

每日一题Java方向day16_12月10日-任栋-测评结果

考生信息



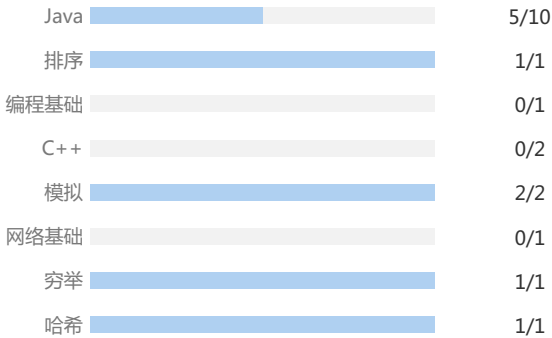
任栋
投递编号：2 | 学校：西安理工大学 | 邮箱：1104580363@qq.com | 职位：2020大四春招冲刺班 |
参考区域: 陕西省西安市 (221.11.20.102) |
做题用时：05:16:35(2020-12-11 22:13:58开始答题，2020-12-12 16:11:42交卷) | 作答设备：PC |
已同意诚信声明和隐私协议

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	25.0	5	63	00:21:48	--
编程	50.0	2	1	04:29:10	--

知识点技能图谱



知识点	得分	正确题数
Java	25.0	5
排序	25.0	1
编程基础	0.0	0
C++	0.0	0
模拟	50.0	2
网络基础	0.0	0
穷举	25.0	1
哈希	25.0	1

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	大四春招冲刺班JavaSE考试	31.0%	26.0/60	单选:26.0分	否	2020-10-28 17:33:16	2020-10-29 10:30:13
2	大四春招冲刺班数据结构考试	77.0%	22.0/60	单选:22.0分 编程:0.0分	否	2020-11-23 11:55:15	2020-11-24 10:40:01
3	Java方向每日一题day02_11月24日	70.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-23 12:10:19	2020-11-25 15:36:05

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
4	Java方向每日一题day03_11月25日	79.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-24 15:19:25	2020-11-24 22:21:11
5	Java方向每日一题day04_11月26日	61.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 14:58:20	2020-11-25 23:17:45
6	Java方向每日一题day05_11月27日	16.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 15:54:38	2020-11-26 23:32:02
7	Java方向每日一题day06_11月28日	43.0%	77.5/100	单选:30.0分 编程:47.5分	否	2020-11-27 14:19:26	2020-11-27 22:53:24
8	Java方向每日一题day07_11月30日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-29 13:58:20	2020-11-30 22:17:40
9	Java方向每日一题day08_12月1日	28.0%	90.0/100	单选:40.0分 编程:50.0分	是，代码抄袭	2020-11-30 10:48:03	2020-12-01 22:29:01
10	Java方向每日一题day09_12月2日	86.0%	50.0/100	单选:35.0分 编程:15.0分	否	2020-12-01 10:43:40	2020-12-02 21:12:07
11	Java方向每日一题day10_12月3日	44.0%	75.0/100	单选:25.0分 编程:50.0分	否	2020-12-02 12:27:01	2020-12-04 09:14:04
12	Java方向每日一题day11_12月4日	72.0%	57.14/100	单选:25.0分 编程:32.14分	否	2020-12-03 10:46:54	2020-12-04 11:05:57
13	Java方向每日一题day12_12月5日	66.0%	60.0/100	单选:35.0分 编程:25.0分	否	2020-12-04 10:43:45	2020-12-05 21:50:32
14	每日一题Java方向day13_12月7日	75.0%	48.57/100	单选:40.0分 编程:8.57分	否	2020-12-05 10:31:45	2020-12-07 22:58:00
15	每日一题Java方向day14_12月8日	62.0%	65.0/100	单选:40.0分 编程:25.0分	否	2020-12-07 12:07:00	2020-12-08 16:49:09

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	03:50:26	Java	355ms	34764K			1%
编程题2	100%	3	00:38:44	Java	29ms	11012K			1%

1

[平均分1.6分 | 27人正确/84人做题 | 用时：<1分 | 得分：0.0 / 5.0]

如果希望监听TCP端口9000，服务器端应该怎样创建socket？

A new Socket("localhost",9000);
B new ServerSocket(9000);
C new Socket(9000);
D new ServerSocket("localhost",9000);

他的回答： D (错误)

正确答案：B

2 [平均分4.4分 | 75人正确/85人做题 | 用时：<1分] 得分：5.0 / 5.0

下列哪个类的声明是正确的？

- A abstract final class HI{}
- B abstract private move(){}
- C protected private number;
- D public abstract class Car{}

