Assignment

Q1. Write a Java program to find the index of an array element.

import java.util.Scanner;

public class firstAssignment {

public static void main(String[] args) {

try (Scanner sc = new Scanner(System.in)) {

int[] array = { 10, 20, 30, 40, 50 };

int index=1, i, element;

System.out.println("Enter element to find index: ");

element = sc.nextInt();

for (i = 0; i < array.length; i++) {

if (array[i] == element) {

System.out.println("element index= " + index);

break;

}

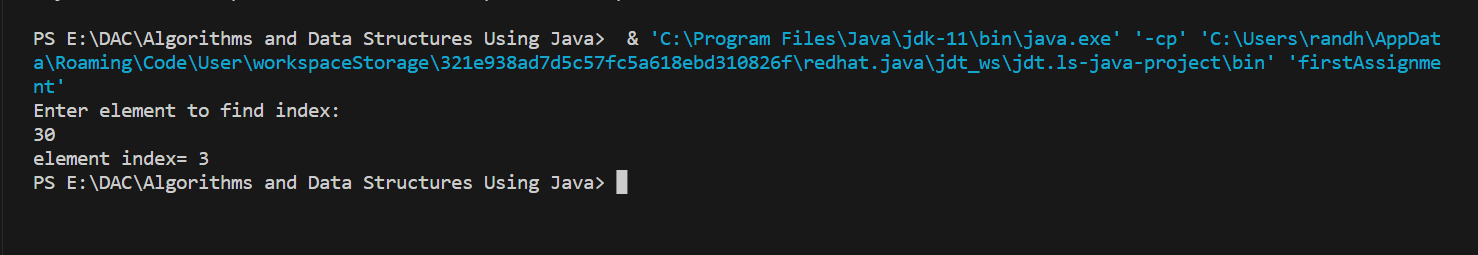
index++;

}

}

}

}



2. You will be given an array and you need to find the third largest

public class secondAssignment {

public static void main(String[] args) {

int[] array = { 50, 20, 30, 10, 25 };

System.out.print("Array elements: ");

for (int i = 0; i < array.length; i++) {

System.out.print(array[i]+" ");

}

for (int i = 0; i < array.length; i++) {

for (int j = 0; j < array.length - i - 1; j++) {

if (array[j] > array[j + 1]) {

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

// for (int i = 0; i < array.length; i++) {

// System.out.print(array[i]+" ");

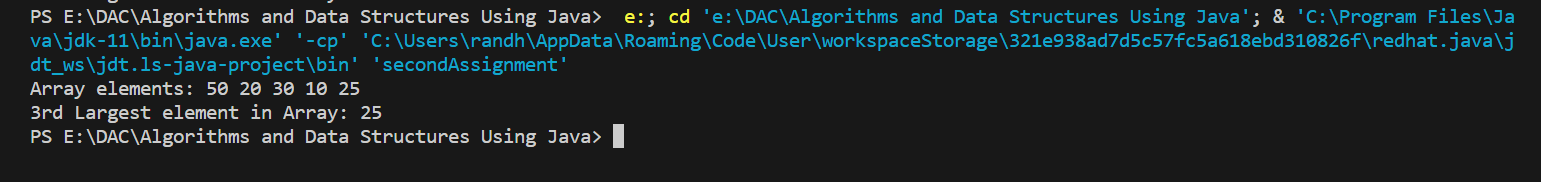
// }

System.out.println();

System.out.println("3rd Largest element in Array: " + array[2]);

}

}



3. You are given array consisting of n integers. Your task is to find the maximum length of an increasing subarray of the given array.

