

每日一题day08_5月28日测评结果

考生信息

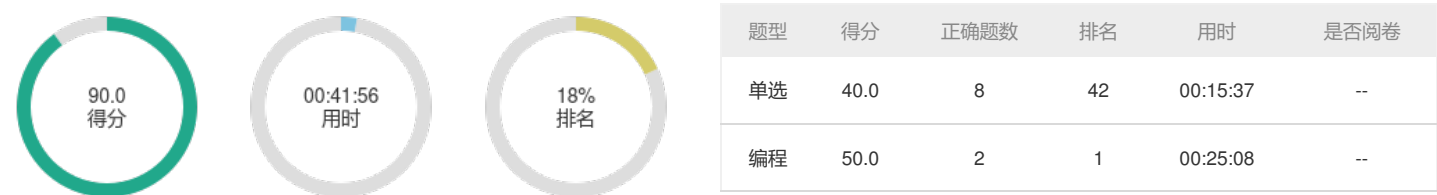


张博翔

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：00:41:56(2019-05-27 23:21:58 - 2019-05-28 00:04:05)

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
5	每日一题day05_5月24日	29.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-05-22 20:39:16
6	每日一题day06_5月25日	10.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-24 10:41:21
7	每日一题day07_5月27日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-26 15:09:41

编码能力



1 [平均分3.7分 | 121人正确/164人做题 | 用时：<1分] 得分：5.0 / 5.0

下面关于"指针"的描述不正确的是()

- A 当使用free释放掉一个指针内容后,指针变量的值被置为NULL
- B 32位系统下任何类型指针的长度都是4个字节
- C 指针的数据类型声明的是指针实际指向内容的数据类型
- D 野指针是指向未分配或者已经释放的内存地址

他的回答：A (正确)

正确答案：A

2 [平均分3.4分 | 111人正确/162人做题 | 用时：7分] 得分：0.0 / 5.0

以下程序统计给定输入中每个大写字母的出现次数(不需要检查输入合法性)

```
void AlphabetCounting(char a[],int n){
    int count[26]={},i,kind=0;
    for(i=0;i<n;++i) (1);
    for(i=0;i<26;++i){
        if(++kind>1) putchar(';');
        printf("%c=%d", (2));
    }
}
```

以下能补全程序，正确功能的选项是()

- A ++count[a[i]-'Z'];'Z'-i,count['Z'-i]
- B ++count['A'-a[i]]; 'A'+i,count[i]
- C ++count[i];i,count[i]
- D ++count['Z'-a[i]]; 'Z'-i,count[i]
- E ++count[a[i]]; 'A'+i,count[a[i]]

他的回答：B (错误)

正确答案：D

3 [平均分4.1分 | 133人正确/164人做题 | 用时：<1分] 得分：5.0 / 5.0

下列关于C/C++的宏定义，不正确的是（ ）

- A 宏定义不检查参数正确性，会有安全隐患
- B 宏定义的常量更容易理解，如果可以使用宏定义常量的话，要避免使用const常量
- C 宏的嵌套定义过多会影响程序的可读性，而且很容易出错
- D 相对于函数调用，宏定义可以提高程序的运行效率

他的回答：B (正确)

正确答案：B

4 [平均分3.5分 | 115人正确/164人做题 | 用时：<1分] 得分：5.0 / 5.0

下面代码会输出()

```
int main(){
    int a[4]={1,2,3,4};
    int *ptr=(int*)(&a+1);
    printf("%d",*(ptr-1));
}
```

A 4

B 1

- C 2
- D 3

他的回答： A (正确)

正确答案： A

5 [平均分2.6分 | 86人正确/164人做题 | 用时：<1分 | 得分：5.0 / 5.0

请找出下面程序中有哪些错误：

```
int main()
{
    int i=10;
    int j=1;
    const int *p1;//(1)
    int const *p2=&i; //(2)
    p2=&j;//(3)
    int *const p3=&i;//(4)
    *p3=20;//(5)
    *p2=30;//(6)
    p3=&j;//(7)
    return 0;
}
```

- A 1,2,3,4,5,6,7
- B 1,3,5,6
- C 6,7
- D 3,5

他的回答： C (正确)

正确答案： C

6 [平均分4.3分 | 140人正确/162人做题 | 用时：<1分 | 得分：5.0 / 5.0

在公有派生的情况下，派生类中定义的成员函数只能访问原基类的()

- A 公有成员和私有成员
- B 私有成员和保护成员
- C 公有成员和保护成员
- D 私有成员,保护成员和公有成员

他的回答： C (正确)

正确答案： C

7 [平均分3.4分 | 111人正确/163人做题 | 用时：<1分 | 得分：5.0 / 5.0

假定有类AB，有相应的构造函数定义，能正确执行

```
AB a(4),b(5),c[3].*p[2]={&a,&b};
```

语句，请问执行完此语句后共调用该类的构造函数次数为____

- A 5
- B 4
- C 3
- D 9

他的回答： A (正确)

正确答案： A

8 [平均分3.9分 | 124人正确/157人做题 | 用时：<1分] 得分：5.0 / 5.0

关于函数的描述正确的是_____。

- A 虚函数是一个static型的函数
- B 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
- C 虚函数是一个非成员函数
- D 基类中说明了虚函数后，派生类中起对应的函数可以不必说明为虚函数

