0分 编程:25.0分	是，摄像头监控异常	2022-10-24 14:11:25	2022-10-25 22:58:05

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
----	------	----	-----	------	------	--------	------

编码能力



题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题1	100%	1	00:18:50	Java	54ms	11052K	良	良	1%
编程题2	78%	7	01:06:37	Java	38ms	10964K	良	良	40%

1 [单选题 | 平均分2.36分 | 41人正确/87人做题 | 用时 : 2分 | 得分 : 5.0 / 5.0

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：

- A 其中一个DELETE操作，一个是SELECT操作
- B 其中两个都是UPDATE
- C 其中一个SELECT，一个是UPDATE
- D 其中一个SELECT
- E 其中一个DELETE，另一个是UPDATE
- F 两个都是DELETE

两个会话，各自执行各自的操作

其中一个是select，另一个的话如果也是select那就没问题。但是如果是删除，修改，更新那就存在问题

两个操作之间存在冲突，就看各自执行之后会不会对对方的结果造成影响。

他的回答： F (正确)

正确答案： F

2 [单选题 | 平均分4.07分 | 70人正确/86人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

计算每位学生的多学科加总成绩的SQL是_____

- A select sum(score) as total,stud_name from [成绩表](nolock)
- B select count(score) as total,stud_name from [成绩表](nolock)
- C select sum(score) as total,stud_name from [成绩表](nolock) group by stud_name
- D select count(score) as total,stud_name from [成绩表](nolock) group by stud_name

他的回答： C (正确)

正确答案： C

3 [单选题 | 平均分1.36分 | 24人正确/88人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0

以下哪个不是与Mysql服务器相互作用的通讯协议()

- A TCP/IP
- B UDP
- C 共享内存
- D Socket

重点就在于MySQL的操作必须保证可靠性，所以UDP是不可以的

他的回答： C (错误)

正确答案： B

4 [单选题 | 平均分4.3分 | 74人正确/86人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

公司中有多个部门和多名职员，每个职员只能属于一个部门，一个部门可以有多个职员。则实体部门和职员间的联系是（ ）

- A 1：1联系

- B m : 1联系
C 1 : m联系
D m : n联系

他的回答： C (正确)

正确答案： C

参考答案：

两个实体集间的联系实际上是实体集间的函数关系，主要有一对一联系（ 1:1 ）、一对多联系（ 1:m ）、多对一联系（ m:1 ）、多对多联系（ m:n ）。对于每一个实体部门，都有多名职员，则其对应的联系为一对多联系（ 1:m ），答案选 C。

5 [单选题 | 平均分1.4分 | 25人正确/89人做题 | 用时：<1分 | 得分： 5.0 / 5.0

设有两个事务T1,T2,其并发操作如下所示,下面评价正确的是()

步骤	T1	T2
1	读A=100	
2		读A=100
3	A=A+10写回	
4		A=A-10写回

- A 该操作不能重复读
B 该操作不存在问题
C 该操作读"脏"数据
D 该操作丢失修改

T2修改的写回覆盖了T1的结果

他的回答： D (正确)

正确答案： D

6 [单选题 | 平均分3.84分 | 66人正确/86人做题 | 用时：<1分 | 得分： 5.0 / 5.0

mysql 数据库有选课表 learn(student_id int,course_id int),字段分别表示学号和课程编号, 现在想获取每个学生所选课程的个数信息,请问如下的 sql 语句正确的是()

- A select student_id,sum(course_id)from learn
B select student_id,count(course_id)from learn group by student_id
C select student_id,count(course_id)from learn
D select student_id,sum(course_id)from learn group by student_id

他的回答： B (正确)

正确答案： B

7 [单选题 | 平均分2.59分 | 45人正确/87人做题 | 用时：2分 | 得分： 5.0 / 5.0

如果事务T获得了数据项Q上的排它锁，则T对Q_____。

- A 只能读不能写
B 只能写不能读
C 既可读又可写
D 不能读不能写
- 排他锁，也就是只有当前获取到这把锁的事务可以进行操作

他的回答： C (正确)

