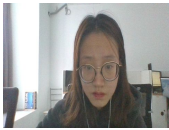


每日一题C++方向day19_11月27日测评结果

考生信息



王婧

考号：2378 | 学校：财经大学 | 邮箱：1031160332@qq.com | 职位：54

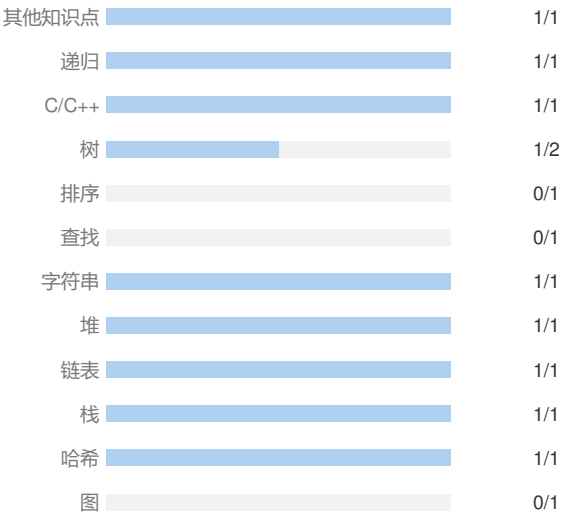
参考区域: 陕西省西安市 (113.200.106.154) | 做题用时：00:24:03(2019-11-27 12:35:28 - 12:59:39) | 作答设备：PC

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	65	00:07:04	--
编程	50.0	2	1	00:11:58	--

知识点技能图谱



知识点	得分	正确题数
其他知识点	25.0	1
递归	5.0	1
C/C++	5.0	1
树	5.0	1
排序	0.0	0
查找	0.0	0
字符串	25.0	1
堆	5.0	1
链表	5.0	1
栈	5.0	1
哈希	5.0	1
图	0.0	0

历史笔记录

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	54班C/C++考试题	11.0%	21.0/60	单选:6.0分 编程:15.0分	否	2019-09-21 17:24:48	2019-09-22 15:38:37

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
2	54班CPP_DS_2_考试卷	38.0%	19.0/60	单选:4.0分 编程:15.0分	否	2019-10-31 17:40:31	2019-11-02 15:40:22
3	每日一题C++方向day02_11月7日	26.0%	70.0/100	单选:20.0分 编程:50.0分	否	2019-11-06 16:54:27	2019-11-07 13:25:19
4	每日一题C++方向day03_11月8日	16.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-07 14:40:49	2019-11-08 12:51:53
5	每日一题C++方向day04_11月9日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-11-08 11:17:08	2019-11-10 17:00:19
6	每日一题C++方向day05_11月10日	64.0%	52.5/100	单选:25.0分 编程:27.5分	是，相似代码	2019-11-09 15:35:20	2019-11-10 17:49:42
7	每日一题C++方向day06_11月11日	54.000004%	65.0/100	单选:40.0分 编程:25.0分	否	2019-11-09 15:38:30	2019-11-11 18:14:20
8	每日一题C++方向day07_11月13日	50.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-12 11:39:41	2019-11-13 14:37:22
9	每日一题C++方向day08_11月14日	8.0%	90.0/100	单选:40.0分 编程:50.0分	是，相似代码	2019-11-13 10:51:46	2019-11-14 17:35:22
10	每日一题C++方向day09_11月15日	79.0%	45.0/100	单选:20.0分 编程:25.0分	否	2019-11-14 18:14:54	2019-11-15 17:52:51
11	每日一题C++方向day10_11月16日	6.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-15 13:29:43	2019-11-16 12:12:48
12	每日一题C++方向day11_11月17日	22.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-11-16 10:59:12	2019-11-17 21:38:02
13	每日一题C++方向day12_11月18日	45.0%	70.0/100	单选:20.0分 编程:50.0分	是，相似代码	2019-11-17 11:32:33	2019-11-18 17:49:04
14	每日一题C++方向day13_11月20日	2.0%	90.0/100	单选:40.0分 编程:50.0分	是，相似代码	2019-11-19 11:13:30	2019-11-20 15:08:03
15	每日一题C++方向day14_11月21日	16.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-20 13:55:05	2019-11-21 13:15:24
16	每日一题C++方向day15_11月22日	21.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-21 12:07:32	2019-11-22 12:56:30
17	每日一题C++方向day16_11月23日	47.0%	55.0/100	单选:5.0分 编程:50.0分	是，相似代码	2019-11-22 14:21:38	2019-11-23 12:51:07
18	每日一题C++方向day17_11月24日	2.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-11-23 14:28:02	2019-11-25 15:04:52
19	每日一题C++方向day18_11月25日	47.0%	70.0/100	单选:20.0分 编程:50.0分	是，相似代码	2019-11-24 11:44:46	2019-11-25 15:42:37

编码能力



1 [平均分2.5分 | 76人正确/155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

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 [平均分5.0分 | 154人正确/155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 [平均分2.2分 | 68人正确/155人做题 | 用时：<1分 | 得分：5.0 / 5.0

求fun（484）的返回值（ ）

```
bool fun(int n){
    int sum=0;
    for(int i=1;n>sum;i=i+2)
        sum=sum+i;
    return (n==sum);
}
```

