**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DATA STRUCTURE LAB**

**Assignment 1**

**Question 1**

**Develop a menu-driven program to demonstrate the following operations of Arrays ——MENU——-**

**1.CREATE**

**2.DISPLAY**

**3.INSERT**

**4.DELETE**

**5.SEARCH**

**6.EXIT**

**SOLUTION CODE**

**import java.util.\*;**

**import java.io.\*;**

**class Main**

**{**

**static Scanner scr = new Scanner(System.in);**

**static int a[];**

**static int m = 10;**

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

**int ch=0;**

**do {**

**ch = menu();**

**switch (ch) {**

**case 1:**

**input();**

**break;**

**case 2:**

**show();**

**break;**

**case 3:**

**insert();**

**break;**

**case 4:**

**delete();**

**break;**

**case 5:**

**search();**

**break;**

**case 6:**

**System.out.println("We are done ");**

**break;**

**default:**

**System.out.println("Not an option \n");**

**}**

**} while (ch != 6);**

**}**

**static int menu() {**

**System.out.println( "——MENU——- \n1.CREATE \n2.DISPLAY\n3.INSERT \n4.DELETE\n5.SEARCH \n6.EXIT\nEnter your choice");**

**int ch = scr.nextInt();**

**return ch;**

**}**

**static void input() {**

**System.out.println("Enter number of elements ");**

**int n = scr.nextInt();**

**a = new int[n + 10];**

**for (int i = 0; i < a.length-m; i++) {**

**System.out.println("Enter value of " + i + "th element ");**

**a[i] = scr.nextInt();**

**}**

**}**

**static void show() {**

**System.out.println("Elements are ");**

**int j = 0;**

**while (j < a.length - m) {**

**System.out.print(a[j] + " ");**

**j = j + 1;**

**}**

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

**}**

**static void insert() {**

**System.out.println("Enter value ");**

**int val = scr.nextInt();**

**System.out.println("Enter position at which we have to insert ");**

**int pos = scr.nextInt();**

**if (pos >= a.length - m)**

**System.out.println("Out of Bounds index ");**

**else {**

**m--;**

**int j = a.length - m;**

**while (j > pos) {**

**a[j] = a[j - 1];**

**j = j - 1;**

**}**

**a[pos] = val;**

**}**

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

