**Patuakhali Science and Technology University**



**Lab Problem 03**

(Java Basic Exercises 151 to 249)

Course Code: CCE 122

Course Title: Object Oriented Programming Sessional

**Submitted By:**

Mehadi Enam Nerjhor

ID: 2202050

REG: 11241

**Submitted To:**

Professor Dr. Md Samsuzzaman

Dept. of Computer and Communication Engineering

Faculty of Computer Science and Engineering

Date: 31 August 2024

Table of Contents

[151. Write a Java program to find the value of a specified expression. 1](#_Toc175961661)

[152. Write a Java program that accepts four integers from the user and prints equal if all four are equal, and not equal otherwise. 1](#_Toc175961662)

[153. Write a Java program that accepts two double variables and test if both strictly between 0 and 1 and false otherwise. 2](#_Toc175961663)

[154. Write a Java program to print the contents of a two-dimensional Boolean array where t represents true and f represents false. 3](#_Toc175961664)

[155. Write a Java program to print an array after changing the rows and columns of a two-dimensional array. 4](#_Toc175961665)

[156. Write a Java program that returns the largest integer but not larger than the base-2 logarithm of a specified integer. 6](#_Toc175961666)

[158. Write a Java program to create a two-dimensional array (m x m) A[][] such that A[i][j] is false if I and j are prime otherwise A[i][j] becomes true. 7](#_Toc175961667)

[159. Write a Java program to find the k largest elements in a given array. Elements in the array can be in any order. 9](#_Toc175961668)

[160. Write a Java program to find the k smallest elements in a given array. Elements in the array can be in any order. 11](#_Toc175961669)

[161. Write a Java program to find the kth smallest and largest element in a given array. Elements in the array can be in any order. 13](#_Toc175961670)

[162. Write a Java program that finds numbers greater than the average of an array. 15](#_Toc175961671)

[163. Write a Java program that will accept an integer and convert it into a binary representation. Now count the number of bits equal to zero in this representation. 17](#_Toc175961672)

[164. Write a Java program to divide the two given integers using the subtraction operator. 18](#_Toc175961673)

[165. Write a Java program to move every positive number to the right and every negative number to the left of a given array of integers. 19](#_Toc175961674)

[166. Write a Java program to transform a given integer into String format. 21](#_Toc175961675)

[167. Write a Java program to move every zero to the right side of a given array of integers. 22](#_Toc175961676)

[168. Write a Java program to multiply two given integers without using the multiply operator (\*). 24](#_Toc175961677)

[169. Write a Java program to reverse a sentence (assume a single space between two words) without reverse every word. 25](#_Toc175961678)

[170. Write a Java program to find the length of the longest consecutive sequence in a given array of integers. 25](#_Toc175961679)

[171. Write a Java program to accept two strings and test if the second string contains the first one. 27](#_Toc175961680)

[173. Write a Java program to find the median of the numbers inside the window (size k) at each step in a given array of integers with duplicate numbers. Move the window to the array start. 28](#_Toc175961681)

[174. Write a Java program to find the maximum number inside the number in the window (size k) at each step in a given array of integers with duplicate numbers. Move the window to the top of the array. 30](#_Toc175961682)

[175. Write a Java program to delete a specified node in the middle of a singly linked list. 32](#_Toc175961683)

[176. Write a Java program that partitions an array of integers into even and odd numbers. 34](#_Toc175961684)

[178. Write a Java program to find the longest increasing continuous subsequence in a given array of integers. 35](#_Toc175961685)

[179. Write a Java program to add one to a positive number represented as an array of digits. 37](#_Toc175961686)

[180. Write a Java program to swap two adjacent nodes in a linked list. 39](#_Toc175961687)

[181. Write a Java program to find the length of the last word in a given string. The string contains upper/lower-case alphabets and empty space characters like ' '. 41](#_Toc175961688)

[182. Write a Java program to check if two binary trees are identical. Assume that two binary trees have the same structure and every identical position has the same value. 42](#_Toc175961689)

[183. Write a Java program to accept a positive number and repeatedly add all its digits until the result has only one digit. 43](#_Toc175961690)

[185. Write a Java program to check if two strings are isomorphic or not. 44](#_Toc175961691)

[186. Write a Java program to check if a number is a strobogrammatic number. The number is represented as a string. 45](#_Toc175961692)

[187. Write a Java program to find the index of the first non-repeating character in a given string. 47](#_Toc175961693)

[189. Write a Java program to two non-negative integers num1 and num2 represented as strings, return the sum of num1 and num2. 49](#_Toc175961694)

[191. Write a Java program to test whether there are two integers x and y such that x^2 + y^2 is equal to a given positive number. 50](#_Toc175961695)

[192. Write a Java program to rearrange the alphabets in the order followed by the sum of digits in a given string containing uppercase alphabets and integer digits (from 0 to 9). 52](#_Toc175961696)

[194. Write a Java program to determine the all positions of a given number in a given matrix. If the number is not found print ("Number not found!"). 54](#_Toc175961697)

[195. Write a Java program to check if three given side lengths (integers) can make a triangle or not. 56](#_Toc175961698)

[197. Write a Java program to test if a given number (positive integer) is a perfect square or not. 57](#_Toc175961699)

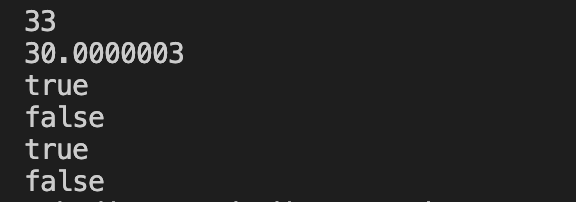
[198. Write a Java program to calculate the position of a given prime number. 58](#_Toc175961700)

