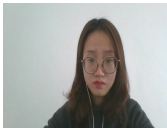


# 每日一题C++方向day18\_11月25日测评结果

## 考生信息



王婧

考号：2378 | 学校：财经大学 | 邮箱：1031160332@qq.com | 职位：54

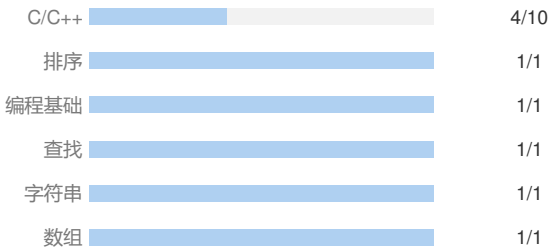
参考区域: 陕西省西安市 ( 36.40.233.151 ) | 做题用时：00:35:24(2019-11-25 15:07:05 - 15:42:37) | 作答设备：PC

## 考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	20.0	4	130	00:05:57	--
编程	50.0	2	1	00:28:41	--

## 知识点技能图谱



知识点	得分	正确题数
C/C++	20.0	4
排序	25.0	1
编程基础	5.0	1
查找	25.0	1
字符串	25.0	1
数组	5.0	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	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

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
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
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
15	每日一题C++方向day14_11月21日	16.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-20 13:55:05	2019-11-21 13:15:24
16	每日一题C++方向day15_11月22日	21.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-21 12:07:32	2019-11-22 12:56:30
17	每日一题C++方向day16_11月23日	47.0%	55.0/100	单选:5.0分 编程:50.0分	是，相似代码	2019-11-22 14:21:38	2019-11-23 12:51:07
18	每日一题C++方向day17_11月24日	2.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-11-23 14:28:02	2019-11-25 15:04:52

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	00:14:39	C++	9ms	492K	优	优	1%
编程题2	100%	2	00:14:02	C++	4ms	500K	优	优	1%

1

[平均分3.7分 | 108人正确/145人做题 | 用时：<1分] 得分：0.0 / 5.0

使用printf函数打印一个double类型的数据，要求：输出为10进制，输出左对齐30个字符，4位精度。以下哪个选项是正确的？

A %-30.4e  
B %4.30e  
C %-30.4f  
D %-4.30f

他的回答：A (错误)  
正确答案：C

2

[平均分2.9分 | 84人正确/146人做题 | 用时：<1分] 得分：0.0 / 5.0

malloc函数进行内存分配是在什么阶段？

A 编译阶段  
B 链接阶段  
C 装载阶段  
D 执行阶段

他的回答：A (错误)  
正确答案：D

3 [平均分3.3分 | 96人正确/146人做题 | 用时：<1分] 得分：0.0 / 5.0

函数作用：将整型数组p中n个数据增大

```
void increment_ints (int p [ ], int n)
{
    assert(p != NULL); /* 确保p不为空指针 */
    assert(n >= 0); /* 确保n不为负数 */
    while (n) /* 循环n次. */
    {
        *p++; /* 增大p*/
        p++, n--; /* p指向下一位, n减1 */
    }
}
```

以上代码的实现有错误，下面哪句话的表述是正确的？

- A \*p++使得p在解引用之前增大，应该改为(\*p)++
- B 数组的值是一个不能改变的值，所以p不能直接被修改。应该使用一个和p相关联的指针来完成这个操作。
- C while循环的条件必须是一个布尔类型的表达式，表达式应该为n!=0.
- D p不应该定义为变长的数组，参数中不应该包含参数n。

他的回答： B (错误)

正确答案： A

4 [平均分3.7分 | 109人正确/146人做题 | 用时：<1分] 得分：0.0 / 5.0

如下函数的f(1)的值为()

```
int f(int n){
    static int i=1;
    if(n>=5)
        return n;
    n=n+i;
    i++;
    return f(n);
}
```

- A 5
- B 6
- C 7
- D 8

他的回答： B (错误)

