### **1. Find the Second Largest Element in an Array**

Write a program to find the second largest element in an array of integers.

* **Example:**
  + Input: [3, 5, 7, 2, 8, 6]
  + Output: 7
* **Test Case 1:**
  + Input: [10, 20, 4, 45, 99, 99]
  + Output: 45
* **Test Case 2:**
  + Input: [1, 2]
  + Output: 1

### **2. Rotate an Array by k Positions**

Write a program to rotate an array to the right by k positions. For example, if k = 2, each element will move two positions to the right, and the last two elements will move to the beginning.

* **Example:**
  + Input: arr = [1, 2, 3, 4, 5, 6, 7], k = 2
  + Output: [6, 7, 1, 2, 3, 4, 5]
* **Test Case 1:**
  + Input: arr = [10, 20, 30, 40, 50], k = 3
  + Output: [30, 40, 50, 10, 20]
* **Test Case 2:**
  + Input: arr = [1, 2, 3, 4, 5], k = 5
  + Output: [1, 2, 3, 4, 5]

### **3. Find the Missing Number in an Array**

Given an array of n-1 integers in the range from 1 to n, find the missing number in the array.

* **Example:**
  + Input: [1, 2, 4, 6, 3, 7, 8]
  + Output: 5
* **Test Case 1:**
  + Input: [1, 2, 3, 5]
  + Output: 4
* **Test Case 2:**
  + Input: [2, 3, 4, 5]
  + Output: 1

### **4. Move Zeros to the End**

Write a program that moves all zeroes in an array to the end while maintaining the relative order of the non-zero elements.

* **Example:**
  + Input: [0, 1, 0, 3, 12]
  + Output: [1, 3, 12, 0, 0]
* **Test Case 1:**
  + Input: [1, 0, 0, 2, 3]
  + Output: [1, 2, 3, 0, 0]
* **Test Case 2:**
  + Input: [0, 0, 0, 1, 2]
  + Output: [1, 2, 0, 0, 0]

### **5. Find the Longest Subarray with Equal Number of 0s and 1s**

Given a binary array (an array consisting of only 0s and 1s), find the length of the longest subarray with an equal number of 0s and 1s.

* **Example:**
  + Input: [0, 1, 0, 1, 0, 1, 1, 0]
  + Output: 8
* **Test Case 1:**
  + Input: [1, 1, 1, 0, 0, 0]
  + Output: 6
* **Test Case 2:**
  + Input: [1, 1, 1, 1]
  + Output: 0

### **6. Find the Maximum Product of Two Elements in an Array**

Write a program that finds the maximum product of two distinct elements in an array of integers.

* **Example:**
  + Input: [3, 4, 5, 2]
  + Output: 20 (4 \* 5)
* **Test Case 1:**
  + Input: [1, 5, 3, 9, 2]
  + Output: 45 (5 \* 9)
* **Test Case 2:**
  + Input: [7, 1, 8, 6]
  + Output: 56 (7 \* 8)

### **7. Merge Two Sorted Arrays**

Write a program to merge two sorted arrays into one sorted array.

* **Example:**
  + Input: arr1 = [1, 3, 5, 7], arr2 = [2, 4, 6, 8]
  + Output: [1, 2, 3, 4, 5, 6, 7, 8]
* **Test Case 1:**
  + Input: arr1 = [10, 20], arr2 = [15, 25]
  + Output: [10, 15, 20, 25]
* **Test Case 2:**
  + Input: arr1 = [1, 2, 3], arr2 = []
  + Output: [1, 2, 3]

### **8. Find All Duplicates in an Array**

Write a program to find all the elements that appear more than once in an array.

* **Example:**
  + Input: [4, 3, 2, 7, 8, 2, 3, 1]
  + Output: [2, 3]
* **Test Case 1:**
  + Input: [1, 2, 3, 1, 2, 4]
  + Output: [1, 2]
* **Test Case 2:**
  + Input: [1, 2, 3, 4]
  + Output: []

### **9. Find the Maximum Sum Subarray (Kadane's Algorithm)**

Write a program to find the maximum sum of a contiguous subarray within a one-dimensional array of integers using Kadane’s Algorithm.

* **Example:**
  + Input: [-2, 1, -3, 4, -1, 2, 1, -5, 4]
  + Output: 6 (subarray [4, -1, 2, 1] has the maximum sum)
* **Test Case 1:**
  + Input: [1, 2, 3, 4, -10]
  + Output: 10
* **Test Case 2:**
  + Input: [-1, -2, -3, -4]
  + Output: -1

### **10. Find the Leaders in an Array**

Write a program to find all the leaders in an array. An element is a leader if it is greater than all the elements to its right side.

* **Example:**
  + Input: [16, 17, 4, 3, 5, 2]
  + Output: [17, 5, 2]
* **Test Case 1:**
  + Input: [1, 2, 3, 4, 0]
  + Output: [4, 0]
* **Test Case 2:**
  + Input: [10, 9, 8, 7]
  + Output: [10, 9, 8, 7]

**11 .Given an unsorted array of integers, sort the array into a wave array. An array arr[0..n-1] is sorted in wave form if: arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= …..**

Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80 } // 2 3 5 6 10 20 80 100

Output: arr[] = {10, 5, 6, 2, 20, 3, 100, 80} // 3 2 6 5 20 10 100 80

Explanation:

here you can see {10, 5, 6, 2, 20, 3, 100, 80} first element is larger than the second and the same thing is repeated again and again. large element – small element-large element -small element and so on .it can be small element-larger element – small element-large element -small element too. all you need to maintain is the up-down fashion which represents a wave. there can be multiple answers.

Input: arr[] = {20, 10, 8, 6, 4, 2}  
Output: arr[] = {20, 8, 10, 4, 6, 2}