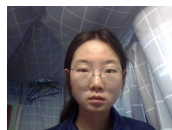


C++方向每日一题day15_5月12日-王梅-测评结果

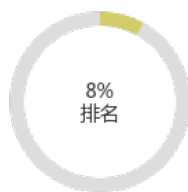
考生信息



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（123.139.235.251） | 做题用时：01:25:11(2021-05-12 12:09:02开始答题，18:58:48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	35.00	7	9	00:51:53	已阅
编程	50.00	2	1	00:30:37	已阅

知识点技能图谱

其他知识点	4/4
C/C++	2/2
C++	2/5
模拟	1/1
贪心	1/1

知识点	得分	正确题数
其他知识点	40.00	4
C/C++	10.00	2
C++	10.00	2
模拟	25.00	1
贪心	25.00	1

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	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
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	2021-04-19 14:53:21	2021-04-20 20:22:40
5	C++方向每日一题day03_4月21日	7.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-20 11:16:00	2021-04-21 16:09:24

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
6	C++方向每日一题day04_4月22日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-21 16:19:39	2021-04-22 16:12:03
7	C++方向每日一题day05_4月23日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 15:09:26	2021-04-23 10:24:19
8	C++方向每日一题day06_4月24日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 15:21:12	2021-04-24 17:35:03
9	C++方向每日一题day07_4月26日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-04-22 16:20:07	2021-04-26 18:10:43
10	C++方向每日一题day08_4月27日	6.0%	95.0/100	单选:50.0分 编程:45.0分	否	2021-04-24 15:13:08	2021-04-27 15:21:45
11	91班&92班CPP1考试试卷	26.0%	68.0/80	单选:18.0分 编程:50.0分	否	2021-04-26 14:46:52	2021-04-28 20:07:07
12	C++方向每日一题day09_4月28日	18.0%	90.0/100	单选:40.0分 编程:50.0分	否	2021-04-27 16:33:22	2021-04-28 14:46:16
13	C++方向每日一题day10_4月29日	35.0%	78.1/100	单选:30.0分 编程:48.08分	否	2021-04-28 11:39:52	2021-04-29 07:43:41
14	C++方向每日一题day11_5月7日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-06 15:16:10	2021-05-06 22:42:17
15	C++方向每日一题day12_5月8日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-07 14:24:30	2021-05-08 12:57:48
16	C++方向每日一题day13_5月10日	1.0%	75.0/100	单选:50.0分 编程:25.0分	否	2021-05-09 11:22:20	2021-05-10 08:10:17
17	C++方向每日一题day14_5月11日	23.0%	70.0/100	单选:40.0分 编程:30.0分	否	2021-05-10 10:13:17	2021-05-10 22:06:18

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%	2	00:04:47	C++	3ms	380K			1%
编程题2	100%	3	00:25:50	C++	2ms	420K			1%

1

[平均分3.45分 | 40人正确/58人做题 | 用时：6分 | 得分：5.00 / 5.00]

对两个字符a和b进行初始化:char a[]="ABCDEF";char b[]={'A','B','C','D','E','F'};则以下叙述正确的是：

A a数组比b数组长度长

B a与b长度相同

C a与b数组完全相同

D a和b中都存放字符串

他的回答：A (正确)

正确答案：A

2 [平均分3.16分 | 36人正确/57人做题 | 用时：5分 得分：0.00 / 5.00

x是一个行列数均为1000二维数组，下面代码效率执行最高的是（ ）

- A for(int j=0;j<1000;j++) for(int i=0;i<1000;i++) x[i][j]+=x[j][i];
- B for(int i=0;i<1000;i++) for(int j=0;j<1000;j++) x[i][j]+=x[j][i];
- C for(int i=0;i<1000;i++) for(int j=0;j<1000;j++) x[j][i]+=x[j][i];
- D for(int i=0;i<1000;i++) for(int j=0;j<1000;j++) x[i][j]+=x[i][j];

他的回答： A (错误)