# **151.** Write a Java program to find the value of a specified expression.

**Code:**

|  |
| --- |
| public class n151 {  public static void main(String[] args) {  System.err.println((101 + 0) / 3);  System.err.println(3.0e-6 \* 10000000.1);  System.err.println(true && true);  System.err.println(false && true);  System.err.println((false && false) || (true && true));  System.err.println((false || false) && (true && true));  }  } |

**Output:**

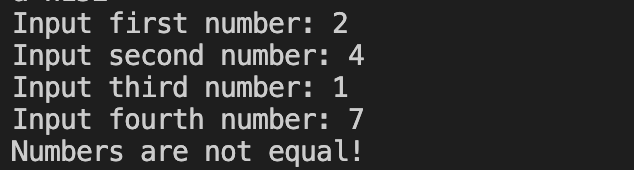
****

# **152.** Write a Java program that accepts four integers from the user and prints equal if all four are equal, and not equal otherwise.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n152 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int a, b, c, d;  System.out.print("Input first number: ");  a = input.nextInt();  System.out.print("Input second number: ");  b = input.nextInt();  System.out.print("Input third number: ");  c = input.nextInt();  System.out.print("Input fourth number: ");  d = input.nextInt();  System.out.println(a == b && a == c && a == d ? "Numbers are equal!" : "Numbers are not equal!");  input.close();  }  } |

**Output:**

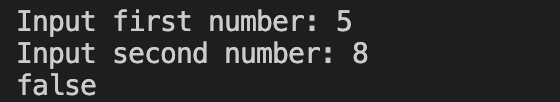
****

# **153.** Write a Java program that accepts two double variables and test if both strictly between 0 and 1 and false otherwise.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n153 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  double a, b;  System.out.print("Input first number: ");  a = input.nextDouble();  System.out.print("Input second number: ");  b = input.nextDouble();  System.out.println((a == 0 || a == 1) && (b == 0 || b == 1));  input.close();  }  } |

**Output:**

****

# **154.** Write a Java program to print the contents of a two-dimensional Boolean array where t represents true and f represents false.

**Code:**

|  |
| --- |
| public class n154 {  public static void main(String[] args) {  boolean[][] arr = {{true, false, true}, {false, true, false}};    for (int i = 0; i < 2; i++) {  for (int j = 0; j < 3; j++) {  System.out.print(arr[i][j] == true ? "t " : "f ");  }  System.out.println();  }  }  } |

**Output:**

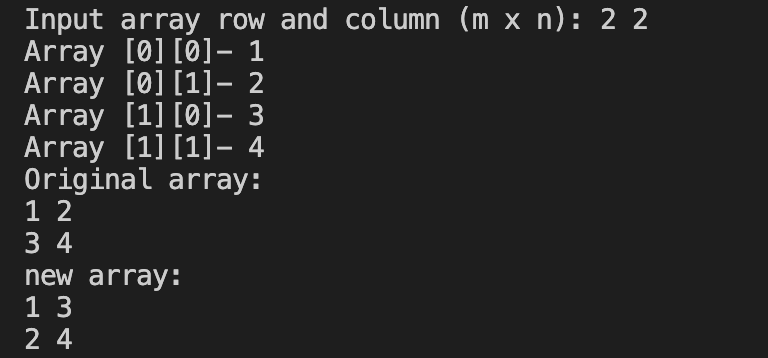
****

# **155.** Write a Java program to print an array after changing the rows and columns of a two-dimensional array.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n155 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int m, n;  System.out.print("Input array row and column (m x n): ");  m = input.nextInt();  n = input.nextInt();  int[][] arr1 = new int[m][n], arr2 = new int[n][m];  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  System.out.printf("Array [%d][%d]- ", i, j);  arr1[i][j] = input.nextInt();  }  }  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  arr2[j][i] = arr1[i][j];  }  }  System.out.println("Original array:");  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  System.out.print(arr1[i][j] + " ");  }  System.out.println();  }  System.out.println("new array:");  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  System.out.print(arr2[i][j] + " ");  }  System.out.println();  }  input.close();  }  } |

**Output:**

****

# **156.** Write a Java program that returns the largest integer but not larger than the base-2 logarithm of a specified integer.

**Code:**

|  |
| --- |
|  |

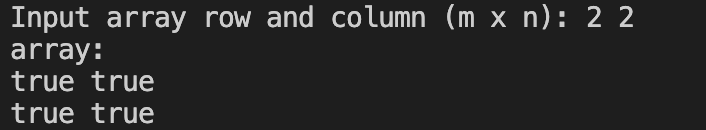
**Output:**

# **158.** Write a Java program to create a two-dimensional array (m x m) A[][] such that A[i][j] is false if I and j are prime otherwise A[i][j] becomes true.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n158 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int m, n, flag1 = 0, flag2 = 0;  System.out.print("Input array row and column (m x n): ");  m = input.nextInt();  n = input.nextInt();  boolean[][] arr = new boolean[m][n];  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  flag1 = 0;  flag2 = 0;  if (i <= 1 || j <= 1) {  flag1 = 1;  flag2 = 1;  }  else {  for (int j2 = 2; j2 < 100; j2++) {  if ((i % j2 == 0) && (i != j2)) {  flag1 = 1;  }  if ((j % j2 == 0) && (j != j2)) {  flag2 = 1;  }  }  }  if (flag1 == 1 || flag2 == 1) {  arr[i][j] = true;  }  else  {  arr[i][j] = false;  }  }  }  System.out.println("array:");  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  System.out.print(arr[i][j] + " ");  }  System.out.println();  }  input.close();  }  } |

