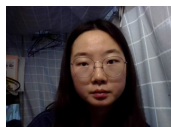


C++方向每日一题day23_5月21日-王梅-测评结果

考生信息



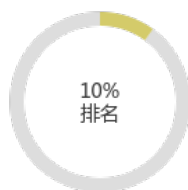
王梅

投递编号：62 | 学校：陕西科技大学 | 邮箱：2910602199@qq.com | 职位：91班 |

参考区域：陕西省西安市（123.139.171.68） | 做题用时：01:34:44(2021-05-21 13:08:05开始答题，21:33:21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	35.0	7	26	00:24:03	已阅
编程	50.0	2	1	01:08:13	已阅

知识点技能图谱

其他知识点	0/1
编程基础	1/1
C++	2/2
字符串	1/1
堆	1/1
数据结构	1/1
哈希	0/1
树	1/1
排序	0/1
递归	1/1
模拟	1/1
查找	0/1
链表	1/1
智力题	1/1

知识点	得分	正确题数
其他知识点	0.0	0
编程基础	5.0	1
C++	10.0	2
字符串	25.0	1
堆	5.0	1
数据结构	5.0	1
哈希	0.0	0
树	5.0	1
排序	0.0	0
递归	5.0	1
模拟	25.0	1
查找	0.0	0
链表	5.0	1
智力题	5.0	1

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	73班C1考试试卷	44.0%	22.0/60	单选:22.0分 编程:0.0分	否	2020-05-17 15:06:33	2020-05-17 23:09:23
2	实习生冲刺班C2考试试卷	19.0%	37.5/60	单选:18.0分 编程:19.5分	否	2020-07-14 10:30:11	2020-07-18 22:56:30
3	C++方向每日一题day01_4月19日	19.0%	88.8/100	单选:45.0分 编程:43.75分	否	2021-04-18 11:21:15	2021-04-19 19:27:23
4	C++方向每日一题day02_4月20日	56.0%	57.5/100	单选:25.0分 编程:32.5分	否	2021-04-19 14:53:21	2021-04-20 20:22:40
5	C++方向每日一题day03_4月21日	7.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-20 11:16:00	2021-04-21 16:09:24
6	C++方向每日一题day04_4月22日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-21 16:19:39	2021-04-22 16:12:03
7	C++方向每日一题day05_4月23日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 15:09:26	2021-04-23 10:24:19
8	C++方向每日一题day06_4月24日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 15:21:12	2021-04-24 17:35:03
9	C++方向每日一题day07_4月26日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 16:20:07	2021-04-26 18:10:43
10	C++方向每日一题day08_4月27日	6.0%	95.0/100	单选:50.0分 编程:45.0分	否	2021-04-24 15:13:08	2021-04-27 15:21:45
11	91班&92班CPP1考试试卷	26.0%	68.0/80	单选:18.0分 编程:50.0分	否	2021-04-26 14:46:52	2021-04-28 20:07:07
12	C++方向每日一题day09_4月28日	18.0%	90.0/100	单选:40.0分 编程:50.0分	否	2021-04-27 16:33:22	2021-04-28 14:46:16
13	C++方向每日一题day10_4月29日	35.0%	78.1/100	单选:30.0分 编程:48.08分	否	2021-04-28 11:39:52	2021-04-29 07:43:41
14	C++方向每日一题day11_5月7日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-06 15:16:10	2021-05-06 22:42:17
15	C++方向每日一题day12_5月8日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-07 14:24:30	2021-05-08 12:57:48
16	C++方向每日一题day13_5月10日	1.0%	75.0/100	单选:50.0分 编程:25.0分	否	2021-05-09 11:22:20	2021-05-10 08:10:17
17	C++方向每日一题day14_5月11日	23.0%	70.0/100	单选:40.0分 编程:30.0分	否	2021-05-10 10:13:17	2021-05-10 22:06:18
18	C++方向每日一题day15_5月12日	8.0%	85.0/100	单选:35.0分 编程:50.0分	否	2021-05-10 14:52:33	2021-05-12 18:58:48
19	C++方向每日一题day16_5月13日	9.0%	82.9/100	单选:35.0分 编程:47.92分	否	2021-05-12 10:20:46	2021-05-12 23:22:37

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
20	91&92班Linux1考试试卷	16.0%	52.0/60	单选:18.0分 不定项选择:4.0分 编程:30.0分	否	2021-05-12 14:37:21	2021-05-15 09:44:02
21	C++方向每日一题day17_5月14日	16.0%	90.0/100	单选:40.0分 编程:50.0分	否	2021-05-12 14:39:41	2021-05-14 17:42:58
22	C++方向每日一题day18_5月15日	39.0%	80.0/100	单选:45.0分 编程:35.0分	否	2021-05-13 11:59:08	2021-05-15 21:28:28
23	C++方向每日一题day19_5月17日	33.0%	80.0/100	单选:30.0分 编程:50.0分	否	2021-05-16 16:34:55	2021-05-17 13:37:47
24	C++方向每日一题day20_5月18日	9.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-17 15:29:48	2021-05-17 23:20:25
25	C++方向每日一题day21_5月19日	4.0%	95.0/100	单选:45.0分 编程:50.0分	是, 代码抄袭	2021-05-18 11:19:57	2021-05-20 00:02:39
26	C++方向每日一题day22_5月20日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-19 09:55:27	2021-05-20 13:37:31

