Experiment Number : 04 **Date: 05.02.2026**

ARRAYS

Aim:

To understand and implement array operations in Java.

PRE LAB EXERCISE

QUESTIONS

- ✓ What is an array?

  An array is a collection of elements of the same data type stored in contiguous memory locations. Each element in an array is accessed using an index value.

- ✓ Why are arrays used?

  Arrays are used to store multiple values of the same data type using a single variable name. They make data storage, access, and manipulation easier and more efficient.

- ✓ What is the difference between array and variable?

Variable	Array
Stores only one value	Stores multiple values
Needs separate variables for each value	Uses a single name with indexes
Simple memory usage	Efficient for large data

IN LAB EXERCISE

Objective:

To perform array operations using simple programs.

PROGRAMS:

1. Program to Read and Print Array Elements

Code:

```
import java.util.Scanner;
public class ReadPrintArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        System.out.println("Array elements are:");
        for(int i = 0; i < 5; i++)
            System.out.print(arr[i] + " ");
    }
}
```

OUTPUT:

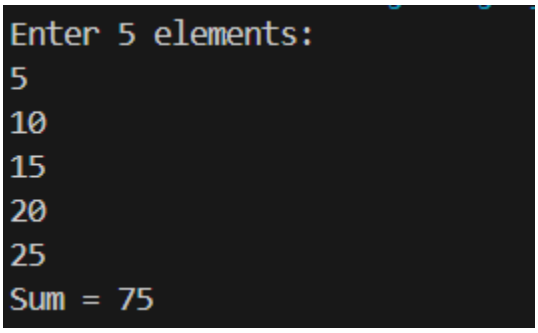
```
Enter 5 elements:
10
20
30
40
50
Array elements are:
10 20 30 40 50
```

2. Program to Find Sum of Array Elements

Code:

```
import java.util.Scanner;
public class SumArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        int sum = 0;
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        for(int i = 0; i < 5; i++)
            sum += arr[i];
        System.out.println("Sum = " + sum);
    }
}
```

OUTPUT:



3. Program to Find Largest Element in an Array

Code:

```
import java.util.Scanner;
public class LargestElement {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        int max = arr[0];
        for(int i = 1; i < 5; i++)
            if(arr[i] > max)
                max = arr[i];
        System.out.println("Largest element = " + max);
    }
}
```

OUTPUT:

```
Enter 5 elements:
12
45
23
9
30
Largest element = 45
```

4. Program to Reverse an Array

Code:

```
import java.util.Scanner;
public class ReverseArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        System.out.println("Reversed array:");
        for(int i = 4; i >= 0; i--)
            System.out.print(arr[i] + " ");
    }
}
```

```
Enter 5 elements:
55
66
77
88
99
Reversed array:
99 88 77 66 55
```

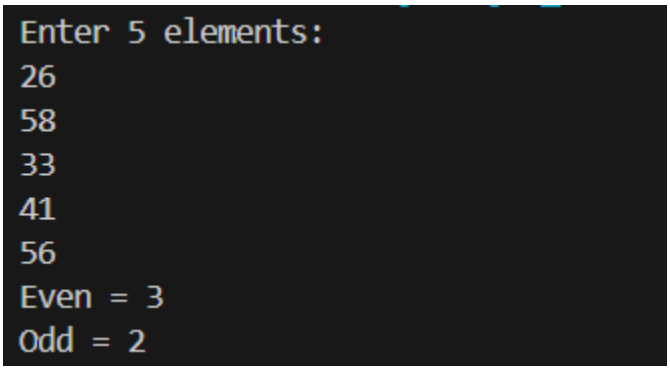
5. Program to Count Even and Odd Numbers

Code:

```
import java.util.Scanner;
public class EvenOddCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
        int[] arr = new int[5];
        int even = 0, odd = 0;
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        for(int i = 0; i < 5; i++) {
            if(arr[i] % 2 == 0)
                even++;
            else
                odd++;}
        System.out.println("Even = " + even);
        System.out.println("Odd = " + odd);
    }
}
```

OUTPUT:



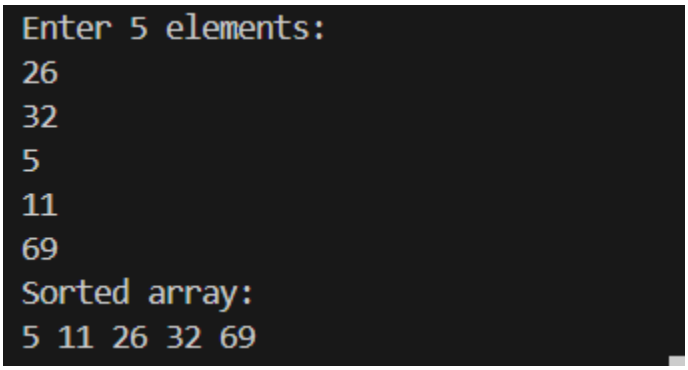
6. Program to Sort Array in Ascending Order

