1. Declare a single-dimensional array of 5 integers inside the main method. Traverse the array to print the default values. Then accept records from the user and print the updated values of the array.

import java.util.Scanner;

public class Q1 {

    public static int[] acceptRecord(int arr[]){

        Scanner sc = new Scanner(System.in);

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

            System.out.print("Enter "+i +" Element: ");

            arr[i] = sc.nextInt();

        }

        return arr;

    }

     static  void print(int arr[]){

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

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

        }

    }

    public static void main(String[] args) {

        int arr[] = new int[5];

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

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

        }

        System.out.println();

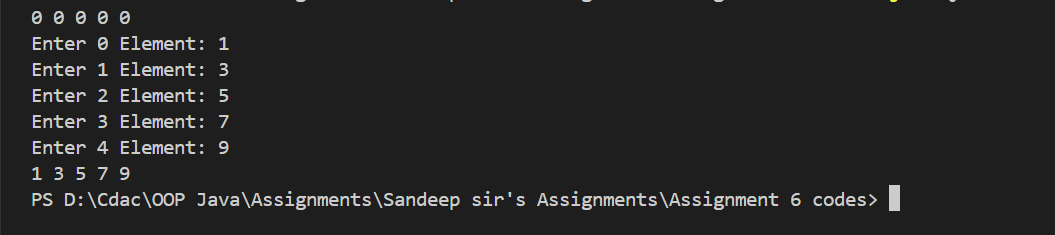
         acceptRecord(arr);

         print(arr);

    }

}

Output:



1. Declare a single-dimensional array of 5 integers inside the main method. Define a method named acceptRecord to get input from the terminal into the array and another method named printRecord to print the state of the array to the terminal.

import java.util.Scanner;

public class Q2 {

    public static int[] acceptRecord(int arr[]){

        Scanner sc = new Scanner(System.in);

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

            System.out.print("Enter "+i +" Element: ");

            arr[i] = sc.nextInt();

        }

        return arr;

    }

    static  void print(int arr[]){

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

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

        }

    }

    public static void main(String[] args) {

        int arr[] = new int[5];

        System.out.println();

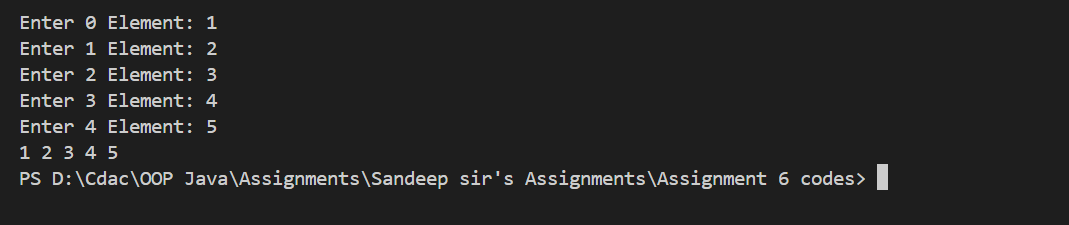
        acceptRecord(arr);

        print(arr);

    }

}

Output:



1. Write a program to find the maximum and minimum values in a single-dimensional array of integers.

public class Q3 {

    public static void main(String[] args) {

        int arr[] = new int[]{88,42,56,79,12,11};

        int max = Integer.MIN\_VALUE;

        int min = Integer.MAX\_VALUE;

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

            if (arr[i] > max){

                max = arr[i];

            } else if (arr[i] < min) {

                min = arr[i];

            }

        }

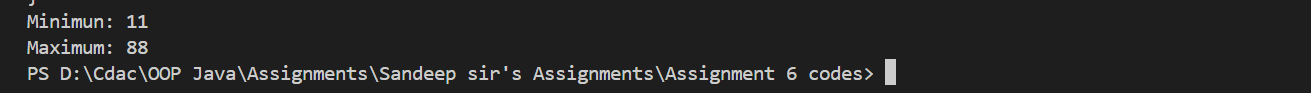
        System.out.println("Minimun: "+ min);

        System.out.println("Maximum: "+ max);

    }

}

Output:



1. Write a program to remove duplicate elements from a single-dimensional array of integers.

import java.util.Arrays;

public class Q4 {

    public static void main(String[] args) {

        int arr[] = new int[]{6,6,2,2,1,3,3,3,4,5};

