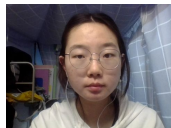


# C++方向每日一题day03\_4月21日-王梅-测评结果

## 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（123.139.160.10） |

做题用时：06:37:57(2021-04-20 21:00:06开始答题，2021-04-21 16:09:24交卷) | 作答设备：PC |

已同意诚信声明和隐私协议

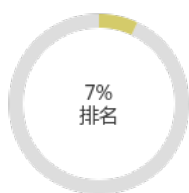
## 考生成绩



95.0  
得分



06:37:57  
用时



7%  
排名

题型	得分	正确题数	排名	用时	是否阅卷
单选	45.0	9	6	00:49:21	已阅
编程	50.0	2	1	05:48:36	已阅

## 知识点技能图谱

其他知识点	1/2
C/C++	2/2
C++	6/6
模拟	1/1
字符串	1/1
贪心	1/1
位运算	1/1
分治	1/1

知识点	得分	正确题数
其他知识点	5.0	1
C/C++	10.0	2
C++	30.0	6
模拟	25.0	1
字符串	25.0	1
贪心	25.0	1
位运算	25.0	1
分治	25.0	1

## 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	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

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	2021-04-19	2021-04-20
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	14:53:21	20:22:40

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%	4	01:11:51	C++	3ms	376K	-	-	1%
编程题2	100%	9	04:36:45	C++	2ms	436K	良	良	1%

1 [平均分3.4分 | 40人正确/58人做题 | 用时：6分] 得分：0.0 / 5.0  
以下程序的输出结果是？

```
#include <stdio.h> main()
{
    char a[10]={ '1','2','3','4','5','6','7','8','9','0'},*p; int i;
    i=8;
    p=a+i;
    printf("%s\n",p-3);
}
```

- A 6
- B 6789
- C '6'
- D 789

他的回答： C (错误)  
正确答案： B

2 [平均分3.9分 | 45人正确/57人做题 | 用时：<1分] 得分：5.0 / 5.0  
能把函数处理结果的二个数据返回给主调函数，在下面的方法中不正确的是（ ）

- A return这二个数据
- B 形参用数组
- C 形参用二个指针
- D 用二个全局变量

他的回答： A (正确)  
正确答案： A

3 [平均分4.1分 | 47人正确/57人做题 | 用时：4分] 得分：5.0 / 5.0  
以下能对二维数组a进行正确初始化的语句是（ ）

- A int ta[2][]={{0,1,2},{3,4,5}};
- B int ta[][3]={{0,1,2},{3,4,5}};
- C int ta[2][4]={{0,1,2},{3,4},{5}};
- D int ta[][3]={{0,,2},{},{3,4,5}};

他的回答： B (正确)  
正确答案： B

4 [平均分3.1分 | 37人正确/59人做题 | 用时：3分  得分：5.0 / 5.0

由多个源文件组成的C程序，经过编辑、预处理、编译、链接等阶段会生成最终的可执行程序。下面哪个阶段可以发现被调用的函数未定义？

- A 预处理
- B 编译
- C 链接
- D 执行

他的回答：C (正确)

正确答案：C

5 [平均分3.5分 | 41人正确/59人做题 | 用时：4分  得分：5.0 / 5.0

下列main()函数执行后的结果为()

```
int func()
{
    int i,j,k=0;
    for(i=0;j=-1;j=0;i++,j++)
    {
        k++;
    }
    return k;
}
int main()
{
    cout<<(func());
}
```

- A -1
- B 0
- C 1
- D 2

他的回答：B (正确)

正确答案：B

6 [平均分4.5分 | 53人正确/59人做题 | 用时：<1分  得分：5.0 / 5.0

以下哪种STL容器中的对象是连续存储的:()

- A list
- B vector
- C map
- D set

他的回答：B (正确)

正确答案：B

7 [平均分3.4分 | 38人正确/56人做题 | 用时：3分  得分：5.0 / 5.0

当一个类对象的生命周期结束后，关于调用析构函数的描述正确的是：( )

- A 如果派生类没有定义析构函数，则只调用基类的析构函数
- B 如果基类没有定义析构函数，则只调用派生类的析构函数
- C 先调用派生类的析构函数，后调用基类的析构函数
- D 先调用基类的析构函数，后调用派生类的析构函数

他的回答：C (正确)

正确答案：C

8 [平均分3.4分 | 39人正确/58人做题 | 用时：9分] 得分：5.0 / 5.0

下面3段程序代码的效果一样吗？

```
int b;  
(1)const int *a = &b;  
(2)int const *a = &b;  
(3)int *const a = &b;
```

- A (2)=(3)
- B (1)=(3)
- C (1)=(2)
- D 都不一样
- E 都一样

他的回答：C (正确)

正确答案：C

9 [平均分3.4分 | 39人正确/58人做题 | 用时：4分] 得分：5.0 / 5.0

下面关于一个类的静态成员描述中,不正确的是()

- A 静态成员变量可被该类的所有方法访问
- B 该类的静态方法只能访问该类的静态成员函数
- C 该类的静态数据成员变量的值不可修改
- D 子类可以访问父类的静态成员
- E 静态成员无多态特性

他的回答：C (正确)

正确答案：C

10 [平均分2.7分 | 31人正确/57人做题 | 用时：10分] 得分：5.0 / 5.0

```
#include <iostream>  
#include <vector>  
using namespace std;  
int main(void)  
{  
    vector<int> array;  
    array.push_back(100);  
    array.push_back(300);  
    array.push_back(300);  
    array.push_back(300);  
    array.push_back(300);  
    array.push_back(500);  
    vector<int>::iterator itor;  
    for(itor=array.begin();itor!=array.end();itor++)  
    {  
        if(*itor==300)  
        {  
            itor=array.erase(itor);  
        }  
    }  
    for(itor=array.begin();itor!=array.end();itor++)  
    {
```

```
    cout<<*itor<<" ";
}
return 0;
}
```

下面这个代码输出的是()

- A 100 300 300 300 300 500
- B 100 300 300 300 500
- C 100 300 300 500
- D 100 300 500
- E 100 500
- F 程序错误

他的回答： C (正确)

正确答案： C

11 ACM编程题 语言限制 [平均分18.8分 | 41人正确/55人做题 | 提交: 4 次] 得分：25.0 / 25.0

标题：字符串中找出连续最长的数字串 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【字符串中找出连续最长的数字串】读入一个字符串str，输出字符串str中的连续最长的数字串

输入描述：

一个测试输入包含1个测试用例，一个字符串str，长度不超过255。

输出描述：

在一行内输出str中里连续最长的数字串。

示例1：

输入

abcd12345ed125ss123456789

输出

123456789

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 75%	使用语言	TA的 C++	平均 00:54:40	答案错误 : 1 编译错误 : 1 答案正确 : 2	
基本测试用例通过率	2/2 (100%)	75%	做题用时	01:11:51	00:54:40		
边缘测试用例通过率	1/1 (100%)	74%	提交次数	4	5		
代码效率					代码规范及可读性		
运行时间	TA的 3ms	参考 1s	代码规范得分 0.0				
占用内存	376K	32768K					

他的代码：

做题用时: 71 分钟 语言：C++ 运行时间：3ms 占用内存：376K 程序状态：答案正确

```
#include<iostream>
using namespace std;
```

```
#include<string>
int main()
{
    string str;
    getline(cin, str);
    int max = 0;
    int tmp, index;
    for (int i = 0; i < str.size(); i++)
    {
        if (str[i] >= '0' && str[i] <= '9')
        {
            int count = 0;
            tmp = i;
            while (str[i] >= '0' && str[i] <= '9')
            {
                count++;
                i++;
            }
            if (max < count)
            {
                max = count;
                index = tmp;
            }
        }
    }
    cout << str.substr(index, max);
    return 0;
}
```



[点此或手机扫描二维码查看代码编写过程](#)

12

完善核心代码

语言限制

[平均分17.3分 | 38人正确/55人做题 | 提交: 9 次]

得分 : 25.0 / 25.0

标题：数组中出现次数超过一半的数字 | 时间限制：1秒 | 内存限制：65536K | 语言限制：[Javascript\_V8, Kotlin, Typescript, Python, C++, Groovy, Rust, C#, Java, Go, C, Scala, Javascript, Ruby, Swift, Php, Python 3]

【数组中出现次数超过一半的数字】数组中有一个数字出现的次数超过数组长度的一半，请找出这个数字。例如输入一个长度为9的数组{1,2,3,2,2,2,5,4,2}。由于数字2在数组中出现了5次，超过数组长度的一半，因此输出2。如果不存在则输出0。

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
TA的 平均	TA的 平均	答案错误 : 6

总通过率	100%	69%	使用语言	C++	编译错误 : 2
基本测试用例通过率	1/1	69%	做题用时	04:36:45 00:27:35	答案正确 : 1
	(100%)		提交次数	9 7	
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	2ms	1s	Line 6: Add #include for sort [build/include_what_you_use] [4]		
占用内存	436K	65536K	Line 3: Add #include for vector<> [build/include_what_you_use] [4]		
			4.6		

他的代码：

做题用时: 276 分钟    语言：C++    运行时间：2ms    占用内存：436K    程序状态：答案正确

```
class Solution {
public:
    int MoreThanHalfNum_Solution(vector<int> numbers) {
        if(numbers.empty())
            return 0;
        sort(numbers.begin(), numbers.end());
        int mid = numbers[numbers.size()/2];
        int count = 0;
        for(int i = 0; i < numbers.size(); i++)
        {
            if(numbers[i] == mid)
                ++count;
        }
        return (count > numbers.size()/2)? mid : 0;
    }
};
```



[点此](#)或手机扫描二维码查看代码编写过程