**}**

**static void delete() {**

**System.out.println("Enter position at which we have to delete ");**

**final int pos = scr.nextInt();**

**int j = pos;**

**if (pos >= a.length - m)**

**System.out.println("Out of Bounds index ");**

**else {**

**m++;**

**while (j < a.length - m) {**

**a[j] = a[j + 1];**

**j = j + 1;**

**}**

**}**

**}**

**static void search()**

**{**

**System.out.println("Enter element to be searched ");**

**final int pos = scr.nextInt();**

**int j=0;**

**int result=-1;**

**while(j<(a.length-m))**

**{**

**if(pos==a[j])**

**{**

**result =j;**

**break;**

**}**

**j=j+1;**

**}**

**if (result != -1)**

**{**

**System.out.println("Element is present at index "+result);**

**}**

**else**

**{**

**System.out.println("Element is not present in array") ;**

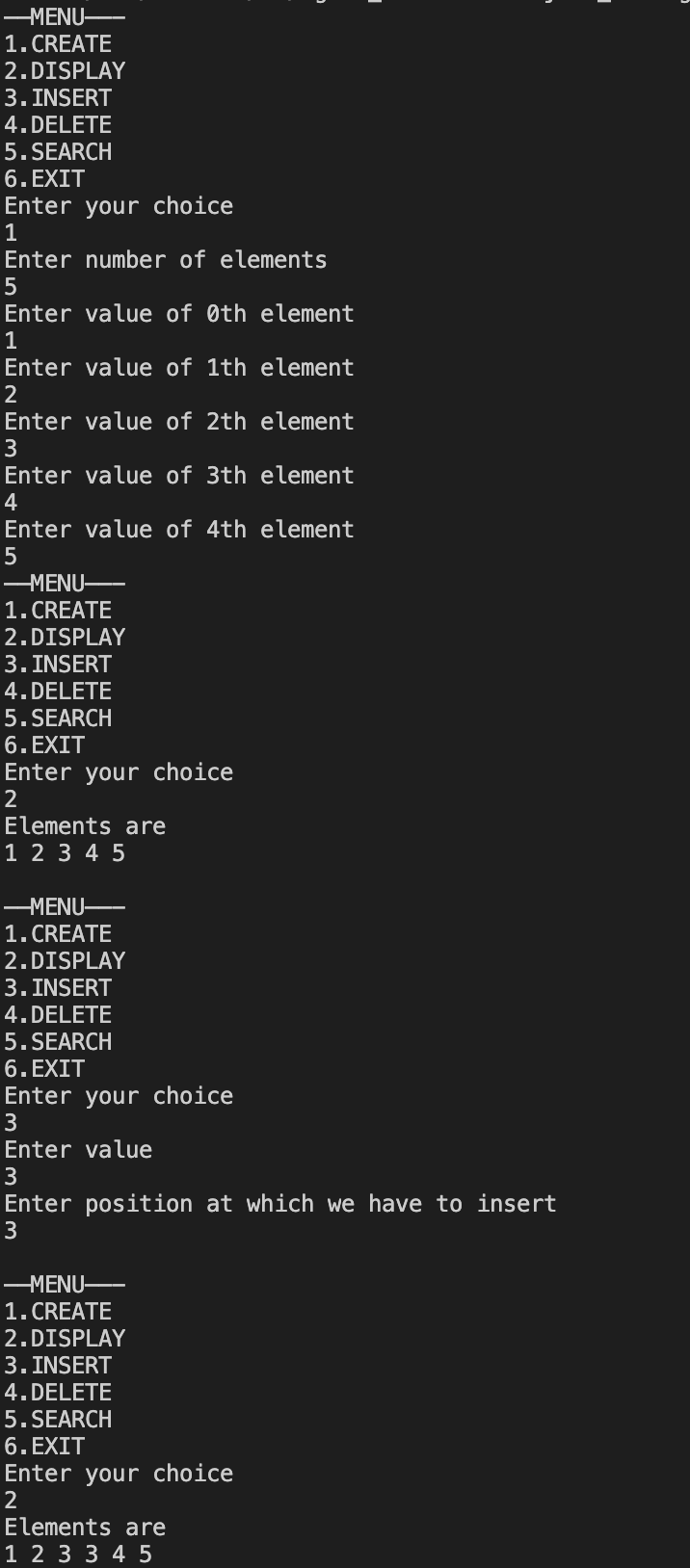
**}**

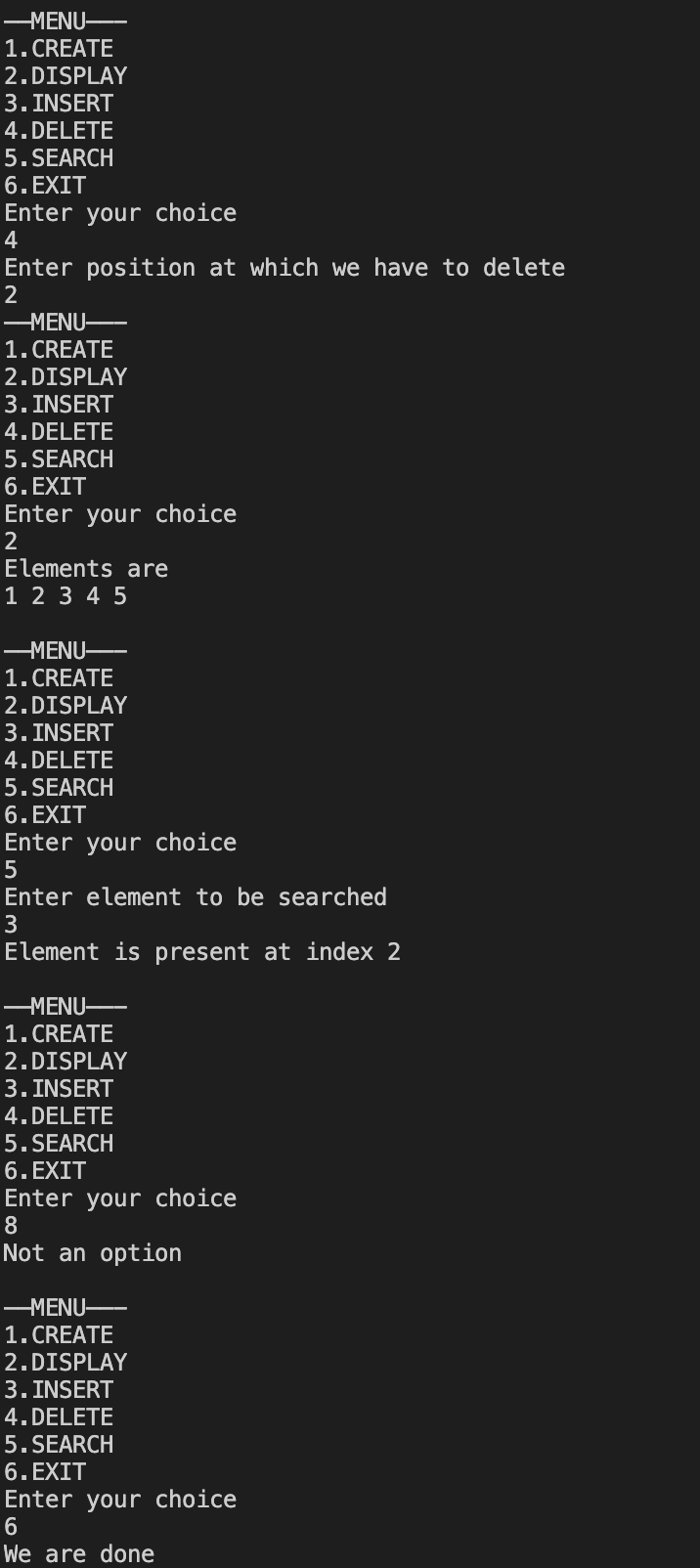
**System.out.println();**

**}**

**}**

**OUTPUT**

****



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 2**

**Design the logic to remove the duplicate elements from an Array and after the deletion, the array should contain the unique elements.**