**Output:**

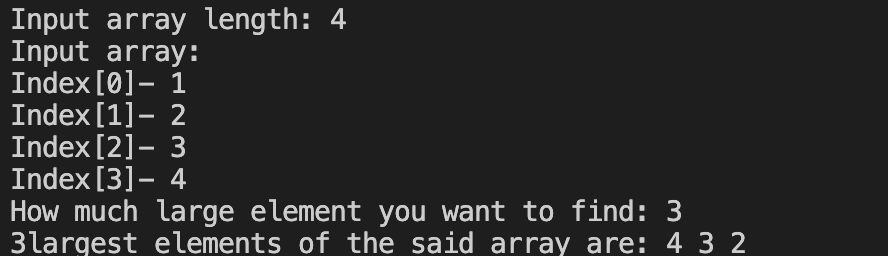
****

# **159.** Write a Java program to find the k largest elements in a given array. Elements in the array can be in any order.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n159 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, n2;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("How much large element you want to find: ");  n2 = input.nextInt();  int[] lElement = new int[n2];  for (int i = 0; i < n2; i++) {  lElement[i] = arr[0];  for (int j = 0; j < arr.length; j++) {  if (i > 0) {  if (arr[j] > lElement[i] && arr[j] < lElement[i-1]) {  lElement[i] = arr[j];  }  }  else {  if (arr[j] > lElement[i]) {  lElement[i] = arr[j];  }  }  }  }  System.out.print(n2 + "largest elements of the said array are: ");  for (int i = 0; i < lElement.length; i++) {  System.out.print(lElement[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

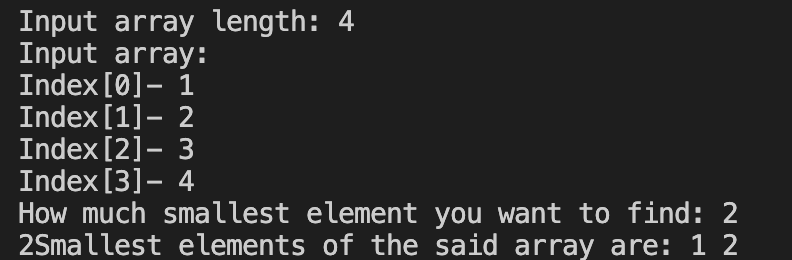
****

# **160.** Write a Java program to find the k smallest elements in a given array. Elements in the array can be in any order.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n160 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, n2;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("How much smallest element you want to find: ");  n2 = input.nextInt();  int[] sElement = new int[n2];    for (int i = 0; i < n2; i++) {  sElement[i] = 999999999;  for (int j = 0; j < arr.length; j++) {  if (i > 0) {  if (arr[j] < sElement[i] && arr[j] > sElement[i-1]) {  sElement[i] = arr[j];  }  }  else {  if (arr[j] < sElement[i]) {  sElement[i] = arr[j];  }  }    }    }  System.out.print(n2 + "Smallest elements of the said array are: ");  for (int i = 0; i < sElement.length; i++) {  System.out.print(sElement[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

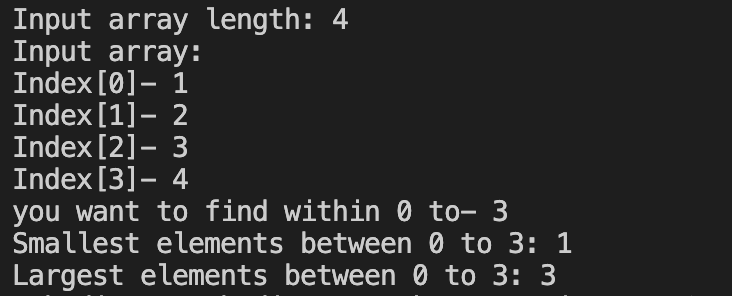
****

# **161.** Write a Java program to find the kth smallest and largest element in a given array. Elements in the array can be in any order.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n161 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, k, smallest, largest;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("you want to find within 0 to- ");  k = input.nextInt();  smallest = arr[0];  largest = arr[0];  for (int i = 0; i < k; i++) {  if (arr[i] < smallest) {  smallest = arr[i];  }  if (arr[i] > largest) {  largest = arr[i];  }  }  System.out.println("Smallest elements between 0 to " + k +": " + smallest);  System.out.println("Largest elements between 0 to " + k +": " + largest);  input.close();  }  } |

**Output:**

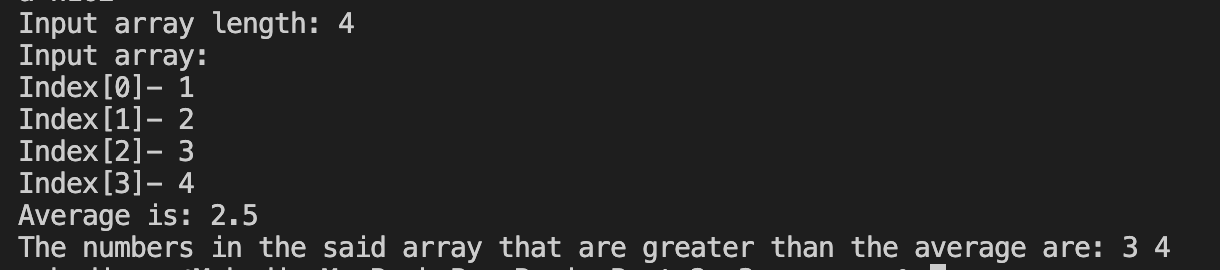
****

# **162.** Write a Java program that finds numbers greater than the average of an array.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n162 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n;  double avg = 0;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  avg += arr[i];  }  avg /= arr.length;  System.out.println("Average is: " + avg);  System.out.print("The numbers in the said array that are greater than the average are: ");  for (int i = 0; i < arr.length; i++) {  if (arr[i] > avg) {  System.out.print(arr[i] + " ");  }  }  System.out.println();  input.close();  }  } |

