Day 88 coding Statement :

Blobo2 is in his practical exam. The teacher gave him a *permutation A* of *N* integers.

The teacher has allowed Blobo2 to make a certain type of operation on the *permutation*. In one operation, he can:

- Apply left shift on the permutation. In other words, he can take the first element of the permutation and move it to the back of the permutation.

The teacher has asked Blobo2 to find the **lexicographically smallest** permutation possible after applying any (possibly zero) number of given operations.

Since Blobo2 wants to impress his teacher, he decided to perform **at most** two swaps in addition to the allowed operation.

Find the **lexicographically smallest** possible permutation Blobo2 can generate after applying **at most two** swaps and any number of given operations.

Note:

- A permutation of size *N* consists of all integers from 1 to *N* exactly once.
- During a swap, Blobo2 can choose any two indices *i* and *j* ($1 \le i,j \le N$) and swap *Ai*? with *Aj*?.
- A permutation *X* is said to be lexicographically smaller than a permutation *Y* if *Xi*?<*Yi*?, where *i* is the first index where both the permutations differ.

Input Format

- The first line of input will contain a single integer *T*, denoting the number of test cases.
- The second line will contain the integer *N*, denoting the size of the permutation.
- The third line contains *N* distinct integers, the elements of the permutation *A*.

Output Format

Output the **lexicographically smallest** possible permutation Blobo2 can generate after applying **at most two** swaps and any number of given operations.

Sample Input

2

5

5 2 1 4 3

5

3 5 2 1 4

Sample Output

1 2 3 4 5

1 2 3 4 5

```
import java.util.*;
import java.lang.*;
import java.io.*;

public class Program {
	static int[] A = new int[100010];
	static int[] tmp = new int[100010];

	static int[] pos = new int[100010];
	static int[] ans = new int[100010];
	static int n;

	static void add(int cnt) {
		for (int i = 1; i <= n; i++)
			pos[tmp[i]] = i;
		for (int i = 1; i <= n; i++) {
			if (cnt == 0)
				break;
			if (tmp[i] == i)
				continue;
			int id = pos[i];
			pos[tmp[i]] = id;
			tmp[id] = tmp[i];
			tmp[i] = i;
			cnt--;
		}
	}

	static int ok() {
		for (int i = 1; i <= n; i++) {
			if (tmp[i] > ans[i])
				return 0;
			else if (tmp[i] < ans[i])
				return 1;
		}
		return 0;
	}

	static void change() {
		if (ok() == 0)
```

```
                return;
        for (int i = 1; i <= n; i++)
                ans[i] = tmp[i];
}

public static void main(String[] args) throws IOException {
        Scanner cin = new Scanner(System.in);
        int t = cin.nextInt();

        while (t-- > 0) {
                n = cin.nextInt();
                for (int i = 1; i <= n; i++)
                        A[i] = cin.nextInt();
                if (n <= 4) {
                        for (int i = 1; i <= n; i++)
                                System.out.print(i + " ");
                        System.out.println();
                        continue;
                }
                for (int i = 1; i <= n; i++)
                        ans[i] = A[i];
                int id = 0;
                for (int i = 1; i <= n; i++)
                        if (A[i] == 1)
                                id = i;
                for (int i = 1; i <= n; i++)
                        tmp[i] = A[(i + id - 2 + 5 * n) % n + 1];
                add(2);
                change();
                for (int i = 1; i <= n; i++)
                        if (A[i] == 2)
                                id = i;
                for (int i = 1; i <= n; i++)
                        tmp[i] = A[(i + id - 3 + 5 * n) % n + 1];
                add(2);
                change();
                for (int i = 1; i <= n; i++)
                        if (A[i] == 3)
                                id = i;
                for (int i = 1; i <= n; i++)
                        tmp[i] = A[(i + id - 4 + 5 * n) % n + 1];
                add(2);
                change();
                for (int i = 1; i <= n; i++)
                        if (A[i] == 4)
                                id = i;
                for (int i = 1; i <= n; i++)
                        tmp[i] = A[(i + id - 5 + 5 * n) % n + 1];
                add(2);
                change();
                id = 0;
                for (int i = 1; i < n; i++) {
                        if (A[i] == 2 && A[i + 1] == 1)
                                id = i;
```

```
                        }
                        if (A[n] == 2 && A[1] == 1)
                                id = n;
                        if (id != 0) {
                                for (int i = 1; i <= n; i++)
                                        tmp[i] = A[(i + id - 2 + 5 * n) % n + 1];
                                add(2);
                                change();
                        }
                        id = 0;
                        for (int i = 1; i < n; i++) {
                                if (A[i] == 3 && A[i + 1] == 2)
                                        id = i;
                        }
                        if (A[n] == 3 && A[1] == 2)
                                id = n;
                        if (id != 0) {
                                for (int i = 1; i <= n; i++)
                                        tmp[i] = A[(i + id - 3 + 5 * n) % n + 1];
                                add(2);
                                change();
                        }
                        id = 0;
                        for (int i = 1; i < n - 1; i++) {
                                if (A[i] == 3 && A[i + 2] == 1)
                                        id = i;
                        }
                        if (A[n - 1] == 3 && A[1] == 1)
                                id = n - 1;
                        if (A[n] == 3 && A[2] == 1)
                                id = n;
                        if (id != 0) {
                                for (int i = 1; i <= n; i++)
                                        tmp[i] = A[(i + id - 2 + 5 * n) % n + 1];
                                add(2);
                                change();
                        }
                        for (int i = 1; i <= n; i++)
                                System.out.print(ans[i] + " ");

                        System.out.println();
                }
        }
}
```