

# 每日一题day05\_5月24日测评结果

## 考生信息

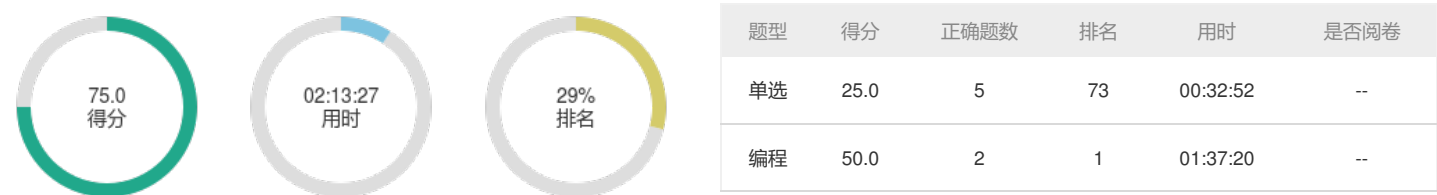


张博翔

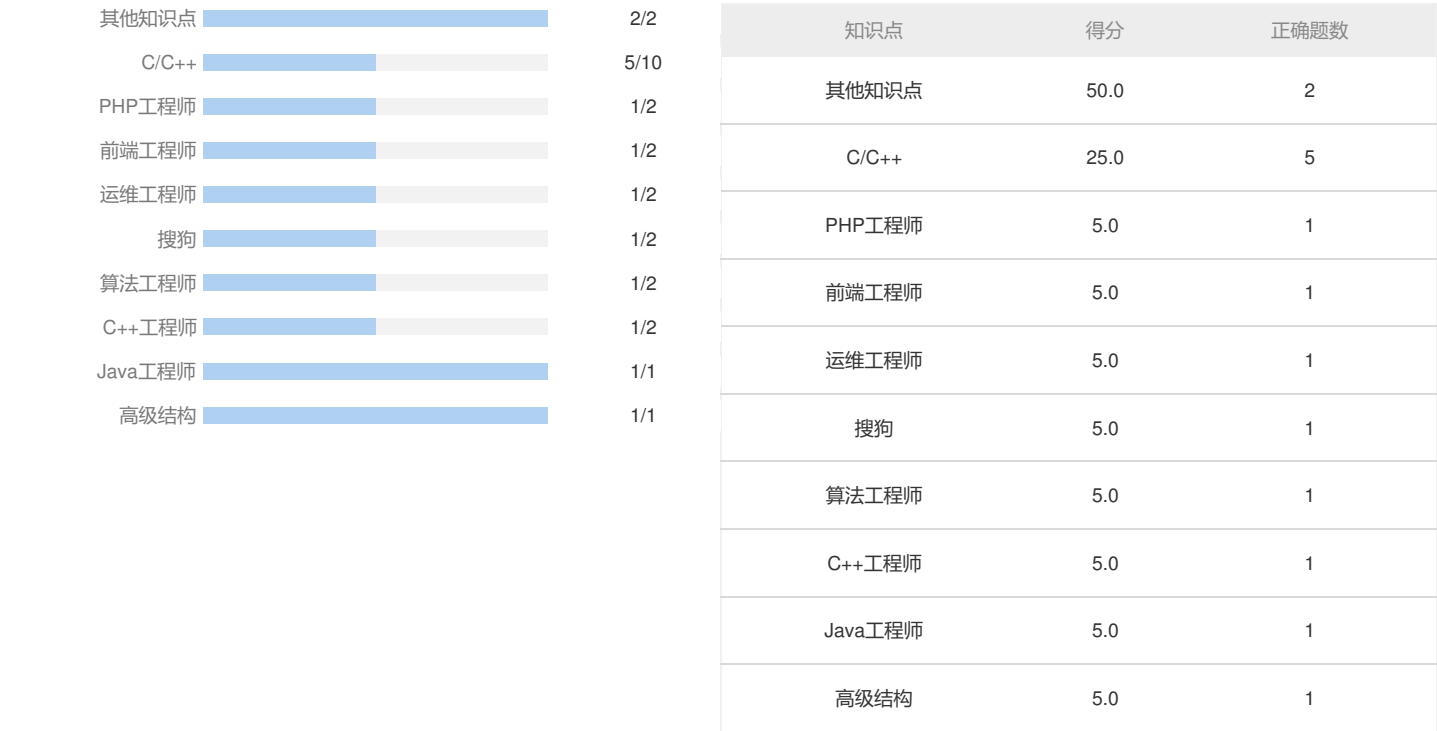
考号：1675 | 学校：陕西科技大学 | 邮箱：1761607418@qq.com | 职位：43班 |

参考区域: 陕西省西安市 ( 111.114.0.2 ) | 做题用时：02:13:27(2019-05-23 23:04:42 - 2019-05-24 01:18:11)