Code:

```
import java.util.Scanner;
public class SortArray {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
        int[] arr = new int[5];
        int temp;
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
            arr[i] = sc.nextInt();
        for(int i = 0; i < 5; i++) {
            for(int j = i + 1; j < 5; j++) {
                if(arr[i] > arr[j]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;}}}
        System.out.println("Sorted array:");
        for(int i = 0; i < 5; i++)
            System.out.print(arr[i] + " ");
    }
}
```

OUTPUT:



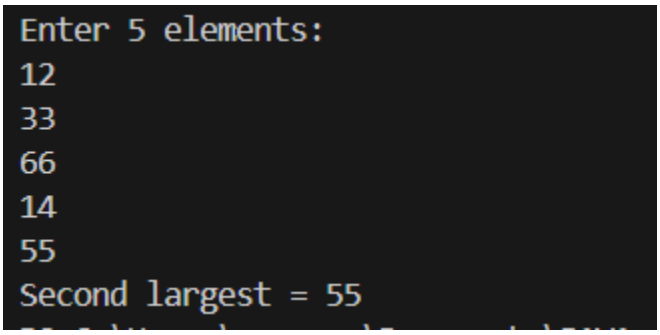
7. Program to Find Second Largest Element

Code:

```
import java.util.Scanner;
public class SecondLargest {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        System.out.println("Enter 5 elements:");
        for (int i = 0; i < 5; i++) {
            arr[i] = sc.nextInt();}
        int largest = Integer.MIN_VALUE;
        int second = Integer.MIN_VALUE;
        for (int i = 0; i < 5; i++) {
            if (arr[i] > largest) {
                second = largest;
                largest = arr[i]; }
            else if (arr[i] > second && arr[i] != largest) {
                second = arr[i]; }
        System.out.println("Second largest = " + second)}}
```

OUTPUT:



8. Program for Matrix Addition (2D Array)

Code:

```
import java.util.Scanner;
public class MatrixAddition {
```

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int[][] a = new int[2][2];
    int[][] b = new int[2][2];
    int[][] sum = new int[2][2];
    System.out.println("Enter elements of matrix A:");
    for(int i = 0; i < 2; i++)
       for(int j = 0; j < 2; j++)
          a[i][j] = sc.nextInt();
    System.out.println("Enter elements of matrix B:");
    for(int i = 0; i < 2; i++)
       for(int j = 0; j < 2; j++)
          b[i][j] = sc.nextInt();
    for(int i = 0; i < 2; i++)
       for(int j = 0; j < 2; j++)
          sum[i][j] = a[i][j] + b[i][j];
    System.out.println("Sum matrix:");
    for(int i = 0; i < 2; i++) {
       for(int j = 0; j < 2; j++)
          System.out.print(sum[i][j] + " ");
       System.out.println();}}}
```

OUTPUT:

```
Enter elements of matrix A:
12
3
46
30
Enter elements of matrix B:
55
23
89
10
Sum matrix:
67 26
135 40
```

POST LAB EXERCISE

- ✓ Why is array indexing usually started from zero instead of one?

  Array indexing starts from zero because it represents the offset from the base memory address. This makes memory access faster and more efficient.

- ✓ What happens if we try to access an array element outside its declared size?

  It causes a **runtime error** called ArrayIndexOutOfBoundsException in Java.

- ✓ How does memory allocation differ for static arrays and dynamic arrays?

  - **Static arrays** have fixed size decided at compile time.
  - **Dynamic arrays** get memory allocated at runtime and can be resized.

- ✓ Why is searching faster in arrays compared to linked lists?

  Arrays allow **direct access** to elements using index values, while linked lists require sequential traversal.

- ✓ What is the difference between contiguous and non-contiguous memory allocation?

Contiguous Memory	**Non-Contiguous Memory**
Memory stored in continuous blocks	Memory stored in scattered locations
Faster access	Slower access
Used by arrays	Used by linked lists

Result:

Thus the array operations were executed successfully.

ASSESSMENT

Description	Max Marks	Marks Awarded
Pre Lab Exercise	**5**	
In Lab Exercise	**10**	
Post Lab Exercise	**5**	
Viva	**10**	
Total	**30**	
	Faculty Signature	