Student Name: Amangeldi Madi

Class and Section CS-2104

Total Points (20 pts) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Due: April 18, 2022. 10 AM**

Data Structures and Algorithms

Astana IT University

**Project 1: Same-Number Subsequence**

Problem Description:

Write an O(n) program that prompts the user to enter a sequence of integers ending with 0 and finds longest subsequence with the same number. Here is a sample run of the program:

<Output>

Enter a series of numbers ending with 0: 2 4 4 8 8 8 8 2 4 4 0

The longest same number sequence starts at index 3 with 4 values of 8

<End Output>

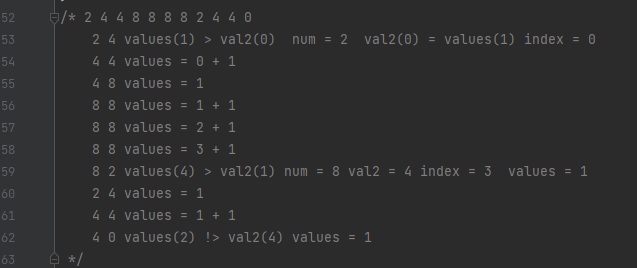
Design Algorithm:

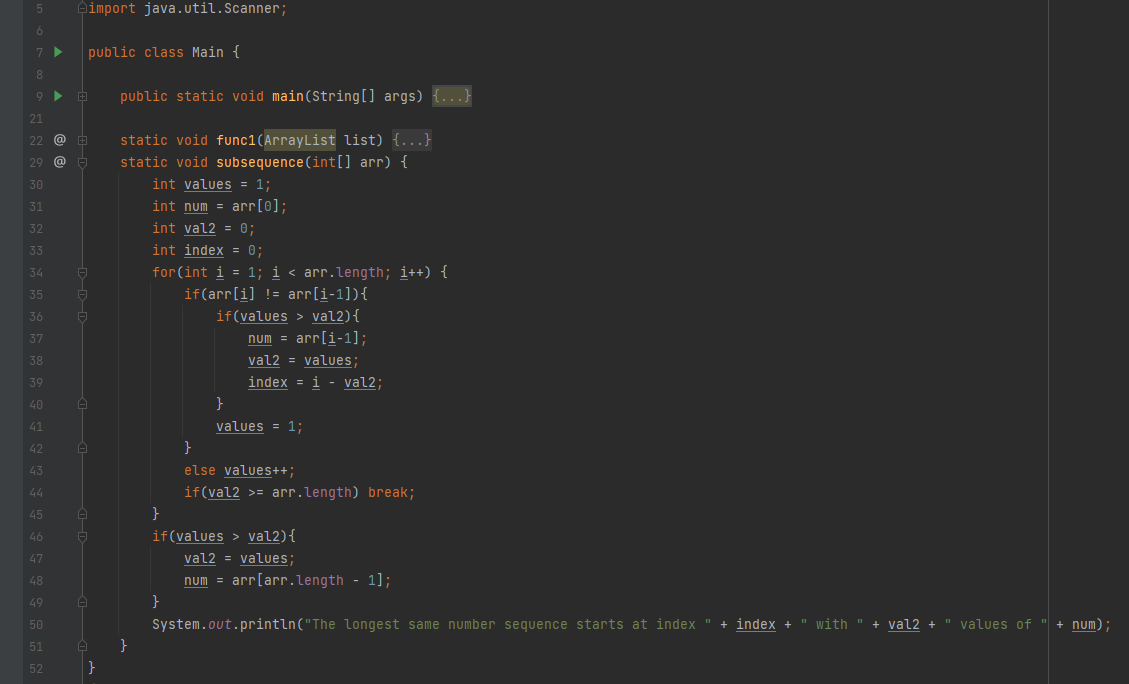
(Describe how your algorithm works)

Starting from the second value, algorithm checks if its equal to the previous value. If not, then it means that it is not a sequence and the value is single. Otherwise, it will increment a number of values and if its more than the first number of same values, it will make the first equal to the second.

Analysis of the Algorithm: (Describe how you analyze the algorithm)

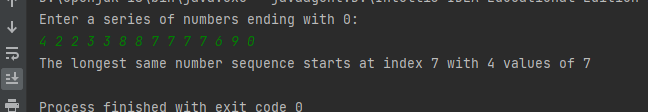
Initially, there are 4 variables declared and 2 of them are the values (quantity) for the similar numbers in sequence. Algorithm consists one if’ inside the loop for comparing max quantity with the current one, and outside for the last non-zero value case.