## 考生成绩



## 知识点技能图谱



## 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	笔试时间
1	每日一题day1_5月20日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-19 16:08:49
2	每日一题day02_5月21日	51.0%	60.0/100	单选:30.0分 编程:30.0分	否	2019-05-20 17:40:56
3	每日一题day03_5月22日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-21 12:59:15
4	每日一题day04_5月23日	8.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-22 10:40:18



1 [平均分4.0分 | 132人正确/165人做题 | 用时：<1分] 得分：5.0 / 5.0  
以下程序的输出结果是？

```
int x = 1;
do{
    printf("%2d\n",x++);
}while(x--);
```

- A 1
- B 无任何输出
- C 2
- D 陷入死循环

他的回答：D (正确)  
正确答案：D

2 [平均分3.3分 | 108人正确/164人做题 | 用时：2分] 得分：5.0 / 5.0  
下面两个结构体

```
struct One{
    double d;
    char c;
    int i;
}
struct Two{
    char c;
    double d;
    int i;
}
```

在#pragma pack(4)和#pragma pack(8)的情况下，结构体的大小分别是  
A 16 24,16 24  
B 16 20,16 20  
C 16 16,16 24  
D 16 16,24 24

他的回答：C (正确)  
正确答案：C

3 [平均分3.8分 | 127人正确/165人做题 | 用时：2分] 得分：5.0 / 5.0  
下列程序的打印结果是？

```
char p1[15]= "abcd",*p2= "ABCD", str[50]= "xyz";
strcpy(str+2,strcat(p1+2,p2+1));
printf("%s",str);
```

- A xyabcAB
- B abcABz
- C ABabcz
- D xycdBCD
- E 运行出错

他的回答： D (正确)

正确答案： D

4 [平均分3.5分 | 117人正确/165人做题 | 用时：2分  得分：5.0 / 5.0

下面程序的输出结果是

```
#include<iostream.h>
void main(){
    int n[][3] = {10,20,30,40,50,60};
    int (*p)[3];
    p=n;
    cout<<p[0][0]<<","<<*(p[0]+1)<<","<<(*p)[2]<<endl;
}
```

- A 10,30,50
- B 10,20,30
- C 20,40,60
- D 10,30,60

他的回答： B (正确)

正确答案： B

5 [平均分1.4分 | 46人正确/163人做题 | 用时：<1分  得分：0.0 / 5.0

```
int main(){
    int a;float b,c;
    scanf("%2d%3f%4f",&a,&b,&c);
    printf("\na=%d,b=%d,c=%f\n",a,b,c);
}
```

若运行时从键盘上输入9876543210I,则上面程序在gcc编译器下的输出结果是

- A a=98,b=765,c=4321.000000
- B a=98,b=0,c=0.000000
- C a=98,b=765.000000,c=4321.000000
- D a=98,b=765.0,c=4321.0

他的回答： A (错误)

正确答案： B

6 [平均分2.2分 | 71人正确/159人做题 | 用时：2分  得分：5.0 / 5.0

STL中的unordered\_map和priority\_queue使用的底层数据结构分别是什么?()

- A rbtree,queue
- B hashtable,heap
- C rbtree,heap
- D hashtable,queue

他的回答： B (正确)

正确答案： B

7 [平均分2.1分 | 68人正确/164人做题 | 用时：10分 | 得分：0.0 / 5.0]

下面说法正确的是()

- A 一个空类默认一定生成构造函数,拷贝构造函数,赋值操作符,引用操作符,析构函数
- B 可以有多个析构函数
- C 析构函数可以为virtual,可以被重载
- D 类的构造函数如果都不是public访问属性,则类的实例无法创建

他的回答： D (错误)

正确答案： A

8 [平均分2.7分 | 89人正确/165人做题 | 用时：<1分 | 得分：0.0 / 5.0]

```
ClassA *pclassa=new ClassA[5];
delete pclassa;
```

c++语言中，类ClassA的构造函数和析构函数的执行次数分别为()

- A 5,1
- B 1,1
- C 5,5
- D 1,5

他的回答： C (错误)

正确答案： A

9 [平均分2.4分 | 75人正确/157人做题 | 用时：3分 | 得分：0.0 / 5.0]

关于重载和多态正确的是

- A 如果父类和子类都有相同的方法,参数个数不同,将子类对象赋给父类后,由于子类继承于父类,所以使用父类指针 调用父类方法时,实际调用的是子类的方法
- B 选项全部都不正确
- C 重载和多态在C++面向对象编程中经常用到的方法,都只在实现子类的方法时才会使用
- D

```
class A{
    void test(float a){cout<<"1";}
};
class B:public A{
    void test(int b){cout<<"2";}
};
A *a=new A;
B *b=new B;
a=b;
a.test(1.1);
结果是1
```

他的回答： D (错误)

正确答案： B

10 [平均分1.2分 | 38人正确/154人做题 | 用时：4分 | 得分：0.0 / 5.0]