**Output:**

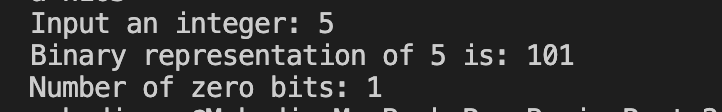
****

# **163.** Write a Java program that will accept an integer and convert it into a binary representation. Now count the number of bits equal to zero in this representation.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n163 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int a, count = 0;  System.out.print("Input an integer: ");  a = input.nextInt();  System.out.println("Binary representation of " + a + " is: " + Integer.toBinaryString(a));  for (int i = 0; i < Integer.toBinaryString(a).length(); i++) {  if (Integer.toBinaryString(a).charAt(i) == '0') {  count++;  }  }  System.out.println("Number of zero bits: " + count);    input.close();  }  } |

**Output:**

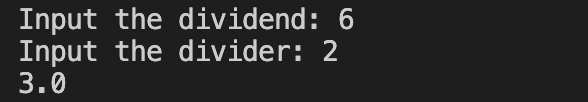
****

# **164.** Write a Java program to divide the two given integers using the subtraction operator.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n164 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int divident, divider, count = 0;  double point = 0;  System.out.print("Input the dividend: ");  divident = input.nextInt();  System.out.print("Input the divider: ");  divider = input.nextInt();  for (int i = divident; i >= divider; ) {  i -= divider;  count++;  if (i < divider) {  point = (double)i / (double)divider;  }  }  System.out.println(count + point);  input.close();  }  } |

**Output:**

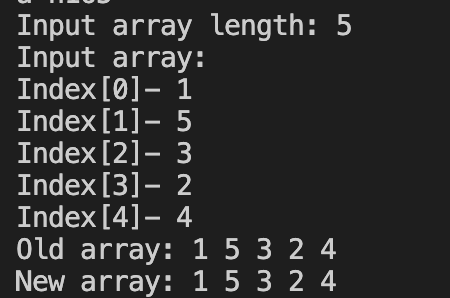
****

# **165.** Write a Java program to move every positive number to the right and every negative number to the left of a given array of integers.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n165 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);    int n, temp;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("Old array: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(arr[i] + " ");  }  for (int i = 0, j = 0; i < arr.length; i++) {  if (arr[i] < 0) {  temp = arr[j];  arr[j] = arr[i];  arr[i] = temp;  j++;  }  }  System.out.print("\nNew array: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(arr[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

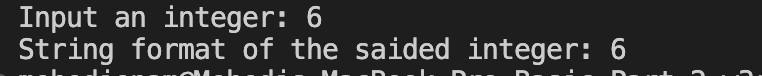
****

# **166.** Write a Java program to transform a given integer into String format.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n166 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int a;  System.out.print("Input an integer: ");  a = input.nextInt();  System.out.println("String format of the saided integer: " + Integer.toString(a));  input.close();  }  } |

**Output:**

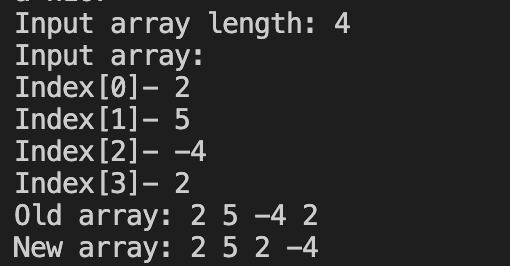
****

# **167.** Write a Java program to move every zero to the right side of a given array of integers.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n167 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);    int n, temp;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input array:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("Old array: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(arr[i] + " ");  }  for (int i = 0, j = 0; i < arr.length; i++) {  if (arr[i] > 0) {  temp = arr[j];  arr[j] = arr[i];  arr[i] = temp;  j++;  }  }  System.out.print("\nNew array: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(arr[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

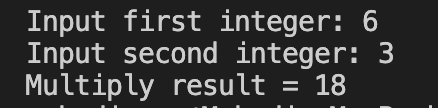
****

# **168.** Write a Java program to multiply two given integers without using the multiply operator (\*).

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n168 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int a, b, multiply = 0;  System.out.print("Input first integer: ");  a = input.nextInt();  System.out.print("Input second integer: ");  b = input.nextInt();  if (b < a) {  for (int i = 0; i < b; i++) {  multiply += a;  }  }  else {  for (int i = 0; i < a; i++) {  multiply += b;  }  }  System.out.println("Multiply result = " + multiply);  input.close();  }  } |

**Output:**

****

# **169.** Write a Java program to reverse a sentence (assume a single space between two words) without reverse every word.

