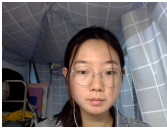


C++方向每日一题day01_4月19日-王梅-测评结果

考生信息



王梅

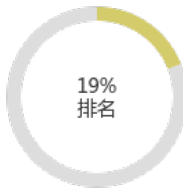
投递编号：62 | 学校：陕西科技大学 | 邮箱：2910602199@qq.com | 职位：91班 |

参考区域: 陕西省西安市 (123.139.93.162) |

做题用时：11:56:33(2021-04-18 21:46:17开始答题，2021-04-19 19:27:23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	45.0	9	5	00:40:46	已阅
编程	43.8	1	32	11:15:47	已阅

知识点技能图谱

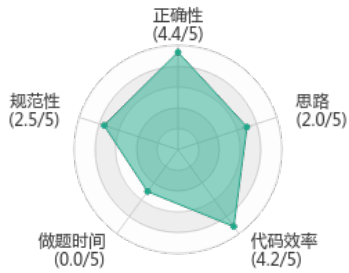


知识点	得分	正确题数
排序	25.0	1
编程基础	20.0	4
Java	5.0	1
C/C++	15.0	3
C++	15.0	3
模拟	18.8	0
字符串	18.8	0
贪心	25.0	1

历史笔记记录

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

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%	15	09:08:45	C++	112ms	1332K			1%
编程题2	75%	29	02:07:02	C++	2ms	412K			61%

1 [平均分3.7分 | 53人正确/72人做题 | 用时：2分] 得分：5.0 / 5.0
以下 for 循环的执行次数是 ()。

```
for(x=0,y=0;(y=123)&&(x<4);x++);
```

- A 是无限循环
- B 循环次数不定
- C 4次
- D 3次

他的回答：C (正确)

正确答案：C

2 [平均分2.7分 | 39人正确/72人做题 | 用时：<1分] 得分：5.0 / 5.0

以下程序的运行结果是 ()

```
int main(void)
{
    printf("%s, %5.3s\n", "computer", "computer");
    return 0;
}
```

%m.ns 输出占m列，但只取字符串中左端n个字符。这n个字符输出在m列的右侧，左补空格

- A computer, puter
- B computer, com
- C computer, computer
- D computer, compu.ter

他的回答：B (正确)

正确答案：B

3 [平均分3.6分 | 52人正确/72人做题 | 用时：<1分] 得分：5.0 / 5.0

int *p[4] 与选择项中的 说明 () 等价

- A int p[4]
- B int *p
- C int *(p[4])
- D int (*p)[4]

[]的优先级比*要高

他的回答：C (正确)

正确答案：C

4 [平均分4.3分 | 62人正确/72人做题 | 用时：9分] 得分：5.0 / 5.0

若有定义语句：int year=1009，*p=&year；以下不能使变量 year 中的值增至 1010 的语句是

- A *p+=1;
- B (*p)++;
- C ++(*p)
- D *p++

他的回答：D (正确)

正确答案：D

5 [平均分4.0分 | 58人正确/72人做题 | 用时：2分 | 得分：5.0 / 5.0]
若有定义语句：int a=10；double b=3.14；则表达式 'A'+a+b 值的类型是（ ）

- A char
- B int
- C double
- D float

他的回答：C (正确)

正确答案：C

6 [平均分4.9分 | 70人正确/72人做题 | 用时：<1分 | 得分：5.0 / 5.0]
在（ ）情况下适宜采用 inline 定义内联函数

- A 函数体含有循环语句
- B 函数体含有递归语句
- C 函数代码少、频繁调用
- D 函数代码多，不常调用

他的回答：C (正确)

正确答案：C

7 [平均分2.8分 | 41人正确/72人做题 | 用时：6分 | 得分：5.0 / 5.0]
在 c++ 语言中，对函数参数默认值描述正确的是（ ）

- A 函数参数的默认值只能设定一个
- B 一个函数的参数若有多个，则参数默认值的设定可以不连续
- C 函数参数必须设定默认值
- D 在设定了参数的默认值后，该参数后面定义的所有参数都必须设定默认值

他的回答：D (正确)

正确答案：D

8 [平均分4.0分 | 58人正确/72人做题 | 用时：4分 | 得分：5.0 / 5.0]
下列静态数据成员的特性中，错误的是（ ）

- A 引用静态数据成员时，要在静态数据成员名前加<类名>和作用域符号
- B 说明静态数据成员时前边要加关键字static来修饰
- C 静态数据成员在类体外进行初始化
- D 静态数据成员不是所有对象所共有的

他的回答：D (正确)

正确答案：D

9 [平均分4.2分 | 60人正确/72人做题 | 用时：2分 | 得分：5.0 / 5.0]
选择表达式 11|10 的结果（ 本题数值均为十进制）

- A 11 1011
- B 10 1010
- C 8 ----
- D 2 1011

他的回答：A (正确)

正确答案：A

10 [平均分1.5分 | 22人正确/72人做题 | 用时 : 7分 | 得分 : 0.0 / 5.0

下面叙述不正确的是 ()

- A 派生类一般都用公有派生
- B 对基类成员的访问必须是无二义性的
- C 赋值兼容规则也适用于多重继承的组合
- D 父类的公有成员在派生类中仍然是公有的

他的回答 : C (错误)
正确答案 : D

C++语言中如果基类的成员是公有的，它被继承后在子类中该成员可能是公有的，也可能不是，主要看继承的权限。

11 ACM编程题 语言限制 [平均分15.0分 | 40人正确/69人做题 | 提交: 15 次 | 得分 : 25.0 / 25.0

标题 : 组队竞赛 | 时间限制 : 1秒 | 内存限制 : 32768K | 语言限制 : 不限

【组队竞赛】牛牛举办了一次编程比赛,参加比赛的有3*n个选手,每个选手都有一个水平值a_i.现在要将这些选手进行组队,一共组成n个队伍,即每个队伍3人.牛牛发现队伍的水平值等于该队伍队员中第二高水平值。

例如:

一个队伍三个队员的水平值分别是3,3,3.那么队伍的水平值是3

一个队伍三个队员的水平值分别是3,2,3.那么队伍的水平值是3

一个队伍三个队员的水平值分别是1,5,2.那么队伍的水平值是2

为了让比赛更有看点,牛牛想安排队伍使所有队伍的水平值总和最大。

样例所示:

如果牛牛把6个队员划分到两个队伍

如果方案为:

team1:{1,2,5}, team2:{5,5,8}, 这时候水平值总和为7.

