Experiment-3

Name: Deepanshu Saini **UID: 23BCS13189**
Section : 23BCS_KRG_01 **Subject Name : DAA**
Subject Code:23CSH-301 **Date:29/07/2025**

---

1. **Aim**: Evaluate the complexity of the developed program to find frequency of elements in a given array.

2. Objective:

- To input elements into an array from the user.
- To read a key value and count its frequency in the array.
- To display the frequency of the given key.

3. PseudoCode:

```
START

  PRINT "Enter the size of array"

  READ n

  DECLARE array arr[n]

  FOR i = 0 TO n-1

    PRINT "Enter element for index i"

    READ arr[i]

  END FOR

  PRINT "Set the key"

  READ key

  SET frequency = 0

  FOR i = 0 TO n-1

    IF arr[i] == key THEN

      frequency = frequency + 1
```

```
            END IF

        END FOR

        PRINT "The frequency of key is" + frequency

    END
```

4. Code:

```
    import java.util.*;
class exp3{
  public static void main(String[] args) {
    System.out.print("Enter the size of array: ");
    Scanner sc=new Scanner(System.in);
    int n=sc.nextInt();
    int[]arr=new int[n];
    for(int i=0;i<n;i++){
      System.out.print("enter for index "+i+" :");
      arr[i]=sc.nextInt();
    }
    System.out.print("set the key: ");
    int key=sc.nextInt();
    int frequency=0;
    for(int i=0;i<n;i++){
      if(arr[i]==key){
        frequency++;
      }
    }
```

```
        System.out.print("the frequency of "+key+" is "+frequency);

        System.out.println("Time Complexity: O(n)");

    }

}
```

5. Output:

```
a\Local\Temp\vscodesws_6ea9e\jdt_ws\jdt.ls-java-project\bin' 'exp3'
Enter the size of array: 5
enter for index 0 :0
enter for index 1 :1
enter for index 2 :2
enter for index 3 :2
enter for index 4 :3
set the key: 2
the frequency of 2 is 2
Time Complexity: O(n)
```

6. Learning Outcomes:

- Understanding how to **take user input** for arrays in Java using `Scanner`.
- Learning to **iterate through arrays** using loops.
- Applying **conditional statements** to compare values within an array.
- Gaining the ability to **count occurrences** (frequency) of a specific element.
- Understanding and identifying the **time complexity** of an algorithm (O(n) in this case).