**Code:**

|  |
| --- |
|  |

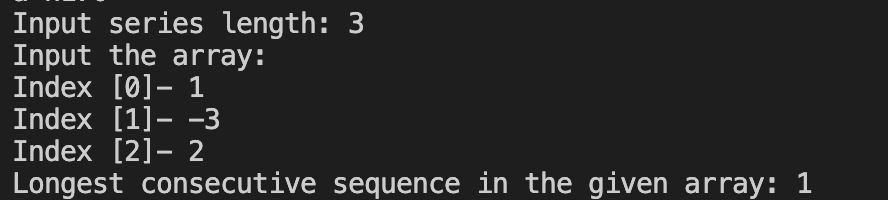
**Output:**

# **170.** Write a Java program to find the length of the longest consecutive sequence in a given array of integers.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n170 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, count = 1, flag = 1;  System.out.print("Input series length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the array: ");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index [%d]- ", i);  arr[i] = input.nextInt();  }  for (int i = 0; i < arr.length-1; i++) {  if ((arr[i] == arr[i+1]) || (arr[i+1] == (arr[i] + 1))) {  count++;  if (count > flag) {  flag = count;  }  }  else {  count = 1;  }    }  System.out.println("Longest consecutive sequence in the given array: " + flag);  input.close();  }  } |

**Output:**

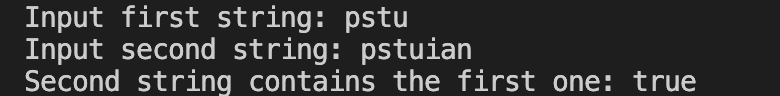
****

# **171.** Write a Java program to accept two strings and test if the second string contains the first one.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n171 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int count = 0;  String str1, str2;  System.out.print("Input first string: ");  str1 = input.nextLine();  System.out.print("Input second string: ");  str2 = input.nextLine();  for (int i = 0, j = 0; i < str2.length(); i++) {  if (str2.charAt(i) == str1.charAt(j)) {  count++;  j++;  if (count == str1.length()) {  break;  }  }  else {  count = 0;  j = 0;  }  }  System.out.println("Second string contains the first one: " + (count == str1.length()));  input.close();  }  } |

**Output:**

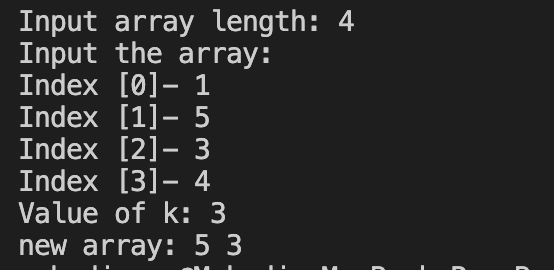
****

# **173.** Write a Java program to find the median of the numbers inside the window (size k) at each step in a given array of integers with duplicate numbers. Move the window to the array start.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n173 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, k;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the array: ");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index [%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("Value of k: ");  k = input.nextInt();  System.out.print("new array: ");  for (int i = k / 2; i < arr.length - (k / 2); i++) {  System.out.print(arr[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

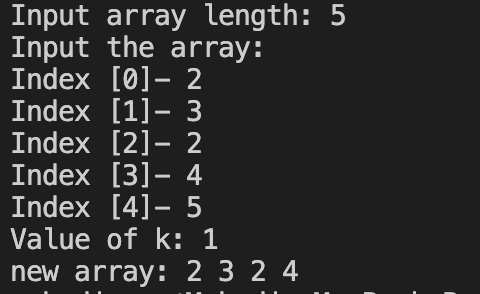
****

# **174.** Write a Java program to find the maximum number inside the number in the window (size k) at each step in a given array of integers with duplicate numbers. Move the window to the top of the array.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n174 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, k, largest;  System.out.print("Input array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the array: ");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Index [%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("Value of k: ");  k = input.nextInt();  System.out.print("new array: ");  for (int i = 1; i < (arr.length - (k / 2)); i++) {  largest = arr[i-1];  for (int j = i-1; j < i + k - 1; j++) {  if (arr[j] > largest) {  largest = arr[j];  }  }  System.out.print(largest + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

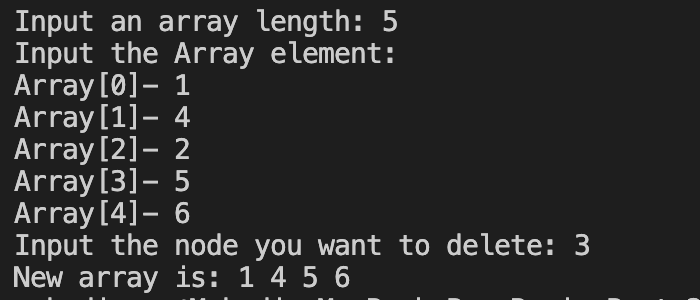
****

# **175.** Write a Java program to delete a specified node in the middle of a singly linked list.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n175 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n;  System.out.print("Input an array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the Array element:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Array[%d]- ", i);  arr[i] = input.nextInt();  }  System.out.print("Input the node you want to delete: ");  int node;  node = input.nextInt();  int[] arr2 = new int[n-1];  for (int i = 0; i < arr2.length; i++) {  if (i >= node-1) {  arr2[i] = arr[i+1];  }  else {  arr2[i] = arr[i];  }  }  System.out.print("New array is: ");  for (int i = 0; i < arr2.length; i++) {  System.out.print(arr2[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

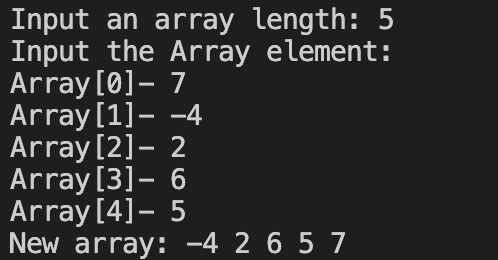
****

# **176.** Write a Java program that partitions an array of integers into even and odd numbers.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n176 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, temp;  System.out.print("Input an array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the Array element:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Array[%d]- ", i);  arr[i] = input.nextInt();  }  for (int i = 0; i < arr.length; i++) {  for (int j = 0; j < arr.length-1; j++) {  if (arr[j] % 2 != 0) {  temp = arr[j];  arr[j] = arr[j+1];  arr[j+1] = temp;  }  }  }  System.out.print("New array: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(arr[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

