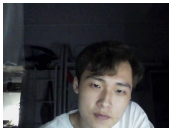


每日一题day18_6月8日测评结果

考生信息



张博翔

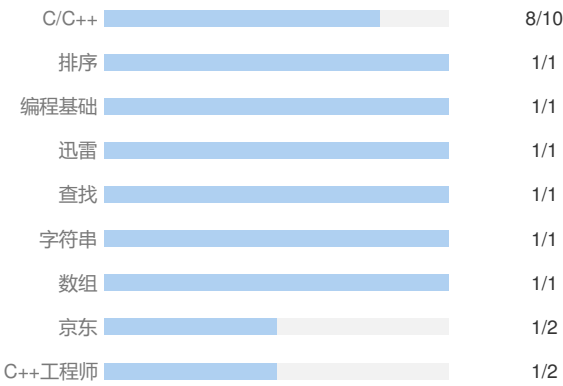
考号：1675 | 学校：陕西科技大学 | 邮箱：1761607418@qq.com | 职位：43班 |
参考区域: 陕西省西安市 (111.114.0.2) | 做题用时：02:27:20(2019-06-08 00:24:39 - 02:52:04)

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	26	00:18:03	--
编程	50.0	2	1	02:07:15	--

知识点技能图谱



知识点	得分	正确题数
C/C++	40.0	8
排序	25.0	1
编程基础	5.0	1
迅雷	5.0	1
查找	25.0	1
字符串	25.0	1
数组	5.0	1
京东	5.0	1
C++工程师	5.0	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	笔试时间
1	每日一题day1_5月20日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-19 16:08:49
2	每日一题day02_5月21日	51.0%	60.0/100	单选:30.0分 编程:30.0分	否	2019-05-20 17:40:56
3	每日一题day03_5月22日	4.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-21 12:59:15
4	每日一题day04_5月23日	8.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-22 10:40:18
5	每日一题day05_5月24日	29.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-05-22 20:39:16

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	笔试时间
6	每日一题day06_5月25日	10.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-24 10:41:21
7	每日一题day07_5月27日	6.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-05-26 15:09:41
8	每日一题day08_5月28日	18.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-27 14:06:31
9	每日一题day09_5月29日	13.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-28 15:11:53
10	每日一题day10_5月30日	6.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-05-29 11:32:20
11	每日一题day11_5月31日	14.0%	80.0/100	单选:30.0分 编程:50.0分	否	2019-05-30 11:22:09
12	每日一题day12_6月1日	31.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-05-31 10:33:31
13	每日一题day13_6月3日	9.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-06-02 15:20:12
14	每日一题day14_6月4日	41.0%	50.0/100	单选:25.0分 编程:25.0分	否	2019-06-03 11:53:13
15	每日一题day15_6月5日	6.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-06-04 12:02:48
16	每日一题day16_6月6日	6.0%	80.0/100	单选:30.0分 编程:50.0分	否	2019-06-05 16:32:52
17	每日一题day17_6月7日	4.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-06-06 10:57:01

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	01:08:06	C++	9ms	476K	优	优	1%
编程题2	100%	6	00:59:09	C++	3ms	360K	优	优	1%

1

[平均分3.6分 | 79人正确/109人做题 | 用时 : <1分 | 得分 : 5.0 / 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： C (正确)

正确答案： C

2

[平均分3.2分 | 70人正确/109人做题 | 用时 : <1分 | 得分 : 5.0 / 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： D (正确)

正确答案： D

3 [平均分3.4分 | 74人正确/108人做题 | 用时：4分 得分：5.0 / 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：A (正确)

正确答案：A

4 [平均分3.5分 | 75人正确/106人做题 | 用时：2分 得分：5.0 / 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：C (正确)

正确答案：C

5 [平均分4.1分 | 86人正确/104人做题 | 用时：3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所指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 mocinnng \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3.1分 | 66人正确/106人做题 | 用时：3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4.6分 | 99人正确/107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 | 95人正确/107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 l;
    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1.8分 | 37人正确/104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23.2分 | 91人正确/99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		

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	
总通过率	TA的 100%	平均 77%	使用语言	TA的 C++	平均	答案错误：5 答案正确：1	
基本测试用例通过率	6/6 (100%)	78%	做题用时	00:59:09	00:52:43		
边缘测试用例通过率	4/4 (100%)	76%	提交次数	6	6		
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	360K	32768K					

他的代码：

做题用时: 59 分钟 语言：C++ 运行时间：3ms 占用内存：360K 程序状态：答案正确

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str1;
    string str2;
    while(cin >> str1 >> str2)
    {
        int i = 0;
        int j = 0;
        while(str1[i] != '\0' && str2[j] != '\0')
        {
            if(str1[i] == '?')
            {
                i++;
                j++;
            }
        }
    }
}
```



```

        continue;
    }
    if(str1[i] == '*')
    {
        i++;
        if(str1[i] == '?')
        {
            i++;
            j++;
        }
        while(str2[j] != str1[i] && str2[j] != '\0')
        {
            j++;
        }
        if(str2[j] != str1[i] && str2[j] == '\0')
        {
            continue;
        }
        else if(str2[j] == str1[i] && str2[j] == '\0')
        {
            continue;
        }
        else
        {
            j++;
            i++;
            while(str2[j] == str2[j - 1] && str2[j] != str1[i])
            {
                j++;
            }
            continue;
        }
    }
    if(str1[i] >= 'A' && str1[i] <= 'Z')
    {
        str1[i] -= ' ';
    }
    if(str2[j] >= 'A' && str2[j] <= 'Z')
    {
        str2[2] -= ' ';
    }
    if(str1[i] != str2[j])
    {
        //cout << str1[i] << " " << str2[j] << endl;
        break;
    }
    i++;
    j++;
}
if(str1[i] == '\0' && str2[j] == '\0')
{
    cout << "true" << endl;
}
else
{
    cout << "false" << endl;
}
}
}

```

