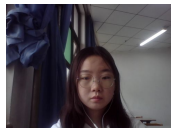


# C++方向每日一题day02\_4月20日-王梅-测评结果

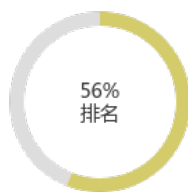
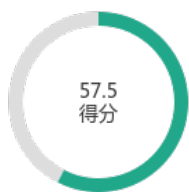
## 考生信息



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（111.114.0.1） | 做题用时：05:41:23(2021-04-20 09:20:09开始答题，20:22:40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	37	00:40:51	已阅
编程	32.5	1	34	05:00:32	已阅

## 知识点技能图谱

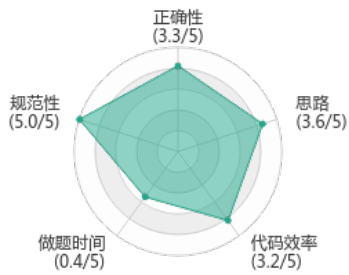
其他知识点	1/1
编程基础	1/3
C/C++	0/1
C++	4/8
字符串	1/1
贪心	0/1

知识点	得分	正确题数
其他知识点	5.0	1
编程基础	5.0	1
C/C++	0.0	0
C++	20.0	4
字符串	25.0	1
贪心	7.5	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	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

## 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	30%	75	03:28:07	C++	31ms	628K			48%
编程题2	100%	25	01:32:25	C++	2ms	424K			1%

1 [平均分3.5分 | 48人正确/68人做题 | 用时: 2分] 得分: 5.0 / 5.0  
有以下定义:

```
int a[10];
char b[80];
```

函数声明为:  
void sss(char[],int[]);

则正确的函数调用形式是 ( )。

- A sss(a,b);
- B sss(char b[],int a[]);
- C sss(b[],a[]);
- D sss(b,a);

他的回答: D (正确)

正确答案: D

2 [平均分3.7分 | 51人正确/69人做题 | 用时: 5分] 得分: 0.0 / 5.0

数组a的定义语句为 "float a[3][4];", 下列 ( ) 是对数组元素不正确的引用方法。

- A a[i][j]
- B \*(a[i]+j)
- C \*(\*(a+i)+j)
- D \*(a+i\*4+j)

他的回答: B (错误)

正确答案: D

```
int main()
{
    float a[3][4];
    cout << a[2][2] << endl;
    cout << *(a[2] + 2) << endl;
    cout << (*(a + 2) + 2) << endl;
    cout << *(a + 2 * 4 + 2) << endl;
}
```

B:  $a[i] + j = *(a + i * 4 + j)$  C:  $*(a + 1) = a[1]$ , D: 改为  $*(a + i * 4 + j)$  就正确了

3 [平均分2.6分 | 35人正确/68人做题 | 用时: 4分] 得分: 5.0 / 5.0

下面叙述错误的是 ( )

```
char acX[] = "abc";
char acY[] = { 'a', 'b', 'c' };
char *szX = "abc";
char *szY = "abc";
```

A acX和acY都是数组, 可以修改

B 因为"abc"是常量字符串, 当它被多次使用时, 编译器不愿意再多分配出额外的内存空间存放多个"abc", 而是重复使用这一个"abc", 所以, szX和szY指针会指向同一个地址。

C 因为szX是字符串数组, 字符串的尾部有一个结束符'\0', 所以szX有四个元素, 内存空间比szY大

D 字符指针指向的是常量字符串, 常量字符串不能修改

他的回答: D (正确)

正确答案: D

4 [平均分2.8分 | 38人正确/68人做题 | 用时: 4分] 得分: 0.0 / 5.0

下列代码的运行结果是 ( )

```
int a[]={1,2,3,4};
int *b=a;
*b+=2;
*(b+2)=2;
b++;
printf( "%d,%d\n" ,*b,*(b+2));
```

- A 1,3
- B 1,2
- C 2,4
- D 3,2

他的回答 : D (错误)

正确答案 : C

5 [平均分2.4分 | 33人正确/69人做题 | 用时 : 2分 | 得分 : 5.0 / 5.0]

在32位cpu上选择缺省对齐的情况下，有如下结构体定义：

```
struct A{
    unsigned a : 19;
    unsigned b : 11;
    unsigned c : 4;
    unsigned d : 29;
    char index;
};
```

32位 :  
19+11  
4  
29  
8  
所以4\*4=16

则sizeof ( struct A ) 的值为 ( )

- A 9
- B 12
- C 16
- D 20

他的回答 : C (正确)

正确答案 : C

6 [平均分3.9分 | 53人正确/68人做题 | 用时 : 7分 | 得分 : 0.0 / 5.0]

以下关于C++的描述中哪一个是正确的：

- A 任何指针都必须指向一个实例
- B 子类指针不可以指向父类实例
- C 任何引用都必须指向一个实例
- D 引用所指向的实例不可能无效

他的回答 : B (错误)

正确答案 : C

7 [平均分3.8分 | 52人正确/69人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0]

以下不是double compare(int,int)的重载函数的是()

- A int compare(double,double)
- B double compare(double,double)
- C double compare(double,int)
- D int compare(int,int) 形式参数个数类型顺序相同了

