20. Sort the array so that whenever nums[i] is odd, i is odd, and whenever nums[i] is even, i is even. Return any answer array that satisfies this condition.

Program:

```
def sort_array_by_parity(nums):
    # Initialize two pointers for even and odd indices
    even_index, odd_index = 0, 1
    # Loop until the even_index is less than the length of nums
    while even_index < len(nums) and odd_index < len(nums):
        # If the number at even_index is even, move to the next even index
        if nums[even_index] % 2 == 0:
            even_index += 2
        # If the number at odd_index is odd, move to the next odd index
        elif nums[odd_index] % 2 == 1:
            odd_index += 2
        # If the number at even_index is odd and the number at odd_index is
even, swap them
        else:
            nums[even_index], nums[odd_index] = nums[odd_index],
nums[even_index]
            even_index += 2
            odd_index += 2
    return nums

# Example usage:
nums = [4, 2, 5, 7]
print(sort_array_by_parity(nums))  # Output: [4, 5, 2, 7] or any other
valid answer
```

Output: