Lab # 7

*Sothea Song*

*100402876*

CPSC 1150 - 003

Instructor: H. Darbandi

Lab Title: Lab 7

Date submitted: July 4 2023

Department: CPSC

Program to create, print, reverse, sort, insert, and find unique in arrays

File Name: Lab 7.java

Purpose: Used to to create, print, reverse, sort, insert, and find unique in arrays.

Input: …

Output: initialized array, reversed array, sorted array, inserted array, unique array

Technical Information:

(You should fill the following information based on compiler and computer you are using).

Compiler: IntelliJ IDEA Community Edition 2023.1.1

Computer: (R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz, 16 GB of RAM

Operating System: Windows 10 Home Single Language

Language: Java

Program Logic (Pseudocode)

**reverse(int[] arr)**Algorithm: Reverse contents of the given array of integers without using a temporary array. The method does this by start reversing the array reversing the first element and the last element and continues on until everything is reversed (halfway through the array)

(definition)

* 1. Int[ ] arr : array to be reversed

START

1. j 🡨 arr.length-1
2. FOR i to arr.length/2
   1. temp 🡨 arr[i]
   2. arr[i] 🡨 arr[j]
   3. arr[j] 🡨 temp
   4. INCREMENT i
   5. DECREMENT j
3. OUTPUT arr

END

**insert(int[] arr, int n, int index)**

Algorithm: Insert number, int n, at index location of the array of int[] arr. The elements after the index is shifted to right, and the last element is dumped. The method returns true in case of success, and false in case of failure.

(definition)

* 1. Int[ ] arr : array to be reversed
  2. int n : element to be inserted
  3. int index : index where the element is inserted

START

1. l 🡨 arr.length
2. IF index <= l

2.1 arr[l-1] 🡨 n

2.2 FOR i = index to l

2.2.1 temp 🡨 arr[i]

2.2.2 arr[i] = arr[l-1]

2.2.3 arr[l-1] 🡨 temp

2.2.4 INCREMENT i

2.3 RETURN True

3. ELSE RETURN False

END

Generate your test cases based on the specifications in your lab assignment. Follow following format for each test case: (Refer to external document of your first lab)

*purpose*

*input*

*output*

*expected value*

*passed or failed*

Test Cases:

Test Case 1: Call the method from main method with size = 50, lowerBound = 10, and upperBound = 20.

Output:

13, 10, 18, 12, 14, 15, 13, 17, 11, 19, 11, 14, 15, 18, 14, 16, 19, 13, 14, 18, 20, 16, 16, 13, 20, 20, 13, 13, 15, 10, 17, 11, 11, 16, 18, 18, 12, 13, 17, 15, 17, 12, 13, 20, 18, 18, 12, 18, 18, 17

Expected:

13, 10, 18, 12, 14, 15, 13, 17, 11, 19, 11, 14, 15, 18, 14, 16, 19, 13, 14, 18, 20, 16, 16, 13, 20, 20, 13, 13, 15, 10, 17, 11, 11, 16, 18, 18, 12, 13, 17, 15, 17, 12, 13, 20, 18, 18, 12, 18, 18, 17

Passed

Test Case 2: Pass the array created in item#1 to this method.

Output:

13, 10, 18, 12, 14, 15, 13, 17, 11, 19, 11, 14, 15, 18, 14, 16, 19, 13, 14, 18, 20, 16, 16, 13, 20, 20, 13, 13, 15, 10, 17, 11, 11, 16, 18, 18, 12, 13, 17, 15, 17, 12, 13, 20, 18, 18, 12, 18, 18, 17

Expected:

13, 10, 18, 12, 14, 15, 13, 17, 11, 19, 11, 14, 15, 18, 14, 16, 19, 13, 14, 18, 20, 16, 16, 13, 20, 20, 13, 13, 15, 10, 17, 11, 11, 16, 18, 18, 12, 13, 17, 15, 17, 12, 13, 20, 18, 18, 12, 18, 18, 17

Passed

Test Case 3: Pass the array created in item#1 to this method to reverse its contents, and then pass it to

method printArray( ... ) to print its contents.