**SOLUTION CODE**

**import java.util.\*;**

**import java.io.\*;**

**class question2**

**{**

**private static int duplicate(int m[])**

**{**

**int c=0;**

**for (int i = 0;i<m.length-c;i++)**

**{**

**for (int j=i+1;j<m.length-c;j++)**

**{**

**if (m[j]==m[i])**

**{**

**c++;**

**delete(m,j);**

**}**

**}**

**}**

**return c;**

**}**

**private static void delete(int m[],int j)**

**{**

**for (int i=j;i<m.length-1;i++)**

**{**

**m[i]=m[i+1];**

**}**

**}**

**static Scanner scr = new Scanner(System.in);**

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

**System.out.println("Enter number of elements ");**

**int n = scr.nextInt();**

**int a[] = new int[n];**

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

**System.out.println("Enter value of " + i+" th element ");**

**a[i]= scr.nextInt();**

**}**

**int m = duplicate(a);**

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

**for (int i =0;i<a.length-m;i++)**

**{**

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

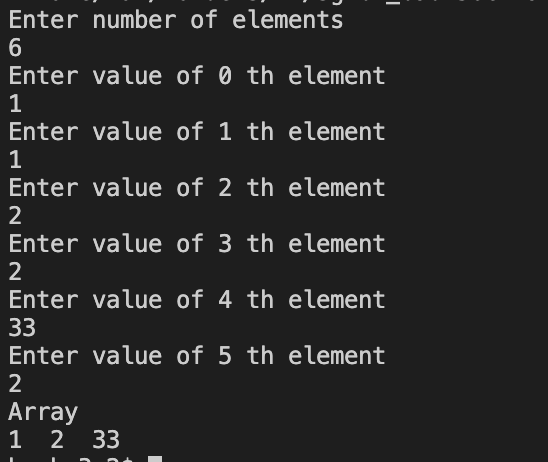
**}**

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

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 3**

**Predict the output of the following program**

**int main()**

**{**

**int i;**

**int arr[5] = {1};**

**for (i = 0; i < 5; i++)**

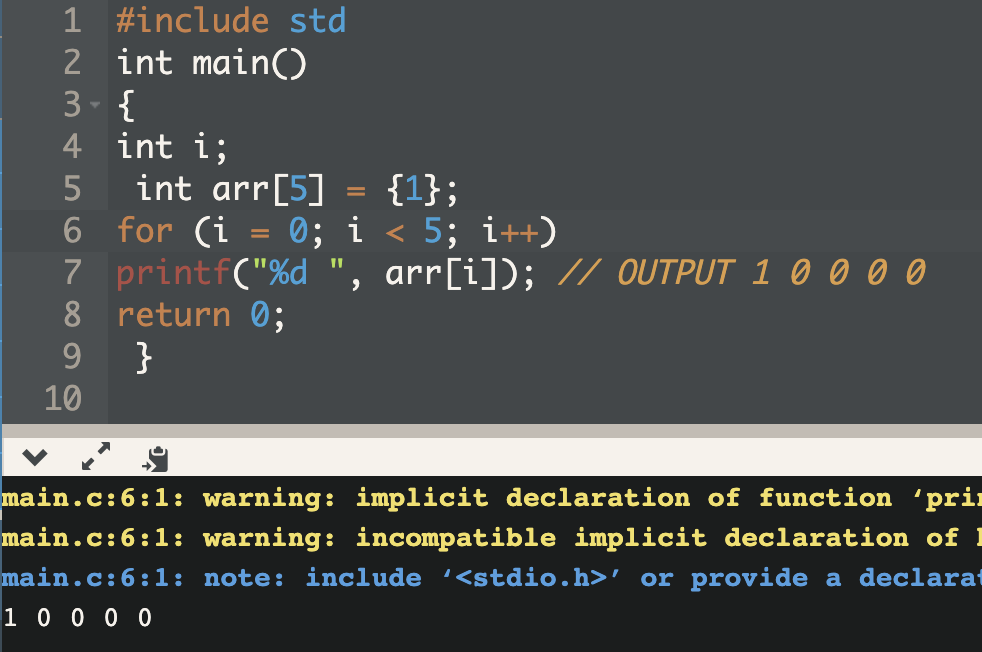
**printf("%d ", arr[i]);**

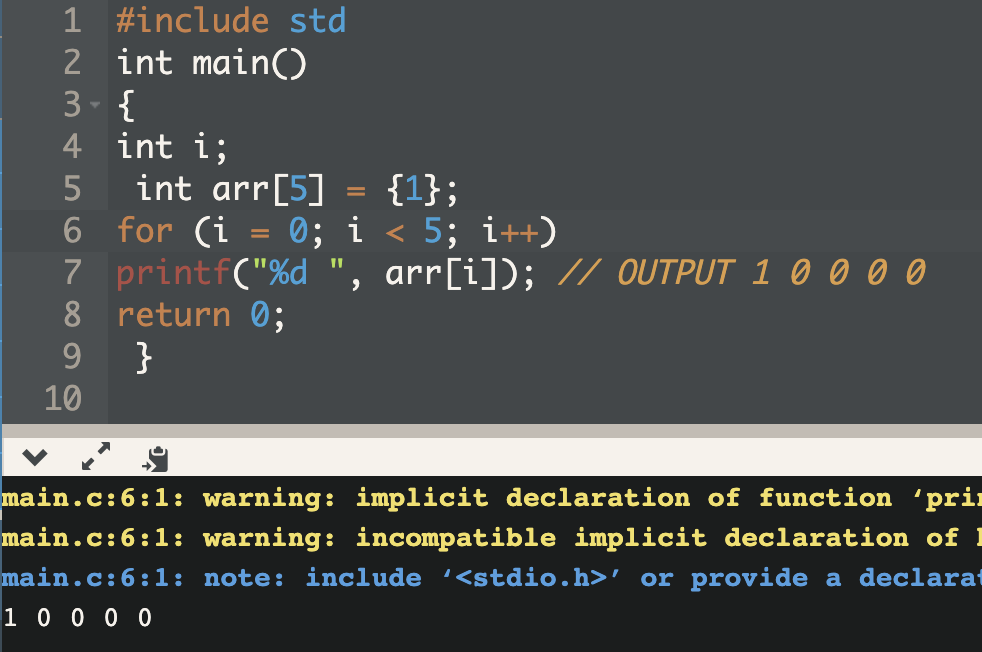
