

# ICA Top 25 Programs in JAVA

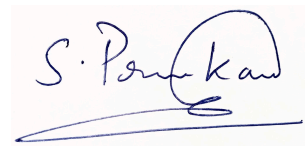


**INNOVATIVE CODES ACADEMY**

*Fly to high*



**Prepared by,**  
Prabakaran S  
Software Engineer



## Table of content

	Topics
	<b>Top 25 Programs in JAVA</b>
<b>0</b>	<b>Sample</b>
<b>1</b>	Leap year program in JAVA
<b>2</b>	Check the given number is an Odd or Even in JAVA
<b>3</b>	Write the program to find whether given number is Armstrong or not in JAVA
<b>4</b>	Write the program to find whether given number is Palindrome or not in JAVA
<b>5</b>	Write the program to Reverse the Number
<b>6</b>	Sum of digits
<b>7</b>	Swapping the two numbers with and without using a third variable in Java
<b>8</b>	Write the program to find whether the the given number is Prime or not in Java
<b>9</b>	Write a Fibonacci sequence using recursion in Java Program
<b>10</b>	Find the factorial of an integer in Java
<b>11</b>	The first occurrence in a string
<b>12</b>	Write a program to Sort the array in descending & Ascending.
<b>13</b>	Write a Java program to check the vowels are found in a given string
<b>14</b>	Reverse each word in the given String
<b>15</b>	Write the program to Reverse a string in JAVA
<b>16</b>	Write the program to Reverse the Array in JAVA
<b>17</b>	Write the program to Remove duplicates in the given String in JAVA

<b>18</b>	Count number of digit
<b>19</b>	Write the JAVA program for Count of each character in the String
<b>20</b>	Write the JAVA program for Star pattern problem - Right RightAlignedStarPattern program :
<b>21</b>	Write the JAVA program for Star pattern problem - Left Left-aligned star pattern:
<b>22</b>	Write the JAVA program for Star pattern problem - Pyramid Pyramid star pattern:
<b>23</b>	write a Java Program to find how many Armstrong numbers are in the array or not
<b>24</b>	Write a Java Program to find how many Palindrome numbers are in the array or not
<b>25</b>	Write a Java Program to find how many prime numbers are in the array or not

*Innovative Codes Academy*

## Software and Tools:

We have used **Notepad++** software for running the JAVA program in the computer before running the program you will install **JDK (Java SE Development Kit 20.0.1)** in your computer.

### Software:

Notepad++ v8.5.3



### Install JDK (JAVA SE Development Kit 20.0.1)

Verify your system: (use the below-mentioned command)

1. Open Command prompt or Windows + R
2. Then enter the command cmd
3. After opening the command prompt use this command (java -version)
4. Then it will show whether any JDK file exists or not, if no files are there you will install JDK in your computer.

```
C:\Users\seeni>java --version
java 20 2023-03-21
Java(TM) SE Runtime Environment (build 20+36-2344)
Java HotSpot(TM) 64-Bit Server VM (build 20+36-2344, mixed mode, sharing)
```

## 0. Introduction to the array

In computer science, an array is a data structure that stores a collection of elements such as values or variables, all elements are stored in an array format of the same data type, and the values are stored in the array in the index format, index value starting from 0 to n values, for example, the array size is 5 the index value is start from 0 and end with 4 (0,1,2,3,4) the value the size of an array is typically fixed at the time of creation, although some programming languages allow dynamic resizing of arrays during runtime.

### Diagrammatic example:

The array size is 5, and the values are 10,20,30,40,50.