Output:

17, 18, 18, 12, 18, 18, 20, 13, 12, 17, 15, 17, 13, 12, 18, 18, 16, 11, 11, 17, 10, 15, 13, 13, 20, 20, 13, 16, 16, 20, 18, 14, 13, 19, 16, 14, 18, 15, 14, 11, 19, 11, 17, 13, 15, 14, 12, 18, 10, 13

Expected:

17, 18, 18, 12, 18, 18, 20, 13, 12, 17, 15, 17, 13, 12, 18, 18, 16, 11, 11, 17, 10, 15, 13, 13, 20, 20, 13, 16, 16, 20, 18, 14, 13, 19, 16, 14, 18, 15, 14, 11, 19, 11, 17, 13, 15, 14, 12, 18, 10, 13

Passed

Test Case 4: Test Case 3: Pass the array created in item#1 to this method to reverse its contents, and then pass it to method printArray( ... ) to print its contents.

Output:

10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13, 13, 13, 13, 13, 14, 14, 14, 14, 15, 15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 17, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 19, 19, 20, 20, 20, 20

Expected:

10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13, 13, 13, 13, 13, 14, 14, 14, 14, 15, 15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 17, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 19, 19, 20, 20, 20, 20

Passed

Test Case 5a: Create a array with length equal to 10, lowerBound = 5, and upperBound = 10, and then pass the array to this method.

int[] arr2=initialize(10, 5, 10);

count(arr2);

Output:

(array: 6, 6, 5, 10, 6, 9, 8, 8, 6, 7)

5 occurs = 1

6 occurs = 4

7 occurs = 1

8 occurs = 2

9 occurs = 1

10 occurs = 1

Expected:

5 occurs = 1

6 occurs = 4

7 occurs = 1

8 occurs = 2

9 occurs = 1

10 occurs = 1

Passed

Test Case 5b: Create another array with length equal to 100, lowerBound = -200, and upperBound = 100, and then pass the array to this method.

int[] arr3=initialize(100, -200, 100);

count(arr3);

Output:

(array: -160, -190, -140, -56, -159, 91, -44, -96, -36, -197, -45, -62, -13, -163, -64, -174, -162, -114, -141, -143, 44, -114, -162, 30, -166, 54, 78, 95, -108, 20, -160, 50, -116, -4, -19, -57, -192, -1, -160, 52, -69, -169, -79, -24, -35, 86, 93, 92, -90, 54, -4, 50, -40, -184, -36, -128, -3, -197, -72, -125, 49, -157, -65, -9, -197, -153, -149, -151, -185, -190, 70, 27, -177, -23, 21, -32, -24, -37, -85, -15, 38, 15, -6, 57, -62, -149, 85, -4, 39, 1, -68, -123, -66, -73, -123, -126, -54, -89, -57, 98)

