1)reverse word using loop

```
public class ReverseWord {
    public static void main(String[] args) {
        String word = "hello";
        String reversed = "";
        for (int i = word.length() - 1; i >= 0; i--) {
            reversed += word.charAt(i);
        }
        System.out.println("Reversed word: " + reversed);
    }
}
```

2)check user name is valid or not

```
import java.util.regex.*;
public class UsernameValidation {
    public static void main(String[] args) {
        String username = "Saveetha@789"; // Sample input username
        if (username.matches("[a-zA-Z0-9@]+")) {
            System.out.println("Valid username");
        } else {
            System.out.println("Invalid username");
        }
    }
}
```

3)reverse a number using loop

```
public class ReverseNumber {
    public static void main(String[] args) {
        int num = 12345, reversed = 0;
        while (num != 0) {
            reversed = reversed * 10 + num % 10;
            num /= 10;
        }
        System.out.println("Reversed number: " + reversed);
    }
}
```

4)eligible for vote or not

```
import java.util.Scanner;
public class VotingEligibility {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter age: ");
        int age = scanner.nextInt();
        if (age >= 18) {
            System.out.println("Eligible to vote.");
        } else {
            System.out.println("Not eligible to vote.");
        }

        scanner.close();
    }
}
```

5)lcm and gcd of numbers

```
import java.util.Scanner;
public class LCMandGCD {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int gcd = gcd(a, b);
        int lcm = (a * b) / gcd;
        System.out.println("GCD: " + gcd);
        System.out.println("LCM: " + lcm);
    }
       static int gcd(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

6)print right triangle * pattern

```
public class RightSideTrianglePattern {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

7)cal simple interest

```
import java.util.Scanner;
public class SimpleInterest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter principal: ");
        double principal = sc.nextDouble();
        System.out.print("Enter rate of interest: ");
        double rate = sc.nextDouble();
        System.out.print("Enter time in years: ");
        double time = sc.nextDouble();
        double interest = calculateSimpleInterest(principal, rate, time);
        System.out.println("Simple Interest: " + interest);
    }
       static double calculateSimpleInterest(double p, double r, double t) {
        return (p * r * t) / 100;
    }
}
```

8)even sum of fibonacci series

```
public class EvenFibonacciSum {
    public static void main(String[] args) {
        int limit = 4000000;
        int sum = 0, a = 0, b = 1;
        while (b <= limit) {
            if (b % 2 == 0) {
                sum += b;
            }
            int temp = a + b;
            a = b;
            b = temp;
        }
        System.out.println("Sum of even Fibonacci numbers: " + sum);
    }
}
```

9)print from m to n skip k numbers

```
public class PrintNumbers {
    public static void main(String[] args) {
        int m = 1, n = 20, k = 3;
        for (int i = m; i <= n; i += k + 1) {
            System.out.print(i + " ");
        }
    }
}
```

10)matrix add

```
public class MatrixAddition {
    public static void main(String[] args) {
        int[][] matrix1 = {{1, 2}, {3, 4}};
        int[][] matrix2 = {{5, 6}, {7, 8}};
        int rows = matrix1.length;
        int cols = matrix1[0].length;
        int[][] result = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                result[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        for (int[] row : result) {
            for (int val : row) {
                System.out.print(val + " ");
            }
            System.out.println();
        }
    }
}
```

11)print rectangle symbol patt

```
public class RectanglePattern {
    public static void main(String[] args) {
        int rows = 5, cols = 7;
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print("* ");
       }
            System.out.println();
        }
    }
}
```

12)program to sort in alphabetical order acc or dec

```
import java.util.*;
public class SortAlphabeticalOrder {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] arr = {"Banana", "Apple", "Mango", "Orange", "Grapes"};
        System.out.print("Enter 1 for Ascending or 2 for Descending order: ");
        int choice = sc.nextInt();
        if (choice == 1) {
            Arrays.sort(arr);
        } else if (choice == 2) {
            Arrays.sort(arr, Collections.reverseOrder());
        }
        System.out.println("Sorted order: " + Arrays.toString(arr));
    }
}
```

13)print following pattern int=1,max=3

```
public class Main {
    public static void main(String[] args) {
        int n = 3;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

14)special char separate & print no of char

```
public class Main {
    public static void main(String[] args) {
        String str = "Hello@123!World#";
        int count = 0;
        for (char c : str.toCharArray()) {
            if (!Character.isLetterOrDigit(c)) {
                System.out.print(c + " ");
                count++;
            }
        }
        System.out.println("\nNumber of special characters: " + count);    }
}
```

15)print all composite num b/w a&b

```
public class Main {
    public static void main(String[] args) {
        int a = 10, b = 30;
        for (int i = a; i <= b; i++) {
            if (isComposite(i)) {
                System.out.print(i + " ");
            }
        }
    }
    public static boolean isComposite(int num) {
        if (num <= 1) return false;
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) return true;
        }
        return false;
    }
}
```

16)inverted full pyramid

```
public class Main {
    public static void main(String[] args) {
        int n = 5;
        for (int i = n; i >= 1; i--) {
            for (int j = 1; j <= n - i; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

17)mean,median,mode from array of num

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
public class Main {
    public static void main(String[] args) {
        int[] nums = {1, 2, 3, 3, 4, 5, 6, 6, 6};
        double mean = Arrays.stream(nums).average().orElse(0);
```

```
           Arrays.sort(nums);
        double median = (nums.length % 2 == 0) ? (nums[nums.length / 2 - 1] + nums[nums.length
/ 2]) / 2.0 : nums[nums.length / 2];
          Map<Integer, Integer> freqMap = new HashMap<>();
        for (int num : nums) {
            freqMap.put(num, freqMap.getOrDefault(num, 0) + 1);
        }
        int mode = nums[0];
        int maxCount = 0;
        for (Map.Entry<Integer, Integer> entry : freqMap.entrySet()) {
            if (entry.getValue() > maxCount) {
                mode = entry.getKey();
                maxCount = entry.getValue();
            }
        }
        System.out.println("Mean: " + mean);
        System.out.println("Median: " + median);
        System.out.println("Mode: " + mode);
    }
}
```

18)factorial of n

```
public class Main {
    public static void main(String[] args) {
        int n = 5;
        long factorial = 1;
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }
        System.out.println("Factorial of " + n + " is: " + factorial);
    }
}
```

19)find year is leap year on not

```
public class Main {
    public static void main(String[] args) {
        int year = 2024; // Example year
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
 }
}
```

20)no of factors for given number

```
public class Main {
    public static void main(String[] args) {
        int num = 12;
        int count = 0;
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                count++;
            }
        }
        System.out.println("Number of factors of " + num + " is: " + count);
    }
}
```

21)print given number is perfect

```
public class Main {
    public static void main(String[] args) {
        int num = 28;
        int sum = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        if (sum == num) {
            System.out.println(num + " is a perfect number.");
        } else {
            System.out.println(num + " is not a perfect number.");
        }
    }
}
```

22)no of vowels in given state

```
public class Main {
    public static void main(String[] args) {
        String statement = "Hello, how are you?";
        int count = 0;
        for (char c : statement.toLowerCase().toCharArray()) {
            if ("aeiou".indexOf(c) != -1) {
                count++;
            }
        }
        System.out.println("Number of vowels: " + count);
    }
}
```

23)print hollow square symbol pattern

```
public class Main {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                if (i == 1 || i == n || j == 1 || j == n) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");                }
            }
            System.out.println();
        }
    }
}
```

24)continents and vowels speriately

```
public class Main {
    public static void main(String[] args) {
        String input = "Hello World";
        String vowels = "";
        String consonants = "";
        for (char c : input.toLowerCase().toCharArray()) {
            if ("aeiou".indexOf(c) != -1) {
                vowels += c; // Append vowels
            } else if (Character.isLetter(c)) {
                consonants += c;
            }
        }
        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
    }
}
```

25)print swap cube

```
public class Main {
    public static void main(String[] args) {
        int a = 3, b = 4; // Example numbers
        System.out.println("Before swap: a = " + a + ", b = " + b);
        int temp = a;
        a = b;
        b = temp;
        System.out.println("After swap: a = " + a + ", b = " + b);
        System.out.println("Cube of a: " + (a * a * a));
        System.out.println("Cube of b: " + (b * b * b));
    }
}
```

26)given char is string or not

```
public class Main {
    public static void main(String[] args) {
        char ch = 'e';
        String str = "Hello World";
        if (str.indexOf(ch) != -1) {
            System.out.println(ch + " is in the string.");
        } else {
            System.out.println(ch + " is not in the string.");
        }
    }
}
```

27)armstrong number,sum of digits ,sq.root of numbers

```
public class Main {
    public static void main(String[] args) {
        int num = 153; // Example number for Armstrong check
        int sumOfDigits = 0, originalNum = num, digit;
                while (num != 0) {
            digit = num % 10;
            sumOfDigits += digit;
            num /= 10;
        }
           num = originalNum;
        int sum = 0, digits = String.valueOf(num).length();
        while (num != 0) {
            digit = num % 10;
            sum += Math.pow(digit, digits);
            num /= 10;
        }
        System.out.println("Sum of digits of " + originalNum + ": " + sumOfDigits);
        if (sum == originalNum) {
            System.out.println(originalNum + " is an Armstrong number.");
        } else {
            System.out.println(originalNum + " is not an Armstrong number.");
        }
              System.out.println("Square root of " + originalNum + ": " + Math.sqrt(originalNum));
    }
}
```