他的回答 : A (错误) 重载：函数名相同，形式参数个数、类型、顺序不同

正确答案：D

8 [平均分1.2分 | 15人正确/65人做题 | 用时：4分 | 得分：0.0 / 5.0]

关于虚函数的描述正确的是()

- A 派生类的虚函数与基类的虚函数具有不同的参数个数和类型
- B 内联函数不能是虚函数
- C 派生类必须重新定义基类的虚函数
- D 虚函数可以是一个static型的函数

他的回答：C (错误)

正确答案：B

9 [平均分2.2分 | 31人正确/69人做题 | 用时：4分 | 得分：5.0 / 5.0]

请将下列构造函数补充完整，使得程序的运行结果是5

```
#include<iostream>
using namespace std;
class Sample{
public:
    Sample(int x){
        p=new int(x);
    }
    ~Sample(){
        if(p) delete p;
    }
    int show(){
        return *p;
    }
private:
    int*p;
};
int main(){
    Sample S(5);
    cout<<S.show()<<endl;
    return 0;
}
```

- A \*p=x;
- B p=new int(x);
- C \*p=new int(x);
- D p=&x;

他的回答：B (正确)

正确答案：B

10 [平均分1.5分 | 20人正确/68人做题 | 用时：3分 | 得分：5.0 / 5.0]

关于c++的inline关键字,以下说法正确的是()

- A 使用inline关键字的函数会被编译器在调用处展开
- B 头文件中可以包含inline函数的声明
- C 可以在同一个项目的不同源文件内定义函数名相同但实现不同的inline函数
- D 定义在Class声明内的成员函数默认是inline函数
- E 优先使用Class声明内定义的inline函数
- F 优先使用Class实现的内inline函数的实现

他的回答：D (正确)

正确答案：D

11 ACM编程题 语言限制 [平均分14.9分 | 32人正确/60人做题 | 提交: 75 次] 得分：7.5 / 25.0

标题：排序子序列 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【排序子序列】牛牛定义排序子序列为一个数组中一段连续的子序列,并且这段子序列是非递增或者非递减排序的。牛牛有一个长度为n的整数数组A,他现在有一个任务是把数组A分为若干段排序子序列,牛牛想知道他最少可以把这个数组分为几段排序子序列。

如样例所示,牛牛可以把数组A划分为[1,2,3]和[2,2,1]两个排序子序列,至少需要划分为2个排序子序列,所以输出2

输入描述：

输入的第一行为一个正整数n( $1 \leq n \leq 10^5$ )  
第二行包括n个整数A<sub>i</sub>( $1 \leq A_i \leq 10^9$ ),表示数组A的每个数字。

输出描述：

输出一个整数表示牛牛可以将A最少划分为多少段排序子序列

示例1：

输入

6  
1 2 3 2 2 1

输出

2

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		
总通过率	30%	59%	使用语言	C++		浮点错误：1	
基本测试用例通过率	3/6 (50%)	61%	做题用时	03:28:07	01:13:13	答案错误：69	
边缘测试用例通过率	0/4 (0%)	57%	提交次数	75	9	运行超时：4	
						编译错误：1	
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	31ms	1s					
占用内存	628K	32768K					

他的代码：

做题用时: 208 分钟    语言：C++    运行时间：31ms    占用内存：628K    程序状态：答案错误

```
#include<iostream>
using namespace std;
#include<vector>
int main()
{
    int n;
    cin >> n;
    vector<int> A(n);
    for (int i = 0; i < n; i++)
    {
        cin >> A[i];
    }
}
```

```
int count = 0;
for(int i = 1; i <= n; i++)
{
    if(A[i] < A[i-1])
    {
        count++;
    }
}
cout << count << endl;
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18.8分 | 47人正确/64人做题 | 提交: 25 次] 得分 : 25.0 / 25.0

标题：倒置字符串 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【倒置字符串】将一句话的单词进行倒置，标点不倒置。比如 I like beijing. 经过函数后变为：beijing. like I

输入描述：

每个测试输入包含1个测试用例：I like beijing. 输入用例长度不超过100

输出描述：

依次输出倒置之后的字符串,以空格分割

示例1：

输入

I like beijing.

输出

beijing. like I

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	答案错误：11 编译错误：11 答案正确：3	
基本测试用例通过率	2/2 (100%)	76%	做题用时	01:32:25	00:55:06		
边缘测试用例通过率	2/2 (100%)	73%	提交次数	25	7		
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 2ms 1s  
占用内存 424K 32768K

他的代码：

做题用时: 92 分钟 语言: C++ 运行时间: 2ms 占用内存: 424K 程序状态: 答案正确

```
#include<iostream>
using namespace std;
#include<string>
#include<algorithm>
int main()
{
    string str;
    getline(cin, str);
    reverse(str.begin(), str.end());
    str.push_back(' ');
    string::iterator it = str.begin();
    string::iterator it1 = str.begin();
    while (it != str.end())
    {
        if (*it == ' ')
        {
            reverse(it1, it);
            it1 = it + 1;
        }
        it++;
    }
    str.pop_back();
    cout << str << endl;
}
```



[点此](#)或手机扫描二维码查看代码编写过程