**Q1 - Find Missing Numbers**

* **Prompt Used:** // find missing numbers from array 1 to n in C#
* **Response Received:** Suggested using a HashSet or in-place marking by modifying the input array using negative signs.
* **Implementation Details:** Applied in-place index marking by negating values at specific indices to mark presence.
* **Adjustments Made:** Added edge case checks for null and empty arrays, and included explanatory comments.

**Q2 - Sort Array by Parity**

* **Prompt Used:** // sort array by parity in-place in C#
* **Response Received:** Two-pointer technique to place evens at the front and odds at the back.
* **Implementation Details:** Created a new array and filled even numbers from the start and odd numbers from the end.
* **Adjustments Made:** Included checks for empty and single-element arrays.

**Q3 - Two Sum**

* **Prompt Used:** // return indices of two numbers adding up to target
* **Response Received:** Dictionary-based lookup (hash map) for complement values.
* **Implementation Details:** Implemented the dictionary approach to store values and their indices for fast lookup.
* **Adjustments Made:** Ensured no duplicate key insertion; handled edge cases for small arrays.

**Q4 - Maximum Product of Three Numbers**

* **Prompt Used:** // find max product of 3 numbers in array
* **Response Received:** Sort the array and compare the product of top 3 vs product of 2 smallest and 1 largest number.
* **Implementation Details:** Used Array.Sort() and Array.Reverse() to simplify sorting logic.
* **Adjustments Made:** Added logic to handle negative numbers affecting product outcomes.

**Q5 - Decimal to Binary**

* **Prompt Used:** // convert decimal to binary string in C#
* **Response Received:** Use modulus and divide with string concatenation.
* **Implementation Details:** Used a while loop with % operator and string prepending.
* **Adjustments Made:** Handled special case for input 0.

**Q6 - Minimum in Rotated Sorted Array**

* **Prompt Used:** // find min in rotated sorted array
* **Response Received:** Binary search approach to find minimum in rotated array.
* **Implementation Details:** Implemented binary search using mid, left, and right pointers.
* **Adjustments Made:** Commented on behavior with duplicates (though not handled explicitly).

**Q7 - Palindrome Number**

* **Prompt Used:** // check if number is palindrome without converting to string
* **Response Received:** Reverse the number and compare with the original.
* **Implementation Details:** Reversed using arithmetic, but later converted to string for simplicity.
* **Adjustments Made:** Early returns for negatives and zero added for clarity.

**Q8 - Fibonacci Number**

* **Prompt Used:** // get nth fibonacci number iteratively
* **Response Received:** Suggested using loop with a, b, and c variables.
* **Implementation Details:** Although response suggested iteration, used recursion for simplicity.
* **Adjustments Made:** Included base cases for n = 0 and n = 1.