AI Assistance Documentation for Assignment 2

# Question 1

* Prompts Used:  
  How to find all missing numbers in array from 1 to n in C#?
* Response Received:  
  Copilot suggested using a loop to mark elements by setting nums[abs(nums[i]) - 1] to negative.
* Implementation Details:  
  I used Copilot's suggestion of modifying the original array in-place by marking seen elements negative.
* Adjustments:  
  Added logic to handle duplicate values and comments explaining the in-place marking technique.

# Question 2

* Prompts Used:  
  C# method to move all even numbers to front and odd numbers to end, in-place
* Response Received:  
  Copilot proposed using two pointers, swapping even and odd values from each end.
* Implementation Details:  
  Used the two-pointer technique to organize the array by parity.
* Adjustments:  
  Added bounds checks and comments for clarity.

# Question 3

* Prompts Used:  
  C# efficient two sum implementation using dictionary
* Response Received:  
  Copilot suggested using a dictionary to store seen values and check for complement.
* Implementation Details:  
  Used dictionary to solve in O(n) time.
* Adjustments:  
  Added comments to explain why it's more efficient than nested loops.

# Question 4

* Prompts Used:  
  Find max product of any 3 numbers in C# array
* Response Received:  
  Copilot recommended sorting and comparing product of top 3 vs two smallest and largest.
* Implementation Details:  
  Implemented both product options and returned the maximum.
* Adjustments:  
  Handled the edge case when array length is exactly 3.

# Question 5

* Prompts Used:  
  Convert decimal to binary string in C# without using Convert.ToString
* Response Received:  
  Copilot provided a loop with n % 2 and reverse of result string.
* Implementation Details:  
  Used loop to manually convert number to binary.
* Adjustments:  
  Handled n == 0 explicitly.

# Question 6

* Prompts Used:  
  C# binary search to find minimum in rotated sorted array
* Response Received:  
  Copilot returned binary search logic comparing mid and right values.
* Implementation Details:  
  Implemented binary search to find minimum efficiently.
* Adjustments:  
  Verified correctness for unrotated arrays.

# Question 7

* Prompts Used:  
  Check if integer is palindrome in C# without converting to string
* Response Received:  
  Copilot suggested reversing half of the number.
* Implementation Details:  
  Used reverse-half approach for better efficiency.
* Adjustments:  
  Handled negatives and trailing zeros appropriately.

# Question 8

* Prompts Used:  
  Iterative Fibonacci function in C#
* Response Received:  
  Copilot suggested an iterative loop with two variables.
* Implementation Details:  
  Implemented loop-based Fibonacci to avoid recursion.
* Adjustments:  
  Used exactly as suggested without modifications.