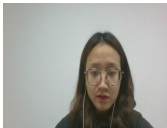


每日一题C++方向day21_11月29日测评结果

考生信息



王婧

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

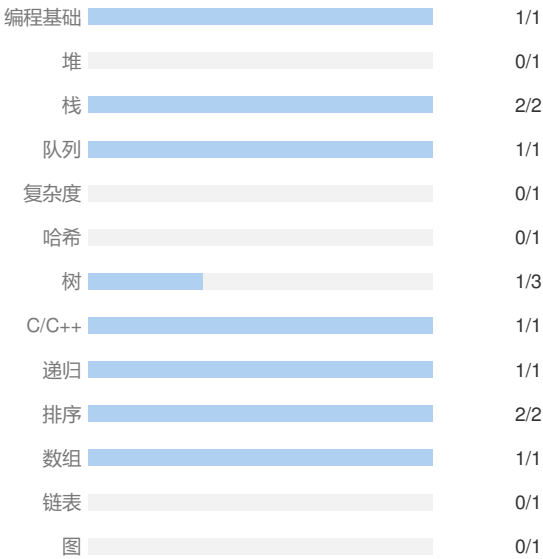
参考区域: 陕西省西安市 (36.40.234.224) | 做题用时：01:29:02(2019-11-29 16:30:26 - 17:59:31) | 作答设备：PC

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	25.0	5	79	00:08:16	--
编程	50.0	2	1	01:18:46	--

知识点技能图谱



知识点	得分	正确题数
编程基础	5.0	1
堆	0.0	0
栈	10.0	2
队列	5.0	1
复杂度	0.0	0
哈希	0.0	0
树	5.0	1
C/C++	5.0	1
递归	5.0	1
排序	30.0	2
数组	25.0	1
链表	0.0	0
图	0.0	0

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
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
19	每日一题C++方向day18_11月25日	47.0%	70.0/100	单选:20.0分 编程:50.0分	是，相似代码	2019-11-24 11:44:46	2019-11-25 15:42:37
20	每日一题C++方向day19_11月27日	22.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-25 14:38:07	2019-11-27 12:59:39
21	每日一题C++方向day20_11月28日	8.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-11-25 18:26:05	2019-11-28 16:59:04

编码能力



1 [平均分3.7分 | 101人正确/138人做题 | 用时：<1分 得分：0.0 / 5.0

设一个有序的单链表中有一个新结点后使得单链表仍然保持有序，则该操作的时间复杂度（ ）

- A $O(\log 2n)$
- B $O(1)$
- C $O(n^2)$
- D $O(n)$

他的回答： B (错误)

正确答案： D

2 [平均分4.1分 | 112人正确/137人做题 | 用时：<1分 | 得分：5.0 / 5.0]

一个栈的初始状态为空。首先将元素5, 4, 3, 2, 1依次入栈，然后退栈一次，再将元素A,B,C,D依次入栈，之后将所有元素全部退栈，则所有元素退栈（包括中间退栈的元素）的顺序为？

- A 1DCAB2345
- B 1DCBA2345
- C 54321ABCD
- D DCBA12345

他的回答： B (正确)

正确答案： B

3 [平均分4.2分 | 114人正确/137人做题 | 用时：<1分 | 得分：5.0 / 5.0]

设栈S和队列Q的初始状态为空，元素e1, e2, e3, e4, e5, e6依次压入栈S,一个元素出栈后即进入队列Q，若出队列的顺序为e2,e4,e3,e6,e5,e1则栈S的容量要求最小值为

- A 2
- B 3
- C 4
- D 5

他的回答： B (正确)

正确答案： B

4 [平均分3.6分 | 98人正确/137人做题 | 用时：<1分 | 得分：5.0 / 5.0]

给定下列程序，那么执行printf("%d\n", foo(20, 13));的输出结果是_____。

```
int foo(int x, int y){
    if (x <= 0 || y <= 0)
        return 1;
    return 3 * foo( x-6, y/2 );
}
```

- A 3
- B 9
- C 27
- D 81

他的回答： D (正确)

正确答案： D

参考答案：

解析：foo(20, 13) = 3 * foo(14, 6) = 3 * 3 * foo(8, 3) = 3 * 3 * 3 * foo(2, 1) = 3 * 3 * 3 * 3 * foo(-4, 0) = 3 * 3 * 3 * 3 * 1 = 81