10	20	30	40	50
----	----	----	----	----

Index -> 0                      1                      2                      3                      4  
Value

## Array Initialization

To initialize an array in Java with user input, you can make use of the Scanner class from the java.util package. The Scanner class is used to get the input from the user. Here's an example of how you can initialize an array with user input in Java:

### Program in JAVA:

```
import java.util.Scanner;

public class ArrayInitialization {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();
        int[] array = new int[size];

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < size; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            array[i] = scanner.nextInt();
        }

        System.out.println("Array elements:");
        for (int i = 0; i < size; i++) {
            System.out.println("Element " + (i + 1) + ": " + array[i]);
        }
        scanner.close();
    }
}
```

### Sample Output:

```
D:\Prabakaran S\Website\ICA_Placement_Programs>javac ArrayInitialization.java
```

```
D:\Prabakaran S\Website\ICA_Placement_Programs>java ArrayInitialization
```

Enter the size of the array: 5  
Enter the elements of the array:  
Enter element 1: 10  
Enter element 2: 20  
Enter element 3: 30  
Enter element 4: 40  
Enter element 5: 50  
Array elements:  
Element 1: 10 Index value:0  
Element 2: 20 Index value:1  
Element 3: 30 Index value:2  
Element 4: 40 Index value:3  
Element 5: 50 Index value:4

### Find the request element in the Array

To find a specific element in an array in Java, you can iterate over the array and check each element until a match is found. Here's an example code that demonstrates how to find a requested element in an array:

#### Program in JAVA:

```
public class ArraySearchExample {  
    public static void main(String[] args) {  
        // Sample array  
        int[] array = {10, 20, 30, 40, 50};  
  
        // Element to search  
        int element = 30;  
  
        // Flag to track if element is found  
        boolean found = false;  
  
        // Iterate over the array  
        for (int i = 0; i < array.length; i++) {  
            if (array[i] == element) {  
                found = true;  
                break;  
            }  
        }  
  
        // Check if the element was found  
        if (found) {  
            System.out.println("Element found in the array.");  
        }  
    }  
}
```

```

    } else {
        System.out.println("Element not found in the array.");
    }
}
}

```

In the above code, we have an array `array` and an `element` that we want to find in the array. We iterate over each element of the array using a for loop and check if the current element is equal to the requested element. If a match is found, we set the `found` flag to `true` and break out of the loop. After the loop, we check the value of the `found` flag to determine if the element was found or not.

### Sample Output:

```

Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

```

```

C:\Users\seeni>D:

```

```

D:\>cd Prabakaran S\Website\ICA_Placement_Programs

```

```

D:\Prabakaran S\Website\ICA_Placement_Programs>javac ArraySearch.java

```

```

D:\Prabakaran S\Website\ICA_Placement_Programs>java ArraySearch
Element 30 found in the array at 2 index

```

```

C:\Users\seeni>D:

D:\>cd Prabakaran S\Website\ICA_Placement_Programs

D:\Prabakaran S\Website\ICA_Placement_Programs>javac ArraySearch.java

D:\Prabakaran S\Website\ICA_Placement_Programs>java ArraySearch
Element 30 found in the array at 2 index

```

### Example - 1: (Two or more occurrences of the same element in the array)

If there are two or more occurrences of the same element in the array and you want to find the first occurrence, you can modify the previous code as follows:

```

public class ArraySearchExample {
    public static void main(String[] args) {
        // Sample array
        int[] array = {10, 20, 30, 40, 30, 50};
    }
}

```

```

// Element to search
int element = 30;

// Index of the found element (-1 if not found)
int foundIndex = -1;

// Iterate over the array
for (int i = 0; i < array.length; i++) {
    if (array[i] == element) {
        foundIndex = i;
        break;
    }
}

// Check if the element was found
if (foundIndex != -1) {
    System.out.println("Element found at index " + foundIndex);
} else {
    System.out.println("Element not found in the array.");
}
}
}

```

*Innovative Codes Academy*

In this modified code, we added a new variable `foundIndex` to keep track of the index where the element is found. Initially, we set it to `-1` to indicate that the element is not found. When a match is found, we update the `foundIndex` variable with the index of the element and break out of the loop. After the loop, we check the value of `foundIndex`. If it is still `-1`, it means the element was not found. Otherwise, we display the index where the element is found.