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	00:14:47	C++	4ms	512K			1%
编程题2	100%	1	00:53:26	C++	3ms	768K			1%

1 [平均分3.17分 | 33人正确/52人做题 | 用时：2分] 得分：5.0 / 5.0

在双向循环链表中，在p指针所指的节点后插入一个指针q所指向的新节点，修改指针的操作是_____。

- A p->next=q;q->prior=p;p->next->prior=q;q->next=q;
- B p->next=q;p->next->prior=q;q->prior=p;q->next=p->next;
- C q->prior=p;q->next=p->next;p->next->prior=q;p->next=q;
- D q->next=p->next;q->prior=;p->next=q;p->next=q;

他的回答： C (正确)
正确答案： C

2 [平均分3.5分 | 35人正确/50人做题 | 用时：<1分] 得分：5.0 / 5.0

以下程序的运行结果是：

```
int main(void){
    int m=5;
    if(m++>5)cout<<m<<endl;
    else cout<<--m;
}
```

A 4

- B 5
- C 7
- D 6

他的回答： B (正确)

正确答案： B

3 [平均分3.7分 | 37人正确/50人做题 | 用时：5分  得分：5.0 / 5.0

```
struct Date
{
    char a;
    int b;
    int64_t c;
    char d;
};
Date data[2][10];
```

在64位系统上，如果Data的地址是x，那么data[1][5].c的地址是（ ）

- A X+195
- B X+365
- C X+368
- D X+215

他的回答： C (正确)

正确答案： C

4 [平均分3.33分 | 34人正确/51人做题 | 用时：<1分  得分：5.0 / 5.0

采用递归方式对顺序表进行快速排序，下列关于递归次数的叙述中，正确的是（ ）

- A 递归次数与初始数据的排列次序无关
- B 每次划分后，先处理较长的分区可以减少递归次数
- C 每次划分后，先处理较短的分区可以减少递归次数
- D 递归次数与每次划分后得到的分区处理顺序无关

他的回答： D (正确)

正确答案： D

5 [平均分4.0分 | 40人正确/50人做题 | 用时：3分  得分：5.0 / 5.0

一棵完全二叉树第六层有9个叶结点（根为第一层），则结点个数最多有（ ）

- A 112
- B 111
- C 107
- D 109

他的回答： D (正确)

正确答案： D

6 [平均分4.42分 | 46人正确/52人做题 | 用时：<1分  得分：5.0 / 5.0

两个人两个小时能组装两辆自行车,要在6小时内组装12辆自行车,需要多少人?

- A 2
- B 3
- C 4
- D 5

他的回答：C (正确)

正确答案：C

7 [平均分4.1分 | 41人正确/50人做题 | 用时：2分 | 得分：5.0 / 5.0]

已知关键字序列5,8,12,19,28,20,15,22是最小堆，插入关键字3，调整后得到的最小堆是()

A 3,8,12,5,20,15,22,28,19

B 3,5,12,19,20,15,22,8,28

C 3,12,5,8,28,20,15,22,19

D 3,5,12,8,28,20,15,22,19

他的回答：D (正确)

正确答案：D

8 [平均分1.84分 | 18人正确/49人做题 | 用时：2分 | 得分：0.0 / 5.0]

已知一个线性表 (38 , 25 , 74 , 63 , 52 , 48) ，假定采用散列函数 $h(\text{key}) = \text{key} \% 7$ 计算散列地址，并散列存储在散列表A【0....6】中，若采用线性探测方法解决冲突，则在该散列表上进行等概率成功查找的平均查找长度为

A 1.5

B 1.7

C 2.0

D 2.3

他的回答：B (错误)

正确答案：C

参考答案：

依次进行取模运算求出哈希地址：

A	0	1	2	3	4	5	6
记录	63	48		38	25	74	52
查找次数	1	3		1	1	2	4

74 应该放在下标为 4 的位置，由于 25 已经放在这个地方，所以 74 往后移动，放在了下标为 5 的位置上了。

由于是等概率查找，所以结果为： $1/6 * (1+3+1+1+2+4) = 2.0$

9 [平均分3.7分 | 37人正确/50人做题 | 用时：<1分 | 得分：0.0 / 5.0]

以30为基准,设一组初始记录关键字序列为 (30,15,40,28,50,10,70), 则第一趟快速排序结果为 ()

A 10 , 28 , 15 , 30 , 50 , 40 , 70

B 10 , 15 , 28 , 30,50 , 40 , 70

C 10 , 28 , 15 , 30 , 40 , 50 , 70

D 10 , 15 , 28 , 30 , 40 , 50 , 70

他的回答：D (错误)

正确答案：B

10 [平均分3.2分 | 32人正确/50人做题 | 用时：<1分 | 得分：0.0 / 5.0]

