**Q1 - Find Missing Numbers**

**Prompt**: // find missing numbers from array 1 to n in C#  
**Response**: Used boolean/marking technique, suggested HashSet or in-place index marking.  
**Implementation**: Used in-place marking with negatives.  
**Adjustment**: Added edge case comments.

**Q2 - Sort Array by Parity**

**Prompt**: // sort array by parity in-place in C#  
**Response**: Used two-pointer method.  
**Implementation**: Swapped even to front, odd to back.  
**Adjustment**: Checked and skipped if already sorted.

**Q3 - Two Sum**

**Prompt**: // return indices of two numbers adding up to target  
**Response**: Dictionary (hash map) approach.  
**Implementation**: Directly used.  
**Adjustment**: Handled duplicates and returns index.

**Q4 - Max Product of Three Numbers**

**Prompt**: // find max product of 3 numbers in array  
**Response**: Sort and compare last 3 vs 2 smallest and largest.  
**Implementation**: Used max between both.  
**Adjustment**: Explained edge case when negatives are involved.

**Q5 - Decimal to Binary**

**Prompt**: // convert decimal to binary string in C#  
**Response**: While loop, use num % 2.  
**Implementation**: Simple loop + string prepend.  
**Adjustment**: Special case for 0.

**Q6 - Min in Rotated Sorted Array**

**Prompt**: // find min in rotated sorted array  
**Response**: Binary search logic.  
**Implementation**: Used binary search with mid checks.  
**Adjustment**: Added comment about duplicates.

**Q7 - Palindrome Number**

**Prompt**: // check if number is palindrome without converting to string  
**Response**: Reverse the number and compare.  
**Implementation**: Used reversed integer approach.  
**Adjustment**: Early return for negatives.

**Q8 - Fibonacci Number**

**Prompt**: // get nth fibonacci number iteratively  
**Response**: Loop method with a/b/c values.  
**Implementation**: Iterative calculation.  
**Adjustment**: Added 0 and 1 direct return.