## 1) Leap year program in JAVA:

```
import java.util.Scanner;
public class LeapYear
{
    public static boolean isLeapYear(int year)
    {
        // A leap year is divisible by 4 and not divisible by 100, or divisible by 400.
        return (year % 4 == 0) && (year % 100 != 0 || year % 400 == 0);
    }

    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int year = scanner.nextInt();

        if (isLeapYear(year))
        {
            System.out.println(year + " is a leap year.");
        }
        else {
            System.out.println(year + " is not a leap year.");
        }
    }
}
```

### Sample Output:

```
Enter a number: 2023
2023 is not a leap year.
```

## 2) Check the given number is an Odd or Even in JAVA:

```
import java.util.Scanner;

public class OddEven {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (number % 2 == 0) {
            System.out.println(number + " is even.");
        }
    }
}
```

```

    } else {
        System.out.println(number + " is odd.");
    }

    scanner.close(); // Remember to close the scanner to prevent resource leaks
}
}

```

### Sample Output:

```

Enter a number: 23
23 is odd.

```

### 3) Write the program to find whether given number is Armstrong or not in JAVA:

```
import java.util.Scanner;
```

```

public class ArmstrongNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.close();
        if (isArmstrong(number)) {
            System.out.println(number + " is an Armstrong number.");
        } else {
            System.out.println(number + " is not an Armstrong number.");
        }
    }
}

```

```

public static boolean isArmstrong(int number) {
    int originalNumber, remainder, result = 0, n = 0;
    originalNumber = number;

```

```

    // Counting the number of digits
    while (originalNumber != 0) {
        originalNumber /= 10;
        ++n;
    }

```

```

    originalNumber = number;

```

```

    // Calculating result
    while (originalNumber != 0) {

```

```

        remainder = originalNumber % 10;
        result += Math.pow(remainder, n);
        originalNumber /= 10;
    }

    // Checking if number is Armstrong
    return result == number;
}
}

```

### Sample Output:

```

Enter a number: 153
153 is an Armstrong number.

```

### 4) Write the program to find whether given number is Palindrome or not in JAVA:

```

import java.util.Scanner;

public class Palindrome {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.close();

        if (isPalindrome(number)) {
            System.out.println(number + " is a palindrome.");
        } else {
            System.out.println(number + " is not a palindrome.");
        }
    }

    public static boolean isPalindrome(int number) {
        int reversedNumber = 0;
        int originalNumber = number;
        // For reverse a number
        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number /= 10;
        }

        return originalNumber == reversedNumber;
    }
}

```

```
}  
}
```

**Sample Output:**

Enter a number: 121

121 is a palindrome.

**5) Write the program to Reverse the Number:**

```
import java.util.Scanner;
```

```
public class ReverseNumber {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter a number: ");  
        int number = scanner.nextInt();  
        scanner.close();  
  
        int reversedNumber = reverse(number);  
        System.out.println("Reversed number: " + reversedNumber);  
    }  
}
```

```
    public static int reverse(int number) {  
        int reversedNumber = 0;  
        while (number != 0) {  
            int digit = number % 10; // Extract the last digit  
            reversedNumber = reversedNumber * 10 + digit; // Append the digit to  
reversed number  
            number /= 10; // Remove the last digit from the original number  
        }  
        return reversedNumber;  
    }  
}
```

**Sample Output:**

Enter a number: 12345

Reversed number: 54321

## 6) Sum of digits:

```
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.close();

        int sum = sumOfDigits(number);
        System.out.println("Sum of digits: " + sum);
    }

    public static int sumOfDigits(int number) {
        int sum = 0;
        while (number != 0) {
            int digit = number % 10; // Extract the last digit
            sum += digit; // Add the digit to the sum
            number /= 10; // Remove the last digit from the original number
        }
        return sum;
    }
}
```