请选择下列程序的运行结果

```
#include<iostream>
```

```
using namespace std;
class B0//基类BO声明
{
public://外部接口
virtual void display()//虚成员函数
{
    cout<<"B0::display0"<<endl;}
};
class B1:public B0//公有派生
{
public:
    void display() { cout<<"B1::display0"<<endl; }
};
class D1: public B1//公有派生
{
public:
    void display(){ cout<<"D1::display0"<<endl; }
};
void fun(B0 ptr)//普通函数
{
    ptr.display();
}
int main()//主函数
{
    B0 b0;//声明基类对象和指针
    B1 b1;//声明派生类对象
    D1 d1;//声明派生类对象
    fun(b0);//调用基类B0函数成员
    fun(b1);//调用派生类B1函数成员
    fun(d1);//调用派生类D1函数成员
}
```

A B0::display0 B0::display0 B0::display0  
 B B0::display0 B0::display0 D1::display0  
 C B0::display0 B1::display0 D1::display0  
 D B0::display0 B1::display0 B1::display0

他的回答： **C (错误)**

正确答案： **A**

11 [平均分22.4分 | 116人正确/138人做题 | 提交: 1 次 | 得分: 25.0 / 25.0]

标题: 统计回文 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【统计回文】“回文串”是一个正读和反读都一样的字符串，比如“level”或者“noon”等等就是回文串。花花非常喜欢这种拥有对称美的回文串，生日的时候她得到两个礼物分别是字符串A和字符串B。现在她非常好奇有没有办法将字符串B插入字符串A使产生的字符串是一个回文串。你接受花花的请求，帮助她寻找有多少种插入办法可以使新串是一个回文串。如果字符串B插入的位置不同就考虑为不一样的办法。

例如：

A = “aba”，B = “b”。这里有4种把B插入A的办法：

- \* 在A的第一个字母之前: “baba” 不是回文
- \* 在第一个字母'a'之后: “abba” 是回文
- \* 在字母'b'之后: “abba” 是回文
- \* 在第二个字母'a'之后 “abab” 不是回文

所以满足条件的答案为2

输入描述：

每组输入数据共两行。

第一行为字符串A

第二行为字符串B

字符串长度均小于100且只包含小写字母

输出描述：

输出一个数字，表示把字符串B插入字符串A之后构成一个回文串的方法数

示例1：

输入

aba  
b

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	答案正确：1	
总通过率	100%	90%	使用语言	C++			
基本测试用例通过率	6/6 (100%)	91%	做题用时	00:28:21	00:44:08		
边缘测试用例通过率	4/4 (100%)	89%	提交次数	1	6		
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	4ms	1s					
占用内存	484K	32768K					

他的代码：

做题用时：28 分钟    语言：C++    运行时间：4ms    占用内存：484K    程序状态：答案正确

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str1;
    cin >> str1;
    string str2;
    cin >> str2;
    int count = 0;
    for(int i = 0; i < str1.size(); i++)
    {
        string temp = str1;
        temp.insert(i, str2);
        string temp2 = temp;
        temp2.assign(temp2.rbegin(), temp2.rend());
        if(temp2 == temp)
        {
            count++;
        }
    }
    string temp = str1;
    temp.append(str2);
    string temp2 = temp;
    temp2.assign(temp2.rbegin(), temp2.rend());
    if(temp2 == temp)
    {
```

```
count++;
}
cout << count << endl;
}
```

12 [平均分19.0分 | 77人正确/133人做题 | 提交: 6 次 | 得分 : 25.0 / 25.0

标题：连续最大和 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【连续最大和】一个数组有 N 个元素，求连续子数组的最大和。例如：[-1,2,1]，和最大的连续子数组为[2,1]，其和为 3

输入描述：

输入为两行。第一行一个整数n(1 <= n <= 100000)，表示一共有n个元素 第二行为n个数，即每个元素,每个整数都在32位int范围内。以空格分隔。

输出描述：

所有连续子数组中和最大的值。

示例1：

输入

3 -1 2 1

输出

3

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 00:52:28	答案错误 : 3
基本测试用例通过率	6/6 (100%)	79%	做题用时	01:08:59	00:52:28	编译错误 : 2
边缘测试用例通过率	4/4 (100%)	70%	提交次数	6	8	答案正确 : 1
代码效率					代码规范及可读性	
运行时间	TA的 40ms	参考 1s				
占用内存	792K	32768K				
			代码规范得分 0.0			

他的代码：

做题用时: 68 分钟    语言：C++    运行时间：40ms    占用内存：792K    程序状态：答案正确

```
#include <iostream>
#include <limits.h>
using namespace std;
int main()
{
    int num = 0;
    cin >> num;
    int curMax = 0;
    int max = INT_MIN;
    for(int i = 0; i < num; i++)
    {
        int temp = 0;
        cin >> temp;
        curMax += temp;
        if(curMax > max)
```

```
{  
    max = curMax;  
}  
if(curMax < 0)  
{  
    curMax = 0;  
}  
}  
cout << max << endl;  
}
```