一棵二叉树的先序遍历为EFHIGJK，中序遍历为HFIEJKG，则后序遍历为 ()

A HIFJKGE

B FHIJKGE

C HIFGJKE

D HIFKJGE

他的回答：A (错误)

正确答案：D

11 完善核心代码 语言限制 [平均分21.28分 | 40人正确/47人做题 | 提交: 1 次] 得分：25.0 / 25.0

标题：微信红包 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【微信红包】

春节期间小明使用微信收到很多个红包，非常开心。在查看领取红包记录时发现，某个红包金额出现的次数超过了红包总数的一半。请帮小明找到该红包金额。写出具体算法思路和代码实现，要求算法尽可能高效。

给定一个红包的金额数组gifts及它的大小n，请返回所求红包的金额。

若没有金额超过总数的一半，返回0。

测试样例：

[1,2,3,2,2],5

返回：2

输入描述：

输出描述：

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 100%	平均 85%	使用语言	TA的 C++	平均 00:14:47	答案正确：1	
基本测试用例通过率	1/1 (100%)	85%	做题用时	00:17:18			
			提交次数	1	4		
代码效率					代码规范及可读性		
运行时间	TA的 4ms	参考 3s	代码规范得分				4.4
占用内存	512K	32768K	Line 16: If/else bodies with multiple statements require braces [readability/braces] [4]				
			Line 4: Add #include for sort [build/include_what_you_use] [4]				
			Line 3: Add #include for vector<> [build/include_what_you_use] [4]				

他的代码：

做题用时: 14 分钟 语言：C++ 运行时间：4ms 占用内存：512K 程序状态：答案正确

```
class Gift {
public:
    int getValue(vector<int> gifts, int n) {
        sort(gifts.begin(), gifts.end());
        int mid = gifts[n / 2];
        int count = 0;
        for (int i = 0; i < n; i++)
        {
            if (gifts[i] == mid)
            {
                count++;
            }
        }
        if (count > n / 2)
            return mid;
        else
            return 0;
    }
}
```



[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5.23分 | 9人正确/43人做题 | 提交: 1 次] 得分: 25.0 / 25.0

标题：计算字符串的距离 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限
【计算字符串的距离】

Levenshtein 距离，又称编辑距离，指的是两个字符串之间，由一个转换成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。编辑距离的算法是首先由俄国科学家Levenshtein提出的，故又叫Levenshtein Distance。
Ex：
字符串A:abcdefg
字符串B: abcdef
通过增加或是删掉字符“g”的方式达到目的。这两种方案都需要一次操作。把这个操作所需要的次数定义为两个字符串的距离。
要求：
给定任意两个字符串，写出一个算法计算它们的编辑距离。

```
请实现如下接口
/* 功能：计算两个字符串的距离
 * 输入：字符串A和字符串B
 * 输出：无
 * 返回：如果成功计算出字符串的距离，否则返回-1
 */
public static int calStringDistance (String charA, String charB)
{
    return 0;
}
```

输入描述：

输入两个字符串

输出描述：

得到计算结果

示例1：

输入

abcdefg
abcdef

输出

1

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		答案正确：1

总通过率	100%	20%	使用语言	C++	
基本测试用例通过率	6/6 (100%)	20%	做题用时	00:53:26	00:27:01
边缘测试用例通过率	4/4 (100%)	20%	提交次数	1	2
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	3ms	1s	Line 34: Add #include for min [build/include_what_you_use]		
占用内存	768K	32768K	[4]		
			4.79167		

他的代码：

做题用时: 53 分钟 语言: C++ 运行时间：3ms 占用内存：768K 程序状态：答案正确

```
#include<iostream>
using namespace std;
#include<string>
#include<vector>
int distance(string str1, string str2)
{
    if (str1.empty() || str2.empty())
    {
        return max(str1.size(), str2.size());
    }
    int len1 = str1.size();
    int len2 = str2.size();
    vector<vector<int>>> f(1 + len1, vector<int>(1 + len2, 0));
    for (int i = 0; i <= len1; ++i)
    {
        f[i][0] = i;
    }
    for (int i = 0; i <= len2; ++i)
    {
        f[0][i] = i;
    }
    for (int i = 1; i <= len1; ++i)
    {
        for (int j = 1; j <= len2; ++j)
        {
            if (str1[i - 1] == str2[j - 1])
            {
                f[i][j] = 1 + min(f[i][j - 1], f[i - 1][j]);
                f[i][j] = min(f[i][j], f[i - 1][j - 1]);
            }
            else
            {
                f[i][j] = 1 + min(f[i][j - 1], f[i - 1][j]);
                f[i][j] = min(f[i][j], 1 + f[i - 1][j - 1]);
            }
        }
    }
    return f[len1][len2];
}
int main()
{
    string str1, str2;
    while (cin >> str1 >> str2)
    {
```



```
cout << distance(str1, str2) << endl;  
}  
return 0;  
}
```



[点此](#)或手机扫描二维码查看代码编写过程