正确答案： C

5 [平均分3.7分 | 107人正确/143人做题 | 用时：<1分] 得分：5.0 / 5.0

下列给定程序中，函数fun的功能是：把形参a所指数组中的最小值放在元素a[0]中，接着把a所指数组中的最大值放在a[1]元素中；再把a所指数组元素中的次小值放在a[2]中，把a索取数组元素中的次大值放在a[3]，以此类推。

例如：若a所指数组中的数据最初排列为：9,1,4,2,3,6,5,8,7;按规则移动后，数据排列为：1,9,2,8,3,7,4,6,5。形参n中存放a所指数组中数据的个数。

规定fun函数中的max存放的当前所找的最大值,px存放当前所找最大值得下标。请在程序的下画线处填入正确的内容并将下画线删除，使程序得出正确的结果。试题程序。

```
#include<stdio.h>
#define N 9
void fun(int a[ ], int n)
{
    int i, j, max, min, px, pn, t;
    for (i = 0; i < n - 1; i += 2)
```

```

{

    max = min = __;
    px = pn = i;
    for (j = i + 1; j < n; j++)
    {

        if (max < __)
        {
            max = a[j];
            px = j;
        }

        if (min > __)
        {
            min = a[j];
            pn = j;
        }
    }
    if (pn != i)
    {
        t = a[i];
        a[i] = min;
        a[pn] = t;
        if (px == i)
            px = pn;
    }
    if (px != i + 1)
    {
        t = a[i + 1];
        a[i + 1] = max;
        a[px] = t;
    }
}
}

int main( )
{
    int b[N] = {9, 1, 4, 2, 3, 6, 5, 8, 7};
    printf("\nThe original data:\n");
    for (int i = 0; i < N; i++)
        printf("%4d", b[i]);
    printf("\n");
    fun(b, N);
    printf("\nThe data after mocinng \n");
    for (int i = 0; i < N; i++)
        printf("%4d", b[i]);
    printf("\n");
}

```

- A 0 a[i] a[i]
- B a[i] a[j] a[j]
- C 0 a[j] a[j]
- D a[i] a[i] a[i]

他的回答： B (正确)

正确答案： B

下面说法正确的是（ ）

- A C++已有的任何运算符都可以重载
- B const对象只能调用const类型成员函数
- C 构造函数和析构函数都可以是虚函数
- D 函数重载返回值类型必须相同

他的回答： B (正确)

正确答案： B

7 [平均分2.7分 | 78人正确/144人做题 | 用时：<1分 | 得分：0.0 / 5.0]

下面关于迭代器失效的描述哪个是错误的（ ）

- A vector的插入操作不会导致迭代器失效
- B map的插入操作不会导致迭代器失效
- C vector的删除操作只会导致指向被删除元素及后面的迭代器失效
- D map的删除操作只会导致指向被删除元素的迭代器失效

他的回答： C (错误)

正确答案： A

8 [平均分4.3分 | 124人正确/144人做题 | 用时：<1分 | 得分：5.0 / 5.0]

下面哪一个是sort的template的正确写法

- A void sort(class A first, class A last, class B pred)
- B void template(class A, class B)sort(A first, A last, B pred)
- C template<class A><class B> void sort(A first, A last, B pred)
- D template<class A, class B> void sort(A first, A last, B pred)

他的回答： D (正确)

正确答案： D

9 [平均分4.4分 | 128人正确/144人做题 | 用时：<1分 | 得分：5.0 / 5.0]

以下程序的运行结果是

```
main()
{
    char a[]="programming",b[]="language";
    char *p1,*p2;
    int i;
    p1=a,p2=b;
    for(i=0;i<7;i++)
        if(*(p1+i)==*(p2+i))
            printf("%c",*(p1+i));
}
```

- A gm
- B rg
- C or
- D ga

他的回答： D (正确)

