Name: Mark Achiles G. Flores Jr. **Assessment:** Pointers Hands-on
Year & Section: BSCS 1-4

Activity 1: **Pointer Arithmetic**
Imagine you have an array of integers, and you need to perform various operations using pointers:

1. Declare an integer array of size 5 and initialize it with some values.
2. Declare an integer pointer and assign it the address of the first element of the array.
3. Use pointer arithmetic to perform the following operations:
    a. Print the value at the third element of the array using the pointer.
    b. Increment the pointer by 2 positions and print the value it points to.
    c. Subtract 1 from the value pointed to by the pointer and print the result.
    d. Move the pointer back to the first element and print the value using the pointer.

```c
#include <stdio.h>

int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int *pointer = NULL;
    pointer = &arr[0];

    pointer = &arr[2];
    printf("Third element of array: %d\n", *pointer);

    pointer += 2;
    printf("Incremented pointer by 2: %d\n", *pointer);

    *pointer -= 1;
    printf("Substract 1 from value at pointer: %d\n", *pointer);

    pointer = &arr[0];
    printf("First element of array: %d\n", *pointer);

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Assessment:** Pointers Hands-on
Year & Section: BSCS 1-4

Activity 2: **Passing Pointers to Functions**
Consider a scenario where you need to write a function that modifies the values in an array:

1. Declare an integer array of size 5 and initialize it with some values.
2. Write a function called **multiplyByTwo** that takes an integer pointer as an argument and multiplies each element in the array by 2.
3. Call the **multiplyByTwo** function, passing the address of the first element of the array.
4. Print the modified array to verify that the values have been multiplied by 2.

```c
#include <stdio.h>

int multiplyByTwo(int *ptr) {
    for (int i=0; i<5; i++) {
        *ptr *= 2;
        ptr++;
    }
}

int main() {
    int arr[5] = {1, 2, 3, 4, 5};

    printf("Input:\n");
    for (int i=0; i<5; i++) {
        printf("element %d - %d\n", i, arr[i]);
    }

    multiplyByTwo(&arr[0]);

    printf("\nOutput:\n");
    for (int i=0; i<5; i++) {
        printf("element %d - %d\n", i, arr[i]);
    }

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Assessment:** Pointers Hands-on
Year & Section: BSCS 1-4

Activity 3: **Swapping Two Pointers**

You are tasked with swapping the values of two pointer variables. Write a program that:

1. Declares two integer pointer variables, **ptrA** and **ptrB**, and assigns them the addresses of two different integer variables.
2. Write a function called **swapPointers** that takes two integer pointers as parameters.
3. Inside the function, swap the values of the two pointers.
4. In the main() function, print the values of **ptrA** and **ptrB** before calling the **swapPointers** function.
5. Call the **swapPointers** function, passing **ptrA** and **ptrB** as arguments.
6. Print the values of **ptrA** and **ptrB** after calling the function. They should be swapped.

```c
#include <stdio.h>
void swapPointers (int **pa, int **pb){
    int *temp = *pa;
    *pa = *pb;
    *pb = temp;
}

int main() {
    int a=1, b=2;
    int *ptrA=NULL, *ptrB=NULL;
    ptrA = &a;
    ptrB = &b;

    printf("Pointer a: %p\n", ptrA);
    printf("Pointer b: %p\n", ptrB);

    swapPointers(&ptrA, &ptrB);

    printf("Pointer a: %p\n", ptrA);
    printf("Pointer b: %p\n", ptrB);

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Assessment:** Pointers Hands-on
Year & Section: BSCS 1-4

Activity 4: **Function Pointers**
Create a program that demonstrates the use of function pointers. Define multiple functions that perform different mathematical operations, such as addition, subtraction, multiplication, and division. Use a function pointer to dynamically select and call one of these functions based on user input.

```c
#include <stdio.h>

int add(int a, int b) {
    return a + b;
}

int subtract(int a, int b) {
    return a - b;
}

int multiply(int a, int b) {
    return a * b;
}

int divide(int a, int b) {
    return a / b;
}

int main() {
    int a, b;
    char operator;
    int (*operation)(int, int); // Function pointer declaration

    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    printf("Enter the operator (+, -, *, /): ");
    scanf(" %c", &operator);

    switch (operator) {
        case '+':
            operation = add;
            break;
        case '-':
            operation = subtract;
            break;
        case '*':
            operation = multiply;
            break;
        case '/':
            operation = divide;
            break;
        default:
            printf("Invalid operator!\n");
            return 1;
    }

    int result = operation(a, b); // Calling the selected operation using the function pointer

    printf("Result: %d\n", result);

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Assessment:** Pointers Hands-on
Year & Section: BSCS 1-4

Activity 5: **Calculate Average**
You have an array of integers data of size 10. Write a program that asks the user to enter ten integers and stores them in the array. Then, using pointers, calculate and display the average value of the elements in the array.

```c
#include <stdio.h>

float average(int *ptr) {
    float sum;
    for (int i=0; i<10; i++) {
        sum += *ptr;
        ptr++;
    }

    return sum / 10.0;
}

int main() {
    int arr[10];

    printf("Enter 10 integers:\n");
    for (int i=0; i<10; i++) {
        printf("element - %d: ", i);
        scanf("%d", &arr[i]);
    }

    float ave = average(&arr[0]);

    printf("The average of the 10 integers is: %.2f", ave);

    return 0;
}
```