### Sample Output:

```
Enter a number: 153
Sum of digits: 9
```

## 7) Swapping the two numbers with and without using a third variable in Java:

### Swapping with a third variable:

```
import java.util.Scanner;

public class SwapWithThirdVariable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int a = scanner.nextInt();
```

```

System.out.print("Enter second number: ");
int b = scanner.nextInt();

System.out.println("Before swapping: a = " + a + ", b = " + b);

// Swapping using a third variable
int temp = a;
a = b;
b = temp;

System.out.println("After swapping with third variable: a = " + a + ", b = " + b);

scanner.close();
}
}

```

### Sample Output:

```

Enter first number: 100
Enter second number: 300
Before swapping: a = 100, b = 300
After swapping with third variable: a = 300, b = 100

```

*Innovative Codes Academy*

### Swapping without using a third variable:

```

import java.util.Scanner;

public class SwapWithoutThirdVariable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int a = scanner.nextInt();

        System.out.print("Enter second number: ");
        int b = scanner.nextInt();

        System.out.println("Before swapping: a = " + a + ", b = " + b);

        // Swapping without using a third variable
        a = a + b;
        b = a - b;
        a = a - b;
    }
}

```

```

        System.out.println("After swapping without third variable: a = " + a + ", b = " +
b);

        scanner.close();
    }
}

```

### Sample Output:

```

Enter first number: 1000
Enter second number: 6000
Before swapping: a = 1000, b = 6000
After swapping without third variable: a = 6000, b = 1000

```

### 8) Write the program to find whether the the given number is Prime or not in Java:

```
import java.util.Scanner;
```

```

public class PrimeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.close();

        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }

    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false; // 1 and numbers less than 1 are not prime
        }
        for (int i = 2; i <= Math.sqrt(number); i++) // math.sprt is used reduce time
complexity
        {
            if (number % i == 0) {
                return false; // If number is divisible by any number other than 1 and itself,
it's not prime
            }
        }
    }
}

```

```

    }
    return true; // If no divisor is found other than 1 and itself, it's prime
}
}

```

**Sample Output:**

Enter a number: 2  
2 is a prime number.

**9) Write a Fibonacci sequence using recursion in JAVA Program:**

```
import java.util.Scanner;
```

```

public class FibonacciRecursion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of terms: ");
        int n = scanner.nextInt();
        scanner.close();

        System.out.println("Fibonacci sequence:");
        for (int i = 0; i < n; i++) {
            System.out.print(fibonacci(i) + " ");
        }
    }

    public static int fibonacci(int n) {
        if (n <= 1) {
            return n; // Base case: If n is 0 or 1, return n
        } else {
            // Recursive call: Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }
}

```

**Sample Output:**

Enter the number of terms: 5  
Fibonacci sequence:  
0 1 1 2 3



## 10) Find the factorial of an integer in Java

```
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a non-negative number: ");
        int n = scanner.nextInt();
        scanner.close();

        if (n < 0) {
            System.out.println("Factorial of a negative number is undefined.");
        } else {
            long factorial = findFactorial(n);
            System.out.println("Factorial of " + n + " is: " + factorial);
        }
    }

    public static long findFactorial(int n) {
        if (n == 0 || n == 1) {
            return 1; // Factorial of 0 and 1 is 1
        } else {
            return n * findFactorial(n - 1); // Calculate factorial using recursion
        }
    }
}
```

### Sample Output:

```
Enter a non-negative number: 5
Factorial of 5 is: 120
```

## 11) First occurrence in a string

```
import java.util.Scanner;

public class FirstOccurrence {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str= sc.nextLine(); //reads string
        System.out.print("Input a character: ");
```

```

char ch = sc.next().charAt(0);

int index = firstOccurrence(str, ch);
if (index != -1) {
    System.out.println("First occurrence of " + ch + " is at index " + index);
} else {
    System.out.println(""" + ch + " not found in the string.");
}
}

public static int firstOccurrence(String str, char ch) {
    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == ch) {
            return i; // Return the index if character is found
        }
    }
    return -1; // Return -1 if character is not found
}
}

```

### Sample Output:

Enter a string: This is a first string  
Input a character: t  
First occurrence of 't' is at index 14

### 12) Write a program to Sort the array in descending & Ascending:

```

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

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        int[] array = new int[size];

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < size; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            array[i] = scanner.nextInt();
        }
    }
}

```

```

// Sorting array in ascending order
SortAscending(array);
System.out.print("Array in ascending order: ");
printArray(array);

// Sorting array in descending order
SortDescending(array);
System.out.print("Array in descending order: ");
printArray(array);
}

```

```

public static void SortAscending(int[] array) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (array[j] < array[minIndex]) {
                minIndex = j;
            }
        }
        // Swap array[i] and array[minIndex]
        int temp = array[i];
        array[i] = array[minIndex];
        array[minIndex] = temp;
    }
}

```

```

public static void SortDescending(int[] array) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (array[j] > array[maxIndex]) {
                maxIndex = j;
            }
        }
        // Swap array[i] and array[maxIndex]
        int temp = array[i];
        array[i] = array[maxIndex];
        array[maxIndex] = temp;
    }
}

```

```

public static void printArray(int[] array) {

```

```

        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }
}

```

### Sample Output:

```

Enter the size of the array: 5
Enter the elements of the array:
Enter element 1: 20
Enter element 2: 46
Enter element 3: 50
Enter element 4: 84
Enter element 5: 14
Array in ascending order: 14 20 46 50 84
Array in descending order: 84 50 46 20 14

```

### 13) Write a Java program to check the vowels are found in a given string:

```
import java.util.Scanner;
```

```

public class VowelCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();
        scanner.close();

        if (containsVowels(str)) {
            System.out.println("Vowels are found in the string.");
        } else {
            System.out.println("No vowels found in the string.");
        }
    }

    public static boolean containsVowels(String str) {
        // Convert the string to lowercase to handle both uppercase and lowercase
        // vowels
        str = str.toLowerCase();
        // Check if the string contains any of the vowels: a, e, i, o, u
        return str.contains("a") || str.contains("e") || str.contains("i") || str.contains("o") ||
        str.contains("u");
    }
}

```

**Sample Output:**

Enter a string: Aeioou

Vowels are found in the string.

**14) Reverse each word in the given String:**

```
import java.util.Scanner;
```

```
public class Main{  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter a string: ");  
        String str = scanner.nextLine();  
        scanner.close();  
  
        String reversedString = reverseEachWord(str);  
        System.out.println("Reversed string: " + reversedString);  
    }  
  
    public static String reverseEachWord(String str) {  
        // Split the string into words  
        String[] words = str.split(" ");  
        StringBuilder reversedString = new StringBuilder();  
  
        // Reverse each word and concatenate them  
        for (String word : words) {  
            StringBuilder reversedWord = new StringBuilder(word);  
            reversedString.append(reversedWord.reverse()).append(" ");  
        }  
  
        // Trim any leading or trailing whitespace  
        return reversedString.toString().trim();  
    }  
}
```

**Sample Output:**

Enter a string: Hello world

Reversed string: olleH dlrow

### 15) Write the program to Reverse a string in JAVA

```
import java.util.Scanner;

public class ReverseString {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();
        scanner.close();

        String reversedString = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            reversedString += str.charAt(i);
        }

        System.out.println("Reversed string: " + reversedString);
    }
}
```

#### Sample Output:

```
Enter a string: Computer
Reversed string: retupmoC
```

### 16) Write the program to Reverse the Array in JAVA

```
import java.util.Arrays;
import java.util.*;

public class ReverseArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        int[] array = new int[size];

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < size; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            array[i] = scanner.nextInt();
        }
    }
}
```

```

    }

    System.out.println("Original array: " + Arrays.toString(array));

    reverseArray(array);

    System.out.println("Reversed array: " + Arrays.toString(array));
}

public static void reverseArray(int[] array) {
    int start = 0;
    int end = array.length - 1;
    while (start < end) {
        // Swap elements at start and end indices
        int temp = array[start];
        array[start] = array[end];
        array[end] = temp;
        // Move start index to the right and end index to the left
        start++;
        end--;
    }
}
}

```

*Innovative Codes Academy*

### Sample Output:

```

Enter the size of the array: 5
Enter the elements of the array:
Enter element 1: 10
Enter element 2: 80
Enter element 3: 52
Enter element 4: 46
Enter element 5: 1
Original array: [10, 80, 52, 46, 1]
Reversed array: [1, 46, 52, 80, 10]

```

### 17) Write the program to Remove duplicates in the given String in JAVA

```
import java.util.Scanner;
```

```

public class RemoveDuplicates {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();
        scanner.close();
    }
}

```

```

String result = removeDuplicates(str);
System.out.println("String after removing duplicates: " + result);
}

public static String removeDuplicates(String str) {
    StringBuilder result = new StringBuilder();

    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        if (result.indexOf(String.valueOf(ch)) == -1) {
            result.append(ch);
        }
    }

    return result.toString();
}
}

```

### Sample Output:

Enter a string: BEEE  
String after removing duplicates: BE

### 18) Count number of digit

```

import java.util.Scanner;

public class CountDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.close();

        int digitCount = countDigits(number);
        System.out.println("Number of digits in " + number + ": " + digitCount);
    }

    public static int countDigits(int number) {
        // Convert the number to a string and get its length
        return String.valueOf(number).length();
    }
}

```

### Sample Output:



Enter a number: 65465  
Number of digits in 65465: 5

### 19) Write the JAVA program for Count of each character in the String

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // Create a Scanner object to take user input
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        // Read the input string
        String str = scanner.nextLine();
        scanner.close();

        // Create an array to store character counts
        int[] charCount = new int[256];

        // Iterate through each character of the input string
        for (int ch : str.toCharArray()) {
            // Increment the count of the character in the array
            charCount[ch]++;
        }

        // Print character counts
        System.out.println("Character counts:");
        for (int i = 0; i < charCount.length; i++) {
            // Print only if the count is non-zero
            if (charCount[i] != 0) {
                // Convert the index (ASCII value) to char and print the character along
                with its count
                System.out.println("'" + (char) i + "' : " + charCount[i]);
            }
        }
    }
}
```

#### Sample Output:

```
Enter a string: Life
Character counts:
'L' : 1
'e' : 1
'f' : 1
```

'i' : 1

**20) Write the JAVA program for Star pattern problem - Right RightAlignedStarPattern program :**

```
import java.util.Scanner;
```

```
public class RightAlignedStarPattern {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of rows: ");  
        int rows = scanner.nextInt();  
        scanner.close();  
  
        // Iterate over each row  
        for (int i = 1; i <= rows; i++) {  
            // Print spaces before printing stars in each row  
            for (int space = 1; space <= rows - i; space++) {  
                System.out.print(" ");  
            }  
            // Print stars in each row  
            for (int j = 1; j <= i; j++) {  
                System.out.print("* ");  
            }  
            // Move to the next line after printing stars in a row  
            System.out.println();  
        }  
    }  
}
```

**Sample Output:**

Enter the number of rows: 5

```
  *  
 * *  
* * *  
* * * *  
* * * * *
```

**21) Write the JAVA program for Star pattern problem - Left  
Left-aligned star pattern:**

```
import java.util.Scanner;

public class LeftAlignedStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        scanner.close();

        // Iterate over each row
        for (int i = 1; i <= rows; i++) {
            // Print stars in each row
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            // Move to the next line after printing stars in a row
            System.out.println();
        }
    }
}
```

**Sample Output:**

**Enter the number of rows: 5**

```
*
**
***
****
*****
```

**22) Write the JAVA program for Star pattern problem - Pyramid  
Pyramid star pattern:**

```
import java.util.Scanner;

public class PyramidStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
```

```

scanner.close();