-197 occurs = 3

-192 occurs = 1

-190 occurs = 2

-185 occurs = 1

-184 occurs = 1

-177 occurs = 1

-174 occurs = 1

-169 occurs = 1

-166 occurs = 1

-163 occurs = 1

-162 occurs = 2

-160 occurs = 3

-159 occurs = 1

-157 occurs = 1

-153 occurs = 1

-151 occurs = 1

-149 occurs = 2

-143 occurs = 1

-141 occurs = 1

-140 occurs = 1

-128 occurs = 1

-126 occurs = 1

-125 occurs = 1

-123 occurs = 2

-116 occurs = 1

-114 occurs = 2

-108 occurs = 1

-96 occurs = 1

-90 occurs = 1

-89 occurs = 1

-85 occurs = 1

-79 occurs = 1

-73 occurs = 1

-72 occurs = 1

-69 occurs = 1

-68 occurs = 1

-66 occurs = 1

-65 occurs = 1

-64 occurs = 1

-62 occurs = 2

-57 occurs = 2

-56 occurs = 1

-54 occurs = 1

-45 occurs = 1

-44 occurs = 1

-40 occurs = 1

-37 occurs = 1

-36 occurs = 2

-35 occurs = 1

-32 occurs = 1

-24 occurs = 2

-23 occurs = 1

-19 occurs = 1

-15 occurs = 1

-13 occurs = 1

-9 occurs = 1

-6 occurs = 1

-4 occurs = 3

-3 occurs = 1

-1 occurs = 1

1 occurs = 1

15 occurs = 1

20 occurs = 1

21 occurs = 1

27 occurs = 1

30 occurs = 1

38 occurs = 1

39 occurs = 1

44 occurs = 1

49 occurs = 1

50 occurs = 2

52 occurs = 1

54 occurs = 2

57 occurs = 1

70 occurs = 1

78 occurs = 1

85 occurs = 1

86 occurs = 1

91 occurs = 1

92 occurs = 1

93 occurs = 1

95 occurs = 1

98 occurs = 1

Expected:

-197 occurs = 3

-192 occurs = 1

-190 occurs = 2

-185 occurs = 1

-184 occurs = 1

-177 occurs = 1

-174 occurs = 1

-169 occurs = 1

-166 occurs = 1

-163 occurs = 1

-162 occurs = 2

-160 occurs = 3

-159 occurs = 1

-157 occurs = 1

-153 occurs = 1

-151 occurs = 1

-149 occurs = 2

-143 occurs = 1

-141 occurs = 1

-140 occurs = 1

-128 occurs = 1

-126 occurs = 1

-125 occurs = 1

-123 occurs = 2

-116 occurs = 1

-114 occurs = 2

-108 occurs = 1

-96 occurs = 1

-90 occurs = 1

-89 occurs = 1

-85 occurs = 1

-79 occurs = 1

-73 occurs = 1

-72 occurs = 1

-69 occurs = 1

-68 occurs = 1

-66 occurs = 1

-65 occurs = 1

-64 occurs = 1

-62 occurs = 2

-57 occurs = 2

-56 occurs = 1

-54 occurs = 1

-45 occurs = 1

-44 occurs = 1

-40 occurs = 1

-37 occurs = 1

-36 occurs = 2

-35 occurs = 1

-32 occurs = 1

-24 occurs = 2

-23 occurs = 1

-19 occurs = 1

-15 occurs = 1

-13 occurs = 1

-9 occurs = 1

-6 occurs = 1

-4 occurs = 3

-3 occurs = 1

-1 occurs = 1

1 occurs = 1

15 occurs = 1

20 occurs = 1

21 occurs = 1

27 occurs = 1

30 occurs = 1

38 occurs = 1

39 occurs = 1

44 occurs = 1

49 occurs = 1

50 occurs = 2

52 occurs = 1

54 occurs = 2

57 occurs = 1

70 occurs = 1

78 occurs = 1

85 occurs = 1

86 occurs = 1

91 occurs = 1

92 occurs = 1

93 occurs = 1

95 occurs = 1

98 occurs = 1

Passed

Test case 6a: inserted into index that is available

insert(arr, 5, 3);

Output:

(array: 10, -11, -1, -1, 1, -14, -12, -3, 0, -16)

10, -11, -1, 5, -1, 1, -14, -12, -3, 0

Expected:

10, -11, -1, 5, -1, 1, -14, -12, -3, 0

Passed

Test case 6b: inserted into index that is not available

insert(arr, 5, 10);

Output:

(array: 6, 9, -2, 0, -10, -11, -11, -3, -4, -6)

Element is not inserted into array.

Expected:

Element is not inserted into array.

Passed

Test case 6c: inserted into index that is not available

insert(arr, 5, -1);

Output:

(array: -12, -1, -9, -4, 2, -3, -9, -3, -14, 3)

Element is not inserted into array.

Expected:

Element is not inserted into array.

Passed