Experiment Number : 04 **Date:**

ARRAYS

Aim:

To understand and implement array operations in Java.

PRE LAB EXERCISE

QUESTIONS

- **What is an array?**
- An array is a data structure that stores multiple values of the same type in a single variable.
- Each value in an array is stored at a specific position called an index.
- **Why are arrays used?**
- Arrays are used to store many related values under one name, reduce code length, and make data processing like searching and sorting easier.
- **What is the difference between array and variable?**
- A variable stores only one value, whereas an array stores multiple values of the same data type in contiguous memory locations.

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:

Input:

10 20 30 40 50

Output:

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:

Input:

5 10 15 20 25

Output:

Sum = 75

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:

Input:

12 45 23 9 30

Output:

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] + " ");
   }
}
```
OUTPUT:
Input:
1 2 3 4 5
Output:
Reversed array:
5 4 3 2 1

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:

Input:
2 7 4 9 10
Output:
Even = 3
Odd = 2

```
Enter 5 elements:
2
9
4
6
8
Even = 4
```

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:
Input:
45 12 78 23 9
Output:
Sorted array:
9 12 23 45 78

```
Enter 5 elements:
2 7 8 12 3
Sorted array:
2 3 7 8 12 %
```

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 = arr[0];
      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);
      sc.close();
   }
}
```

OUTPUT:

Input:

10 45 23 89 67

Output:

Second largest = 67

```
Enter 5 elements:
12233 567 8990 887 333
Second largest = 8990
```

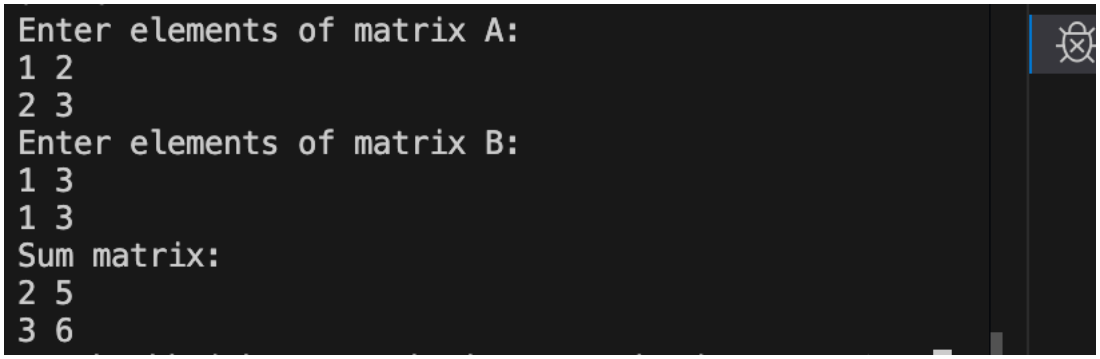
8. Program for Matrix Addition (2D Array)

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:

Matrix A:
1 2
3 4
Matrix B:
5 6
7 8
Sum matrix:
6 8
10 12



POST LAB EXERCISE

- **Why is array indexing usually started from zero instead of one?**
  Array indexing starts from zero because the index represents the offset from the base memory address.The first element is at offset 0, the second at offset 1, and so on.This makes address calculation faster and simpler for the computer.
- **What happens if we try to access an array element outside its declared size?**
  Accessing an array outside its declared size causes undefined behavior.
  it may return garbage values, overwrite other memory, or crash the program.
  Some languages may detect this and throw a runtime error.
- **How does memory allocation differ for static arrays and dynamic arrays?**
  Static arrays are allocated at compile time and have a fixed size.
  Dynamic arrays are allocated at runtime and their size can be changed.
  Static arrays use stack memory, while dynamic arrays usually use heap memory.
- **Why is searching faster in arrays compared to linked lists?**
  Arrays store elements in contiguous memory locations.
  This allows direct access to any element using its index.
  Linked lists require sequential traversal, making searching slower.
- **What is the difference between contiguous and non-contiguous memory allocation?**
  In contiguous allocation, memory blocks are stored next to each other.
  In non-contiguous allocation, memory blocks are scattered in different locations.
  Arrays use contiguous memory, while linked lists use non-contiguous memory.

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		