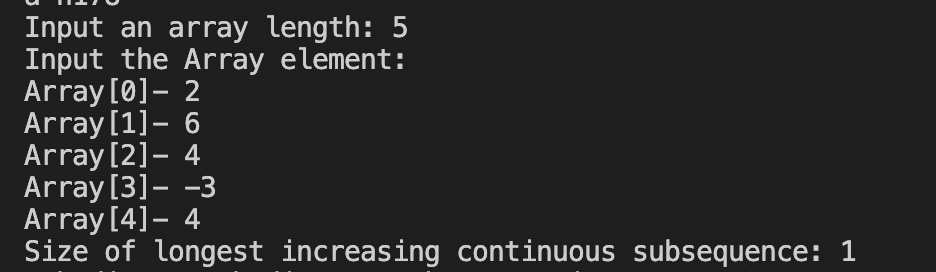
****

# **178.** Write a Java program to find the longest increasing continuous subsequence in a given array of integers.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n178 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, tempCount = 1, count = 1;  System.out.print("Input an array length: ");  n = input.nextInt();  int[] arr = new int[n];  System.out.println("Input the Array element:");  for (int i = 0; i < arr.length; i++) {  System.out.printf("Array[%d]- ", i);  arr[i] = input.nextInt();  }  for (int i = 0; i < arr.length-1; i++) {  if (arr[i] + 1 == arr[i+1]) {  tempCount++;  if (tempCount > count) {  count++;  }  }  else {  tempCount = 1;  }  }  System.out.println("Size of longest increasing continuous subsequence: " + count);  input.close();  }  } |

**Output:**

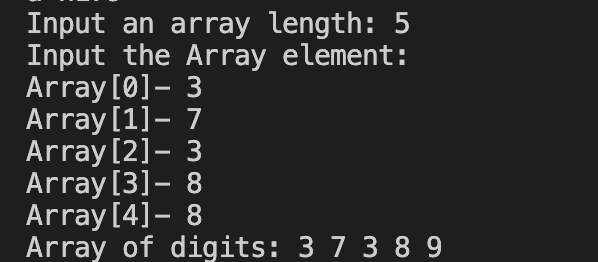
****

# **179.** Write a Java program to add one to a positive number represented as an array of digits.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n179 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, sum = 0;  System.out.print("Input an array length: ");  n = input.nextInt();  int[] arr1 = new int[n];  System.out.println("Input the Array element:");  for (int i = 0; i < arr1.length; i++) {  System.out.printf("Array[%d]- ", i);  arr1[i] = input.nextInt();  if (i == 0) {  sum = arr1[i];  }  else {  sum \*= 10;  sum += arr1[i];  }  }  // sum /= 10;  sum++;  String num = Integer.toString(sum);  // System.out.println(num);  int[] arr2 = new int[num.length()];  System.out.print("Array of digits: ");  for (int i = 0; i < arr2.length; i++) {  arr2[i] = Character.getNumericValue(num.charAt(i));  System.out.print(arr2[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

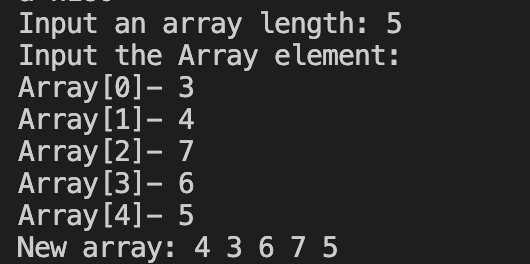
****

# **180.** Write a Java program to swap two adjacent nodes in a linked list.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n180 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, temp;  System.out.print("Input an array length: ");  n = input.nextInt();  int[] arr1 = new int[n];  System.out.println("Input the Array element:");  for (int i = 0; i < arr1.length; i++) {  System.out.printf("Array[%d]- ", i);  arr1[i] = input.nextInt();  }  for (int i = 0; i < arr1.length-1; i += 2) {  temp = arr1[i];  arr1[i] = arr1[i+1];  arr1[i+1] = temp;  }  System.out.print("New array: ");  for (int i = 0; i < arr1.length; i++) {  System.out.print(arr1[i] + " ");  }  System.out.println();  input.close();  }  } |

**Output:**

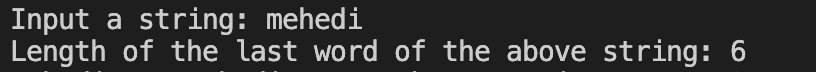
****

# **181.** Write a Java program to find the length of the last word in a given string. The string contains upper/lower-case alphabets and empty space characters like ' '.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n181 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int count = 0;  String str;  System.out.print("Input a string: ");  str = input.nextLine();  for (int i = str.length()-1; i >= 0; i--) {  if (str.charAt(i) == ' ') {  break;  }  count++;  }  System.out.println("Length of the last word of the above string: " + count);  input.close();  }  } |

