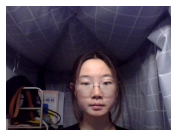


# C++方向每日一题day21\_5月19日-王梅-测评结果

## 考生信息



王梅

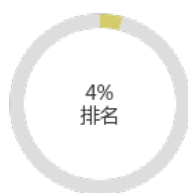
投递编号：62 | 学校：陕西科技大学 | 邮箱：2910602199@qq.com | 职位：91班 |

参考区域：陕西省西安市（117.136.87.38） |

做题用时：01:06:25(2021-05-18 23:52:39开始答题，2021-05-20 00:02:39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	45.00	9	3	00:24:17	已阅
编程	50.00	2	1	00:42:08	已阅

## 知识点技能图谱

编程基础	1/1
堆	1/1
栈	2/2
穷举	1/1
队列	1/1
复杂度	1/1
哈希	1/1
树	3/3
C/C++	1/1
递归	1/1
排序	1/2
模拟	1/1
数组	1/1
链表	1/1
图	1/1

知识点	得分	正确题数
编程基础	5.00	1
堆	5.00	1
栈	10.00	2
穷举	25.00	1
队列	5.00	1
复杂度	5.00	1
哈希	5.00	1
树	15.00	3
C/C++	5.00	1
递归	5.00	1
排序	25.00	1
模拟	25.00	1
数组	25.00	1
链表	5.00	1

知识点	得分	正确题数
图	5.00	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	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
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
16	C++方向每日一题day13_5月10日	1.0%	75.0/100	单选:50.0分 编程:25.0分	否	2021-05-09 11:22:20	2021-05-10 08:10:17
17	C++方向每日一题day14_5月11日	23.0%	70.0/100	单选:40.0分 编程:30.0分	否	2021-05-10 10:13:17	2021-05-10 22:06:18

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
18	C++方向每日一题day15_5月12日	8.0%	85.0/100	单选:35.0分 编程:50.0分	否	2021-05-10 14:52:33	2021-05-12 18:58:48
19	C++方向每日一题day16_5月13日	9.0%	82.9/100	单选:35.0分 编程:47.92分	否	2021-05-12 10:20:46	2021-05-12 23:22:37
20	91&92班Linux1考试试卷	16.0%	52.0/60	单选:18.0分 不定项选择:4.0分 编程:30.0分	否	2021-05-12 14:37:21	2021-05-15 09:44:02
21	C++方向每日一题day17_5月14日	16.0%	90.0/100	单选:40.0分 编程:50.0分	否	2021-05-12 14:39:41	2021-05-14 17:42:58
22	C++方向每日一题day18_5月15日	39.0%	80.0/100	单选:45.0分 编程:35.0分	否	2021-05-13 11:59:08	2021-05-15 21:28:28
23	C++方向每日一题day19_5月17日	33.0%	80.0/100	单选:30.0分 编程:50.0分	否	2021-05-16 16:34:55	2021-05-17 13:37:47
24	C++方向每日一题day20_5月18日	9.0%	95.0/100	单选:45.0分 编程:50.0分	否	2021-05-17 15:29:48	2021-05-17 23:20:25

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	00:16:23	C++	7ms	504K			1%
编程题2	100%	2	00:25:45	C++	3ms	380K			1%

1 [平均分3.23分 | 31人正确/48人做题 | 用时：<1分] 得分：5.00 / 5.00  
设一个有序的单链表中n个结点，现要求插入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： D (正确)  
正确答案： D

2 [平均分4.49分 | 44人正确/49人做题 | 用时：<1分] 得分：5.00 / 5.00  
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4.18分 | 41人正确/49人做题 | 用时：<1分] 得分：5.00 / 5.00

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3.65分 | 35人正确/48人做题 | 用时：2分] 得分：5.00 / 5.00

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3.67分 | 36人正确/49人做题 | 用时：<1分] 得分：5.00 / 5.00

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： A (正确)

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，...，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,...,2n。故叶子结点个数为 n，本题答案为 A 选项。

6 [平均分2.45分 | 23人正确/47人做题 | 用时：10分] 得分：5.00 / 5.00

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2.08分 | 20人正确/48人做题 | 用时：2分  得分：5.00 / 5.00

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： D (正确)

正确答案： D

8 [平均分3.83分 | 36人正确/47人做题 | 用时：2分  得分：5.00 / 5.00

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： D (正确)

正确答案： D

9 [平均分2.60分 | 25人正确/48人做题 | 用时：<1分  得分：0.00 / 5.00

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： A (错误)

正确答案： C

10 [平均分1.98分 | 19人正确/48人做题 | 用时：<1分  得分：5.00 / 5.00

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： D (正确)

正确答案： D

11 ACM编程题 语言限制 [平均分17.93分 | 33人正确/46人做题 | 提交: 2 次  得分：25.00 / 25.00

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a1,a2,...,a2n( $1 \leq a_i \leq 1000000000$ )。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	
总通过率	TA的	平均		使用语言	TA的	平均		格式错误	: 1
	100%	71%			C++			答案正确	: 1
	6/6	71%		做题用时	00:16:23	00:57:42			
	(100%)			提交次数	2	5			
基本测试用例通过率									
	4/4	71%							
边缘测试用例通过率	(100%)								
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	7ms	1s							
占用内存	504K	32768K							