答案：D

5 [平均分2.7分 | 73人正确/136人做题 | 用时：<1分 | 得分：0.0 / 5.0]

在具有 2n 个结点的完全二叉树中，叶子结点个数为（ ）

- A n
- B n+1
- C n-1
- D n/2

他的回答： B (错误)

正确答案： A

参考答案：

完全二叉树是指除最后一层外，每一层上的结点数均达到最大值，在最后一层上只缺少右边的若干结点。根据完全二叉树性质，如果共 $2n$ 个结点，从根结点开始按层序用自然数 $1, 2, \dots, 2n$ 给结点编号，则编号为 n 的结点左子结点编号为 $2n$ ，因此叶子结点编号为 $n+1, n+2, \dots, 2n$ 。故叶子结点个数为 n ，本题答案为 A 选项。

6 [平均分2.5分 | 63人正确/126人做题 | 用时：<1分 | 得分：5.0 / 5.0]

有权值分别为11，8，6，2，5的叶子结点生成一棵哈夫曼树，它的带权路径长度为_____。

- A 24
- B 71
- C 48
- D 53

他的回答： B (正确)

正确答案： B

7 [平均分1.9分 | 51人正确/134人做题 | 用时：<1分 | 得分：0.0 / 5.0]

下述二叉树中,哪一种满足性质:从任一结点出发到根的路径上所经过的结点序列按其关键字有序()

- A 二叉排序树
- B 哈夫曼树
- C AVL树
- D 堆

他的回答： B (错误)

正确答案： D

8 [平均分3.2分 | 84人正确/133人做题 | 用时：<1分 | 得分：0.0 / 5.0]

为提高散列（Hash）表的查找效率，可以采取的正确措施是（ ）。

- I . 增大装填（载）因子
- II . 设计冲突（碰撞）少的散列函数
- III . 处理冲突（碰撞）时避免产生聚集（堆积）现象

- A 仅 I
- B 仅 II
- C 仅 I、 II
- D 仅 II、 III

他的回答： C (错误)

正确答案： D

9 [平均分2.2分 | 57人正确/131人做题 | 用时：2分 | 得分：5.0 / 5.0]

将整数数组（7-6-3-5-4-1-2）按照堆排序的方式原地进行升序排列，请问在第一轮排序结束之后，数组的顺序是_____。

- A 2-6-3-5-4-1-7
- B 6-2-3-5-4-1-7
- C 6-5-3-2-4-1-7
- D 1-5-3-2-4-6-7
- E 5-4-3-2-1-6-7
- F 5-1-3-2-4-6-7

他的回答： C (正确)

正确答案： C

10 [平均分2.0分 | 53人正确/134人做题 | 用时：<1分 | 得分：0.0 / 5.0

要连通具有 n 个顶点的有向图，最少需要（ ）条边。

- A $n+1$
- B $n-1$
- C $2n$
- D n

他的回答： B (错误)

正确答案： D

11 [平均分22.2分 | 71人正确/80人做题 | 提交: 2 次 | 得分：25.0 / 25.0

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

【洗牌】洗牌在生活中十分常见，现在需要写一个程序模拟洗牌的过程。现在需要洗 $2n$ 张牌，从上到下依次是第1张，第2张，第3张一直到第 $2n$ 张。首先，我们把这 $2n$ 张牌分成两堆，左手拿着第1张到第 n 张（上半堆），右手拿着第 $n+1$ 张到第 $2n$ 张（下半堆）。接着就开始洗牌的过程，先放下右手的最后一张牌，再放下左手的最后一张牌，接着放下右手的倒数第二张牌，再放下左手的倒数第二张牌，直到最后放下左手的第一张牌。接着把牌合并起来就可以了。例如有6张牌，最开始牌的序列是1,2,3,4,5,6。首先分成两组，左手拿着1,2,3；右手拿着4,5,6。在洗牌过程中按顺序放下了6,3,5,2,4,1。把这六张牌再次合成一组牌之后，我们按照从上往下的顺序看这组牌，就变成了序列1,4,2,5,3,6。现在给出一个原始牌组，请输出这副牌洗牌 k 次之后从上往下的序列。

