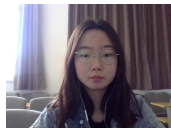


C++方向每日一题day09_4月28日-王梅-测评结果

考生信息



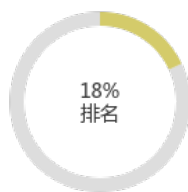
王梅

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（123.139.92.224） | 做题用时：01:56:40(2021-04-28 12:49:25开始答题，14:46:16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	40.0	8	16	00:34:48	已阅
编程	50.0	2	1	01:17:13	已阅

知识点技能图谱

C/C++	<div><div></div></div>	3/4
C++	<div><div></div></div>	5/6
字符串	<div><div></div></div>	1/1
位运算	<div><div></div></div>	1/1

知识点	得分	正确题数
C/C++	15.0	3
C++	25.0	5
字符串	25.0	1
位运算	25.0	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
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

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
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	28.0%	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

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	00:03:01	C++	3ms	520K	优	优	1%
编程题2	100%	15	01:14:12	C++	2ms	396K	优	优	1%

1 [平均分3.7分 | 39人正确/53人做题 | 用时：5分] 得分：5.0 / 5.0

请声明一个指针，其所指向的内存地址不能改变，但内存中的值可以被改变。

A const int const *x = &y;
B int * const x = &y;
C const int *x = &y;
D int const *x = &y;
E const int * const x = &y;

他的回答：B (正确)

正确答案：B

2 [平均分4.5分 | 47人正确/52人做题 | 用时：<1分] 得分：0.0 / 5.0

以下说法中正确的是（ ）。

A C++程序中的main()函数必须放在程序的开始部分
B C++程序的入口函数是main函数
C 在C++程序中，要调用的函数必须在main()函数中

他的回答：C (错误)

正确答案：B

3 [平均分4.1分 | 42人正确/51人做题 | 用时：4分] 得分：5.0 / 5.0

下面哪个指针表达式可以用来引用数组元素a[i][j][k][l]

A (((a+i)+j)+k)+l
B *((*(a+i)+j)+k)+l
C (((a+i)+j)+k+l)

D ((a+i)+j+k+l)

他的回答： B (正确)

正确答案： B

4 [平均分4.3分 | 43人正确/50人做题 | 用时：5分  得分：5.0 / 5.0

```
int fun(int a){
    a^=(1<<5)-1;
    return a;
}
```

fun(21)运行结果是()

A 10

B 5

C 3

D 8

他的回答： A (正确)

正确答案： A

5 [平均分2.8分 | 29人正确/52人做题 | 用时：2分  得分：5.0 / 5.0

以下程序的输出结果是：

```
#include <iostream>
using namespace std;
void func(char **m){
    ++m;
    cout<<*m<<endl;
}
int main(){
    static char *a[]={"morning", "afternoon", "evening"};
    char **p;
    p=a;
    func(p);
    return 0;
}
```

A afternoon

B 字符o的起始地址

C 字符o

D 字符a的起始地址

他的回答： A (正确)

正确答案： A

6 [平均分3.9分 | 41人正确/53人做题 | 用时：<1分  得分：5.0 / 5.0

下面对析构函数的正确描述是：

A 系统不能提供默认的析构函数

B 析构函数必须由用户定义

C 析构函数没有参数

D 析构函数可以设置默认参数

他的回答： C (正确)

正确答案：C

7 [平均分3.2分 | 33人正确/51人做题 | 用时：2分 | 得分：0.0 / 5.0]

某函数申明如下：

```
void Func(int &nVal1);
```

有int a,下面使用正确的为：

- A Func(a)
- B Func(&a)
- C Func(*a)
- D Func(&(*a))

他的回答：C (错误)

正确答案：A

8 [平均分3.2分 | 33人正确/51人做题 | 用时：<1分 | 得分：5.0 / 5.0]

当一个类的某个函数被说明为virtual，则在该类的所有派生类中的同原型函数_____？

- A 只有 被重新说明时才识虚函数
- B 只有被重新说明为virtual时才是虚函数
- C 都不是虚函数
- D 都是虚函数

他的回答：D (正确)

正确答案：D

9 [平均分3.6分 | 36人正确/50人做题 | 用时：4分 | 得分：5.0 / 5.0]

有如下程序：

```
#include<iostream>
using namespace std;
class MyClass{
public:
    MyClass(int i=0){cout<<1;}
    MyClass(const MyClass&x){cout<<2;}
    MyClass& operator=(const MyClass&x){cout<<3; return*this;}
    ~MyClass(){cout<<4;}
};
int main(){
    MyClass obj1(1),obj2(2),obj3(obj1);
    return 0;
}
```