正确答案： D

10 [平均分2.0分 | 58人正确/144人做题 | 用时 : <1分] 得分 : 0.0 / 5.0

下列程序的输出结果 :

```
#include <iostream>
using namespace std;
class A
{
public:
    void print()
    {
        cout << "A:print()";
    }
};
class B: private A
{
public:
    void print()
    {
        cout << "B:print()";
    }
};
class C: public B
{
public:
    void print()
    {
        A:: print();
    }
};
int main()
{
    C b;
    b.print();
}
```

- A A:print()
- B B:print()
- C 编译出错

他的回答 : A (错误)

正确答案 : C

11 [平均分23.3分 | 121人正确/130人做题 | 提交: 2 次] 得分 : 25.0 / 25.0

标题: 统计每个月兔子的总数 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【统计每个月兔子的总数】

有一只兔子，从出生后第3个月起每个月都生一只兔子，小兔子长到第三个月后每个月又生一只兔子，假如兔子都不死，问每个月的兔子总数为多少？

```
/**
 * 统计出兔子总数。
 *
 * @param monthCount 第几个月
 * @return 兔子总数
 */
public static int getTotalCount(int monthCount)
{
    return 0;
}
```

输入描述 :

输入int型表示month

输出描述：

输出兔子总数int型

示例1：

输入
9
输出
34

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：1
总通过率 100% 93%	使用语言 C++		答案正确：1
基本测试用例通过率 6/6 (100%) 93%	做题用时 00:14:39 00:28:01		
边缘测试用例通过率 4/4 (100%) 93%	提交次数 2 3		
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 9ms 1s			
占用内存 492K 32768K			

他的代码：

做题用时: 14 分钟    语言：C++    运行时间：9ms    占用内存：492K    程序状态：答案正确

```
#include<iostream>
using namespace std;
int Sum(int month)
{
    if(month==1||month==2)
        return 1;
    return Sum(month-1)+Sum(month-2);
}
int main()
{
    int month;
    while(cin>>month)
    {
        int num=Sum(month);
        cout<<num<<endl;
    }
    return 0;
}
```

12 [平均分19.0分 | 65人正确/104人做题 | 提交: 2 次] 得分：25.0 / 25.0

标题：字符串通配符 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【字符串通配符】

问题描述：在计算机中，通配符一种特殊语法，广泛应用于文件搜索、数据库、正则表达式等领域。现要求各位实现字符串通配符的算法。  
要求：

实现如下2个通配符：  
\*：匹配0个或以上的字符（字符由英文字母和数字0-9组成，不区分大小写。下同）  
?：匹配1个字符

输入：  
通配符表达式；  
一组字符串。

输出：  
返回匹配的结果，正确输出true，错误输出false  
输入描述：

先输入一个带有通配符的字符串，再输入一个需要匹配的字符串

输出描述：

返回匹配的结果，正确输出true，错误输出false

示例1：

输入

te?t\*. \*  
txt12.xls

输出

false

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	编译错误：1
总通过率	100%	76%	使用语言	C++		答案正确：1
基本测试用例通过率	6/6 (100%)	76%	做题用时	00:14:02	00:49:57	
边缘测试用例通过率	4/4 (100%)	75%	提交次数	2	5	
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	4ms	1s				
占用内存	500K	32768K				

他的代码：

做题用时: 14 分钟    语言：C++    运行时间：4ms    占用内存：500K    程序状态：答案正确

```
#include<iostream>
#include<string>
using namespace std;
bool pipei(const char* str1,const char* str2)
{
    if(*str1=='\0'&&*str2=='\0')
        return true;
    if(*str1=='\0'||*str2=='\0')
        return false;
    if(*str1=='?')
        return pipei(str1+1,str2+1);
    if(*str1=='*')
        return pipei(str1+1,str2)||pipei(str1+1,str2+1)||pipei(str1,str2+1);
    else if(*str1==*str2)
        return pipei(str1+1,str2+1);
}
```



```
    return false;
}
int main()
{
    string str1,str2;
    while(cin>>str1>>str2)
    {
        bool ret=pipei(str1.c_str(),str2.c_str());
        if(ret)
            cout<<"true"<<endl;
        else
            cout<<"false"<<endl;
    }
    return 0;
}
```