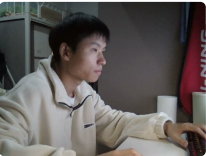


比特就业课Java方向Java班笔试强训48天day19_11月7日（105Java班）-王世国-测评结果

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: 湖北武汉 (113.57.53.192) |
做题用时：05:22:53(2022-11-06 20:15:17开始答题，2022-11-07 11:29:43交卷) | 作答设备：PC |
已同意诚信声明和隐私协议

72.5 分 / 95分

在本次考试中，考生总成绩为72.5分/95分，评级为C（排名前47%），编程能力良好（1题通过，1题部分通过，分数排名前51%），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，无视频监控截图，未开启摄像头。

考生成绩



作弊风险

高风险

未开启摄像头

考生考试过程中未开启摄像头

知识点技能图谱



| | 知识点得分情况 | | |
|--|---------|------|------|
| | 知识点 | 得分 | 正确题数 |
| | 链表 | 5.0 | 1 |
| | 栈 | 5.0 | 1 |
| | 数学 | 25.0 | 1 |
| | 哈希 | 0.0 | 0 |

历史笔试记录

| 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 |

| 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 |
|----|------------------------------------|------------|----------|--------------------|----------------|---------------------|---------------------|
| 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 |
| 20 | 比特就业课105期Java方向笔试强训48天day16_10月27日 | 13.0% | 89.4/100 | 单选:45.0分 编程:44.44分 | 是,摄像头监控异常 | 2022-10-24 14:25:44 | 2022-10-26 21:33:12 |
| 21 | 比特就业课105期Java方向笔试强训48天day17_10月28日 | 24.0% | 83.3/100 | 单选:40.0分 编程:43.33分 | 是,摄像头监控异常 | 2022-10-24 14:29:13 | 2022-10-28 22:18:13 |
| 22 | 比特就业课105期Java方向笔试强训48天day18_10月29日 | 12.0% | 84.1/100 | 单选:40.0分 编程:44.12分 | 是,摄像头监控异常 | 2022-10-24 14:31:43 | 2022-10-29 15:39:41 |

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% | 7 | 00:24:18 | Java | 42ms | 11152K | 良 | 良 | 1% |
| 编程题2 | 30% | 3 | 04:14:01 | Java | 60ms | 13580K | 中 | 良 | 51% |

1 [单选题 | 平均分2.92分 | 49人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0]
下列关于线性链表的叙述中，正确的是（ ）

- A 各数据结点的存储空间可以不连续，但它们的存储顺序与逻辑顺序必须一致
- B 各数据结点的存储顺序与逻辑顺序可以不一致，但它们的存储空间必须连续
- C 进行插入与删除时，不需要移动表中的元素
- D 以上说法均不正确

对于顺序表例如数组arr[0]一定是在arr[1]前面的，他们的逻辑顺序和存储顺序是一致的，但是链表的存储顺序和逻辑顺序可能是不一样的，因为链表的存储空间不连续，第二个节点可能是存在第一个节点的前面

他的回答： C (正确)
正确答案： C

参考答案：
一般来说，在线性表的链式存储结构中，各数据结点的存储序号是不连续的，并且各结点在存储空间中的位置关系与逻辑关系也不一致。线性链表中数据的插入和删除都不需要移动表中的元素，只需改变结点的指针域即可。

2 [单选题 | 平均分4.76分 | 80人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0]
一个栈的初始状态为空。现将元素 1,2,3,A,B,C 依次入栈，然后再依次出栈，则元素出栈的顺序是（ ）
A 1,2,3,A,B,C
B C,B,A,1,2,3
C C,B,A,3,2,1
D 1,2,3,C,B,A

他的回答： C (正确)
正确答案： C

参考答案：

栈的修改是按后进先出的原则进行的，所以顺序应与入栈顺序相反，故选 C。

3 [单选题 | 平均分4.17分 | 70人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0

下列数据结构中，不能采用顺序存储结构的是（ ）

A 非完全二叉树

B 堆

C 队列

D 栈

在二叉树中，只有完全二叉树可以采用顺序存储，因为它的相邻节点之间不可能存在空的，可以理解为层序遍历过来是连续的。

属于顺序表

他的回答：A (正确)

正确答案：A

参考答案：

根据完全二叉树的性质 6，满二叉树和完全二叉树可以按层序进行顺序存储，但一般的二叉树不适用。堆可以用一维数组来存储也可以用完全二叉树来直观地表示堆的结构。队列、栈本身就是顺序存储的。故本题答案为 A 选项。

4 [单选题 | 平均分3.99分 | 67人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0

递归函数最终会结束，那么这个函数一定？

A 使用了局部变量

B 有一个分支不调用自身 也就是递归的终止条件

C 使用了全局变量或者使用了一个或多个参数

D 没有循环调用

他的回答：B (正确)