而如果方案为:

team1:{2,5,8}, team2:{1,5,5}, 这时候水平值总和为10.

没有比总和为10更大的方案,所以输出10.

输入描述 :

输入的第一行为一个正整数n(1 ≤ n ≤ 10^5)

第二行包括3*n个整数a_i(1 ≤ a_i ≤ 10^9),表示每个参赛选手的水平值.

输出描述 :

输出一个整数表示所有队伍的水平值总和最大值.

示例1 :

输入

2
5 2 8 5 1 5

输出

10

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 60%	使用语言	TA的 C++	平均	答案错误 : 5
基本测试用例通过率	6/6 (100%)	61%	做题用时	09:08:45	00:56:39	编译错误 : 6
边缘测试用例通过率	4/4 (100%)	58%	提交次数	15	8	答案正确 : 4
代码效率			代码规范及可读性			

运行时间 112ms 1s
占用内存 1332K 32768K

他的代码：

做题用时: 548 分钟 语言: C++ 运行时间: 112ms 占用内存: 1332K 程序状态: 答案正确

```
#include<iostream>
using namespace std;
#include<vector>
#include<algorithm>
void test()
{
    int n; //定义n为组数
    cin >> n; //输入n ,
    vector<int> v(3*n);
    for(int i = 0; i < n*3; i++)
    {
        cin >> v[i]; //输入水平值
    }
    sort(v.begin(), v.end()); //给所有水平值排序：从小到大

    long long count = 0;
    //贪心算法：局部最优解：三个数中取第二个
    //总体：将最小的n个数去掉，在剩余的2n个数中，从小向大取，每隔一个数取出一个，即为组水平值
    //例如：1 2 3 4 5 8
    //去掉1 2 两个数，剩余的数中取出3 5，即代表两个队各自的水平值
    for(int i = n; i <= (3*n)-2; i += 2)
    {
        count += v[i]; //取出的所有水平值相加
    }
    cout << count << endl;
}

int main()
{
    test();
    return 0;
}
```

```
1 #include<iostream>
2 using namespace std;
3 #include<algorithm>
4 #include<vector>
5 int main()
6 {
7     int n;
8     long long sum = 0;
9     cin >> n;
10    vector<int> a;
11    a.resize(3*n);
12    for(int i = 0; i < 3*n; i++)
13    {
14        cin >> a[i];
15    }
16    sort(a.begin(), a.end());
17    for(int i = 0; i < n; i++)
18    {
19        sum = sum + a[a.size() - 2*(i+1)];
20    }
21    cout << sum << endl;
22    return 0;
23 }
24 }
```



[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18.3分 | 44人正确/69人做题 | 提交: 29 次] 得分: 18.8 / 25.0

标题: 删除公共字符 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【删除公共字符】输入两个字符串，从第一字符串中删除第二个字符串中所有的字符。例如，输入“ They are students.” 和“ aeiou”，则删除之后的第一个字符

串变成“Thy r stdnts.”

输入描述：

每个测试输入包含2个字符串

输出描述：

输出删除后的字符串

示例1：

输入

They are students. aeiou

输出

Thy r stdnts.

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		75%	73%	使用语言	C++			答案错误	: 8
基本测试用例通过率	1/2 (50%)	72%		做题用时	02:07:02	00:49:35		运行超时	: 2
边缘测试用例通过率	2/2 (100%)	73%		提交次数	29	9		段错误	: 10
								编译错误	: 9
代码效率							代码规范及可读性		
		TA的	参考				代码规范得分		
运行时间	2ms	1s					5.0		
占用内存	412K	32768K							

他的代码：

做题用时: 127 分钟 语言：C++ 运行时间：2ms 占用内存：412K 程序状态：段错误

```
#include<iostream>
using namespace std;
#include<string>
int main()
{
    string str,str2;
    getline(cin,str);
    getline(cin,str2); //输入字符串
    string::iterator it2 = str2.begin();
    while(it2 != str2.end())
    {
        string::iterator it = str.begin();
        while(it != str.end())
        {
            if(*it == *it2)
            {
                str.erase(it);
            }
            it++;
        }
        it2++;
    }
    for(auto& e : str)
```

```
1 #include<iostream>
2 using namespace std;
3 #include<string>
4 int main()
5 {
6     string str1, str2;
7     getline(cin, str1);
8     getline(cin, str2);
9     int hash[256] = {0};
10    for(int i = 0; i < str2.size(); i++)
11    {
12        hash[str2[i]]++;
13    }
14    string ret = "";
15    for(int i = 0; i < str1.size(); i++)
16    {
17        if(hash[str1[i]] == 0)
18        {
19            ret += str1[i];
20        }
21    }
22    cout << ret << endl;
23    return 0;
24 }
25
```

```
{  
    cout << e;  
}  
return 0;  
}
```



[点此](#)或手机扫描二维码查看代码编写过程