AI Assistance Documentation -Prompts doc

# Q1: Find Missing Numbers

Prompt Used:  
find missing numbers in array C#

Response Received:  
Copilot suggested using a HashSet to keep track of seen numbers and loop through 1 to n to check what's missing.

Implementation Details:  
I used the HashSet idea but renamed variables to 'seen' and 'missing' (later updated to something else) and rewrote the loop for clarity.

Adjustments:  
Changed variable names and added meaningful comments for uniqueness.

# Q2: Sort Array by Parity

Prompt Used:  
sort array by parity two pointer approach C#

Response Received:  
Copilot proposed using two pointers from both ends and swapping when the left is odd and right is even.

Implementation Details:  
I accepted the suggestion but used tuple-swapping syntax for brevity and clarity. Changed variable names to i and j.

Adjustments:  
Added better comments and ensured clarity without copying the original style.

# Q3: Two Sum

Prompt Used:  
C# two sum using dictionary

Response Received:  
Copilot generated a dictionary-based approach storing indices of elements while checking for the complement.

Implementation Details:  
I reused the logic but restructured comments and variable names like 'indexMap', 'required', etc.

Adjustments:  
Improved readability and ensured originality in naming and comments.

# Q4: Maximum Product of Three Numbers

Prompt Used:  
maximum product of three numbers C# array

Response Received:  
Copilot suggested sorting the array and checking the product of top 3 or two smallest with largest.

Implementation Details:  
Kept the same idea but cleaned up the logic and added explanation of both scenarios in comments.

Adjustments:  
Modified comment structure to make it more instructive.

# Q5: Decimal to Binary

Prompt Used:  
convert decimal to binary C# using stack

Response Received:  
Copilot suggested using a stack to store binary digits and popping to create the final binary string.

Implementation Details:  
I followed this approach but adjusted variable naming and simplified the return using string.Concat.

Adjustments:  
Edge case handling for zero was added explicitly.

# Q6: Find Minimum in Rotated Sorted Array

Prompt Used:  
find min in rotated sorted array binary search C#

Response Received:  
Copilot recommended a binary search comparing mid and right elements.

Implementation Details:  
I rewrote the loop with descriptive variables (low, high) and added step-by-step comments.

Adjustments:  
Changed the flow slightly to make it more readable.

# Q7: Palindrome Number

Prompt Used:  
is number a palindrome in C#

Response Received:  
Copilot suggested reversing the number and comparing it with the original.

Implementation Details:  
I applied this logic while using a clear while loop and renaming variables like 'reverse'.

Adjustments:  
Handled the negative edge case early in the method.

# Q8: Fibonacci Number

Prompt Used:  
fibonacci iterative C#

Response Received:  
Copilot returned an iterative approach using two variables for previous and current.

Implementation Details:  
I used that and renamed variables to 'prev' and 'curr', with a loop up to n.

Adjustments:  
Edge case for n < 2 was handled separately and comments were added.