JavaPrepForNothing 🚀

The full document can be accessed through githuhb.com/DinethDilhara and for the direct access, the following link can be utilized.

https://github.com/DinethDilhara/java-prep-for-nothing

This repository contains a collection of over 50 Java questions along with their answers. These questions cover various fundamental topics in Java programming, including:

- Control statements 🛠️
- Repetitive loops 🔁
- Random and math methods 🎲
- Methods 📚
- Recursive functions 🔃
- Arrays 📊
- Input validation ✅
- String manipulation with the String class 🔤

Each question is designed to enhance your thought process and teach you how to approach problems with suitable logic. Whether you're a beginner looking to learn Java or an experienced developer seeking to refresh your skills, this repository provides valuable practice and insight into Java programming concepts.

Feel free to explore, practice, and improve your Java skills with these questions! 💡

1. Write a Java program to calculate the sum of the first 100 natural numbers. 🧮

2. Create a Java program to print the multiplication table of a given number. 🔄

3. Write a Java program to find the square root of a given number. √

4. Implement a Java program to calculate the factorial of a number without using recursion. ➕

5. Create a Java program to check whether a given year is a leap year or not. 📆

6. Write a Java program to find the GCD (Greatest Common Divisor) of two numbers. 🤝

7. Implement a Java program to swap two numbers without using a temporary variable. ↔

8. Create a Java program to check whether a given string is a palindrome or not (ignore case). 🔄

9. Write a Java program to count the number of vowels and consonants in a given string. 🔡

10. Implement a Java program to find the largest and smallest elements in an array. 📊

11. Create a Java program to reverse the elements of an array. 🔃

12. Write a Java program to calculate the factorial of a number using recursion. ➕

13. Implement a Java program to check whether a given number is a prime number or not. 🔢

14. Create a Java program to find the sum of all even numbers between 1 and 100. ➕

15. Write a Java program to generate a random number between 1 and 100. 🎲

16. Implement a Java program to check whether a given string is an anagram of another string. 🔄

17. Create a Java program to find the length of a string without using the length() method. 🔡

18. Write a Java program to remove all white spaces from a given string. 🚫

19. Implement a Java program to count the frequency of each character in a given string. 🔡

20. Create a Java program to find the average of an array of numbers. 📊

21. Write a Java program to check whether a given number is a perfect number or not. ✔

22. Implement a Java program to check whether a given string is a valid palindrome or not (considering case and punctuation). 🔄

23. Create a Java program to find the Fibonacci series up to n terms. 🔢

24. Write a Java program to reverse a sentence word by word. 🔁

25. Implement a Java program to find the second largest element in an array. 📊

26. Create a Java program to check whether a given number is an Armstrong number or not. 💪

27. Write a Java program to check whether a given char is a lowercase or uppercase. 🔡

28. Implement a Java program to find the reverse of a number. 🔢

29. Create a Java program to find the sum of digits of a given number. ➕

30. Print the first Armstrong number between 1-500. 💪

31. Create a Java program to find the sum of natural numbers divisible by 3 or 5 up to n. ➕

32. Implement a Java program to convert a decimal number to binary. 🔄

33. Write a Java program to print the Pascal's triangle up to n rows. ▲

34. Implement a Java program to find the longest word in a given sentence. 📏

35. Implement a Java program to find the sum of digits of a number without using loops. ➕

36. Implement a Java program to check whether a given string is a valid email address or not. ✉

37. Write a Java program to sort an array of strings in alphabetical order. 🔤

38. Write a Java program to remove duplicates from an array. 🚫

39. Write a Java program to check if two arrays are equal or not. ⚖

40. Find a given element's position in the array. 🔍

41. Write a Java program to simulate a dice roll. 🎲

42. Write a Java program to generate a random password of a specified length. 🔐

43. Validate Integer input between 1-20. ✅

44. Write a Java program to find the area and circumference of a circle given its radius. 🔄

45. Write a Java program to convert temperature from Celsius to Fahrenheit using a method. 🌡

46. Write a Java program to calculate a person's age in years, months, and days based on their birth date and the current date. 🎂

47. Write a Java program to calculate the power of a number using recursion. 🔢

48. Write a simple Java program for a random number game. 🎲

49. Given a password guessing game with only 4 attempts. 🔐

50. Write a Java program to perform matrix addition. 📊

Table of Contents

Q01 – Sum of First 100 integers

```
public class Q01_SumOf1st100 {
/*
    1. Write a Java program to calculate the sum of the first 100 natural
numbers.
*/
    public static void main(String[] args) {
        sumOfFirstDigits(100);
        sumOfFirstDigits(5);
    }

    static void sumOfFirstDigits(int n){
        int sum = 0;

        for (int i = 0; i <= n ; i++) {
            sum+=i;
        }

        System.out.printf("Sum of first %d number is %d %n",n,sum);
    }
}
```

Q02 - multiplication table of a given number.

```
public class Q02_MultiplicationTable {
/*
    2. Create a Java program to print the multiplication table of a given
number.
*/

    public static void main(String[] args) {
        calTable(10);
        System.out.println();
        calTable(1);
    }

    static void calTable(int n){
        for (int i = 0; i <= 12; i++) {
            System.out.printf("%d X %d = %d %n",i,n,(i*n));
        }
    }
}
```

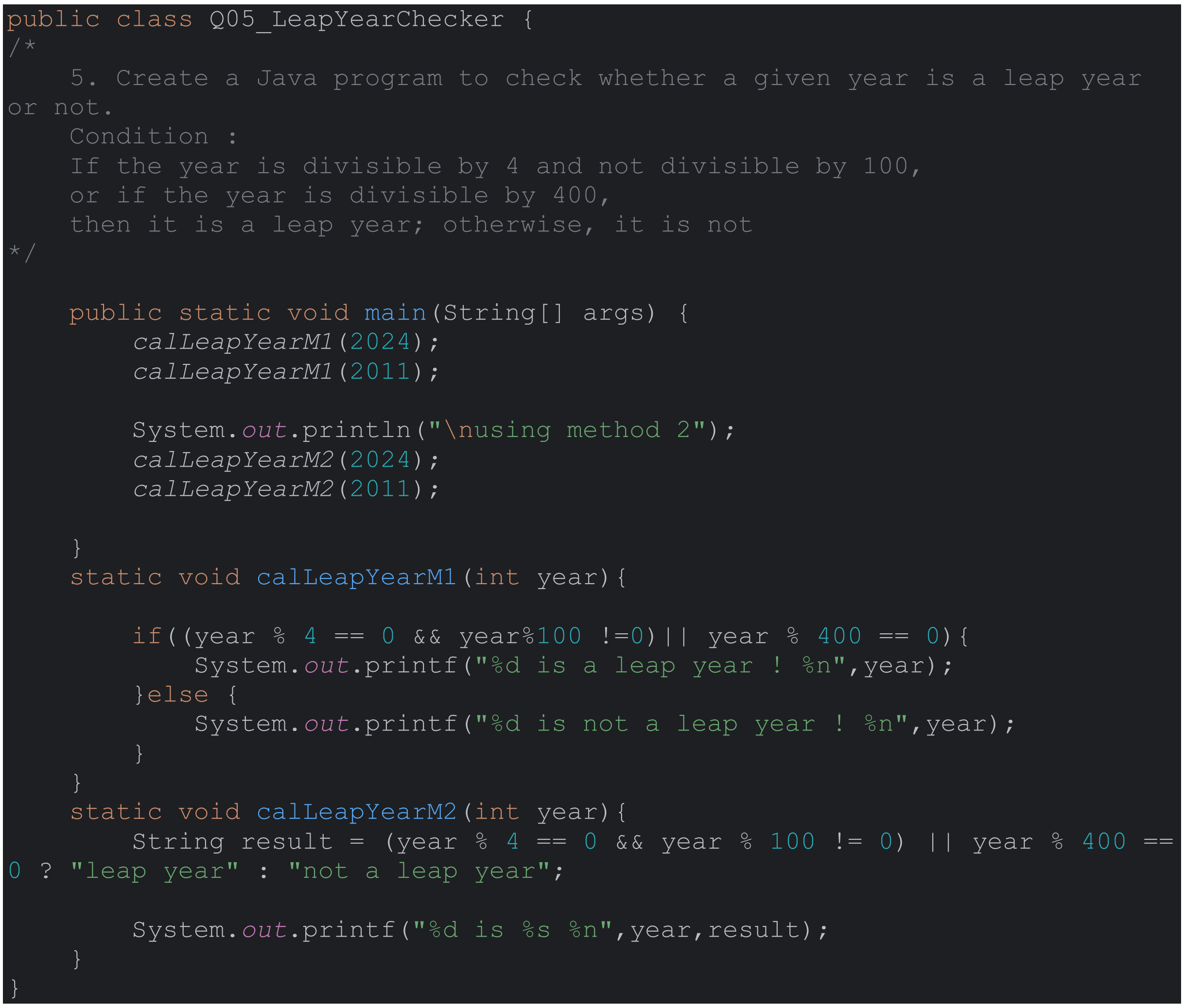
Q03 - Find the square root of a given number.

```
public class Q03_SquareRoot {
/*
    3. Write a Java program to find the square root of a given number.
*/
    public static void main(String[] args) {
        calSquareRoot(64);
        calSquareRoot(16);
    }
    static void calSquareRoot(int number){

        double ans = Math.sqrt(number);

        System.out.printf("square root of %d is %f %n",number,ans);
    }
}
```

