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 ArrayExample{
    public static void main(String[] args){
        int[] arr=new int[5];

        System.out.println("Default Values:");
        for(int i=0;i<=arr.length;i++){
            System.out.println(arr[i]);
        }

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter 5 Integers");
        for(int i=0;i<=arr.length;i++){
            arr[i]=sc.nextInt();
        }
        System.out.println("Updated Values");
                for(int i=0;i<=arr.length;i++){
                System.out.println(arr[i]);
                }



    }
}
```

24. 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 ArrayWithMethods{
   public static void main (String[] args){
   int[] arr=new int[5];

   }

public static void printRecord(int[] arr){
Scanner sc=new Scanner(System.in);
System.out.println("Enter 5 Integers");
for(int i=0; i<arr.length;i++){
   arr[i]=scanner.nextInt();
}
}

public static void printRecord(int[] arr){
    System.out.println("Array Values:");
    for(int val: arr){
        System.out.println(val);
    }
}
}
```

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

```
public class MaxMinArray{
    public static void main(String[] args){
    int[] arr={3,5,6,3,8};

    int max=arr[0];
    int min=arr[0];

    for(int i=1; i<arr.length;i++){
    if(arr[i]>max) max=arr[i];
    if(arr[i]<min) min=arr[i];
    }

    System.out.println("Maximum: " +max);
    System.out.println("Minimun: " +min);
}
}
```

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

```
import java.util.Arrays;
public class RemoveDuplicates{
    public static void main(String[] args){
        int[] arr={3,5,7,3,6,3,7};
        Arrays.sort(arr);

        int[] result=new int[arr.length];

        int j=0;

        for(int i=0; i<arr.length -1;i++){
            if (arr[i] !=arr[i+1]){
                result[j++]=arr[i];
            }

        }
        result[j++]=arr[arr.length-1];
        System.out.println("Array without Dublicates: ");
        for(int i=0;i<j;i++){
            System.out.print(result[i]+ " ");
        }

    }
}
```

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

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

```
public class MissingNumber{

    public static void main(String[] args){

        int[] arr={3,6,5,7,9};


        int n=arr.length+1;

        int expectedSum=n*(n+1)/2;

        int realSum=0;

        for (int num:arr){

            realSum += num;


        }

        int missingNumber= expectedSum -realSum;


        System.out.println("Missing Number:" +missingNumber);
    }

}
```

29. 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;

public class ArrayInClass {
    private int[] arr;

    public ArrayInClass() {
        arr = new int[5];  // Initialize array in constructor
    }

    public void acceptRecord() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter 5 integers:");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = scanner.nextInt();
        }
    }

    public void printRecord() {
        System.out.println("Array values:");
        for (int val : arr) {
            System.out.println(val);
        }
    }

    public static void main(String[] args) {
        ArrayInClass obj = new ArrayInClass();
        obj.acceptRecord();
        obj.printRecord();
    }
}
```

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

```
import java.util.Scanner;



public class ArrayWithGetSet {

    private int[] arr;
```

```
public ArrayWithGetSet() {
    arr = new int[5];
}

public void setRecord(int[] arr) {
    this.arr = arr;
}

public int[] getRecord() {
    return this.arr;
}

public static void main(String[] args) {
    ArrayWithGetSet obj = new ArrayWithGetSet();
    int[] newArr = new int[5];

    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter 5 integers:");
    for (int i = 0; i < newArr.length; i++) {
        newArr[i] = scanner.nextInt();
    }

    obj.setRecord(newArr);
```

```
        System.out.println("Array values:");
        int[] arrFromObj = obj.getRecord();
        for (int val : arrFromObj) {
            System.out.println(val);
        }
    }
}
```

31. 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.

```
public class AirplaneSeats {
   private boolean[][] seats;

   public AirplaneSeats(int rows, int cols) {
      seats = new boolean[rows][cols];  // Initialize all seats as available
   }

   public void bookSeat(int row, int col) {
      if (seats[row][col]) {
         System.out.println("Seat is already booked.");
      } else {
         seats[row][col] = true;
         System.out.println("Seat booked successfully.");
      }
   }

   public void cancelBooking(int row, int col) {
      if (seats[row][col]) {
         seats[row][col] = false;
         System.out.println("Booking cancelled.");
      } else {
         System.out.println("Seat is not booked.");
      }
   }
```

```
    public void checkSeat(int row, int col) {
        if (seats[row][col]) {
            System.out.println("Seat is occupied.");
        } else {
            System.out.println("Seat is available.");
        }
    }

    public void displaySeats() {
        System.out.println("Seating arrangement:");
        for (int i = 0; i < seats.length; i++) {
            for (int j = 0; j < seats[i].length; j++) {
                System.out.print(seats[i][j] ? "X " : "O ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        AirplaneSeats airplane = new AirplaneSeats(3, 3);

        airplane.bookSeat(1, 1);
        airplane.cancelBooking(1, 1);
        airplane.checkSeat(1, 1);
        airplane.displaySeats();
    }
}
```