ASSIGNMENT NO.7

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.

   CODE:

```
package org.assign6;

import java.util.Scanner;

public class Problem1 {

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

            System.out.println("Default values of the array:");
      for (int value : arr) {
         System.out.println(value);
      }

      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 of the array:");
      for (int value : arr) {
         System.out.println(value);
      }
        sc.close();
    }
}
```

   OUTPUT:

```
Default values of the array:
0
0
0
0
0
Enter 5 integers:
1
2
4
6
8
Updated values of the array:
1
2
4
6
8
```

2. 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.

CODE:

```
package org.assign6;

import java.util.Scanner;

public class Program2 {

    public static void acceptRecord(int[] arr) {
        Scanner sc = new Scanner(System.in);

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

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

    }
```

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

            acceptRecord(arr);
            printRecord(arr);
    }

}
```

OUTPUT:

```
Enter 5 Integers:
3
6
2
8
1
Array Values are:
3
6
2
8
1
```

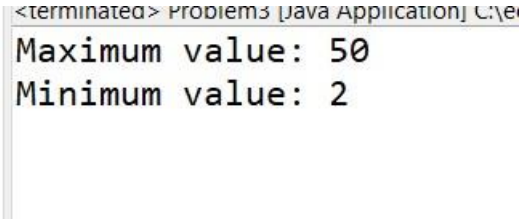
3. Write a program to find the maximum and minimum values in a single-dimensional array of integers.

CODE:

```
package org.assign6;

public class Problem3 {
    public static int findMax(int[] arr) {
        int max = arr[0];
        for (int value : arr) {
            if (value > max) {
                max = value;
            }
        }
        return max;
    }

    public static int findMin(int[] arr) {
        int min = arr[0];
        for (int value : arr) {
            if (value < min) {
                min = value;
```

```
            }
        }
        return min;
    }

    public static void main(String[] args) {

        int[] arr = {15, 20, 2, 50, 6};

        int max = findMax(arr);
        int min = findMin(arr);

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

OUTPUT:



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

CODE:

```
package org.assign6;

import java.util.Arrays;

public class Program4 {

    public static void main(String[] args) {
    int[] arr = {10,20,30,10,10,20};
    int[] result = removeDuplicates(arr);

    System.out.println("Array without duplicates: " + Arrays.toString(result));
  }

  public static int[] removeDuplicates(int[] arr) {
    return Arrays.stream(arr).distinct().toArray();
  }
}
```

OUTPUT:

```
<terminated> Program4 (1) [Java Application] C:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.
Array without duplicates: [10, 20, 30]
```

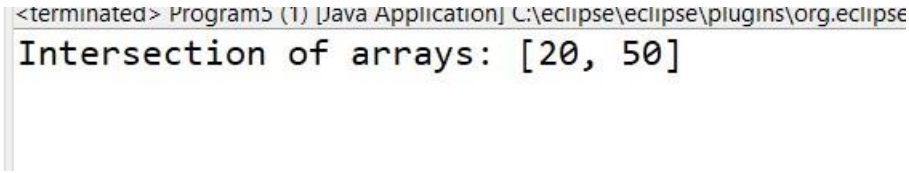
5. Write a program to find the intersection of two single-dimensional arrays.

CODE:

```
package org.assign6;


import java.util.Arrays;


public class Program5 {


        public static void main(String[] args) {
    int[] arr1 = {10, 20, 50};
    int[] arr2 = {20, 50, 40};


    int[] intersection = findIntersection(arr1, arr2);


    System.out.println("Intersection of arrays: " + Arrays.toString(intersection));
  }


  public static int[] findIntersection(int[] arr1, int[] arr2) {
        int[] result = new int[Math.min(arr1.length, arr2.length)];
    int index = 0;



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

```
        for (int j = 0; j < arr2.length; j++) {
            if (arr1[i] == arr2[j]) {

                boolean isDuplicate = false;
                for (int k = 0; k < index; k++) {
                    if (result[k] == arr1[i]) {
                        isDuplicate = true;
                        break;
                    }
                }

                if (!isDuplicate) {
                    result[index++] = arr1[i];
                }
            }
        }
    }

    return Arrays.copyOf(result, index); //Arrays.copyOf() helps us return only the relevant portion of the result array, i.e., the part that contains the common elements.