正确答案： C

参考答案：

共享 (S)锁：多个事务可封锁一个共享页；任何事务都不能修改该页；通常是该页被读取完毕，S锁立即被释放。

排它 (X)锁：仅允许一个事务封锁此页；其他任何事务必须等到X锁被释放才能对该页进行访问；X锁一直到事务结束才能被释放。

更新 (U)锁：用来预定要对此页施加X锁，它允许其他事务读，但不允许再施加U锁或X锁；当被读取的页将要被更新时，则升级为X锁；U锁一直到事务结束时才能被释放。

8 [单选题 | 平均分3.56分 | 62人正确/87人做题 | 用时：<1分 | 得分： 5.0 / 5.0

在关系模型中，实现“表示了两个关系之间的相关联系”的约束是通过（ ）

- A 候选键
- B 主键
- C 外键
- D 超键

可以唯一表示一条记录的字段就可以是超键，候选键是最小的超键，也就是说候选键中任意再去除掉一个字段就不是超键了

他的回答： C (正确)

正确答案： C

9 [单选题 | 平均分3.22分 | 56人正确/87人做题 | 用时：<1分 | 得分： 5.0 / 5.0

学生关系模式S(S#,Sname,Sex,Age)，S的属性分别表示学生的学号、姓名、性别、年龄。要在表S中删除一个属性“年龄”，可选用的SQL语句是（ ）

- A ALTER TABLE S DROP Age
- B ALTER TABLE S 'Age'
- C UPDATE S Age
- D DELETE Age from S

他的回答： A (正确)

正确答案： A

10 [单选题 | 平均分3.74分 | 65人正确/87人做题 | 用时：<1分 | 得分： 5.0 / 5.0

下列选项中,不属于SQL约束的是:

- A UNIQUE
- B PRIMARY KEY
- C FOREIGN KEY
- D BETWEEN

他的回答： D (正确)

正确答案： D

11 ACM编程题 语言限制 [编程题 | 平均分21.84分 | 76人正确/87人做题 | 提交: 1 次 | 得分： 25.0 / 25.0

标题：完全数计算 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【完全数计算】

完全数（Perfect number），又称完美数或完备数，是一些特殊的自然数。

它所有的真因子（即除了自身以外的约数）的和（即因子函数），恰好等于它本身。

例如：28，它有约数1、2、4、7、14、28，除去它本身28外，其余5个数相加，1+2+4+7+14=28。

输入n，请输出n以内(含n)完全数的个数。

$$1 \leq n \leq 5 \times 10^5$$

数据范围：

输入描述：

输入一个数字n

输出描述：

输出不超过n的完全数的个数

示例1：

输入

1000

输出

3

代码片段						
功能实现			代码提交统计			代码执行统计
总通过率	TA的	平均				答案正确 : 1
	100%	87%	使用语言	Java		
	12/12	87%	做题用时	00:18:50	00:15:41	
	(100%)		提交次数	1	1	
基本测试用例通过率						
边缘测试用例通过率	8/8 (100%)	87%				

代码效率			代码规范及可读性		
	TA的	参考	代码规范得分		
运行时间	54ms	1s	4.2		
占用内存	11052K	32768K	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]		
			Line 3:42: Parameter name 'n' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]		
			Line 15: 'METHOD_DEF' should be separated from previous statement. [EmptyLineSeparator]		
			Line 17:13: Local variable name 'n' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]		

他的代码：

做题用时: 18 分钟 语言 : Java 运行时间 : 54ms 占用内存 : 11052K 程序状态 : 答案正确

```
import java.util.*;
public class Main {
    public static boolean isComplete(int n){
        int sum = 1;//每个数1肯定是它的约数
        for(int i = 2;i <= (int)(Math.sqrt(n));i++){
            if(n % i == 0){
                sum += (i + n/i);
            }
        }
        if(sum == n){
            return true;
        }
        return false;
    }
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        int count = 0;
        for(int i = 4;i <= n;i++){
            if(isComplete(i)){
                count++;
            }
        }
        System.out.println(count);
    }
}
```