**return 0;**

**}**

**ANSWER**

**CODE**

****

***OUTPUT***

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 4 i)**

**Reverse the elements of an array**

**SOLUTION CODE**

**/\***

**#Implement the logic to**

**#Reverse the elements of an array\*/**

**import java.util.Scanner;**

**class Main1**

**{**

**static Scanner scr = new Scanner(System.in);**

**static int a[];**

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

**System.out.println("Enter number of elements ");**

**final int n = scr.nextInt();**

**a = new int[n];**

**int i;**

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

**System.out.println("Enter value of " + i + "th element ");**

**a[i] = scr.nextInt();**

**}**

**final int m = (int) (n / 2);**

**for (i = 0; i < m; i++) {**

**final int d = a[i];**

**a[i]= a[n-1-i];**

**a[n-i-1] = d;**

**}**

**for (i=0;i<a.length;i++)**

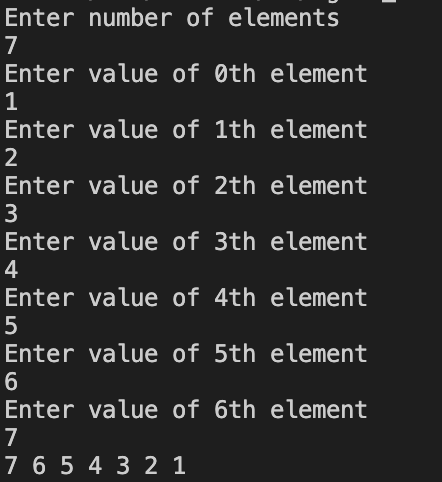
**System.out.print(a[i]+" ");**

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

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 4 ii)**

**Find the matrix multiplication.**

**SOLUTION CODE**

**/\***

**\* # ASSIGNMENT 4 2D Matrix Multiplication**

**#Implement the logic to Find the matrix multiplication**

**# A basic code for matrix input from the user**

**\*/**

**import java.util.Scanner;**

**class question4**

**{**

**static Scanner scr = new Scanner(System.in);**

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