正确答案：B

参考答案：

直接排除AD，注意力集中在B和C。

B肯定是对的，只有一次循环满足某个条件，不调用自己就返回，递归才会一层一层向上返回。

那么C呢，想一下，全局变量和参数确实可以用来控制递归的结束与否。

该不该选C呢？再仔细看一下题目（说实话，我很讨厌这种文字游戏），“这个函数一定...” ，所以，问题集中在，是否是一定会使用这两种方式呢？显然不是的。

除了C中提到的两种情况外，还有如下控制递归的方式：

1. 局部静态变量是可以控制递归函数最终结束的 2. 可能通过异常来控制递归的结束。 3. 可以利用BIOS或OS的一些数据或一些标准库的全局值来控制递归过程的终止。 4. 可以把一些数据写入到BIOS或OS的系统数据区，也可以把数据写入到一个文件中，以此来控制递归函数的终止。

所以，答案为B

5 [单选题 | 平均分4.05分 | 68人正确/84人做题 | 用时：28分 | 得分：5.0 / 5.0

已知二叉树后序遍历序列是bfegcda，中序遍历序列是badefcg，它的前序遍历序列是（ ）

A abcdefg

B abdcefg

C adbcfeg

D abecdffg

画图，解题其实排除法只需要还原一部分二叉树就可以得到结果

他的回答：B (正确)

正确答案：B

参考答案：

分析：很有代表性的一道题目，去年参加微软笔试的时候也有类似的题目。后序遍历中的最后一个元素是根节点，a，然后查找中序中a的位置，把中序遍历分成 b a defcg，易知左子树为b，右子树为defcg，再递归求解，可画出原始二叉树，故知前序遍历序列为B。

6 [单选题 | 平均分4.52分 | 76人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0

某完全二叉树按层次输出（同一层从左到右）的序列为 ABCDEFGH。该完全二叉树的前序序列为（ ）

A ABDHECFG

B ABCDEFGH

C HDBEAFCG
D HDEBFGCA

他的回答： A (正确)
正确答案： A

参考答案：
前序遍历：访问根结点在访问左子树和访问右子树之前。即先访问根结点，然后遍历左子树，最后遍历右子树；并且在遍历左子树和右子树时，仍然先访问根结点，然后遍历左子树，最后遍历右子树。 中序遍历：访问根结点在访问左子树和访问右子树两者之间。即先遍历左子树，然后访问根结点，最后遍历右子树。并且在遍历左子树和右子树时，仍然首先遍历左子树，然后访问根结点，最后遍历右子树。 后序遍历：访问根结点在访问左子树和访问右子树之后。即首先遍历左子树，然后遍历右子树，最后访问根结点；并且在遍历左子树和右子树时，仍然首先遍历左子树，然后遍历右子树，最后访问根结点。 完全二叉树是指除最后一层外，每一层上的结点数均达到最大值，在最后一层上只缺少右边的若干结点。 因此此完全二叉树可能的形状为： 则前序遍历序列为： ABDHECFG 。故本题答案为 A 选项。

7 [单选题 | 平均分3.1分 | 52人正确/84人做题 | 用时：3分 得分： 0.0 / 5.0

设有一组记录的关键字为{19,14,23,1,68,20,84,27,55,11,10,79}，用链地址法构造哈希表,哈希函数为H(key)=key MOD 13，哈希地址为1的链中有（ ）个记录

- A 1
B 2
C 3
D 4

不要把1掉了

他的回答： C (错误)
正确答案： D

8 [单选题 | 平均分3.87分 | 65人正确/84人做题 | 用时：<1分 得分： 5.0 / 5.0

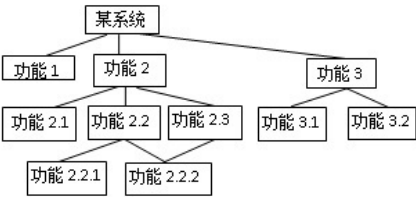
假设你只有100Mb的内存，需要对1Gb的数据进行排序，最合适的算法是（ ）

- 多路归并排序 当内存中放不下我们要排序的数据的时候就要借助外部磁盘空间，也就是外部排序了(多路归并排序)
A 归并排序 将外部数据拆解为小份之后读入内存利用内部排序(快排，插排等等)排序好之后，最后在外部分进行这些小份排序好的数据的归并过程。
B 插入排序
C 快速排序
D 冒泡排序

他的回答： A (正确)
正确答案： A

9 [单选题 | 平均分4.17分 | 70人正确/84人做题 | 用时：<1分 得分： 5.0 / 5.0

某系统总体结构如下图所示, 该系统结构图的深度是（ ）



- A 4
B 3
C 2
D 1

他的回答： A (正确)
正确答案： A

参考答案：
系统结构图的深度是指表示控制的层数。从图中可见该系统结构的深度为 4 层。故本题答案为 A 选项。

标题：汽水瓶 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【汽水瓶】

某商店规定：三个空汽水瓶可以换一瓶汽水，允许向老板借空汽水瓶（但是必须要归还）。

小张手上有n个空汽水瓶，她想知道自己最多可以喝到多少瓶汽水。

$1 \leq n \leq 100$

数据范围：输入的正整数满足

注意：本题存在多组输入。输入的 0 表示输入结束，并不用输出结果。

输入描述：

输入文件最多包含 10 组测试数据，每个数据占一行，仅包含一个正整数 n (1<=n<=100)，表示小张手上的空汽水瓶数。n=0 表示输入结束，你的程序不应当处理这一行。