        Arrays.sort(arr);

        int i = 0;

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

            if (arr[i] != arr[j]){

                i++;

                arr[i] = arr[j];

            }

        }

        for (int j = i+1; j < arr.length; j++) {

            arr[j] = 0;

        }

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

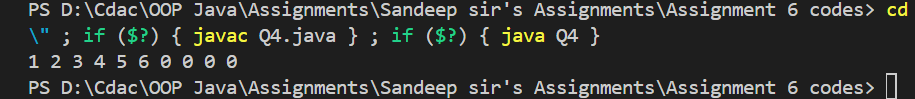
            System.out.print(arr[j] +" ");

        }

    }

}

Output:



1. Write a program to find the intersection of two single-dimensional arrays.

public class Q5 {

    public static void main(String[] args) {

        int arr1[] = {1,3,2,5,4,8};

        int arr2[] = {4,5,6,7,8};

        for (int i = 0; i < arr1.length; i++) {

            for (int j = 0; j < arr2.length; j++) {

                if (arr1[i] == arr2[j]){

                    System.out.print(arr2[j]+" ");

                }

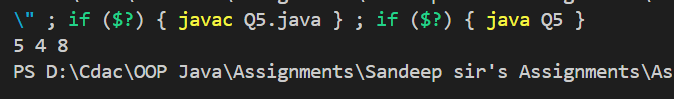
            }

        }

    }

}

Output:



1. Write a program to find the missing number in an array of integers ranging from 1 to N.

public class Q6 {

    public static void main(String[] args) {

        int arr[] = {2,1,5,3,4,7,8};

        int n = arr.length+1;

        int sum\_n = n\*(n+1)/2;

        int curr\_sum = 0;

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

            curr\_sum += arr[i];

        }

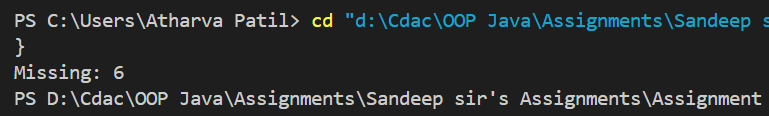
        int Missing = sum\_n - curr\_sum;

        System.out.println("Missing: "+Missing);

    }

}

Output:



1. Declare a single-dimensional array as a field inside a class and instantiate it inside the class constructor. Define methods named acceptRecord and printRecord within the class and test their functionality.

import java.util.Scanner;

class Test{

    int arr[];

    Test(int size){

        arr = new int[size];

    }

    void acceptRecord(){

        Scanner sc = new Scanner(System.in);

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

            System.out.print("Enter "+i+" Element:");

            arr[i] = sc.nextInt();

        }

    }

    void printRecord(){

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

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

        }

    }

}

public class Q7 {

    public static void main(String[] args) {

        Test t1 = new Test(5);

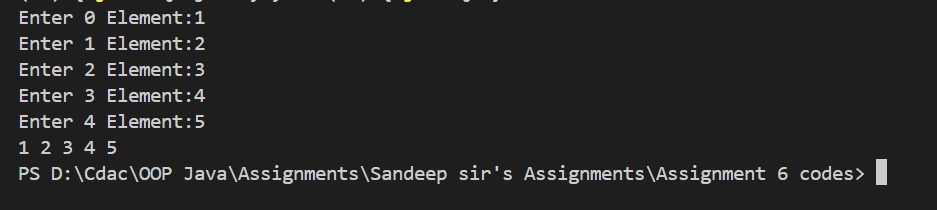
        t1.acceptRecord();

        t1.printRecord();

    }

}

Output:



1. Modify the previous assignment to use getter and setter methods instead of acceptRecord and printRecord.

import java.util.Scanner;

class Test2{

    int arr[];

    Test2(int size){

        arr = new int[size];

    }

    void setArr(){

        Scanner sc = new Scanner(System.in);

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

            System.out.print("Enter "+i+" Element:");

            arr[i] = sc.nextInt();

        }

    }

    void getArr(){

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

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

        }

    }

}

public class Q8 {

    public static void main(String[] args) {

        Test2 t2 = new Test2(6);