// Iterate over each row
for (int i = 1; i <= rows; i++) {
    // Print spaces before printing stars in each row
    for (int space = 1; space <= rows - i; space++) {
        System.out.print(" ");
    }
    // Print stars in each row
    for (int j = 1; j <= 2 * i - 1; j++) {
        System.out.print("*");
    }
    // Move to the next line after printing stars in a row
    System.out.println();
}
}
}

```

### Sample Output:

Enter the number of rows: 5

```

*
***
*****
*****
*****

```

**23) Write a Java Program to find how many Armstrong numbers are in the array or not**

```
import java.util.Scanner;
```

```
public class ArmstrongNumber1 {
```

```
    // Function to check if a number is Armstrong or not
    static boolean isArmstrong(int num) {
```

```
        int originalNum = num;
        int sum = 0;
```

```
        // Count number of digits
```

```
        int numDigits = 0;
        while (originalNum != 0) {
            numDigits++;
            originalNum /= 10;
```

```
        }
```

```

originalNum = num;

// Calculate sum of digits raised to the power of numDigits
while (originalNum != 0) {
    int digit = originalNum % 10;
    int power = 1;
    for (int i = 0; i < numDigits; i++) {
        power *= digit;
    }
    sum += power;
    originalNum /= 10;
}

// Check if the number is Armstrong
return sum == num;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Input array size
    System.out.print("Enter the size of the array: ");
    int size = scanner.nextInt();

    // Input array elements
    int[] arr = new int[size];
    System.out.println("Enter the elements of the array:");
    for (int i = 0; i < size; i++) {
        arr[i] = scanner.nextInt();
    }

    int count = 0; // Counter for Armstrong numbers

    // Check each number in the array
    for (int num : arr) {
        if (isArmstrong(num)) {
            count++;
        }
    }

    System.out.println("Number of Armstrong numbers in the array: " + count);
}
}

```

### Sample Output:

```
Enter the size of the array: 5
Enter the elements of the array:
153
373
201
845
407
Number of Armstrong numbers in the array: 2
```

### 24) Write a Java Program to find how many Palindrome numbers are in the array or not

```
import java.util.Scanner;

public class PalindromeNumberArray {
    // Function to check if a number is a palindrome or not
    static boolean isPalindrome(int num) {
        int originalNum = num;
        int reversedNum = 0;

        // Reverse the number
        while (originalNum != 0) {
            int remainder = originalNum % 10;
            reversedNum = reversedNum * 10 + remainder;
            originalNum /= 10;
        }

        // Check if the number is equal to its reverse
        return num == reversedNum;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input array size
        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();
    }
}
```

```

// Input array elements
int[] arr = new int[size];
System.out.println("Enter the elements of the array:");
for (int i = 0; i < size; i++) {
    arr[i] = scanner.nextInt();
}

int count = 0; // Counter for palindrome numbers

// Check each number in the array
for (int num : arr) {
    if (isPalindrome(num)) {
        count++;
    }
}

System.out.println("Number of palindrome numbers in the array: " + count);
}
}

```

**Sample Output:**

```

Enter the size of the array: 5
Enter the elements of the array:
121
400
000
231
333
Number of palindrome numbers in the array: 3

```

**25) Write a Java Program to find how many prime numbers are in the array or not**

```

import java.util.Scanner;

public class PrimeNumberCount {

    // Function to check if a number is prime
    static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
    }
}

```

```

    for (int i = 2; i <= Math.sqrt(num); i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Input array size
    System.out.print("Enter the size of the array: ");
    int size = scanner.nextInt();

    // Input array elements
    int[] arr = new int[size];
    System.out.println("Enter the elements of the array:");
    for (int i = 0; i < size; i++) {
        arr[i] = scanner.nextInt();
    }

    int count = 0; // Counter for prime numbers

    // Check each number in the array
    for (int num : arr) {
        if (isPrime(num)) {
            count++;
        }
    }

    System.out.println("Number of prime numbers in the array: " + count);
}
}

```

**Sample Output:**

```

Enter the size of the array: 5
Enter the elements of the array:
1
2
6
5
7
Number of prime numbers in the array: 3

```



*Innovative Codes Academy*

***Thank You for Reading***