# Seneca Final SBA- Python track

**1.** Write a Swift program that accept two integer values and return true if one of them is 50 or if their sum is 50.

Solution:

```
func check_50(x: Int, y: Int) -> Bool {

    if x + y == 50 || x == 50 || y == 50

    {

        return true

    }

    else

    {

        return false

    }

}
```

**2.** Write a Swift program to test whether the last digit of the two given non-negative integer values are same or not.

```
func check_last_Digit(a: Int, b: Int) -> Bool {
    guard a < 0, b < 0
    else
     {
        if a % 10 == b % 10
        {
            return true
        }
        else
        {
            return false
        }
    }
    return false
}
```

3. Write a Swift program to check if 10 appears as either the first or last element in a given array of integers.

```
func first_last_5( _ arra:[Int]) -> Bool {
    if arra.first == 10 || arra.last == 10
```

```
    {
        return true
    }
    else
    {
        return false
    }
}
 print(first_last_10([1, 2, 10]))
print(first_last_10([10, 1, 2, 3, 4]))
print(first_last_10([10, 6, 1, 2, 10]))
print(first_last_10([1, 2, 6, 10, 3, 7]))
```

4. Write a Swift program to check whether the first element and the last element of a given array of integers are equal.

```
func check_first_last(arr: [Int]) -> Bool {
    guard arr.count > 0 else
    {
        return false
    }
    if arr.first == arr.last
    {
        return true
    } else
    {
        return false
    }
}
print(check_first_last([1, 2, 3]))
print(check_first_last([1, 2, 3, 1]))
print(check_first_last([1, 2, 2, 1]))
print(check_first_last([1]))
```

5. Write a Swift program to test if an array of integers does not contain a 30 or a 50.

```
func no35(_ a: [Int]) -> Bool {
    if a.contains(30) || a.contains(50)
    {
        return false
    }
    else
    {
        return true
    }
}
print(no35([20, 50]))
print(no35([20, 30, 50]))
print(no35([20, 40, 70]))
```

6. Write a Swift program to swap the first and last elements of a given array of integers. Return the modified array.

```swift
func swap_elements(_ a: [Int]) -> [Int] {
    var new_array = a
    let first_element = new_array.removeFirst()
    let last_element = new_array.removeLast()
    new_array.insert(last_element, at: new_array.startIndex)
    new_array.append(first_element)

    return new_array
}

print(swap_elements([1, 2, 3, 4]))
print(swap_elements([1, 2, 3]))
print(swap_elements([11, 12, 13, 14, 15]))
```