**System.out.println("Enter the number of rows for both the matrix: ") ;**

**int R = scr.nextInt();**

**System.out.println("Enter the number of columns for both the matrix: ") ;**

**int C = scr.nextInt();**

**// Initialize matrix**

**int multiplied[][] = new int [R][C];**

**int matrixA[][] = new int [R][C];**

**int matrixB[][] = new int [R][C];**

**System.out.println("Enter the entries rowwise: for first matrix") ;**

**int i,j;**

**//For user input**

**for (i=0;i<R;i++)**

**{**

**for (j=0;j<C;j++)**

**{**

**matrixA[i][j]=scr.nextInt();**

**}**

**}**

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

**System.out.println("Enter the entries rowwise: for second matrix") ;**

**for (i=0;i<R;i++)**

**{**

**for (j=0;j<C;j++)**

**{**

**matrixB[i][j]=scr.nextInt();**

**}**

**}**

**for (i=0;i<matrixA.length;i++)**

**{**

**for (j=0;j<matrixB[0].length;j++)**

**{**

**for (int k =0;k<matrixB.length;k++)**

**multiplied[i][j] += (matrixA[i][k] \* matrixB[k][j]);**

**}**

**}**

**System.out.println("\nThe final answer is ") ;**

**for (i=0;i<R;i++)**

**{**

**for (j=0;j<C;j++)**

**{**

**System.out.print(multiplied[i][j]+" ");**

**}**

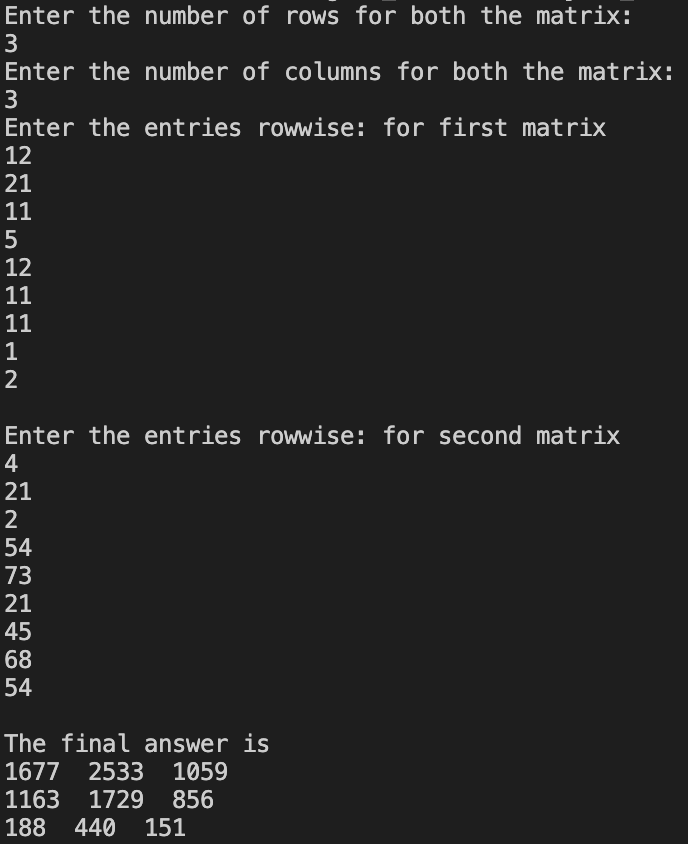
**System.out.println("");**

**}**

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 5**

**Implement the Binary search algorithm regarded as a fast search algorithm with run-time complexity of Ï(log n) in comparison to the Linear Search.**