正确答案： D

3 [平均分3.02分 | 35人正确/58人做题 | 用时：<1分 得分：5.00 / 5.00

C++中关于堆和栈的说法，哪个是错误的:

- A 堆的大小仅受操作系统的限制，栈的大小一般一般较小
- B 在堆上频繁的调用new/delete容易产生内存碎片，栈没有这个问题
- C 堆和栈都可以静态分配
- D 堆和栈都可以动态分配

他的回答： C (正确)

正确答案： C

4 [平均分2.59分 | 30人正确/58人做题 | 用时：2分 得分：5.00 / 5.00

下面程序会输出什么：

```
static int a=1;
void fun1(void){ a=2; }
void fun2(void){ int a=3; }
void fun3(void){ static int a=4; }
int main(int argc,char** args){
    printf( "%d" ,a);
    fun1( );
    printf( "%d" ,a);
    fun2( );
    printf( "%d" , a);
    fun3( );
    printf( "%d" ,a);
}
```

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

他的回答： B (正确)

正确答案： B

5 [平均分2.41分 | 27人正确/56人做题 | 用时：17分 得分：5.00 / 5.00

In the main() function, after ModifyString(text) is called, what' s the value of 'text' ?

```
int FindSubString( char* pch )
{
    int count = 0;
    char * p1 = pch;
    while ( *p1 != '\0' )
```

```

{
    if ( *p1 == p1[1] - 1 )
    {
        p1++;
        count++;
    }else {
        break;
    }
}
int count2 = count;
while ( *p1 != '\0' )
{
    if ( *p1 == p1[1] + 1 )
    {
        p1++;
        count2--;
    }else {
        break;
    }
}
if ( count2 == 0 )
    return(count);
return(0);
}
void ModifyString( char* pText )
{
    char * p1  = pText;
    char * p2  = p1;
    while ( *p1 != '\0' )
    {
        int count = FindSubString( p1 );
        if ( count > 0 )
        {
            *p2++ = *p1;
            sprintf( p2, "%i", count );
            while ( *p2 != '\0' )
            {
                p2++;
            }
            p1 += count + count + 1;
        }else {
            *p2++ = *p1++;
        }
    }
}
void main( void )
{
    char text[32] = "XYBCDCBABABA";
    ModifyString( text );
    printf( text );
}

```

- A XYBCDCBABABA
- B XYBCBCDA1BAA
- C XYBCDCBA1BAA
- D XYBCDDBA1BAB

他的回答： C (正确)

正确答案： C

6 [平均分4.25分 | 51人正确/60人做题 | 用时：<1分] 得分：5.00 / 5.00

所谓数据封装就是将一组数据和与这组数据有关操作组装在一起，形成一个集合，这集合也就是（ ）

- A 类
- B 对象
- C 函数体
- D 数据块

他的回答：A (正确)

正确答案：A

7 [平均分1.81分 | 21人正确/58人做题 | 用时：2分] 得分：0.00 / 5.00

关于以下代码，哪个说法是正确的？

```
myClass::foo(){
    delete this;
}
..
void func(){
    myClass *a = new myClass();
    a->foo();
}
```

- A 它会引起栈溢出
- B 都不正确
- C 它不能编译
- D 它会引起段错误

他的回答：A (错误)

正确答案：B

8 [平均分0.86分 | 10人正确/58人做题 | 用时：3分] 得分：0.00 / 5.00

假定CSomething是一个类，执行下面这些语句之后，内存里创建了___个CSomething对象。

```
CSomething a();
CSomething b(2);
CSomething c[3];
CSomething &ra = b;
CSomething d=b;
CSomething *pA = c;
CSomething *p = new CSomething(4);
```

- A 10
- B 9
- C 8
- D 7
- E 6
- F 5

他的回答：F (错误)