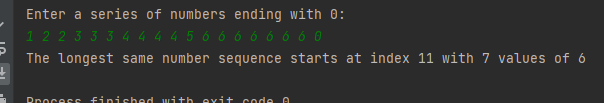


Coding: (Copy source code here)

Testing:

(Describe how you test your program, include screenshots of the output)





2. Fill in self-evaluation:

1. Can your program find the largest same number sequence? Yes.
2. Can your program find the correct start index of the largest same number sequence? Yes.
3. Is your algorithm O(n)? maybe

**Project 2: Reverse list**

Problem Description:

Implement the following two methods in O(n) time.

// Reverse the list and return it in O(n) time

**public** **static** <E> ArrayList<E> reverse(ArrayList<E> list)

// Reverse the list and return it in O(n) time

**public** **static** <E> LinkedList<E> reverse(LinkedList<E> list)

Use the following code to test these methods:

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.print("Enter 10 numbers: ");

ArrayList<Integer> list = **new** ArrayList<>();

**for** (**int** i = 0; i < 10; i++) {

list.add(input.nextInt());

}

*reverse*(list);

**for** (**int** i: list) {

System.***out***.print(i + " ");

}

System.***out***.println();

LinkedList<Integer> list1 = **new** LinkedList<>(list);

*reverse*(list1);

**for** (**int** i: list1) {

System.***out***.print(i + " ");

}

System.***out***.println();

}

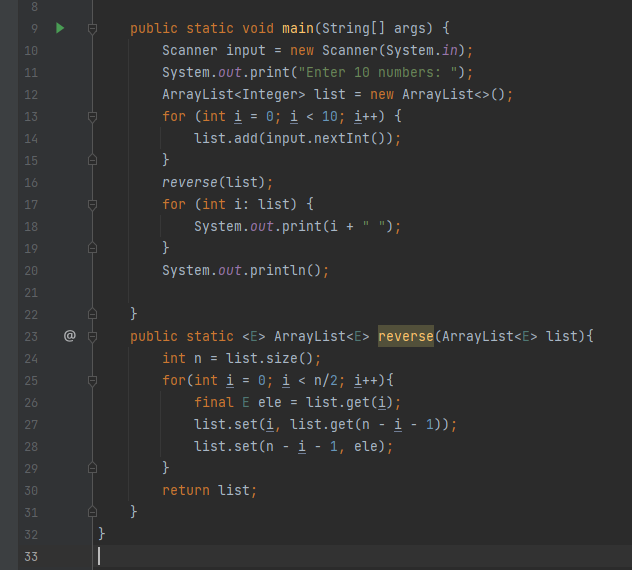
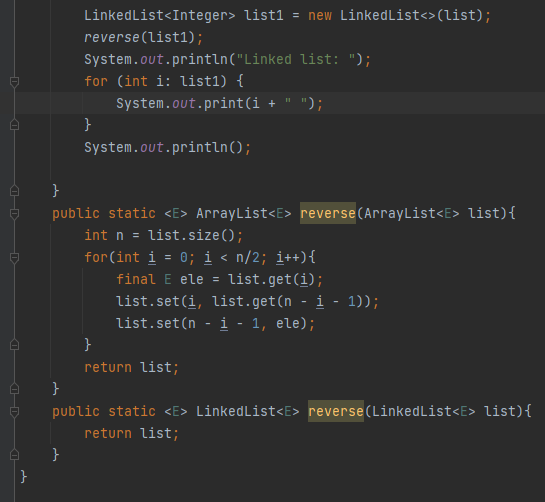
Analysis of the Algorithm: (Describe how you analyze the algorithm)

For ArrayList the loop swaps i-th element with a

‘length – i’-th element. I-th element is stored in a temp for this. Also loop is half the size for time consumption.

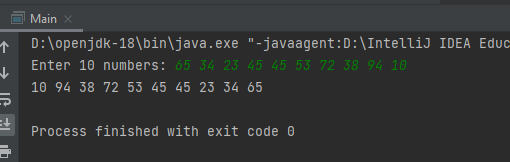
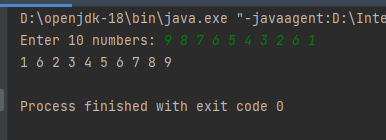
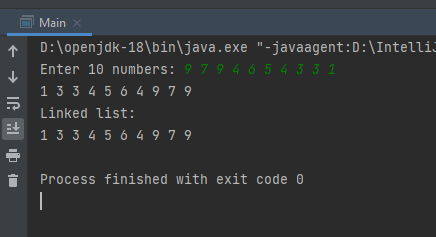
Linked list is mapped from arraylist. Couldn’t get how

to get the nodes and reverse them. Don’t know really much, but its possible in O(n) time.

Coding: (Copy source code here)

Testing:

(Describe how you test your program, include screenshots of the output)



What to submit?

1. Complete and submit the PDF of this document to Moodle.

2. Submit your Java file and PDF file as one ZIP file.