**SOLUTION CODE**

**// Java implementation of iterative Binary Search**

**import java.util.\*;**

**class question6binary {**

**// Returns index of x if it is present in arr[],**

**// else return -1**

**static Scanner scr = new Scanner(System.in);**

**static int binarySearch(int arr[], int x)**

**{**

**int l = 0, r = arr.length - 1;**

**while (l <= r) {**

**int m = l + (r - l) / 2;**

**if (arr[m] == x)**

**return m;**

**if (arr[m] < x)**

**l = m + 1;**

**else**

**r = m - 1;**

**}**

**return -1;**

**}**

**// Driver method to test above**

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

**{**

**System.out.println("Enter number of elements ");**

**int n = scr.nextInt();**

**int arr[] = new int[n];**

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

**System.out.println("Enter value of " + i + "th element ");**

**arr[i] = scr.nextInt();**

**}**

**System.out.println("Enter search element ");**

**int x = scr.nextInt();**

**int result = binarySearch(arr, x);**

**if (result == -1)**

**System.out.println("Element not present");**

**else**

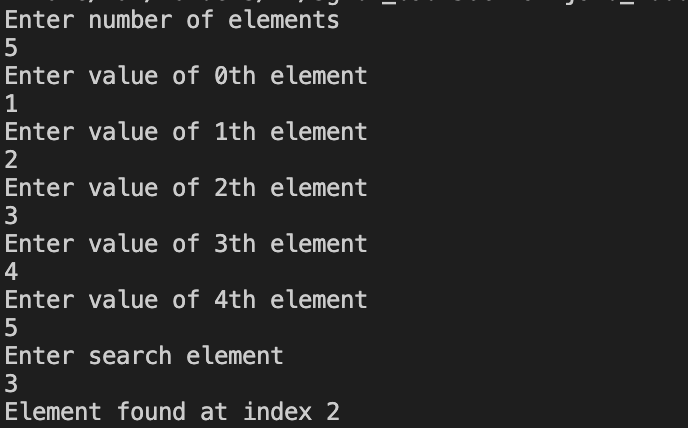
**System.out.println("Element found at "**

**+ "index " + result);**

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 6**

**Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. Code the Bubble sort with the following elements:**

**64 34 25 12 22 11 90**

**SOLUTION CODE**

**/\*Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements: 64 34 25 12 22 11 90\*/**

**class question6**

**{**

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

**int arr[] = {64, 34, 25, 12, 22, 11, 90};**

**bubbleSort(arr);**

**System.out.println("The sorted array is :") ;**

**int i,j;**

**//For user input**

**for (i=0;i<arr.length;i++)**

**{**

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

**}**

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

**}**

**static void bubbleSort(int arr[])**

**{**

**int n = arr.length;**

**for (int i = 0; i < n-1; i++)**

**for (int j = 0; j < n-i-1; j++)**

**if (arr[j] > arr[j+1])**

**{**

**// swap arr[j+1] and arr[i]**

**int temp = arr[j];**

**arr[j] = arr[j+1];**

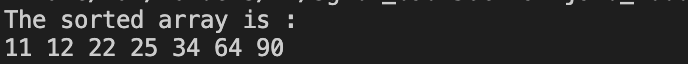
**arr[j+1] = temp;**

**}**

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 7**

**Design the Logic to Find a Missing Number in a Sorted Array from 1 to n**

**CODE**

**// #DesigntheLogictoFindaMissingNumberinaSortedArray.**

**import java.util.\*;**

**import java.io.\*;**

**class question7**

**{**

**static Scanner scr = new Scanner(System.in);**

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

**System.out.println("Enter number of elements ");**

**int n = scr.nextInt();**

**int arr[] = new int [n];**

**int miss[] = new int [n-1];**

**int i;**

**for (i=0;i<arr.length;i++)**

**arr[i]= i+1;**

**int sum= n\*(n+1)/2;**

**int usersum=0;**

**System.out.println("Enter the rest of elements ");**

**for (i=0;i<miss.length;i++)**

**{**

**System.out.println("Enter value of "+i+"th element ");**

**miss[i]=scr.nextInt();**

**usersum+=miss[i];**

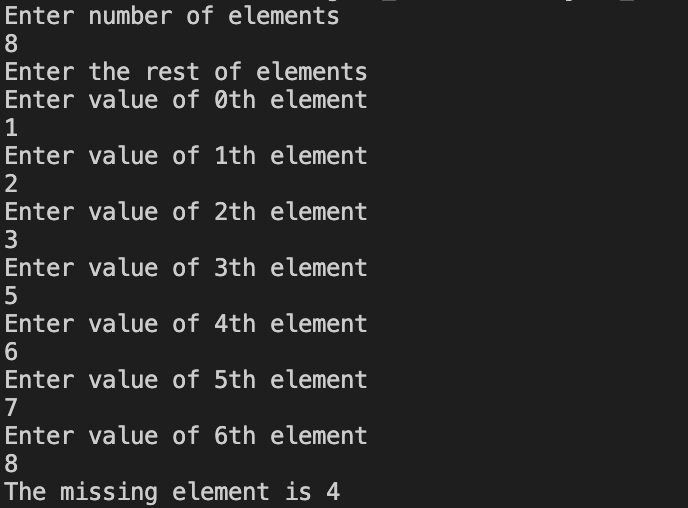
**}**

**System.out.println("The missing element is "+(sum-usersum));**

**}**

**}**

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**