他的回答：D (正确)

正确答案：D

3 [平均分3.8分 | 65人正确/85人做题 | 用时：<1分] 得分：5.0 / 5.0

java 接口的修饰符可以为（ ）

- A private
- B protected
- C final
- D abstract

他的回答：D (正确)

正确答案：D

4 [平均分1.7分 | 29人正确/84人做题 | 用时：<1分] 得分：0.0 / 5.0

jre 判断程序是否执行结束的标准是（ ）

- A 所有的前台线程执行完毕
- B 所有的后台线程执行完毕
- C 所有的线程执行完毕
- D 和以上都无关

他的回答：C (错误)

正确答案：A

5 [平均分3.8分 | 63人正确/84人做题 | 用时：16分] 得分：0.0 / 5.0

int i=5;
int s=(i++)+(++i)+(i--)+(--i);
s=()/s 的值是什么？

- A 28
- B 25
- C 21
- D 26
- E 24
- F 23

他的回答：C (错误)

正确答案：E

6 [平均分3.7分 | 62人正确/84人做题 | 用时：<1分] 得分：0.0 / 5.0

如下语句通过算术运算和逻辑运算之后i和j的结果是（ ）

```
int i=0;  
int j=0;  
if((++i>0)||(++j>0))
```

```
{  
    //打印出i和j的值。  
}
```

- A i=0;j=0
- B i=1;j=1
- C i=0;j=1
- D i=1;j=0

他的回答： **B (错误)**

正确答案： **D**

7 [平均分3.3分 | 55人正确/84人做题 | 用时：<1分 | 得分：0.0 / 5.0]

下面不属于Object类中方法的是:

- A hashCode()
- B finally()
- C wait()
- D toString()

他的回答： **C (错误)**

正确答案： **B**

8 [平均分3.2分 | 54人正确/85人做题 | 用时：<1分 | 得分：5.0 / 5.0]

下列哪项不属于jdk1.6垃圾收集器？

- A Serial收集器
- B parNew收集器
- C CMS收集器
- D G1收集器

他的回答： **D (正确)**

正确答案： **D**

9 [平均分2.5分 | 42人正确/84人做题 | 用时：<1分 | 得分：5.0 / 5.0]

instanceof运算符能够用来判断一个对象是否为:

- A 一个类的实例
- B 一个实现指定接口的类的实例
- C 全部正确
- D 一个子类的实例

他的回答： **C (正确)**

正确答案： **C**

10 [平均分4.6分 | 78人正确/85人做题 | 用时：<1分 | 得分：5.0 / 5.0]

对于java类型变量char c,short s,float f,double d,表达式c*s+f+d的结果类型为 ()

- A float
- B char
- C short
- D double

他的回答： D (正确)

正确答案： D

11 ACM编程题 语言限制 [平均分19.5分 | 46人正确/59人做题 | 提交: 7 次] 得分： 25.0 / 25.0

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

【洗牌】洗牌在生活中十分常见，现在需要写一个程序模拟洗牌的过程。现在需要洗2n张牌，从上到下依次是第1张，第2张，第3张一直到第2n张。首先，我们把这2n张牌分成两堆，左手拿着第1张到第n张（上半堆），右手拿着第n+1张到第2n张（下半堆）。接着就开始洗牌的过程，先放下右手的最后一张牌，再放下左手的最后一张牌，接着放下右手的倒数第二张牌，再放下左手的倒数第二张牌，直到最后放下左手的第一张牌。接着把牌合并起来就可以了。例如有6张牌，最开始牌的序列是1,2,3,4,5,6。首先分成两组，左手拿着1,2,3；右手拿着4,5,6。在洗牌过程中按顺序放下了6,3,5,2,4,1。把这六张牌再次合成一组牌之后，我们按照从上往下的顺序看这组牌，就变成了序列1,4,2,5,3,6。现在给出一个原始牌组，请输出这副牌洗牌k次之后从上往下的序列。

输入描述：

第一行一个数T($T \leq 100$)，表示数据组数。对于每组数据，第一行两个数n,k($1 \leq n,k \leq 100$)，接下来一行有2n个数a1,a2,...,a2n($1 \leq a_i \leq 1000000000$)。表示原始牌组从上到下的序列。