输入描述：

第一行一个数 T ($T \leq 100$)，表示数据组数。对于每组数据，第一行两个数 n, k ($1 \leq n, k \leq 100$)，接下来一行有 $2n$ 个数 a_1, a_2, \dots, a_{2n} ($1 \leq a_i \leq 1000000000$)。表示原始牌组从上到下的序列。

输出描述：

对于每组数据，输出一行，最终的序列。数字之间用空格隔开，不要在行末输出多余的空格。

示例1：

输入

3 3 1 1 2 3 4 5 6 3 2 1 2 3 4 5 6 2 2 1 1 1 1

输出

1 4 2 5 3 6 1 5 4 3 2 6 1 1 1 1

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	100%	88%	使用语言	C++
基本测试用例通过率	6/6 (100%)	88%	做题用时	00:45:39 00:49:02
边缘测试用例通过率	4/4 (100%)	88%	提交次数	2 4
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	16ms	1s		
占用内存	376K	32768K		

他的代码：

做题用时: 45 分钟 语言：C++ 运行时间：16ms 占用内存：376K 程序状态：答案正确

//思路：首先T表示牌组数；n表示总牌数的一半，即一个手中牌的个数；k表示洗牌的次数

//先记住洗牌前的下标，在记住洗牌一次后的下标

//会发现，对下标小于n的数，洗牌后它的下标数为2*下标

//对大于n得数，洗牌后他的下标为2*（下标-n）+1；

```
#include<iostream>
```

```
#include<vector>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int T,n,k;
```

```
    cin>>T;
```

```
    while(T--)
```

```
    {
```

```
        cin>>n>>k;
```

```
        int num=2*n;//牌的总数
```

```
        vector<int> card(num);
```

```
        for(int i=0;i<num;i++)
```

```
        {
```

```
            cin>>card[i];//将左右手的牌都方入vector数组中
```

```
        }
```

```
        while(k--)
```

```
        {
```

```
            vector<int> n1(card.begin(),card.end());//将card数组中的元素拷贝到n1中
```

```
            for(int i=0;i<n;i++)
```

```
            {
```

```
                card[2*i]=n1[i];
```

```
                card[2*i+1]=n1[i+n];
```

```
            }
```

```
        }
```

```
        for(int i=0;i<num-1;i++)
```

```
        {
```

```
            cout<<card[i]<<" ";
```

```
        }
```

```
        cout<<card[num-1]<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

12 [平均分20.0分 | 59人正确/78人做题 | 提交: 6次 | 得分: 25.0 / 25.0

标题：MP3光标位置 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

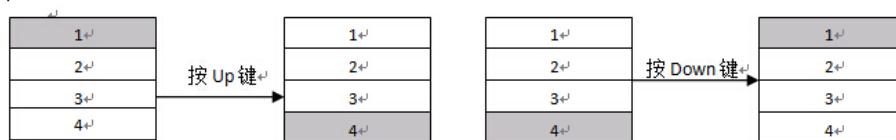
【MP3光标位置】

MP3 Player因为屏幕较小，显示歌曲列表的时候每屏只能显示几首歌曲，用户要通过上下键才能浏览所有的歌曲。为了简化处理，假设每屏只能显示4首歌曲，光标初始的位置为第1首歌。

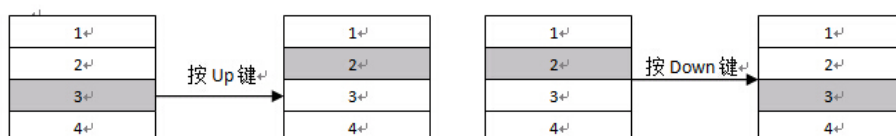
现在要实现通过上下键控制光标移动来浏览歌曲列表，控制逻辑如下：

歌曲总数 ≤ 4 的时候，不需要翻页，只是挪动光标位置。

光标在第一首歌曲上时，按Up键光标挪到最后一首歌曲；光标在最后一首歌曲时，按Down键光标挪到第一首歌曲。

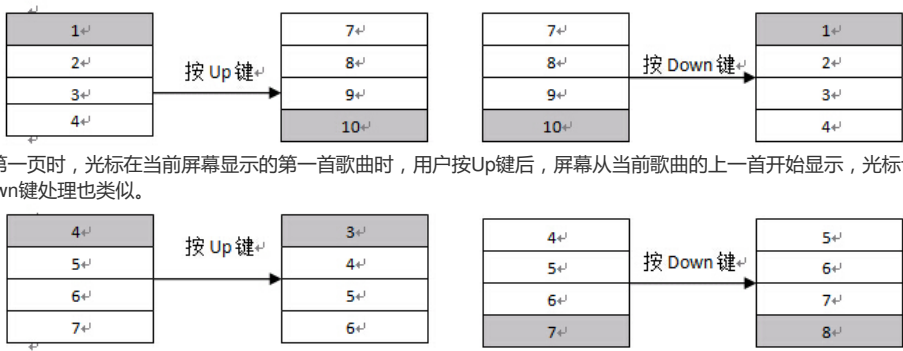


其他情况下用户按Up键，光标挪到上一首歌曲；用户按Down键，光标挪到下一首歌曲。



2. 歌曲总数大于4的时候（以一共有10首歌为例）：