**Output:**

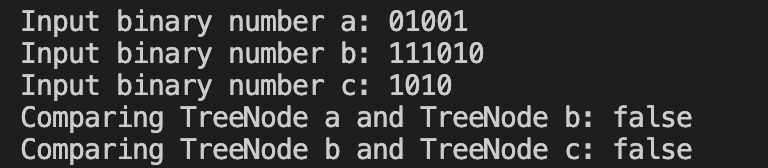
****

# **182.** Write a Java program to check if two binary trees are identical. Assume that two binary trees have the same structure and every identical position has the same value.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n182 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  String a, b, c;  System.out.print("Input binary number a: ");  a = input.nextLine();  System.out.print("Input binary number b: ");  b = input.nextLine();  System.out.print("Input binary number c: ");  c = input.nextLine();  System.out.println("Comparing TreeNode a and TreeNode b: " + (a.compareTo(b) == 0 ? "true" : "false"));  System.out.println("Comparing TreeNode b and TreeNode c: " + (b.compareTo(c) == 0 ? "true" : "false"));  input.close();  }  } |

**Output:**

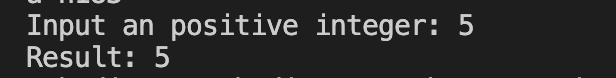
****

# **183.** Write a Java program to accept a positive number and repeatedly add all its digits until the result has only one digit.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n183 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, count = 0;  System.out.print("Input an positive integer: ");  n = input.nextInt();  for (int i = n; i > 0; ) {  count += i % 10;  i = i / 10;  }  System.out.println("Result: " + count);  input.close();  }  } |

**Output:**

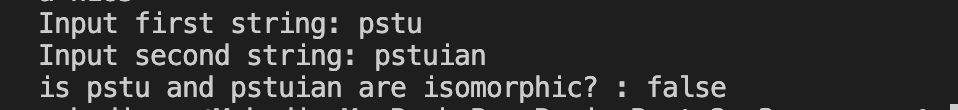
****

# **185.** Write a Java program to check if two strings are isomorphic or not.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n185 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  String a, b;  System.out.print("Input first string: ");  a = input.nextLine();  System.out.print("Input second string: ");  b = input.nextLine();  System.out.println("is " + a + " and " + b + " are isomorphic? : " + ((a.charAt(0) == a.charAt(a.length()-1)) && (b.charAt(0) == b.charAt(b.length()-1))));  input.close();  }  } |

**Output:**

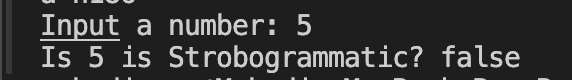
****

# **186.** Write a Java program to check if a number is a strobogrammatic number. The number is represented as a string.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n186 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int flag = 0, l;  String s;  System.out.print("Input a number: ");  s = input.nextLine();  l = s.length();  if (l > 1) {  for (int i = 0, j = l-1; i < (l/2); i++, j--) {  if ((s.charAt(i) == '1' && s.charAt(j) == '1') || ((s.charAt(i) == '6') && (s.charAt(j) == '9')) || ((s.charAt(i) == '9') && (s.charAt(j) == '6')) || ((s.charAt(i) == '8') && (s.charAt(j) == '8')) || ((s.charAt(i) == '0') && (s.charAt(j) == '0'))) {  if (l%2 != 0) {  if ((s.charAt(l/2) == '1') || (s.charAt(l/2) == '0') || (s.charAt(l/2) == '8')) {  flag = 0;  }  else {  flag = 1;  break;  }  }  else {  flag = 0;  }  }  else {  flag = 1;  break;  }  }  }  else {  if ((s.charAt(0) == '1') || (s.charAt(0) == '0') || (s.charAt(0) == '8')) {  flag = 0;  }  else {  flag = 1;  }  }  System.out.println("Is " + s + " is Strobogrammatic? " + (flag == 0));  input.close();  }  } |

**Output:**

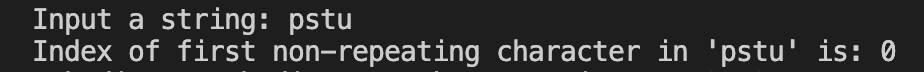
****

# **187.** Write a Java program to find the index of the first non-repeating character in a given string.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n187 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int flag = 0;  String s;  System.out.print("Input a string: ");  s = input.nextLine();  for (int i = 0; i < s.length(); i++) {  for (int j = 0; j < s.length(); j++) {  if ((i != j) && (s.charAt(i) == s.charAt(j))) {  flag = 1;  break;  }  else {  flag = 0;  }  }  if (flag == 0) {  System.out.println("Index of first non-repeating character in '" + s + "' is: " + i);  break;  }  }  input.close();  }  } |

**Output:**

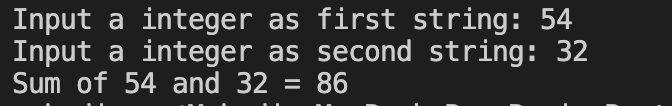
****

# **189.** Write a Java program to two non-negative integers num1 and num2 represented as strings, return the sum of num1 and num2.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n189 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int s1Sum = 0, s2Sum = 0;  String s1, s2, s3 = "0123456789";  System.out.print("Input a integer as first string: ");  s1 = input.nextLine();  System.out.print("Input a integer as second string: ");  s2 = input.nextLine();  for (int i = 0; i < s1.length(); i++) {  for (int j = 0; j < s3.length(); j++) {  if (s1.charAt(i) == s3.charAt(j)) {  s1Sum \*= 10;  s1Sum += j;  break;  }  }  }  for (int i = 0; i < s2.length(); i++) {  for (int j = 0; j < s3.length(); j++) {  if (s2.charAt(i) == s3.charAt(j)) {  s2Sum \*= 10;  s2Sum += j;  break;  }  }  }  System.out.printf("Sum of %s and %s = %d\n", s1, s2, (s1Sum+s2Sum));  // System.err.println(s1Sum);  // System.err.println(s2Sum);  input.close();  }  } |

