Group Exercises: Pointers and Memory Management

- **This is a group exercise. Each group must collaborate to complete all 7 exercises.**
- **Each solution must be written in pseudocode style and include clear variable names, comments (optional), and proper indentation.**

Exercise 1: Pointer-Based Calculator

Write a function that takes two pointers and an operation symbol (+, -, *, /) and returns the result of the operation.
You should ask the user to enter two numbers and the operation, then display the result.

Exercise 2: Find the Maximum Value in an Array

Dynamically allocate an array of 5 integers.
Use a pointer to:

- Read values into the array.
- Traverse the array.
- Find and return the **maximum** value.

Exercise 3: Pointer-Based Average Calculator

Using Pointers:

- Dynamically allocate memory for 5 student scores.
- Read the values.
- Calculate and print the average score.
- Free the allocated memory after use.

Exercise 4: Reverse an Array Using Pointers

Write a program to:

- Allocate an array of **n** elements (size can be fixed or input by the user).
- Read values into the array.
- Reverse the array in-place using pointers only (**e.g., p[0] ↔ p[n-1]).**
- Print the reversed array.

Exercise 5: Count Even and Odd Numbers

- Allocate memory for 6 integers.
- Use a pointer to read the values.
- Count how many are **even** and how many are **odd**.
- Print both totals.

Exercise 6: Debug This Pseudocode

The following pseudocode contains errors. Work as a group to:

- Identify and fix the issues.
- Explain what each pointer is doing.

```
Variable x
Pointer p ← &x
Write *p
x ← 15
Free(p)    // Error?
```

Exercise 7: Dynamic Table Generator

Using pointers, write a program that:

- Allocates memory for 10 numbers.
- Reads the numbers from the user.
- Prints each number and its square in a table format:

E.g. Below:

Number	Square
2	4
5	25

- Free memory at the end.