他的回答： D (正确)

正确答案： D

9 [平均分3.7分 | 122人正确/163人做题 | 用时：<1分] 得分：5.0 / 5.0

有如下程序，执行后输出的结果是()

```
#include <iostream.h>
class cla{
    static int n;
public:
    cla(){n++;}
    ~cla(){n--;}
    static int get_n(){return n;}
};
int cla::n= 0;
int main()
{
    cla *p =new cla;
    delete p;
    cout<<"n="<<cla::get_n()<<endl;
    return 0;
}
```

- A n=3
- B n=4
- C n=1
- D n=0

他的回答： D (正确)

正确答案： D

10 [平均分2.1分 | 63人正确/153人做题 | 用时：<1分] 得分：0.0 / 5.0

以下程序输出结果是_____

```
class A
{
public:
    A():m_iVal(0){test();}
    virtual void func() { std::cout<<m_iVal<<' ';}
    void test(){func();}
public:
    int m_iVal;
};
class B : public A
{
public:
    B(){test();}
    virtual void func()
    {
        ++m_iVal;
    }
}
```

```
        std::cout<<m_iVal<<' ';\n    }\n};\n\nint main(int argc ,char* argv[])\n{\n    A*p = new B;\n    p->test();\n    return 0;\n}
```

- A 1 0
- B 0 1
- C 0 1 2
- D 2 1 0
- E 不可预期
- F 以上都不对

他的回答： E (错误)

正确答案： C

11 [平均分20.9分 | 91人正确/134人做题 | 提交: 2 次 | 得分： 25.0 / 25.0

标题：两种排序方法 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【两种排序方法】考拉有n个字符串字符串，任意两个字符串长度都是不同的。考拉最近学习到有两种字符串的排序方法：1.根据字符串的字典序排序。例如：

"car" < "carriage" < "cats" < "doggies" < "koala"

2.根据字符串的长度排序。例如：

"car" < "cats" < "koala" < "doggies" < "carriage"

考拉想知道自己的这些字符串排列顺序是否满足这两种排序方法，考拉要忙着吃树叶，所以需要你来帮忙验证。

输入描述：

输入第一行为字符串个数n($n \leq 100$)
接下来的n行,每行一个字符串,字符串长度均小于100，均由小写字母组成

输出描述：

如果这些字符串是根据字典序排列而不是根据长度排列输出"lexicographically",

如果根据长度排列而不是字典序排列输出"lengths",

如果两种方式都符合输出"both"，否则输出"none"

示例1：

输入

3
a
aa
bbb

输出

both

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：2
总通过率	100%	84%	使用语言	C++		
基本测试用例通过率	6/6 (100%)	84%	做题用时	00:20:35	01:01:23	
边缘测试用例通过率	4/4 (100%)	83%	提交次数	2	7	

代码效率	代码规范及可读性
TA的 参考 运行时间 4ms 1s 占用内存 504K 32768K	代码规范得分 5.0

他的代码：

做题用时: 20 分钟 语言：C++ 运行时间：4ms 占用内存：504K 程序状态：答案正确

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    int num = 0;
    cin >> num;
    int lex = 1, len = 1;
    string str1, str2;
    cin >> str1;
    for(int i = 1; i < num; i++)
    {
        str2 = "";
        cin >> str2;
        if(str1 > str2)
        {
            lex = 0;
        }
        if(str1.size() > str2.size())
        {
            len = 0;
        }
        str1 = str2;
    }
    if(lex == 1 && len == 1)
    {
        cout << "both" << endl;
    }
    if(lex == 1 && len != 1)
    {
        cout << "lexicographically" << endl;
    }
    if(lex != 1 && len == 1)
    {
        cout << "lengths" << endl;
    }
    if(lex == 0 && len == 0)
    {
        cout << "none" << endl;
    }
}
```

12 [平均分24.7分 | 148人正确/152人做题 | 提交: 1 次 得分：25.0 / 25.0

标题：求最小公倍数 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【求最小公倍数】

正整数A和正整数B 的最小公倍数是指 能被A和B整除的最小的正整数，设计一个算法，求输入A和B的最小公倍数。

输入描述：

输入两个正整数A和B。

输出描述：

输出A和B的最小公倍数。

示例1：

输入

5
7

输出

35

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%	99%	使用语言		C++			
基本测试用例通过率		6/6 (100%)	99%	做题用时		00:04:33	00:19:11		
边缘测试用例通过率		4/4 (100%)	99%	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 5.0		
运行时间		4ms	1s						
占用内存		616K	32768K						

他的代码：

做题用时: 4 分钟 语言：C++ 运行时间：4ms 占用内存：616K 程序状态：答案正确

```
#include <iostream>
using namespace std;
int main()
{
    int num1 = 0;
    int num2 = 0;
    cin >> num1 >> num2;
    int max = 1;
    for(int i = 2; i <= (num1 < num2 ? num1 : num2); i++)
    {
        if(num1 % i == 0 && num2 % i == 0)
        {
            max = i;
        }
    }
    int min = num1 * num2 / max;
    cout << min << endl;
}
```