**Output:**

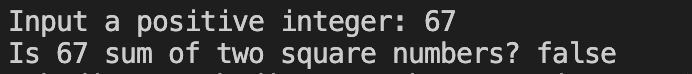
****

# **191.** Write a Java program to test whether there are two integers x and y such that x^2 + y^2 is equal to a given positive number.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n191 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, flag = 0;  System.out.print("Input a positive integer: ");  n = input.nextInt();  for (int i = 1; i\*i <= n; i++) {  for (int j = 1; ((i\*i) + (j\*j)) <= n; j++) {  if (((i\*i) + (j\*j)) == n) {  flag = 1;  break;  }  }  if (flag == 1) {  break;  }  }  System.out.println("Is " + n + " sum of two square numbers? " + (flag == 1));  input.close();  }  } |

**Output:**

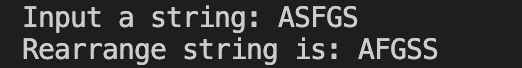
****

# **192.** Write a Java program to rearrange the alphabets in the order followed by the sum of digits in a given string containing uppercase alphabets and integer digits (from 0 to 9).

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n192 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  String s, rs = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ";  System.out.print("Input a string: ");  s = input.nextLine();  // String[] s2 = new String[s.length()];  int[] arr = new int[s.length()];  for (int i = 0; i < s.length(); i++) {  for (int j = 0; j < rs.length(); j++) {  if (s.charAt(i) == rs.charAt(j)) {  arr[i] = j;  }  }  }  // char temp1;  int temp;  for (int i = 0; i < arr.length; i++) {  for (int j = 0; j < arr.length-1; j++) {  if (arr[j] > arr[j+1]) {  // temp1 = s.charAt(j);  temp = arr[j];  arr[j] = arr[j+1];  arr[j+1] = temp;  }  }  }  System.out.print("Rearrange string is: ");  for (int i = 0; i < arr.length; i++) {  System.out.print(rs.charAt(arr[i]));  }  System.out.println();  // System.out.println(arr[2]);  input.close();  }  } |

**Output:**

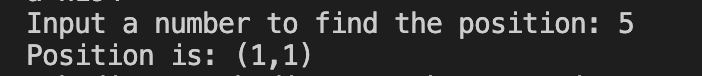
****

# **194.** Write a Java program to determine the all positions of a given number in a given matrix. If the number is not found print ("Number not found!").

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n194 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int flag = 0;  int[][] matrix = new int[3][3];  matrix[0][0] = 1;  matrix[0][1] = 2;  matrix[0][2] = 3;  matrix[1][0] = 4;  matrix[1][1] = 5;  matrix[1][2] = 6;  matrix[2][0] = 7;  matrix[2][1] = 8;  matrix[2][2] = 9;  System.out.print("Input a number to find the position: ");  int position = input.nextInt();  for (int i = 0; i < matrix.length; i++) {  for (int j = 0; j < matrix.length; j++) {  if (matrix[i][j] == position) {  System.out.printf("Position is: (%d,%d)\n", i, j);  flag = 0;  break;  }  else {  flag = 1;  }  }  if (flag == 0) {  break;  }  }  if (flag == 1) {  System.out.println("Number not found!");  }  input.close();  }  } |

**Output:**

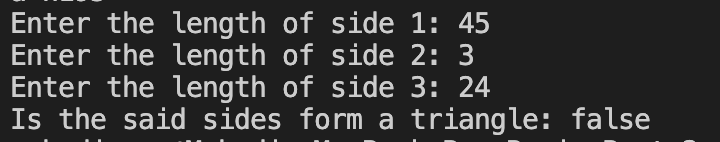
****

# **195.** Write a Java program to check if three given side lengths (integers) can make a triangle or not.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n195 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int side1, side2, side3;  System.out.print("Enter the length of side 1: ");  side1 = input.nextInt();  System.out.print("Enter the length of side 2: ");  side2 = input.nextInt();  System.out.print("Enter the length of side 3: ");  side3 = input.nextInt();  System.out.println("Is the said sides form a triangle: " + ((side1 + side2 > side3) && (side1 + side3 > side2) && (side2 + side3 > side1)));  input.close();  }  } |

**Output:**

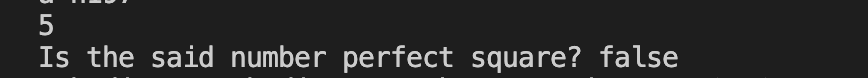
****

# **197.** Write a Java program to test if a given number (positive integer) is a perfect square or not.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n197 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int n, flag = 0;  n = input.nextInt();  for (int j = 0; j <= n; j++) {  if (j\*j == n) {  System.out.println("Is the said number perfect square? " + (j\*j == n));  flag = 1;  break;  }  }  if (flag == 0) {  System.out.println("Is the said number perfect square? " + (flag != 0));  }  input.close();  }  } |

**Output:**

****

# **198.** Write a Java program to calculate the position of a given prime number.

**Code:**

|  |
| --- |
| import java.util.Scanner;  public class n198 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int primeNumber, flag = 0, count = 0;  System.out.print("Input a positive integer: ");  primeNumber = input.nextInt();  for (int i = 2; i <= primeNumber; i++) {  flag = 0;  for (int j = 2; j <= i; j++) {  if ((i % j == 0) && (i != j)) {  flag = 1;  break;  }  else {  flag = 0;  // break;  }  }  if (flag == 0) {  count++;  // System.out.println(i + " ");  }  if ((flag == 0) && (i == primeNumber)) {  // System.out.println(count);  break;  }  }  System.out.println("Position of the said Prime number: " + count);  input.close();  }  } |

**Output:**

****