

136. single number

```
class Solution {
    public int singleNumber(int[] nums) {
        int ans = 0;

        for (final int num : nums)
            ans ^= num;

        return ans;
    }
}
```

60. permutation sequence

```
class Solution {
    public String getPermutation(int n, int k) {
        StringBuilder sb = new StringBuilder();
        List<Integer> nums = new ArrayList<>();
        int[] fact = new int[n + 1]; // fact[i] := i!

        for (int i = 1; i <= n; ++i)
            nums.add(i);

        Arrays.fill(fact, 1);
        for (int i = 2; i <= n; ++i)
            fact[i] = fact[i - 1] * i;

        --k; // 0-indexed

        for (int i = n - 1; i >= 0; --i) {
            final int j = k / fact[i];
            k %= fact[i];
            sb.append(nums.get(j));
            nums.remove(j);
        }

        return sb.toString();
    }
}
```