Q04 - Calculate the factorial of a number.

```
public class Q04_FactorialWithoutRecursion {
/*
    4. Implement a Java program to calculate the factorial of a number
without using recursion.
*/
    public static void main(String[] args) {
        calFact(5);
        calFact(0);
    }
    static void calFact(int n){
        int ans = 1;

        for (int i = n; i >= 1 ; i--) {
            ans*=i;
        }
        System.out.printf("Factorial of %d = %d %n",n,ans);

    }
}
```

Q05 - Check whether a given year is a leap year or not.



```
public class Q05_LeapYearChecker {
/*
    5. Create a Java program to check whether a given year is a leap year
or not.
    Condition :
    If the year is divisible by 4 and not divisible by 100,
    or if the year is divisible by 400,
    then it is a leap year; otherwise, it is not
*/

    public static void main(String[] args) {
        calLeapYearM1(2024);
        calLeapYearM1(2011);

        System.out.println("\nusing method 2");
        calLeapYearM2(2024);
        calLeapYearM2(2011);

    }
    static void calLeapYearM1(int year){

        if((year % 4 == 0 && year%100 !=0)|| year % 400 == 0){
            System.out.printf("%d is a leap year ! %n",year);
        }else {
            System.out.printf("%d is not a leap year ! %n",year);
        }
    }
    static void calLeapYearM2(int year){
        String result = (year % 4 == 0 && year % 100 != 0) || year % 400 ==
0 ? "leap year" : "not a leap year";

        System.out.printf("%d is %s %n",year,result);
    }
}
```

Q06 - Greatest Common Divisor of two numbers.

```java
public class Q06_GreatestCommonDivisor {
/*
    6. Write a Java program to find the GCD (Greatest Common Divisor) of
two numbers.

    Condition :

    - If (b) is equal to 0, the GCD is (a).
    - Otherwise, recursively calculate the GCD of (b) and the remainder of
(a) divided by (b),
      until (b) becomes 0.
    - The GCD is the value obtained when \(b\) becomes 0."
*/
    public static void main(String[] args) {

        System.out.println(gcdM1(12,4));

        System.out.println(gcdM2(3,10));

    }
    // GCD of two numbers without recursion
     static int gcdM1(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    // GCD of two numbers using recursion
    static int gcdM2(int a, int b) {
        return b == 0 ? a : gcdM2(b, a % b);
    }
}
```

Q07 - Swap two numbers without using a temporary variable.

```
public class Q07_SwapTwoNumWithoutTemp {
/*
    7. Implement a Java program to swap two numbers without using a
temporary variable.
*/
    public static void main(String[] args) {
        swapTwoNum(10,5);
        System.out.println();
        swapTwoNum(2,4);
    }
    static void swapTwoNum(int num1, int num2){

        System.out.printf("Before Swap number 1 : %d and number 2 : %d
%n",num1,num2);

                             // num1 = 10  num2 = 5
        num1 = num1 + num2;  // num1 = 10 + 5 => num1 = 15
        num2 = num1 - num2;  // num2 = 15 - 5 => num2 = 10
        num1 = num1 - num2;  // num1 = 15 - 10 => num1 =5

        System.out.printf("After Swap number 1 : %d and number 2 : %d
%n",num1,num2);
    }
}
```

Q08 - Check whether a given string is a palindrome.

```
public class Q08_PalindromeCheckerIgnoreCase {
/*
    8. Create a Java program to check whether a given string is a
palindrome or not (ignore case).
*/
    public static void main(String[] args) {
        palindromeChecker("dineth");
        palindromeChecker("dinethhtenid");
    }
    static void palindromeChecker(String str){

        StringBuilder revStr = new StringBuilder(str);

        if(str.equalsIgnoreCase(String.valueOf(revStr.reverse()))){
            System.out.printf("%s is a palindrome word %n",str);
        }else {
            System.out.printf("%s is not a palindrome word %n",str);
        }
    }
}
```

