1. Convert Array into Zig Zag Array

arr [] ={4,3,7,8,6,2,1

output : {3,7,4,8,2,6,1}

first<second>third and so on.

**CODE :**

import java.util.Arrays;  
public class ZigZag {  
  
 public static void main(String[] args) {  
 int[] arr = {4, 3, 7, 8, 6, 2, 1};  
 System.*out*.println("Original Array: " + Arrays.*toString*(arr));  
  
 *convertToZigZag*(arr);  
 System.*out*.println("Zig Zag Array: " + Arrays.*toString*(arr));  
 }  
  
 public static void convertToZigZag(int[] arr) {  
 for (int i = 0; i < arr.length - 1; i++) {  
 if (i % 2 == 0 && arr[i] > arr[i + 1]) {  
 *swap*(arr, i, i + 1);  
 } else if (i % 2 == 1 && arr[i] < arr[i + 1]) {  
 *swap*(arr, i, i + 1);  
 }  
 }  
 }  
  
 public static void swap(int[] arr, int i, int j) {  
 int temp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = temp;  
 }  
 }

**Output :**

A screenshot of a computer code

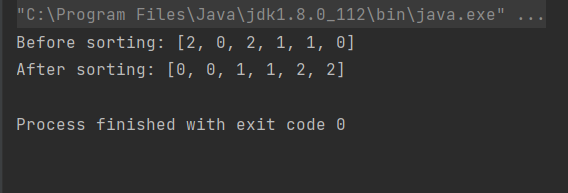
Description automatically generated with low confidence

2. DNF Problem

**CODE :**

import java.util.Arrays;  
  
public class DNF\_Problem {  
  
 public static void sortColors(int[] nums) {  
  
  
 int low = 0;  
 int mid = 0;  
 int high = nums.length - 1;  
  
 while (mid <= high) {  
 switch (nums[mid]) {  
 case 0: // Red  
 *swap*( nums,low, mid);  
 low++;  
 mid++;  
 break;  
 case 1: // White  
 mid++;  
 break;  
 case 2: // Blue  
 *swap*( nums,mid, high);  
 high--;  
 break;  
 }  
 }  
 }  
  
 public static void swap(int[] nums,int i, int j) {  
 int temp = nums[i];  
 nums[i] = nums[j];  
 nums[j] = temp;  
 }  
  
 public static void main(String[] args) {  
 int[] nums = {2, 0, 2, 1, 1, 0};  
  
 System.*out*.println("Before sorting: " + Arrays.*toString*(nums));  
 *sortColors*(nums);  
 System.*out*.println("After sorting: " + Arrays.*toString*(nums));  
 }  
}

**Output :**

****