他的代码：

做题用时: 16 分钟    语言：C++    运行时间：7ms    占用内存：504K    程序状态：答案正确

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

void test(int n, vector<vector<int>>> v)
{
    v.resize(n);
    for (int i = 0; i < n; i++)
    {
        int a = 0, b = 0;
        cin >> a >> b;
        v[i].resize(2 * a);
        for (int j = 0; j < 2 * a; j++)
        {
            cin >> v[i][j];
        }
        vector<int> v1(2 * a);
        for (int k = 0; k < b; k++)
```

```

{
    int start = a - 1, end = 2 * a - 1;
    int count = end;
    while (start >= 0)
    {
        v1[count--] = v[i][end--];
        v1[count--] = v[i][start--];
    }
    v[i] = v1;
}
for (auto& e : v1)
{
    cout << e << " ";
}
    cout << endl;
}
}
int main()
{
    int n;
    while (cin >> n)
    {
        vector<vector<int>> v;
        v.resize(n);
        test(n, v);
    }
    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13.01分 | 22人正确/44人做题 | 提交: 2 次] 得分: 25.00 / 25.00

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

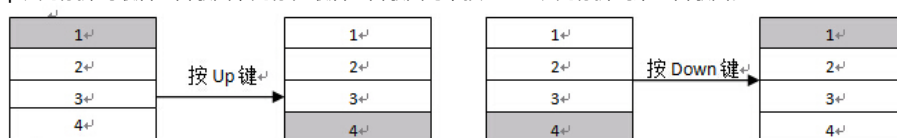
【MP3光标位置】

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。

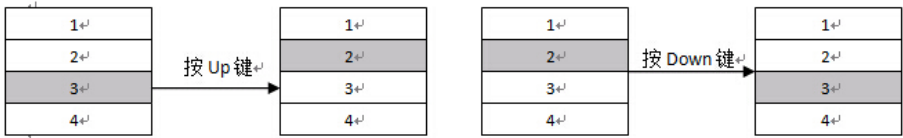
现在要实现通过上下键控制光标移动来浏览歌曲列表，控制逻辑如下：

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。

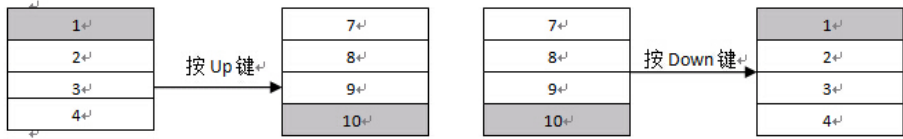


其他情况下用户按Up键，光标挪到上一首歌曲；用户按Down键，光标挪到下一首歌曲。

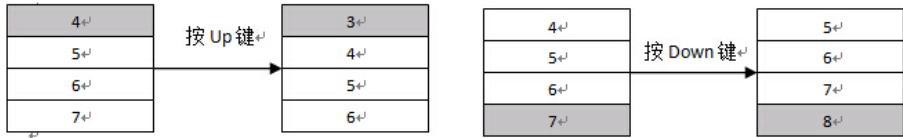


2. 歌曲总数大于4的时候（以一共有10首歌为例）：

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							
总通过率	TA的	平均	使用语言	TA的	平均	做题用时	答案正确：2							
	100%	52%		C++										
	6/6			提交次数										
	(100%)	52%		2	4									
基本测试用例通过率	4/4	51%												
边缘测试用例通过率	(100%)													
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	3ms	1s	4.34426											
占用内存	380K	32768K	Line 24: If an else has a brace on one side, it should have it on both [readability/braces] [5]											
			Line 30: If an else has a brace on one side, it should have it on both [readability/braces] [5]											
			Line 34: 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: 25 分钟

语言: C++

运行时间: 3ms

占用内存: 380K

程序状态: 答案正确

```
#include<iostream>
using namespace std;
#include<string>
void test(int n, string str)
{
    int num = 1, first = 1;
    if (n <= 4) {
        for (int i = 0; i < str.length(); ++i) {
            if (num == 1 && str[i] == 'U')
                num = n;

            else if (num == n && str[i] == 'D')
                num = 1;
            else if (str[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 < str.length(); ++i) {
            if (first == 1 && num == 1 && str[i] == 'U') {
                first = n - 3;;
                num = n;
            }
            else if (first == n - 3 && num == n && str[i] == 'D') {
                first = 1;
                num = 1;
            }
            else if (first != 1 && num == first && str[i] == 'U') {
                first--;
                num--;
            }
            else if (first != n - 3 && num == first + 3 && str[i] == 'D') {
                first++;
                num++;
            }
            else if (str[i] == 'U')
                num--;
            else
                num++;
        }
        for (int i = first; i < first + 3; ++i)
            cout << i << ' ';
        cout << first + 3 << endl;
        cout << num << endl;
    }
}
int main()
```

```
{  
    int n;  
    string str;  
    while (cin >> n, cin >> str)  
    {  
        test(n, str);  
    }  
    return 0;  
}
```



[点此](#)或手机扫描二维码查看代码编写过程