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 ≤ 100)，表示数据组数。对于每组数据，第一行两个数n,k(1 ≤ n,k ≤ 100)，接下来一行有2n个数a1,a2,...,a2n(1 ≤ ai ≤ 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。

##### **输入例子:**

3

3 1

1 2 3 4 5 6

3 2

1 2 3 4 5 6

2 2

1 1 1 1

##### **输出例子:**

1 4 2 5 3 6

1 5 4 3 2 6

1 1 1 1

小明同学把1到n这n个数字按照一定的顺序放入了一个队列Q中。现在他对队列Q执行了如下程序：

while(!Q.empty()) //队列不空，执行循环{

int x=Q.front(); //取出当前队头的值x

Q.pop(); //弹出当前队头

Q.push(x); //把x放入队尾

x = Q.front(); //取出这时候队头的值

printf("%d\n",x); //输出x

Q.pop(); //弹出这时候的队头

}

做取出队头的值操作的时候，并不弹出当前队头。

小明同学发现，这段程序恰好按顺序输出了1,2,3,...,n。现在小明想让你构造出原始的队列，你能做到吗？[注：原题样例第三行5有错，应该为3，以下已修正]

输入描述:

第一行一个整数T（T ≤ 100）表示数据组数，每组数据输入一个数n（1 ≤ n ≤ 100000），输入的所有n之和不超过200000。

输出描述:

对于每组数据，输出一行，表示原始的队列。数字之间用一个空格隔开，不要在行末输出多余的空格.

输入例子:

4

1

2

3

10

输出例子:

1

2 1

2 1 3

8 1 6 2 10 3 7 4 9 5