- A True
- B False

他的回答：A (正确)

正确答案：A

4 [平均分4.0分 | 124人正确/154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
- 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 [平均分4.0分 | 121人正确/151人做题 | 用时：<1分 | 得分：0.0 / 5.0

已知二叉树后序遍历序列是bfehgda，中序遍历序列是badehcg，它的前序遍历序列是：

- A abcdefg
- B abdcefg
- C adbcfeg
- D abecdffg

他的回答： D (错误)

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 [平均分3.9分 | 117人正确/150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 [平均分4.2分 | 126人正确/149人做题 | 用时：<1分 | 得分：5.0 / 5.0]
以下序列不是堆的是()

- A (100,85,98,77,80,60,82,40,20,10,66)
- B (100,98,85,82,80,77,66,60,40,20,10)
- C (10,20,40,60,66,77,80,82,85,98,100)
- D (100,85,40,77,80,60,66,98,82,10,20)

他的回答： D (正确)

正确答案： D

8 [平均分2.4分 | 71人正确/148人做题 | 用时：2分 | 得分：5.0 / 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

他的回答： D (正确)

正确答案： D

9 [平均分3.3分 | 100人正确/152人做题 | 用时：<1分 | 得分：0.0 / 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？

- A 归并排序
- B 插入排序

- C 快速排序
- D 冒泡排序

他的回答：C (错误)

正确答案：A

10 [平均分2.8分 | 83人正确/148人做题 | 用时：<1分 | 得分：0.0 / 5.0

下列哪种图的邻接矩阵是对称矩阵（ ）。

- A 有向图
- B 无向图
- C AOV图
- D AOE图

他的回答：C (错误)

正确答案：B

11 [平均分23.8分 | 123人正确/129人做题 | 提交: 1 次 | 得分：25.0 / 25.0

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】

有这样一道智力题：“某商店规定：三个空汽水瓶可以换一瓶汽水。小张手上有十个空汽水瓶，她最多可以换多少瓶汽水喝？”答案是5瓶，方法如下：先用9个空瓶子换3瓶汽水，喝掉3瓶满的，喝完以后4个空瓶子，用3个再换一瓶，喝掉这瓶满的，这时候剩2个空瓶子。然后你让老板先借给你一瓶汽水，喝掉这瓶满的，喝完以后用3个空瓶子换一瓶满的还给老板。如果小张手上有n个空汽水瓶，最多可以换多少瓶汽水喝？

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	
	TA的	平均		TA的	平均	答案正确：1	
总通过率	100%	95%	使用语言	C++			
基本测试用例通过率	6/6 (100%)	95%	做题用时	00:07:24	00:25:42		
边缘测试用例通过率	4/4 (100%)	95%	提交次数	1	3		
代码效率					代码规范及可读性		
	TA的	参考				代码规范得分 5.0	
运行时间	5ms	1s					
占用内存	380K	32768K					

他的代码：

做题用时: 7 分钟

语言: C++

运行时间: 5ms

占用内存: 380K

程序状态: 答案正确

```
#include<iostream>
#include<vector>
using namespace std;
int calc(int number)
{
    if(number==1||number==0)
    {
        return 0;
    }
    if(number==2)
    {
        return 1;
    }
    if(number>=3)
    {
        return (number/3)+calc(number%3+(number/3));
    }
}
int main()
{
    vector<int> data;
    int index=0;
    while(1)
    {
        int temp=0;
        cin>>temp;
        data.push_back(temp);
        if(data[index]==0)
        {
            break;
        }
        else
        {
            index++;
        }
    }
    for(int i=0;i<index;i++)
    {
        cout<<calc(data[i])<<endl;
    }
    return 0;
}
```

12 [平均分19.0分 | 79人正确/110人做题 | 提交: 1 次] 得分: 25.0 / 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。若有多个，输出在较短串中最先出现的那个。

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

abcdefghijklmnop
abcsafjklmnopqrstuvw

输出

jklmnop

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%	76%	使用语言	C++				
基本测试用例通过率	6/6 (100%)	76%	做题用时	00:04:34	00:48:31			
边缘测试用例通过率	4/4 (100%)	75%	提交次数	1	5			
代码效率					代码规范及可读性			
	TA的	参考				代码规范得分		4.4
运行时间	4ms	1s				Line 18: Else clause should be indented at the same level as if. Ambiguous nested if/else chains require braces. [readability/braces] [4]		
占用内存	376K	32768K				Line 18: Using deprecated casting style. Use static_cast(...) instead [readability/casting] [4]		
						Line 11: Add #include for swap [build/include_what_you_use] [4]		

他的代码：

做题用时: 4 分钟 语言：C++ 运行时间：4ms 占用内存：376K 程序状态：答案正确

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string a, b;
    while (cin >> a >> b)
    {
        if (a.size() > b.size())
            swap(a, b);
        string str_m;//存储最长公共子串
        for (int i = 0; i < a.size(); i++)
        {
            for (int j = i; j < a.size(); j++)
            {
                string temp = a.substr(i, j - i + 1);
                if (int(b.find(temp))<0)
                    break;
                else if (str_m.size() < temp.size())
                    str_m = temp;
            }
        }
        cout << str_m << endl;
    }
    return 0;
}
```