  }

}
```

OUTPUT:



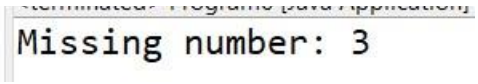
6. Write a program to find the missing number in an array of integers ranging from 1 to N.

CODE:

```
package org.assign6;


public class Program6 {


	public static void main(String[] args) {
	int[] arr = {1, 2, 4, 5};
	int missingNumber = findMissingNumber(arr, 5);


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


    public static int findMissingNumber(int[] arr, int n) {
	int totalSum = n * (n + 1) / 2;
	int arrSum = 0;
	for (int value : arr) {
	    arrSum += value;
	}


	return totalSum - arrSum;
    }
```

}

OUTPUT:



7. 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.

CODE:

```
package org.assign6;


import java.util.Scanner;


class ArrayField{


        private int[] arr;


   public ArrayField(int size) {

      arr = new int[size];

   }


   public void acceptRecord() {

      Scanner sc = new Scanner(System.in);

      System.out.println("Enter " + arr.length + " integers:");

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

         arr[i] = sc.nextInt();

      }
```

```
        sc.close();
    }

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

public class Program7 {
 public static void main(String[] args) {
            ArrayField f = new ArrayField(5);
     f.acceptRecord();
     f.printRecord();
 }
}
```

OUTPUT:

```
Enter 5 integers:
12
23
11
2
3
Array values:
12
23
11
2
3
```

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

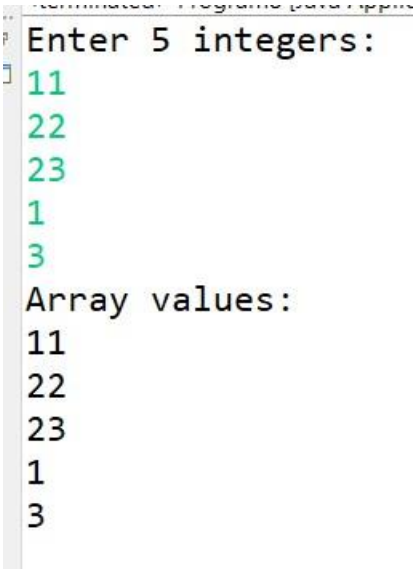
CODE:

```
package org.assign6;


import java.util.Scanner;


class FieldArray{

	private int[] arr;


	public FieldArray(int size) {

		arr = new int[size];

	}


	public int[] getArr() {

		return arr;

	}
```

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

}

public class Program8 {

	public static void main(String[] args) {
		FieldArray a = new FieldArray(5);
		Scanner sc = new Scanner(System.in);

		System.out.println("Enter 5 integers:");
	int[] temp = new int[5];
	for (int i = 0; i < temp.length; i++) {
	   temp[i] = sc.nextInt();
	}
	a.setArr(temp);

	int[] arr = a.getArr();
	System.out.println("Array values:");
	for (int value : arr) {
	   System.out.println(value);
	}
```

```
        sc.close();

            }

}
```

OUTPUT:



9. 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.

CODE:

```
package org.assign6;

import java.util.Scanner;

class AirplaneSeat{
        private char[][] seats;
        private int rows;
        private int cols;

        public AirplaneSeat(int rows, int cols) {
                this.rows = rows;
                this.cols = cols;
                seats = new char[rows][cols]; // Creating the 2D array with the size
```

```
    initializeSeats();
}

public void initializeSeats() {
        for(int i=0; i < rows; i++) {
                for(int j=0; j<cols; j++) {
                        seats[i][j] = 'A';
                }
        }
}

public boolean bookSeat(int row, int col) {
    if (isSeatValid(row, col) && seats[row][col] == 'A') {
        seats[row][col] = 'B';  // 'B' means Booked
        return true;
    }
    return false;
}

public boolean cancelBooking(int row, int col) {
    if (isSeatValid(row, col) && seats[row][col] == 'B') {
        seats[row][col] = 'A';  // Set back to 'A' for Available
        return true;
    }
    return false;
}

public boolean isSeatAvailable(int row, int col) {
    if (isSeatValid(row, col)) {
        return seats[row][col] == 'A';
    }
    return false;
}

public void displaySeatingChart() {
    System.out.println("Seating Chart:");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            System.out.print(seats[i][j] + " ");
        }
        System.out.println();
    }
}

private boolean isSeatValid(int row, int col) {
    return row >= 0 && row < rows && col >= 0 && col < cols;
}
```

```
}

public class Program9 {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);
	    System.out.println("Enter number of rows and columns for the airplane:");
	    int rows = scanner.nextInt();
	    int cols = scanner.nextInt();

	    AirplaneSeat manager = new AirplaneSeat(rows, cols);
	    manager.displaySeatingChart();

	    // Example booking, canceling, and checking
	    manager.bookSeat(1, 2); // Book seat at row 1, column 2
	    manager.bookSeat(0, 0); // Book seat at row 0, column 0
	    manager.cancelBooking(1, 2); // Cancel booking at row 1, column 2

	    manager.displaySeatingChart(); // Display updated seating chart
	    System.out.println("Seat at (1, 2) is available: " + manager.isSeatAvailable(1, 2));
	    System.out.println("Seat at (0, 0) is available: " + manager.isSeatAvailable(0, 0));

	    scanner.close();
	}

}
```

OUTPUT:

```
Enter number of rows and columns for the airplane:
4
4
Seating Chart:
A A A A
A A A A
A A A A
A A A A
Seating Chart:
B A A A
A A A A
A A A A
A A A A
Seat at (1, 2) is available: true
Seat at (0, 0) is available: false
```