Ojas Bardiya

UID: 505145284

1. One of the main problems I encountered while developing this program was correctly developing the code for the shiftRight function. Taking into account that the array parameter closely resembles the call-by reference parameter, it was necessary to make sure that previous elements of array didn’t changed after they had been set to the desired value. I managed to this in the end by moving from the last array element to the first, instead of first to last.
2. Data types that could be used for testing:
3. For hasDuplicates:

Array used:

alphabet[7] = { “alpha” , “gamma” , “delta” , “beta”, “theta” , “Beta“, “alpha”)

* (alphabet[7], 3) – should return false as there is no occurrence of duplicates amongst the first three array elements .
* (alphabet[7], -1) – should return false as size < 0.
* (alphabet[7], 0) – should return false as size is 0.
* (alphabet[7], 6) – should return false as there is no occurrence of duplicates amongst the first six array elements .
* (alphabet[7], 7) – should return true as a[0] == a[6].

1. For countallDigits:

Array used:

random[6] == { “Samuel\_1” , “level”, “George”, “curious100”, “delig88t”, “country0”}

* (random[6], 0) – should return -1.
* (random[6], -8) – should return -1.
* (random[6], 3) – should return 1.
* (random[6], 5) – should return 6.
* (random[6], 6) – should return 7.
* (random[6], 4) – should return 4.

1. For isInDecreasingOrder:

Array used:

order[6] == { “samuel\_1” , “level”, “George”, “curious100”, “Delig88t”, “country0”,}

* (order[6], 0) – should return true.
* (order[6], 1) – should return true.
* (order[6], -5) – should return false.
* (order[6], 2) – should return true.
* (order[6], 3) – should return true.
* (order[6], 4) – should return false.
* (order[6], 5) – should return false.
* (order[6], 6) – should return false.

1. For Shiftright:

Array used:

order[6] == { “samuel\_1” , “level”, “George”, “curious100”, “Delig88t”, “country0”,}

* (order[6], 6 , 0, “tom”) – should return 0.
* (order[6], 0 , 4, “tom”) – should return 0.
* (order[6], 5, -7, “tom”) – should return -1.
* (order[6], -3 , 5, “tom”) – should return -1.
* (order[6], 6, 3, “tom”) – should return 3 and array should be changed to { “tom” , “tom”, “tom”, “samuel\_1”, “level”, “George” }
* (order[6], 6, 4, “tom”) – should return 4 and array should be changed to { “tom” , “tom”, “tom”, “tom”, “samuel\_1”, “level” }
* (order[6], 5, 3, “tom”) – should return 3 and array should be changed to { “tom” , “tom”, “tom”, “samuel\_1”, “level”, “country0” }
* (order[6], 6, 12, “tom”) – should return 6 and array should be changed to { “tom” , “tom”, “tom”, “tom”, “tom”, “tom” }
* (order[6], 5, 12, “tom”) – should return 6 and array should be changed to { “tom” , “tom”, “tom”, “tom”, “tom”, “country0” }

1. For find:

Array used:

order[6] == { “samuel\_1” , “level”, “George”, “curious100”, “Delig88t”, “country0”,}

* (order[6], -1, “level”) – should return -1.
* (order[6], 0, “George”) – should return -1.
* (order[6], 3, “George”) – should return 2.
* (order[6], 3, “country0”) – should return -1.
* (order[6], 6, “country0”) – should return 5.
* (order[6], 6, “countRy0”) – should return -1.

1. For replaceAllCharacters:

Array used:

order[6] == { “samuel\_1” , “level”, “George”, “curious100”, “Delig88t”, “country0”,}

* (order[6], 0, ‘e’, ‘#’) – should return -1.
* (order[6], -4, ‘e’, ‘#’) – should return -1.
* (order[6], 6, ‘e’, ‘#’) – should return 6 and array should be changed to

{ “samu#l\_1” , “l#v#l”, “G#org#”, “curious100”, “D#lig88t”, “country0”,}

* (order[6], 4, ‘e’, ‘#’) – should return 5 and array should be changed to

{ “samu#l\_1” , “l#v#l”, “G#org#”, “curious100”, “Delig88t”, “country0”,}