输出描述：

对于每组测试数据，输出一行，表示最多可以喝的汽水瓶数。如果一瓶也喝不到，输出0。

示例1：

输入

3
10
81
0

输出

1
5
40

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 | | TA的 | 平均 | 答案错误：2 答案正确：5 | |
| | 100% | 89% | 使用语言 | Java | | | |
| | 6/6 | 89% | 做题用时 | 00:24:18 | 00:17:56 | | |
| | (100%) | | 提交次数 | 7 | 1 | | |
| 基本测试用例通过率 | | | | | | | |
| 边缘测试用例通过率 | 4/4 | 89% | | | | | |
| | (100%) | | | | | | |

| 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 | 42ms | 1s | 4.6 | |
| 占用内存 | 11152K | 32768K | Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator] Line 6:17: 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: 24 分钟 语言：Java 运行时间：42ms 占用内存：11152K 程序状态：答案正确

```
import java.util.*;
public class Main {
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        while(scan.hasNextInt()){
```

```

int n = scan.nextInt();
if(n == 0){
    return ;
}
int count = 0;//记录喝的瓶数
int emptyNum = n;//记录手中空瓶数
while(emptyNum >= 2){
    if(emptyNum == 2){//只有等于2的时候可以借了能还回去
        emptyNum += 1;//向老板借一瓶
    }
    int num = emptyNum/3;//当前空瓶可以换的瓶数
    count += num;//换的都可以喝掉，就加上count
    emptyNum = num + emptyNum - 3*num;//更新一下剩余空瓶数，换的喝掉的空瓶 + 本身不能换掉的剩余的
}
System.out.println(count);
}
}
}

```



[点此](#)或手机扫描二维码查看代码编写过程

11 ACM编程题 语言限制 [编程题 | 平均分12.6分 | 38人正确/86人做题 | 提交: 3 次] 得分: 7.5 / 25.0

标题: 查找两个字符串a,b中的最长公共子串 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【查找两个字符串a,b中的最长公共子串】

查找两个字符串a,b中的最长公共子串。若有多组，输出在较短串中最先出现的那个。

注：子串的定义：将一个字符串删去前缀和后缀（也可以不删）形成的字符串。请和“子序列”的概念分开！

$$1 \leq length \leq 300$$

数据范围：字符串长度

$$O(n^3)$$

$$O(n)$$

进阶：时间复杂度：，空间复杂度：

输入描述：

输入两个字符串

输出描述：

返回重复出现的字符

示例1：

输入

abcdefghijklmnp
abcsafjklmnopqrstuvw

输出

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 30% | 平均 50% | 使用语言 | TA的 Java | 平均 | 答案错误 : 3 | |
| 基本测试用例通过率 | 4/12 (33%) | 51% | 做题用时 | 04:14:01 | 00:42:54 | | |
| 边缘测试用例通过率 | 2/8 (25%) | 48% | 提交次数 | 3 | 2 | | |
| 代码效率 | | | | | 代码规范及可读性 | | |
| 运行时间 | TA的 60ms | 参考 1s | 代码规范得分 Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator] | | | 4.77778 | |
| 占用内存 | 13580K | 32768K | | | | | |

他的代码：

做题用时: 254 分钟 语言：Java 运行时间：60ms 占用内存：13580K 程序状态：答案错误

```
import java.util.*;
public class Main{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String str1 = scan.nextLine();
        String str2 = scan.nextLine();
        if(str1.length() > str2.length()){
            String tmp = str1;
            str1 = str2;
            str2 = tmp;
        }

        String ret = "";
        for(int i = 0;i < str1.length();i++){
            int tmp = i;
            int flag = 0;//标志位
            int left = 0;
            int right = 0;
            for(int j = 0;j < str2.length();j++){
                if(str2.charAt(j) == str1.charAt(tmp)){
                    if(flag == 0){
                        flag = 1;
                        left = tmp;
                    }
                    if(tmp == str1.length() - 1){
                        right = str1.length();
                        if(str1.substring(left,right).length() > ret.length()){
                            ret = str1.substring(left,right);
                        }
                        break;
                    }
                    tmp++;
                }else{
                    if(flag == 1){
                        flag = 0;
                        right = tmp;
                    }
                    if(str1.substring(left,right).length() > ret.length()){
```



```
        ret = str1.substring(left,right);  
    }  
    }  
    }  
    }  
    System.out.println(ret);  
  
    }  
}
```



[点此](#)或手机扫描二维码查看代码编写过程

监控截图
