

## 每日一题C++方向day21\_11月29日

### 一. 单选

1. 设一个有序的单链表中，有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  $O(n)$

正确答案：D

2. 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

正确答案：B

3. 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

正确答案：B

4. 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

正确答案：D

5.

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  $n/2$

正确答案：A

6. 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

正确答案：B

7. 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

正确答案：D

8.

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

正确答案：D

9. 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

正确答案：C

10. 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  $n$

正确答案：D

## 二. 编程

1. 标题：洗牌 | 时间限制：1秒 | 内存限制：32768K

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

正确答案：

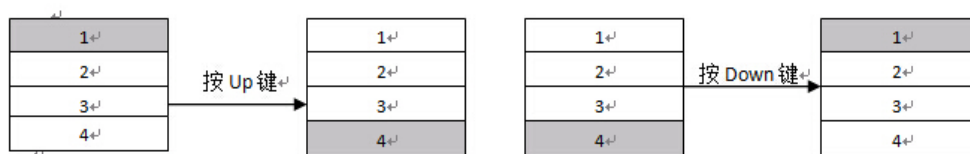
2. 标题：MP3光标位置 | 时间限制：1秒 | 内存限制：32768K

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。

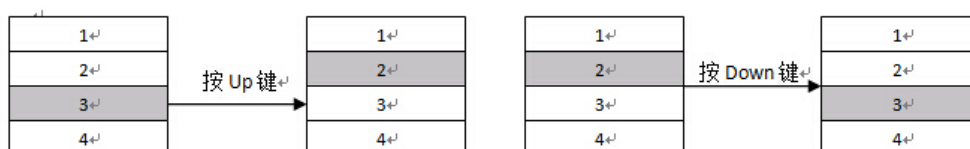
现在要实现通过上下键控制光标移动来浏览歌曲列表，控制逻辑如下：

歌曲总数 $\leq 4$ 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。

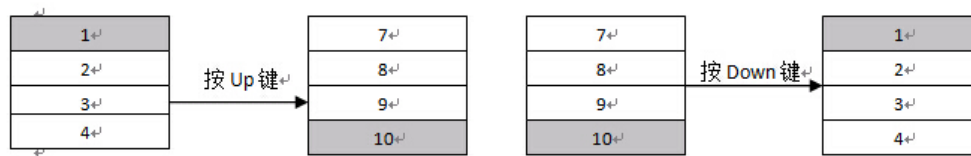


其他情况下用户按Up键，光标挪到上一首歌曲；用户按Down键，光标挪到下一首歌曲。

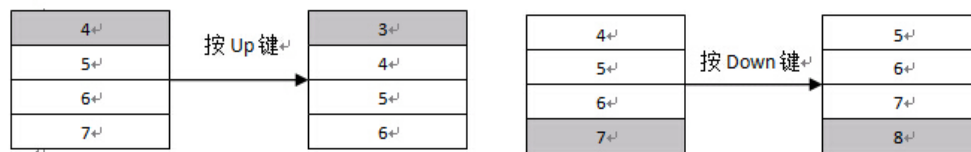


2. 歌曲总数大于4的时候（以一共有10首歌为例）：

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

正确答案：