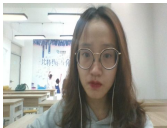


每日一题C++方向day14_11月21日测评结果

考生信息



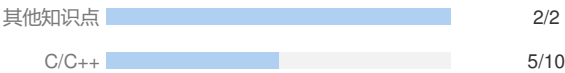
王婧
考号：2378 | 学校：财经大学 | 邮箱：1031160332@qq.com | 职位：54 |
参考区域: 陕西省西安市 (113.139.120.150) | 做题用时：00:55:55(2019-11-21 12:19:22 - 13:15:24) | 作答设备：PC

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	25.0	5	65	00:06:59	--
编程	50.0	2	1	00:48:12	--

知识点技能图谱



知识点	得分	正确题数
其他知识点	50.0	2
C/C++	25.0	5

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
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

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
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

编码能力



1 [平均分3.0分 | 90人正确/148人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0]

定义char dog[]="wang\0miao";那么sizeof(dog)与strlen(dog)分别是多少 :

- A 10,4
- B 4,4
- C 9,9
- D 9,4

他的回答 : A (正确)

正确答案 : A

2 [平均分3.2分 | 93人正确/145人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0]

假设寄存器为8位, 用补码形式存储机器数, 包括一位符号位, 那么十进制数-25在寄存器表示为 :

- A 67H
- B 99H
- C E6H
- D E7H

他的回答 : C (错误)

正确答案 : D

3 [平均分3.1分 | 91人正确/148人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0]

下面代码的执行结果是什么 :

```
char ccString1[]="Is Page Fault?";
char ccString2[]="No Page Fault?";
strcpy(ccString1,"No");
if(strcmp(ccString1,ccString2)==0)
    cout<<ccString2;
else
    cout<<ccString1;
```

- A No
- B No Page Fault??
- C Is Page Fault??
- D 其他三项都错

他的回答： A (正确)

正确答案： A

4 [平均分2.1分 | 58人正确/140人做题 | 用时：<1分 | 得分：0.0 / 5.0

```
int Function(unsigned int n) {  
  
    n = (n & 0x55555555) + ((n >> 1) & 0x55555555);  
    n = (n & 0x33333333) + ((n >> 2) & 0x33333333);  
    n = (n & 0x0f0f0f0f) + ((n >> 4) & 0x0f0f0f0f);  
    n = (n & 0x00ff00ff) + ((n >> 8) & 0x00ff00ff);  
    n = (n & 0x0000ffff) + ((n >> 16) & 0x0000ffff);  
  
    return n;  
}
```

输入参数为197时，函数返回多少？

- A 2
- B 3
- C 4
- D 5

他的回答： B (错误)

正确答案： C

5 [平均分2.2分 | 63人正确/144人做题 | 用时：<1分 | 得分：5.0 / 5.0

下面程序的功能是输出数组的全排列。请填空。

```
void perm(int list[], int k, int m)  
{  
    if ( )  
    {  
        copy(list,list+m,ostream_iterator<int>(cout," "));  
        cout<<endl;  
        return;  
    }  
    for (int i=k; i<=m; i++)  
    {  
        swap(&list[k],&list[i]);  
        ( );  
        swap(&list[k],&list[i]);  
    }  
}
```

- A $k!=m$ 和 $\text{perm}(\text{list}, k+1, m)$
- B $k==m$ 和 $\text{perm}(\text{list}, k+1, m)$
- C $k!=m$ 和 $\text{perm}(\text{list}, k, m)$
- D $k==m$ 和 $\text{perm}(\text{list}, k, m)$

他的回答： B (正确)

正确答案： B

6 [平均分3.9分 | 116人正确/147人做题 | 用时：<1分 | 得分：0.0 / 5.0

C++中以下关于函数调用的说法哪个是正确的？

- A 传地址后实参和形参指向不同的对象
- B 传引用后实参和形参是不同的对象
- C 传值后对形参的修改会改变实参的值
- D 其他三项都不对

他的回答： A (错误)

正确答案： D

7 [平均分1.7分 | 50人正确/147人做题 | 用时：<1分 | 得分：0.0 / 5.0

有一个类B继承自类A，他们数据成员如下：

```
class A {  
    ...  
private :  
    int a;  
};  
class B : public A {  
    ...  
private:  
    int a;  
public:  
    const int b;  
    A &c;  
    static const char* d;  
    B* e;  
}
```

则构造函数中，成员变量一定要通过初始化列表来初始化的是_____。

- A a b c
- B b c e
- C b c d e
- D c e
- E b d
- F b c

他的回答： E (错误)

正确答案： F

8 [平均分2.2分 | 66人正确/147人做题 | 用时：<1分 | 得分：5.0 / 5.0

以下代码编译有错误，哪个选项能解决编译错误？

```
class A {  
    public:  
        int GetValue() const {  
            vv = 1;  
            return vv;  
        }  
    private:  
        int vv;  
};
```

- A 改变成员变量"vv"为"mutable int vv"
- B 改变成员函数"GetValue"的声明，以使其不是const的
- C 都不能修复编译错误
- D 都可以修复编译错误

他的回答： D (正确)

正确答案：D

9 [平均分2.4分 | 72人正确/147人做题 | 用时：<1分] 得分：0.0 / 5.0

下面程序运行后的结果为：

```
char str[] = "glad to test something";
char *p = str;
p++;
int *p1 = reinterpret_cast<int *>(p);
p1++;
p = reinterpret_cast<char *>(p1);
printf("result is %s\n", p);
```

