# AI Usage Documentation for Assignment 2

Tool Used: GitHub Copilot (Visual Studio Extension)

## Question 1

* Prompt Given to Copilot:

Help me to write a C# method that finds which numbers are missing from an array that’s supposed to have all numbers from 1 to n — but some numbers might be missing and others could show up more than once?

* Code Suggested by Copilot:

Used a HashSet<int> to store seen values and looped from 1 to n to find missing elements.

* Changes Made:

No changes. The suggested solution was clean and handled edge cases.

## Question 2

* Prompt Given to Copilot:

Help me to rearrange the numbers in an array so that all the even numbers come first, and then the odd ones.

* Code Suggested by Copilot:

Used a two-pointer technique to rearrange even numbers before odd numbers.

* Changes Made:

No major changes. Added a comment to handle edge cases where the array is already sorted by parity.

## Question 3

* Prompt Given to Copilot:

Give me a hint to Implement the two-sum algorithm in C# using a dictionary.

* Code Suggested by Copilot:

Used a dictionary to track complements during a single pass loop.

* Changes Made:

Added a condition to avoid adding duplicate keys to the dictionary.

## Question 4

* Prompt Given to Copilot:

How to find the maximum product of any three numbers in an integer array in C#.

* Code Suggested by Copilot:

Sorted array and compared product of last three and two smallest with the largest.

* Changes Made:

No changes. Covered cases with negative numbers correctly.

## Question 5

* Prompt Given to Copilot:

How to convert a decimal number to binary string in C# without using Convert.ToString.

* Code Suggested by Copilot:

Used a while loop and remainder method to convert decimal to binary.

* Changes Made:

No changes. Clean and correct logic provided.

## Question 6

* Prompt Given to Copilot:

Find the minimum element in a rotated sorted array using binary search in C#.

* Code Suggested by Copilot:

Implemented binary search comparing mid with right pointer to find pivot/minimum.

* Changes Made:

None. Edge cases were naturally handled by the loop.

## Question 7

* Prompt Given to Copilot:

Check if a given integer is a palindrome without converting to a string in C#.

* Code Suggested by Copilot:

Used reverse integer logic to compare original and reversed numbers.

* Changes Made:

Added overflow protection for integer reversal.

## Question 8

* Prompt Given to Copilot:

Write an iterative function to return the nth Fibonacci number in C#.

* Code Suggested by Copilot:

Used a loop with two variables to compute Fibonacci values.

* Changes Made:

No changes. Efficient and straightforward logic.