运行时的输出结果是

- A 121,444
- B 112,444
- C 11,114,444
- D 11,314,445
- E 11,314,444

他的回答：B (正确)

正确答案：B

10 [平均分1.8分 | 18人正确/51人做题 | 用时 : 5分 得分 : 5.0 / 5.0

代码执行后，a和b的值分别为？

```
class Test{
public:
    int a;
    int b;
    virtual void fun() {}
    Test(int temp1 = 0, int temp2 = 0)
    {
        a=temp1 ;
        b=temp2 ;
    }
    int getA()
    {
        return a;
    }
    int getB()
    {
        return b;
    }
};

int main()
{
    Test obj(5, 10);
    // Changing a and b
    int* pInt = (int*)&obj;
    *(pInt+0) = 100;
    *(pInt+1) = 200;
    cout << "a = " << obj.getA() << endl;
    cout << "b = " << obj.getB() << endl;
    return 0;
}
```

- A 200 10
- B 5 10
- C 100 200
- D 100 10

他的回答： A (正确)

正确答案： A

11 完善核心代码 语言限制 [平均分20.9分 | 41人正确/49人做题 | 提交: 2 次 得分 : 25.0 / 25.0

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

【另类加法】

请编写一个函数，将两个数字相加。不得使用+或其他算数运算符。

给定两个int **A**和**B**。请返回A + B的值

测试样例：

1,2

返回：3

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
总通过率	TA的 100%	平均 83%	使用语言	TA的 C++	平均	答案正确 : 2
基本测试用例通过率	1/1 (100%)	83%	做题用时	00:03:01	00:21:47	
			提交次数	2	3	
代码效率					代码规范及可读性	
运行时间	TA的 3ms	参考 3s	代码规范得分 5.0			
占用内存	520K	32768K				

他的代码：

做题用时: 3 分钟 语言: C++ 运行时间: 3ms 占用内存: 520K 程序状态: 答案正确

```
class UnusualAdd {
public:
    int addAB(int A, int B) {
        for(int i = B; B>0; B--)
        {
            A++;
        }
        return A;
    }
};
```

```
class UnusualAdd {
public:
    int addAB(int A, int B) {
        if(A == 0)
            return B;
        if(B == 0)
            return A;
        int a = A^B; //求和后当前位的数据
        int b = (A&B)<<1; //求和后进位的数据
        return addAB(a, b); //递归两个数进行相加，任意为0时截止
    }
};
```



[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17.0分 | 31人正确/47人做题 | 提交: 15 次] 得分: 25.0 / 25.0

标题：201301 JAVA 题目2-3级 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【201301 JAVA 题目2-3级】

请编写一个函数（允许增加子函数），计算 $n \times m$ 的棋盘格子（ n 为横向的格子数， m 为竖向的格子数）沿着各自边缘线从左上角走到右下角，总共有多少种走法，要求不能走回头路，即：只能往右和往下走，不能往左和往上走。

输入描述：

输入两个正整数

输出描述：

返回结果

示例1：

输入

2
2

输出

6

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		答案错误	: 7
总通过率	100%	67%	使用语言	C++		返回非零	: 1	
基本测试用例通过率	6/6 (100%)	68%	做题用时	01:14:12	00:53:07	运行超时	: 1	
边缘测试用例通过率	4/4 (100%)	67%	提交次数	15	6	运行错误	: 1	
						编译错误	: 3	
						答案正确	: 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	2ms	1s						
占用内存	396K	32768K						

他的代码：

做题用时: 74 分钟 语言: C++ 运行时间: 2ms 占用内存: 396K 程序状态: 答案正确

```
#include<iostream>
using namespace std;
#include<vector>
int chess(int n, int m)
{
    if (n == 0 || m == 0)
    {
        return 1;
    }
    return chess(n - 1, m) + chess(n, m - 1);
}
int main()
{
    int n, m;
    while (cin >> m >> n)
    {
        cout << chess(m, n) << endl;
    }
    return 0;
}

#include<iostream>
using namespace std;
int path(int n, int m)
{
    if(n > 1 && m > 1)
    {
        //b情况，递归
        return path(n-1, m) + path(n, m-1);
    }
    else if(((n >= 1) && (m == 1)) || ((n == 1) && (m >=1)))
    {
        //a情况，终止条件
        return n+m;
    }
    else
        //格子为0时，路径为0
        return 0;
}
int main()
{
    int n, m;
    while(cin >> n >> m)
    {
        cout << path(n, m)<< endl;
    }
    return 0;
}
```



[点此](#)或手机扫描二维码查看代码编写过程