



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。

```
1  #include<iostream>
2  #include<vector>
3  using namespace std;
4
5  int main()
6  {
7      int T, n, k;
8      cin >> T;
9      while (T--)
10     {
11         cin >> n >> k;
12         int num = 2 * n;
13         vector<int> table(num);
14         for(int i = 0; i < num; ++i)
15             cin >> table[i];
16         while (k--)
17         {
18             vector<int> n1(table.begin(), table.end());
19             for (int i = 0; i < n; ++i)
20             {
21                 table[2 * i] = n1[i];
22                 table[2 * i + 1] = n1[i + n];
23             }
24         }
25         for(int i = 0; i < num - 1; ++i)
26             cout << table[i] << " ";
27         cout << table[num - 1] << endl;
28     }
29     return 0;
30 }
```

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]

```
1  import java.util.LinkedList;
2  import java.util.Scanner;
3  public class NewTest{
4      public static LinkedList<Integer> func(int n){
5          LinkedList<Integer> help=new LinkedList<Integer>();
6          for(int i=n;i>=1;i--){
7              help.addFirst(i);
8              help.addFirst(help.removeLast());
9          }
10         return help;
11     }
12     public static void main(String[] args){
13         int t;
14         Scanner scan = new Scanner(System.in);
15         t=scan.nextInt();
```



```
16         int n;  
17         LinkedList<Integer> res;  
18         while(t-->0) {  
19             n=scan.nextInt();  
20             res=func(n);  
21             for(int i=0;i<n-1;i++) {  
22                 System.out.print(res.removeFirst()+" ");  
23             }  
24             System.out.println(res.removeFirst());  
25         }  
26     }  
27 }
```