特殊翻页：屏幕显示的是第一页（即显示第1-4首）时，光标在第一首歌曲上，用户按Up键后，屏幕要显示最后一页（即显示第7-10首歌），同时光标放到最后一首歌上。同样的，屏幕显示最后一页时，光标在最后一首歌曲上，用户按Down键，屏幕要显示第一页，光标挪到第一首歌上。



一般翻页：屏幕显示的不是第一页时，光标在当前屏幕显示的第一首歌曲时，用户按Up键后，屏幕从当前歌曲的上一首开始显示，光标也挪到上一首歌曲。光标当前屏幕的最后一首歌曲时的Down键处理也类似。

其他情况，不用翻页，只是挪动光标就行。

输入描述：

- 输入说明：
- 1 输入歌曲数量

2 输入命令 U或者D

输出描述：

- 输出说明
- 1 输出当前列表

2 输出当前选中歌曲

示例1：

输入
10 UUUU
输出
7 8 9 10 7

代码片段

功能实现	代码提交统计	代码执行统计
TA的 平均 总通过率 100% 79% 基本测试用例通过率 6/6 (100%) 79% 边缘测试用例通过率 4/4 (100%) 79%	TA的 平均 使用语言 C++ 做题用时 00:33:07 01:17:53 提交次数 6 4	答案错误 ：2 编译错误 ：2 答案正确 ：2
代码效率		代码规范及可读性
TA的 参考 运行时间 4ms 1s 占用内存 488K 32768K	代码规范得分 4.84848 Line 32: If an else has a brace on one side, it should have it on both [readability/braces] [5]	

他的代码：

做题用时: 33 分钟 语言：C++ 运行时间：4ms 占用内存：488K 程序状态：答案正确

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
    int n;
    string exe;
```

```

while(cin>>n>>exe)
{
    int num=1,first=1;
    if(n<=4)
    {
        for(int i=0;i<exe.size();i++)
        {
            if(num==1&&exe[i]=='U')
            {
                num=n;
            }
            else if(num==n&&exe[i]=='D')
            {
                num=1;
            }
            else if(exe[i]=='U')
            {
                num--;
            }
            else
            {
                num++;
            }
        }
        for(int i=1;i<=n-1;i++)
        {
            cout<<i<<' ';
        }
        cout<<n<<endl;
        cout<<num<<endl;
    }
    else{
        for(int i=0;i<exe.size();i++)
        {
            if(first==1&&num==1&&exe[i]=='U')
            {
                first=n-3;
                num=n;
            }
            else if(first==n-3&&num==n&&exe[i]=='D')
            {
                first=1;
                num=1;
            }
            else if(first!=1&&num==first&&exe[i]=='U')
            {
                first--;
                num--;
            }
            else if(first!=n-3&&num==first+3&&exe[i]=='D')
            {
                first++;
                num++;
            }
            else if(exe[i]=='U')
            {
                num--;
            }
            else
            {
                num++;
            }
        }
        for(int i=first;i<first+3;i++)
        {
            cout<<i<<' ';
        }
        cout<<first+3<<endl;
        cout<<num<<endl;
    }
}
return 0;
}

```