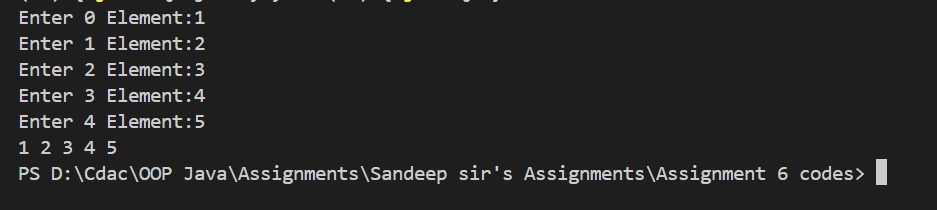
        t2.setArr();

        t2.getArr();

    }

}

Output:



1. You need to implement a system to manage airplane seat assignments. The airplane has seats arranged in rows and columns. Implement functionalities to:

* Initialize the seating arrangement with a given number of rows and columns.
* Book a seat to mark it as occupied.
* Cancel a booking to mark a seat as available.
* Check seat availability to determine if a specific seat is available.
* Display the current seating chart.
* import java.util.Arrays;
* import java.util.Scanner;
* class seat{
* public static void availability(int seatno,String seats[][] ){
* int c = 1;
* for (int i = 0; i < seats.length; i++) {
* for (int j = 0; j < seats[0].length; j++) {
* if (c == seatno){
* if (seats[i][j].equals( "S:"+c +" NA")){
* System.out.println("Seat not Available");
* }else{
* System.out.println("Seat Available");
* }
* }
* c++;
* }
* }
* }
* public static void cancelBooking(int seatno,String seats[][] ){
* int c = 1;
* for (int i = 0; i < seats.length; i++) {
* for (int j = 0; j < seats[0].length; j++) {
* if (c == seatno){
* if (seats[i][j].equals( "S:"+c +" NA")){
* seats[i][j] = "S:"+c +" A";
* System.out.println("Booking cancled");
* }else{
* System.out.println("Not a booked seat");
* }
* }
* c++;
* }
* }
* }
* public static String[][] bookSeat(int seatNo,String seats[][]){
* int c = 1;
* for (int i = 0; i < seats.length; i++) {
* for (int j = 0; j < seats[0].length; j++) {
* if (c == seatNo){
* if (seats[i][j].equals( "S:"+c +" NA")){
* System.out.println("Seat not Available");
* }else{
* System.out.println("Seat Booked");
* }
* seats[i][j] = "S:"+c +" NA";
* }
* c++;
* }
* }
* return seats;
* }
* }
* public class Q9 {
* public static void print(String seats[][]){
* for (int i = 0; i < seats.length; i++) {
* for (int j = 0; j < seats[0].length; j++) {
* System.out.print(seats[i][j]+"   ");
* }
* System.out.println();
* }
* }
* public  static  void initialize(String seatchart[][]){
* int seatno = 1;
* for (int i = 0; i < seatchart.length; i++) {
* for (int j = 0; j < seatchart[0].length; j++) {
* if (seatno < 10){
* seatchart[i][j] = " S:"+seatno++ +" A";
* }else{
* seatchart[i][j] = "S:"+seatno++ +" A";
* }
* }
* }
* }
* public static void main(String[] args) {
* int row = 10;
* int col = 3;
* String seatchart[][] = new String[row][col];
* initialize(seatchart);
* Scanner sc = new Scanner(System.in);
* int choice;
* while (true) {
* System.out.println("0.Exit");
* System.out.println("1.Book seat");
* System.out.println("2.Cancel booking");
* System.out.println("3.Seat Availability");
* System.out.println("4.Show chart");
* System.out.println("Enter your choice:");
* choice = sc.nextInt();
* if (choice == 0) break;
* switch (choice) {
* case 1:
* System.out.println("Enter Seat No:");
* int seatNo = sc.nextInt();
* seat.bookSeat(seatNo, seatchart);
* break;
* case 2:
* System.out.println("Enter Seat No:");
* int cancelseatNo = sc.nextInt();
* seat.cancelBooking(cancelseatNo, seatchart);
* break;
* case 3:
* System.out.println("Enter Seat No:");
* int AvailableseatNo = sc.nextInt();
* seat.availability(AvailableseatNo, seatchart);
* break;
* case 4:
* print(seatchart);
* break;
* default:
* System.out.println("Invalid choice, please try again.");
* }
* }
* }
* }

Output:

