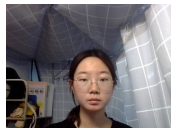


C++方向每日一题day13_5月10日-王梅-测评结果

考生信息



王梅

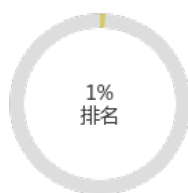
投递编号：62 | 学校：陕西科技大学 | 邮箱：2910602199@qq.com | 职位：91班 |

参考区域：陕西省西安市（123.139.169.172） |

做题用时：01:48:32(2021-05-09 21:55:30开始答题，2021-05-10 08:10:17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	50.00	10	1	00:32:05	已阅
编程	25.00	1	1	01:02:43	已阅

知识点技能图谱

C/C++	4/4
C++	6/6
字符串	0/1
贪心	1/1
动态规划	1/1

知识点	得分	正确题数
C/C++	20.00	4
C++	30.00	6
字符串	0.00	0
贪心	25.00	1
动态规划	25.00	1

历史笔记录录

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	得分	得分详情	作弊嫌疑	2021-04-21 交卷时间	2021-04-22 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

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	0%	5	00:11:41	C++	0ms	OK	差	优	-
编程题2	100%	3	00:51:02	C++	120ms	724K	良	良	1%

1 [平均分3.30分 | 33人正确/50人做题 | 用时: <1分] 得分: 5.00 / 5.00

在嵌套使用if语句时，C语言规定else总是_____。

A 和之前与其具有相同缩进位置的if配对
B 和之前与其最近的if配对
C 和之前与其最近的且不带else的if配对
D 和之前的第一个if配对

他的回答: C (正确)
正确答案: C

2 [平均分4.10分 | 41人正确/50人做题 | 用时: <1分] 得分: 5.00 / 5.00

以下系统中，int类型占几个字节，指针占几个字节，操作系统可以使用的最大内存空间是多大：

A 32位下：4,4,2^32 64位下：8,8,2^64
B 32位下：4,4,不限制 64位下：4,8,不限制
C 32位下：4,4,2^32 64位下：4,8,2^64

D 32位下 : $4,4,2^{32}$ 64位下 : $4,4,2^{64}$

他的回答 : C (正确)

正确答案 : C

3 [平均分4.71分 | 48人正确/51人做题 | 用时 : <1分] 得分 : 5.00 / 5.00

以下程序的输出结果是_____。

```
#include <stdio.h>
int main()
{
    int i,a[10];
    for(i=9;i>=0;i--) a[i]=10-i;
    printf("%d%d%d",a[2],a[5],a[8]);
    return 0;
}
```

A 258

B 741

C 852

D 369

他的回答 : C (正确)

正确答案 : C

4 [平均分3.33分 | 34人正确/51人做题 | 用时 : <1分] 得分 : 5.00 / 5.00

下面C程序的输出结果:

```
int i=0,a=1,b=2,c=3;
i=++a||++b||++c;
printf("%d %d %d %d",i,a,b,c);
```

A 1 2 2 3

B 1 2 3 4

C 3 2 3 4

D 3 3 3 4

他的回答 : A (正确)

正确答案 : A

5 [平均分1.98分 | 19人正确/48人做题 | 用时 : 13分] 得分 : 5.00 / 5.00

下面代码不能正确输出hello的选项为

```
#include<stdio.h>
struct str_t{
    long long len;
    char data[32];
};
struct data1_t{
    long long len;
    int data[2];
};
struct data2_t{
    long long len;
    char *data[1];
};
```

```

struct data3_t{
    long long len;
    void *data[];
};
int main(void)
{
    struct str_t str;
    memset((void*)&str,0,sizeof(struct str_t));
    str.len=sizeof(struct str_t)-sizeof(int);
    snprintf(str.data,str.len,"hello");//VS下为_snprintf
    _____;
    _____;
    return 0;
}

```

- A struct data3_t *pData=(struct data3_t*)&str; printf("data:%s%s\n",str.data,(char*)&(pData->data[0]));
- B struct data2_t *pData=(struct data2_t*)&str; printf("data:%s%s\n",str.data,(char*)(pData->data[0]));
- C struct data1_t *pData=(struct data1_t*)&str;printf("data:%s%s\n",str.data,(char*)(pData->data));
- D struct str_t *pData=(struct str_t*)&str; printf("data:%s%s\n",str.data,(char*)(pData->data));

他的回答： B (正确)

正确答案： B

6 [平均分3.17分 | 33人正确/52人做题 | 用时：<1分 | 得分：5.00 / 5.00]

哪个操作符不能被重载？

- A , (逗号)
- B ()
- C . (点)
- D []
- E ->

他的回答： C (正确)

正确答案： C

7 [平均分4.40分 | 44人正确/50人做题 | 用时：<1分 | 得分：5.00 / 5.00]

下列对重载函数的描述中，_____是错误的。

- A 重载函数中不允许使用默认参数
- B 重载函数中编译时根据参数表进行选择
- C 构造函数重载将会给初始化带来多种方式
- D 不要使用重载函数来描述毫无相干的函数

他的回答： A (正确)

正确答案： A

8 [平均分4.40分 | 44人正确/50人做题 | 用时：<1分 | 得分：5.00 / 5.00]

下列关于多态性说法不正确的是（ ）。

- A 多态性是指同名函数对应多种不同的实现
- B 重载方式仅有函数重载
- C 重载方式包含函数重载和运算符重载
- D 多态性表现为静态和动态两种方式

他的回答： B (正确)

正确答案： B

9 [平均分2.60分 | 26人正确/50人做题 | 用时：3分] 得分：5.00 / 5.00

分析一下这段程序的输出

```
#include<iostream>
using namespace std;
class B
{
public:
    B()
    {
        cout << "default constructor" << " ";
    }
    ~B()
    {
        cout << "destructed" << " ";
    }
    B(int i): data(i)
    {
        cout << "constructed by parameter" << data << " ";
    }
private: int data;
};
B Play( B b)
{
    return b;
}
int main(int argc, char *argv[])
{
    B temp = Play(5);
    return 0;
}
```

A constructed by parameter5 destructed destructed

B constructed by parameter5 destructed

C default constructor" constructed by parameter5 destructed

D default constructor" constructed by parameter5 destructed destructed

他的回答：A (正确)

正确答案：A

10 [平均分4.12分 | 42人正确/51人做题 | 用时：5分] 得分：5.00 / 5.00

求输出结果

```
#include <iostream>

using namespace std;

class A
{
public:
    virtual void print()
    {
        cout << "A::print()" << "\n";
    }
};

class B: public A
```

```

{
public: virtual void print()
{
    cout << "B::print()" << "\n";
}
};

class C: public A
{
public: virtual void print()
{
    cout << "C::print()" << "\n";
}
};

void print(A a)
{
    a.print();
}

int main()
{
    A a, *aa, *ab, *ac;
    B b;
    C c;
    aa = &a;
    ab = &b;
    ac = &c;
    a.print();
    b.print();
    c.print();
    aa->print();
    ab->print();
    ac->print();
    print(a);
    print(b);
    print(c);
}

```

A C::print() B::print() A::print() A::print() B::print() C::print() A::print() A::print() A::print()
 B A::print() B::print() C::print() A::print() B::print() C::print() A::print() A::print() A::print()
 C A::print() B::print() C::print() A::print() B::print() C::print() B::print() B::print() B::print()
 D C::print() B::print() A::print() A::print() B::print() C::print() C::print() C::print() C::print()

他的回答： B (正确)

正确答案： B

11 ACM编程题 语言限制 [平均分0.00分 | 0人正确/47人做题 | 提交: 5 次] 得分：0.00 / 25.00

标题：参数解析 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【参数解析】

在命令行输入如下命令：

xcopy /s c:\ d:\ ,

各个参数如下：

参数1：命令字xcopy

参数2：字符串/s

参数3：字符串c:\

参数4: 字符串d:\

请编写一个参数解析程序，实现将命令行各个参数解析出来。

- 解析规则：
- 1.参数分隔符为空格
 - 2.对于用 "" 包含起来的参数，如果中间有空格，不能解析为多个参数。比如在命令行输入xcopy /s "C:\program files" "d:\\" 时，参数仍然是4个，第3个参数应该是字符串C:\program files，而不是C:\program，注意输出参数时，需要将 "" 去掉，引号不存在嵌套情况。
 - 3.参数不定长
 - 4.输入由用例保证，不会出现不符合要求的输入

输入描述：

输入一行字符串，可以有空格

输出描述：

输出参数个数，分解后的参数，每个参数都独占一行

示例1：

输入

xcopy /s c:\ d:\

输出

4
xcopy
/s
c:\
d:\

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	内部错误：5
总通过率	0%	0%	使用语言	C++		
基本测试用例通过率	0/6 (0%)	0%	做题用时	00:11:41	00:52:56	
边缘测试用例通过率	0/4 (0%)	0%	提交次数	5	9	
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	0ms	1s				
占用内存	0K	32768K				

他的代码：

做题用时: 11 分钟 语言: C++ 运行时间: 0ms 占用内存: 0K 程序状态: 内部错误

```
#include<iostream>
using namespace std;
#include<string>
void test(string str)
{
    int count = 1;
    for (int i = 0; i < str.size(); i++)
    {
        if (str[i] == ' ')
        {
            count++;
            str[i] = '\n';
        }
    }
}
```

```

    }
    cout << count << endl;
    for (int i = 0; i < str.size(); i++)
    {
        cout << str[i];
    }
}

int main()
{
    string str;
    getline(cin, str);
    test(str);
    return 0;
}

```



[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9.84分 | 12人正确/47人做题 | 提交: 3 次] 得分: 25.00 / 25.00

标题: 跳石板 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【跳石板】小易来到了一条石板路前，每块石板上从1挨着编号为：1、2、3.....

这条石板路要根据特殊的规则才能前进：对于小易当前所在的编号为K的 石板，小易单次只能往前跳K的一个约数(不含1和K)步，即跳到K+X(X为K的一个非1和本身的约数)的位置。 小易当前处在编号为N的石板，他想跳到编号恰好为M的石板去，小易想知道最少需要跳跃几次可以到达。

例如：

N = 4，M = 24：

4->6->8->12->18->24

于是小易最少需要跳跃5次，就可以从4号石板跳到24号石板

输入描述：

输入为一行，有两个整数N，M，以空格隔开。 ($4 \leq N \leq 100000$) ($N \leq M \leq 100000$)

输出描述：

输出小易最少需要跳跃的步数,如果不能到达输出-1

示例1：

输入

4 24

输出

5

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 39%	使用语言	TA的 C++	平均	运行超时 : 1
基本测试用例通过率	6/6 (100%)	42%	做题用时	00:51:02	00:41:31	编译错误 : 1
边缘测试用例通过率	4/4 (100%)	34%	提交次数	3	4	答案正确 : 1
代码效率			代码规范及可读性			
TA的 运行时间	参考 120ms	1s	代码规范得分			
占用内存	724K	32768K	4.44444			
			Line 4: Found C system header after C++ system header. Should be: 38208851_1620731585162.h, c system, c++ system, other. [build/include_order] [4]			
			Line 10: Is this a non-const reference? If so, make const or use a pointer: vector &divNum [runtime/references] [2]			
			Line 36: Add #include for min [build/include_what_you_use] [4]			

他的代码：

做题用时: 51 分钟 语言：C++ 运行时间：120ms 占用内存：724K 程序状态：答案正确

```
#include<iostream>
using namespace std;
#include<vector>
#include<math.h>
int min(int a, int b)
{
    return a > b ? b : a;
}
// 计算约数，求除了1和本身的约数
void div(int n, vector<int> &divNum) {
    for (int i = 2; i <= sqrt(n); ++i) {
        if (n % i == 0) {
            divNum.push_back(i);
            // 非平方数时还有另一个数也要加上
            if (n / i != i)
                divNum.push_back(n / i);
        }
    }
}
int Jump(int N, int M) {
    // 存储的到达此stepNum点的步数，初始N为1步，从N到N为1步
    vector<int> stepNum(M + 1, 0);
    stepNum[N] = 1;

    for (int i = N; i < M; ++i) {
        // N的所有约数，即为从本身这个点开始能走的数量
        vector<int> divNum;
        // 0代表这个点不能到
        if (stepNum[i] == 0)
            continue;
        // 求出所有走的步数储存在divNum的容器中
        div(i, divNum);
        for (int j = 0; j < divNum.size(); ++j) {
            // 由位置i出发能到达的点为stepNum[divNum[j] + i]
            if ((divNum[j] + i) <= M && stepNum[divNum[j] + i] != 0)
                stepNum[divNum[j] + i] = min(stepNum[divNum[j] + i], stepNum[i] + 1);
            else if ((divNum[j] + i) <= M)
                stepNum[divNum[j] + i] = stepNum[i] + 1;
        }
    }
}
```

```
}  
if (stepNum[M] == 0)  
    return -1;  
else  
    // 初始化时多给了一步，故需要减1  
    return stepNum[M] - 1;  
}  
  
int main()  
{  
    int n, m;  
    while (cin >> n >> m)  
    {  
        cout << Jump(n, m) << endl;  
    }  
    return 0;  
}
```



[点此](#)或手机扫描二维码查看代码编写过程