LAB 06
CLOSURE

iOS App development

Example: calculateSum

- Create a function called `calculateSum` that takes an array of integers and a closure as parameters.
- The closure should accept two integers and return an integer representing their sum.
- The function should use the closure to calculate the sum of all elements in the array.

```
// Function Definition
func calculateSum(_ numbers: [Int], _ closure: (Int, Int) -> Int) -> Int {
    var sum = 0
    for number in numbers {
        sum = closure(sum, number)
    }
    return sum
}

// Example Usage
let numbers = [1, 2, 3, 4, 5]
let sum = calculateSum(numbers) { $0 + $1 }
print(sum) // Output: 15
```

1. Calculate Average

- Create a function called `calculateAverage` that takes an array of integers and a closure as parameters.
- The closure should accept an array of integers and return an integer representing the average of those numbers.
- The function should use the closure to traverse and calculate the average of the elements in the integer array.

```
// Example Usage
  let numbers = [1, 2, 3, 4, 5]
  let average =   //???

  print(average) // Output: 3
```

2. Map to Uppercase

- Create a function called `mapToUppercase` that takes an array of strings and a closure as parameters.
- The closure should accept a string and return a string representing the uppercase version of that string.
- The function should return a new array containing the uppercase versions of the strings from the original array.

```
// Example Usage
let strings = ["apple", "banana", "orange"]
let uppercaseStrings = // ???

print(uppercaseStrings)   // Output: ["APPLE", "BANANA", "ORANGE"]
```

3. Filter Odd Numbers

- Create a function called `filterOddNumbers` that takes an array of integers and a closure as parameters.
- The closure should accept an integer and return a boolean indicating whether the number is odd.
- The function should return a new array containing only the odd numbers from the original array.

```
// Example Usage
  let numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  let oddNumbers =   //????


  print(oddNumbers) // Output: [1, 3, 5, 7, 9]
```

4. Find Longest String

- Create a function called `findLongestString` that takes an array of strings and a closure as parameters.
- The closure should accept two strings and return a boolean indicating which string is longer.
- The function should return the longest string from the original array.

```
// Example Usage
let strings = ["apple", "banana", "orange", "kiwi"]
let longestString = ///???
    print(longestString) // Output: "banana"
```