- A result is glad to test something
- B result is ad to test something
- C result is test something
- D result is to test something

他的回答：C (错误)

正确答案：D

10 [平均分2.0分 | 59人正确/146人做题 | 用时：<1分] 得分：5.0 / 5.0

以下程序的输出是

```
class Base {
public:
    Base(int j): i(j) {}
    virtual ~Base() {}
    void func1() {
        i *= 10;
        func2();
    }
    int getValue() {
        return i;
    }
protected:
    virtual void func2() {
        i++;
    }
protected:
    int i;
};

class Child: public Base {
public:
    Child(int j): Base(j) {}
    void func1() {
        i *= 100;
        func2();
    }
protected:
    void func2() {
        i += 2;
    }
};

int main() {
    Base * pb = new Child(1);
```

```
pb->func1();
cout << pb->getValue() << endl; delete pb; }
```

- A 11
- B 101
- C 12
- D 102

他的回答： C (正确)

正确答案： C

11 [平均分22.6分 | 117人正确/133人做题 | 提交: 14 次 | 得分: 25.0 / 25.0

标题：计算日期到天数转换 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【计算日期到天数转换】

根据输入的日期，计算是这一年的第几天。。

详细描述：

输入某年某月某日，判断这一天是这一年的第几天？。

接口设计及说明：

/******

Description : 数据转换

Input Param : year 输入年份

Month 输入月份

Day 输入天

Output Param :

Return Value : 成功返回0，失败返回-1（如：数据错误）

*****/

public static int iConverDateToDay(int year, int month, int day)

{

/* 在这里实现功能，将结果填入输入数组中*/

return 0;

}

/******

Description :

Input Param :

Output Param :

Return Value : 成功:返回outDay输出计算后的第几天;

失败:返回-1

*****/

public static int getOutDay()

{

return 0;

}

输入描述：

输入三行，分别是年，月，日

输出描述：

成功:返回outDay输出计算后的第几天;

失败:返回-1

示例1：

输入

2012

12

31

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	100%	90%	使用语言	C++		答案错误 : 8	
基本测试用例通过率	6/6 (100%)	90%	做题用时	00:40:00	00:33:03	编译错误 : 4	
边缘测试用例通过率	4/4 (100%)	90%	提交次数	14	6	答案正确 : 2	
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	484K	32768K					
					代码规范得分 5.0		

他的代码：

做题用时: 40 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>
using namespace std;
const int days[] = { 0,31,59,90,120,151,181,212,243,273,304,334,365 };
int main() {
    int y, m, d;
    while (cin >> y >> m >> d) {
        int ans = days[m - 1] + d;
        if (((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) && m > 2)
            ans++;
        cout << ans << endl;
    }
    return 0;
}
```

12 [平均分15.3分 | 56人正确/96人做题 | 提交: 1 次 | 得分: 25.0 / 25.0]

标题：幸运的袋子 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【幸运的袋子】一个袋子里面有n个球，每个球上面都有一个号码(拥有相同号码的球是无区别的)。如果一个袋子是幸运的当且仅当所有球的号码的和大于所有球的号码的积。

例如：如果袋子里面的球的号码是{1, 1, 2, 3}，这个袋子就是幸运的，因为1 + 1 + 2 + 3 > 1 * 1 * 2 * 3

你可以适当从袋子里移除一些球(可以移除0个,但是别移除完)，要使移除后的袋子是幸运的。现在让你编程计算一下你可以获得的多少种不同的幸运的袋子。

输入描述：

```
第一行输入一个正整数n(n ≤ 1000)
第二行为n个数正整数xi(xi ≤ 1000)
```

输出描述：

```
输出可以产生的幸运的袋子数
```

示例1：

输入

```
3
```

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%	61%	使用语言	C++			
基本测试用例通过率	6/6 (100%)	62%	做题用时	00:08:12	00:54:11		
边缘测试用例通过率	4/4 (100%)	59%	提交次数	1	4		
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	15ms	1s					
占用内存	376K	32768K					

他的代码：

做题用时: 8 分钟 语言: C++ 运行时间: 15ms 占用内存: 376K 程序状态: 答案正确

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
int cont = 0;
void ContOne(int, vector<int>, int, int, int); // ( 1 的个数, 非1数, 当前位数, 和, 积 )
int main(void)
{
    int n, numb, OneCont(0);    //OneCont用来记录1的总个数
    vector<int> list;
    cin >> n;
    while (n-->0)
    {
        cin >> numb;
        if (numb > 1)
            list.push_back(numb);
        else
            OneCont++;
    }
    cont = OneCont - 1; //将纯1数组先计算出来
    if (!list.empty())
    {
        sort(list.begin(), list.end()); //对数组排序
        for (int i = 1; i <= OneCont; i++) ContOne(i, list, 0, i, 1);
    }
    cout << cont << endl;
    return 0;
}

void ContOne(int OneNub, vector<int> list, int nowBit, int sum, int mult)
{
    for (int i = nowBit; i < list.size(); i++) //迭代计算和与积
    {
        sum += list[i], mult *= list[i];
        if (sum <= mult) //如果不符合，回溯至上一位
```



```
        break;
    else
        cont++;
    ContOne(OneNub, list, i + 1, sum, mult);
    sum -= list[i], mult /= list[i]; //回到上一位后需要恢复数值
    while (i < list.size() - 1 && list[i] == list[i + 1])
        i++; //寻找不重复
    }
}
```