## 1. Swapping Two Variables Using Pointers

Declare two integer variables `a` and `b` with initial values.

Use a function that takes pointers to `a` and `b` as arguments to swap their values.

Print the values of `a` and `b` before and after the swap.

## 2. Accessing Array Elements Using a Pointer

Declare an array of integers, for example, `int arr[5] = {10, 20, 30, 40, 50};`.

Use a pointer to print the second and fourth elements in the array without directly accessing `arr[1]` or `arr[3]`.

## 3. Passing Pointers to Functions

Write a function `updateValue` that takes a pointer to an integer as its parameter.

Inside the function, set the integer's value to 100.

In `main`, declare an integer variable, call `updateValue` with the address of this variable, and print the variable's value before and after the function call.

## 4. Pointer Arithmetic

Declare an array of floats, `float nums[3] = {1.1, 2.2, 3.3};`.

Declare a pointer `p` that points to the first element of `nums`.

Use pointer arithmetic to print each element of `nums` by moving the pointer.

## 5. Comparing Two Variables Using Pointers

Declare two integer variables, `x` and `y`, with different values.

Write a function `compare` that takes two pointers to integers.

Inside the function, compare the values and print the larger of the two.

Call the function from `main` and observe the output.

## 6. Copying a Value Using Pointers

Declare an integer variable `source` with a value, and another integer `destination`.

Write a function `copyValue` that takes pointers to both variables and copies the value from `source` to `destination`.

Print both variables to confirm the copy.

## 7. Finding the Maximum in an Array Using a Pointer

Declare an array of integers and initialize it with some values.

Write a function `findMax` that takes a pointer to the array and its size as parameters.

Using pointer dereferencing, find and return the maximum value in the array.