输出描述：

对于每组数据，输出一行，最终的序列。数字之间用空格隔开，不要在行末输出多余的空格。

示例1：

输入

3 3 1 1 2 3 4 5 6 3 2 1 2 3 4 5 6 2 2 1 1 1 1

输出

1 4 2 5 3 6 1 5 4 3 2 6 1 1 1 1

代码片段

功能实现

	TA的	平均
总通过率	100%	77%
基本测试用例通过率	6/6 (100%)	77%
边缘测试用例通过率	4/4 (100%)	77%

代码提交统计

	TA的	平均
使用语言	Java	
做题用时	03:50:26	01:03:26
提交次数	7	5

代码执行统计

格式错误	: 4
答案错误	: 1
返回非零	: 1
答案正确	: 1

代码效率

	TA的	参考
运行时间	355ms	1s
占用内存	34764K	32768K

代码规范及可读性

代码规范得分	3.66667
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]	
Line 8:21: Local variable name 'm' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]	
Line 9:21: Local variable name 'b' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]	
Line 19:52: Parameter name 'b' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	
Line 32:13: 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: 230 分钟 语言：Java 运行时间：355ms 占用内存：34764K 程序状态：答案正确

```

import java.util.*;
public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNext()){
            int n = scanner.nextInt();
            for (int i = 0; i < n; i++) {
                int m = scanner.nextInt();
                int b = scanner.nextInt();
                LinkedList<Integer> list = new LinkedList<>();
                for (int j = 0; j < m*2; j++) {
                    list.add(scanner.nextInt());
                }
                func2(list,b);
            }
        }

        private static void func2(LinkedList list, int b) {
            for (int i = 0; i < b; i++) {
                list = func3(list);
            }
            String str = list.toString();
            str = str.replace('[', ' ').replace(']', ' ')
                .replaceAll(", ", "").trim();
            System.out.println(str);
        }

        private static LinkedList func3(LinkedList list) {
            Stack<Integer> stack1 = new Stack<>();
            Stack<Integer> stack2 = new Stack<>();
            int n = list.size()/2;
            for (int i = 0; i < n; i++) {
                stack1.add((Integer) list.poll());
            }
            for (int i = 0; i < n; i++) {
                stack2.add((Integer) list.poll());
            }
            while (!stack1.isEmpty()){
                if (!stack2.isEmpty()){
                    list.addFirst(stack2.pop());
                }
                list.addFirst(stack1.pop());
            }
            return list;
        }
    }
}

```



[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 [平均分22.5分 | 69人正确/77人做题 | 提交: 3 次] 得分: 25.0 / 25.0

标题: 统计同成绩学生人数 | 时间限制: 1秒 | 内存限制: 65536K | 语言限制: 不限

【统计同成绩学生人数】读入N名学生的成绩，将获得某一给定分数的学生人数输出。

输入描述:

测试输入包含若干测试用例，每个测试用例的格式为

第1行: N

第2行: N名学生的成绩，相邻两数字用一个空间隔。

第3行: 给定分数

当读到N=0时输入结束。其中N不超过1000，成绩分数为(包含) 0到100之间的一个整数。

输出描述:

对每个测试用例，将获得给定分数的学生人数输出。

示例1:

输入

```
3
80 60 90
60
2
85 66
0
5
60 75 90 55 75
75
0
```

输出

```
1
0
2
```

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	使用语言	TA的	平均	答案正确 : 3
	100%	89%		Java		
	6/6 (100%)	89%		做题用时	00:38:44 00:24:47	
基本测试用例通过率			提交次数	3	3	

边缘测试用例通过率 4/4
(100%) 89%

代码效率

代码规范及可读性

TA的 参考
运行时间 29ms 1s
占用内存 11012K 65536K

代码规范得分 0.0

他的代码：

做题用时: 38 分钟 语言 : Java 运行时间 : 29ms 占用内存 : 11012K 程序状态 : 答案正确

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNext()){
            int n = scanner.nextInt();
            scanner.nextLine();
            String str = scanner.nextLine();
            int num = scanner.nextInt();
            func(n,str,num);
        }
    }
    private static void func(int n, String str, int num) {
        HashMap<String,Integer> hashMap = new HashMap<>();
        String[] strarr = str.split(" ");
        for (int i = 0; i < strarr.length; i++) {
            if (!hashMap.containsKey(strarr[i])){
                hashMap.put(strarr[i],1);
            }else {
                int val = hashMap.get(strarr[i]);
                hashMap.put(strarr[i],val+1);
            }
        }
        if (hashMap.get(String.valueOf(num))!= null) {
            System.out.println(0);
        }else {
            System.out.println(hashMap.get(String.valueOf(num)));
        }
    }
}
```