正确答案：E

9 [平均分2.98分 | 34人正确/57人做题 | 用时：6分] 得分：5.00 / 5.00

下面这段代码运行时会出现什么问题？

```
class A
{
```

```

public:
    void f()
    {
        printf("A\n");
    }
};

class B: public A
{
public:
    virtual void f()
    {
        printf("B\n");
    }
};

int main()
{
    A *a = new B;
    a->f();
    delete a;
    return 0;
}

```

- A 没有问题，输出B
- B 不符合预期的输出A
- C 程序不正确
- D 以上答案都不正确

他的回答： B (正确)

正确答案： B

10 [平均分2.54分 | 29人正确/57人做题 | 用时：3分] 得分：5.00 / 5.00

下面这段代码会打印出什么？

```

class A
{
public:
    A()
    {
        printf("A ");
    }

    ~A()
    {
        printf("deA ");
    }
};

class B
{
public:
    B()
    {
        printf("B ");
    }
}

```

```
    }
    ~B()
    {
        printf("deB ");
    }
};

class C: public A, public B
{
public:
    C()
    {
        printf("C ");
    }
    ~C()
    {
        printf("deC ");
    }
};

int main()
{
    A *a = new C();
    delete a;
    return 0;
}
```

A A B C deA
B C A B deA
C A B C deC
D C A B deC

他的回答： A (正确)

正确答案： A

11 ACM编程题 语言限制 [平均分20.54分 | 46人正确/56人做题 | 提交: 2 次 得分：25.00 / 25.00]

标题：查找输入整数二进制中1的个数 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【查找输入整数二进制中1的个数】

请实现如下接口

```
public static int findNumberOf1( int num)
{
    /* 请实现 */
    return 0;
} 譬如：输入5，5的二进制为101，输出2
```

涉及知识点：

输入描述：

输入一个整数

输出描述：

计算整数二进制中1的个数

示例1：

输入

5

输出

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 82%	使用语言	C++		答案错误 : 1 答案正确 : 1	
基本测试用例通过率	6/6 (100%)	82%	做题用时	00:04:47	00:10:53		
边缘测试用例通过率	4/4 (100%)	82%	提交次数	2	4		
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			5.0	
占用内存	380K	32768K					

他的代码：

做题用时: 4 分钟 语言: C++ 运行时间: 3ms 占用内存: 380K 程序状态: 答案正确

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    while(cin >> n)
    {
        int count = 0;
        while (n)
        {
            if (n % 2 == 1)
            {
                count++;
            }
            n = n / 2;
        }
        cout << count << endl;
    }
    return 0;
}
```



标题：手套 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【手套】

在地下室里放着n种颜色的手套，手套分左右手，但是每种颜色的左右手手套个数不一定相同。A先生现在要出门，所以他要去地下室选手套。但是昏暗的灯光让他无法分辨手套的颜色，只能分辨出左右手。所以他会多拿一些手套，然后选出一双颜色相同的左右手手套。现在的问题是，他至少要拿多少只手套(左手加右手)，才能保证一定能选出一双颜色相同的手套。

给定颜色种数n(1≤n≤13),同时给定两个长度为n的数组left,right,分别代表每种颜色左右手手套的数量。数据保证左右的手套总数均不超过26，且一定存在至少一种合法方案。

测试样例：

4,[0,7,1,6],[1,5,0,6]

返回：10(解释：可以左手手套取2只，右手手套取8只)

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	答案正确 : 3	
总通过率		100%	45%	使用语言		C++			
基本测试用例通过率		1/1 (100%)	45%	做题用时		00:25:50 00:34:33			
				提交次数		3 3			
代码效率							代码规范及可读性		
		TA的	参考				代码规范得分		4.2
运行时间		2ms	3s				Line 8: Else clause should be indented at the same level as if. Ambiguous nested if/else chains require braces. [readability/braces] [4]		
占用内存		420K	32768K				Line 10: If/else bodies with multiple statements require braces [readability/braces] [4]		
							Line 13: Add #include for sort [build/include_what_you_use] [4]		
							Line 4: Add #include for vector<> [build/include_what_you_use] [4]		



[点此](#)或手机扫描二维码查看代码编写过程