A subarray is the sequence of consecutive elements of the array. Subarray is called increasing if each element of this subarray strictly greater than previous.

i/p-5 1 7 2 11 15

o/p -3

public class thirdAssignment {

public static void main(String[] args) {

int[] array = {5,1,7,2,11,15};

System.out.print("Array elements: ");

for (int i = 0; i < array.length; i++) {

System.out.print(array[i] + " ");

}

System.out.println();

int count = 1;

for (int j = 0; j < array.length-1; j++) {

if (array[j] < array[j + 1])

count++;

else

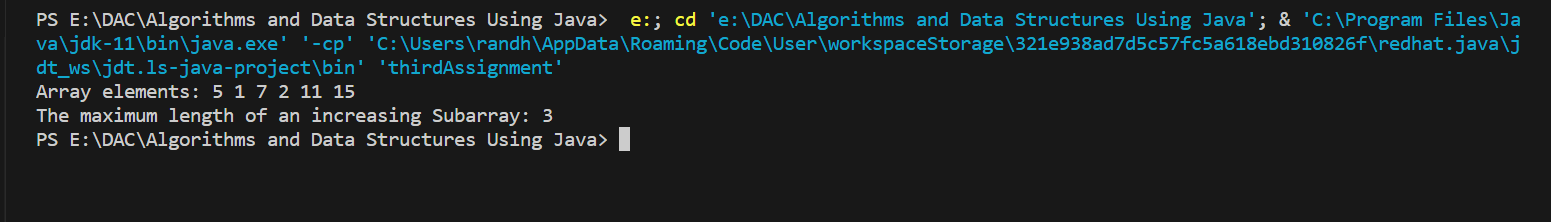
count = 1;

}

System.out.print("The maximum length of an increasing Subarray: "+count);

}

}



4. Rearrange Everything

Given an array of size n where all elements are in range from 0 to n-1, change contents of arr[] so that arr[i] = j is changed to arr[j] = i.

i/p - 22 13 9 41

o/p - 22 9 13 41

5) LL

Implement Linled List function

1. insertAtBeg

2. insertAtLAst

1. deleteAtBeg

2. deleteAtLAst

5. display

package day1\_linkedlist;

public class Node {

private int data;

private Node next;

public Node(int data) {

super();

this.data = data;

this.next = null;

}

public int getData() {

return data;

}

public void setData(int data) {

this.data = data;

}

public Node getNext() {

return next;

}

public void setNext(Node next) {

this.next = next;

}

}

package day1\_linkedlist;

public class LinkedList {

private Node head;

private Node temp;

Node del;

public boolean insertAtBeg(int data) {

Node nn=new Node(data);

if(nn==null) {

System.out.println("There are no value in the New node");

return false;

}

if(head==null) {

this.head=nn;

return true;

}

if(head!=null) {

temp=head;

nn.setNext(temp);

head=nn;

}

return true;

}

public void display() {

temp=head;

while(temp!=null) {

System.out.print(temp.getData()+" ");

temp=temp.getNext();

}

System.out.println("\n");

}

public boolean insertAtLast(int data) {

Node nn=new Node(data);

if(nn==null) {

System.out.println("nn is null");

return false;

}

if(head==null) {

this.head=nn;

return true;

}

temp=head;

while(temp.getNext()!=null) {

temp=temp.getNext();

}

temp.setNext(nn);

return true;

}

public boolean deleteAtBeg() {

if(head==null) {

System.out.println("No value is present in the list");

return false;

}

temp=head;

if(temp.getNext()==null) {

head=null;

return true;

}

if(temp.getNext()!=null) {

temp=temp.getNext();

head=temp;

return true;

}

return true;

}

public boolean deleteAtLast() {

if(head==null) {

System.out.println("No value is present in the list");

return false;

}

temp=head;

while(temp.getNext()!=null) {

del=temp;

temp=temp.getNext();

}

del.setNext(null);

return true;

}

}

package day1\_linkedlist;

public class Main {

public static void main(String[] args) {

LinkedList l1=new LinkedList();

l1.insertAtBeg(10);

l1.display();

l1.insertAtBeg(20);

l1.display();

l1.insertAtBeg(30);

l1.display();

l1.insertAtLast(40);

l1.display();

l1.insertAtLast(50);

l1.display();

l1.insertAtLast(60);

l1.display();

l1.insertAtBeg(70);

l1.display();

l1.deleteAtBeg();

l1.display();

l1.deleteAtLast();

l1.display();

}

}

