1. The Reverse Engineering

Goal: Practice indexing and swapping.

Create an array of 5 integers (e.g., 1, 2, 3, 4, 5). Without using the built-in Array.Reverse() method, write a loop that reverses the order of the elements within that same array.

- **Challenge:** Try to do this "in-place" (swap the first with the last, second with second-to-last, etc.).

2. Linear Search: Find the Index

Goal: Master the "search" pattern.

Create an array of strings containing 6 different colors. Ask the user to type a color name.

- Loop through the array to find if that color exists.
- If found, print: "Found at index [i]!".
- If the loop finishes and the color wasn't there, print: "Color not found.".

3. The Even/Odd Split

Goal: Logic branching within an array.

Initialize an array with 10 random integers between 1 and 100.

- Create a script that counts how many **even** numbers and how many **odd** numbers are in the array.
- Print the final counts.
- *Bonus:* Calculate the sum of only the even numbers.

4. Find the Minimum & Maximum

Goal: Comparative logic and state tracking.

Given an array: int[] numbers = { 34, 7, 23, 99, 1, 56, 12 };

- Write a program that identifies the smallest and largest numbers in the array.
- **Rule:** Do not use numbers.Min() or numbers.Max(). You must use a foreach or for loop and a "tracking" variable.

5. The Grade Normalizer

Goal: Data manipulation.

A teacher has an array of 5 student test scores (0–100). The test was too hard, so the teacher wants to "curve" the grades by adding 5 points to every score.

- Update the original array by adding 5 to each element.
- **Constraint:** If a score is already 98, it should cap at 100 (no scores above 100).
- Print the "Before" and "After" arrays to the console.