点此或手机扫描二维码查看代码编写过程

12 ACM编程题 语言限制 [编程题 | 平均分13.25分 | 20人正确/85人做题 | 提交: 7 次] 得分: 19.44 / 25.0

标题: 扑克牌大小 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【扑克牌大小】

扑克牌游戏大家应该都比较熟悉了，一副牌由54张组成，含3~A、2各4张，小王1张，大王1张。牌面从小到大用如下字符和字符串表示（其中，小写joker表示小王，大写JOKER表示大王）：

3 4 5 6 7 8 9 10 J Q K A 2 joker JOKER

输入两手牌，两手牌之间用“-”连接，每手牌的每张牌以空格分隔，“-”两边没有空格，如：4 4 4 4-joker JOKER。

请比较两手牌大小，输出较大的牌，如果不存在比较关系则输出ERROR。

基本规则：

- （1）输入每手牌可能是个子、对子、顺子（连续5张）、三个、炸弹（四个）和对王中的一种，不存在其他情况，由输入保证两手牌都是合法的，顺子已经从小到大排列；
- （2）除了炸弹和对王可以和所有牌比较之外，其他类型的牌只能跟相同类型的存在比较关系（如，对子跟对子比较，三个跟三个比较），不考虑拆牌情况（如：将对子拆分成个子）；
- （3）大小规则跟大家平时了解的常见规则相同，个子、对子、三个比较牌面大小；顺子比较最小牌大小；炸弹大于前面所有的牌，炸弹之间比较牌面大小；对王是最大的牌；
- （4）输入的两手牌不会出现相等的情况。

数据范围：字符串长度： $3 \leq s \leq 10$

输入描述：

输入两手牌，两手牌之间用“-”连接，每手牌的每张牌以空格分隔，“-”两边没有空格，如 4 4 4 4-joker JOKER。

输出描述：

输出两手牌中较大的那手，不含连接符，扑克牌顺序不变，仍以空格隔开；如果不存在比较关系则输出ERROR。

示例1：

输入

4 4 4 4-joker JOKER

输出

joker JOKER

代码片段									
功能实现				代码提交统计				代码执行统计	
		TA的	平均			TA的	平均		
总通过率		78%	53%	使用语言		Java		答案错误：5 编译错误：2	
基本测试用例通过率		8/11 (73%)	51%	做题用时		01:06:37 00:51:23			
边缘测试用例通过率		6/7 (86%)	55%	提交次数		7 3			
代码效率							代码规范及可读性		
		TA的	参考				代码规范得分		
运行时间		38ms	1s				4.53488		

占用内存 10964K 32768K

Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]
Line 29:16: Local variable name 's' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]

他的代码：

做题用时: 66 分钟 语言 : Java 运行时间 : 38ms 占用内存 : 10964K 程序状态 : 答案错误

```
import java.util.*;
public class Main {
    public static void comPareCard(String card1,String card2){
        String[] s1 = card1.split(" ");
        String[] s2 = card2.split(" ");
        int len1 = s1.length;
        int len2 = s2.length;
        String cards = "34567891JQKA2";
        //注意每手牌只可能是个子，对子，顺子等中的一种，没有组合情况
        if(len1 == len2){
            //这个时候就可以直接比较
            //只取第一个字符比较
            if(cards.indexOf(s1[0].charAt(0)) > cards.indexOf(s2[0].charAt(0))){
                System.out.println(card1);
            }else{
                System.out.println(card2);
            }
        }else if(len1 == 4){//都是炸弹相等的时候就会比较了
            System.out.println(card1);
        }else if(len2 == 2){
            System.out.println(card2);
        }else{
            System.out.println("ERROR");
        }
    }
}

public static void main(String[] args){
    Scanner scan = new Scanner(System.in);
    String s = scan.nextLine();
    String[] card = s.split("-");
    //存在大小王的那一手牌就是最大的
    if(card[0].indexOf("joker") >= 0 && card[0].indexOf("JOKER") >= 0){
        System.out.println(card[0]);
        return;
    }
    if(card[1].indexOf("joker") >= 0 && card[1].indexOf("JOKER") >= 0){
        System.out.println(card[1]);
        return;
    }
    //走到了这里说明没有大小王的存在
    comPareCard(card[0],card[1]);
}
}
```




[点此](#)或手机扫描二维码查看代码编写过程

监控截图

