Difficulties I Overcame

1. Handling cases in my functions in which n was negative or zero.
2. Not losing my value of the original variable when swapping. (I used another variable to store my value)
3. Figuring out the separate function. I had to create a new array.
4. Figuring out the subsequence function. I had to loop until I found a match and then hold the position of the match in a separate variable.

Test Cases

Function: appendToAll

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

Check for a negative value for n: (cast, -1, “!!!”)

Check for a positive value for n: (cast, 1, “!!!”)

Check for when n is 0; (cast, 0, “!!!”)

Function: lookup

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is 0: (names, 0, “rick”)

When n is positive: (names, 3, “rick”)

When target not present: (names, 3, “yoyoyo”)

If more than one position is equal to target: (names, 3, “rick”)

Function: positionOfMax

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is negative: (cast, -2)

When n is 0: (cast, 0)

When n is positive: (cast, 2)

When there is more than one max value: (cast, 3)

Function: rotateLeft

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is negative: (cast, -1, 1)

When pos is negative: (cast, 3, -1)

When pos is greater than n: (cast, 3, 4)

When pos is equal to n: (cast, 3, 3)

When n is 0: (cast, 0, 1)

When n is positive: (cast, 2, 1)

When position is 0: (cast, 2, 0)

Function: rotateRight

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is negative: (cast, -1, 1)

When pos is negative: (cast, 3, -1)

When pos is greater than n: (cast, 3, 4)

When pos is equal to n: (cast, 3, 3)

When n is 0: (cast, 0, 1)

When n is positive: (cast, 2, 1)

When position is 0: (cast, 2, 0)

Function: flip

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is negative: (roles, -2)

When n is zero: (roles, 0) //my code returns negative 1

When n is positive: (roles, 2)

Function: differ

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n1 and n2 is equal to 0: (roles, 0, roles2, 0)

When just n1 is equal to 0: (roles, 0, roles2, 3)

When just n2 is equal to 0: (roles, 3, roles2, 0)

When n1 is smaller than n2: (roles, 1, roles2, 4)

When n2 is smaller than n1: (roles, 3, roles2, 1)

When n1 or n2 is negative or both: (roles, -2, roles2, 1) or (roles, 2, roles2, -2) or (roles, -2, roles2, -2)

When n1 and n2 are equal: (roles, 3, roles2, 3)

Function: subsequence

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When arguments are legitimate: (names, 5, names1, 3)

When n1 or n2 is negative: (names, -1, names1, 2) or (names, 1, names1, -2)

(names, -1, names1, -1) return -1

When n1 is 0 and n2 is 0: (names, 0, names1, 0) return 0

When just n1 is 0: (names, 0, names 1, 2) return -1

When just n2 is 0: (names, 3, names1, 0) return 0

When n2 is greater than n1: (names, 3, names1, 5) return -1

\*also test if the subsequence that you are looking for starts to appear in the array but then stops matching, but then reappears in full later on in the array.

Function: lookupAny

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n1 or n2 is negative: (names, -2, set1, 4) or (names, 2, set1, -3)

Returns -1

When n1 is 0: (names, 0, set1, 4) returns -1

When n2 is 0: (names, 3, set1, 0) returns -1

When no elements are equal: (names, 4, names1, 3)

When there is more than 1 match: (names, 5, set1, 4) only return position of first match

Function: separate

Reason: to make sure the desired effect is achieved along with outputting the proper return value.

When n is 0: (cast2, 0, “daryl) returns -1

When n is negative: (cast2, -2, “daryl”) return -1

When separator is not in array: (cast, 6, “glenn”)

When separator is in the array: (cast2, 4, “daryl”)