Q09 - count the number of vowels in string.

```
public class Q09_VowelsCounter {
/*
    9. Write a Java program to count the number of vowels and consonants in
a given string.
*/
    static char[] vowels ={'A','E','I','O','U'};

    public static void main(String[] args) {
        vowelCounter("dineth");
        vowelCounter("aeiou");
    }
    static void vowelCounter(String str){
        int counter = 0;

        for (int i = 0; i < str.length(); i++) {
            for ( char j : vowels ){
                if(str.toUpperCase().charAt(i)==j){
                    counter +=1;
                }
            }
        }
        System.out.printf("%s has %d number of vowels %n",str,counter);
    }
}
```

Q10 - largest and smallest elements in an array.

```
public class Q10_LargestAndSmallestEleInArray {
/*
10. Implement a Java program to find the largest and smallest elements in
an array.
*/
    static int[] array = {1,2,3,4,5,6,7,8,9,10};

    public static void main(String[] args) {
        largestNumAndSmallestNum();
    }
    static void largestNumAndSmallestNum(){
        int bn = array[0];
        int sn = array[0];

        for (int i : array){
            if(bn < i){
                bn=i;
            }
            if(sn > i){
                sn=i;
            }
        }
        System.out.printf("Largest number is %d %n",bn);
        System.out.printf("Smallest number is %d %n",sn);
    }
}
```

Q11 - Reverse the elements of an array.

```
public class Q11_ReverseAnArray {
/*
    11. Create a Java program to reverse the elements of an array.
*/
    static int[] array = {1,2,3,4,5,6,7,8,9,10};

    public static void main(String[] args) {
        printReverse();
    }
    static void printReverse(){

        for (int i = array.length - 1; i >= 0; i--) {
            System.out.print(array[i]+ " ");
        }
    }
}
```

Q12 - Factorial of a number using recursion.

```
public class Q12_FactorialWithRecursion {
/*
    12. Write a Java program to calculate the factorial of a number using
recursion.
*/
public static void main(String[] args) {
        int n = 5;
        long fact = factorialWithRecursion(n);
        System.out.printf("Factorial of %d : %d %n",n,fact);
    }
    public static long factorialWithRecursion(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorialWithRecursion(n - 1);
        }
    }
}
```

Q13 - Check whether a given number is a prime

```
public class Q13_PrimeNumberChecker {
/*
    13. Implement a Java program to check whether a given number is a prime
number or not.
*/
    public static void main(String[] args) {

        int num = 11;
        boolean is = isPrime(num);
        primeChecker(is);
    }
     static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
    static void primeChecker(boolean is){
        if (is){
            System.out.println("is a prime number ");
        }else {
            System.out.println(" not a prime number");
        }
    }
}
```

Q14 - Sum of all even numbers between 1 and 100.

```
public class Q14_Sumof1st100Even {
/*
    14. Create a Java program to find the sum of all even numbers between 1
and 100.
*/
    public static void main(String[] args) {
        sumOfEven(100);
    }
    static void sumOfEven(int n) {
        int sum = 0;

        for (int i = 0; i <= n; i += 2) {
            sum += i;
        }
        System.out.println("Sum of even number between 1-100 : " + sum);
    }
}
```

Q15 – Generate a random number between 1 and 100

```java
import java.util.Random;

public class Q15_GenerateRandomNumber {
/*
    15. Write a Java program to generate a random number between 1 and 100.
*/
    public static void main(String[] args) {

    Random random = new Random();
    int randomNum = random.nextInt(100) + 1;

    System.out.println("random number between 1-100 is "+ randomNum);
   }
}
```

Q16 - string is an anagram of another String.

```java
import java.util.Arrays;

public class Q16_AnagramChecker {
/*
    16. Implement a Java program to check whether a given string is an anagram of another string.
*/
    public static void main(String[] args) {
        areAnagrams("sing","sign");

        areAnagrams("dineth","dilhara");
    }

     static void areAnagrams(String str1, String str2) {
        // Remove whitespace and convert strings to lowercase for case-insensitive comparison
        str1 = str1.replaceAll("\\s", "").toLowerCase();
        str2 = str2.replaceAll("\\s", "").toLowerCase();

        // Sort the characters in both strings
        char[] charArray1 = str1.toCharArray();
        char[] charArray2 = str2.toCharArray();
        Arrays.sort(charArray1);
        Arrays.sort(charArray2);

        boolean result =Arrays.equals(charArray1, charArray2);

        if (result) {
            System.out.println("\"" + str1 + "\" and \"" + str2 + "\" are anagrams.");
        } else {
            System.out.println("\"" + str1 + "\" and \"" + str2 + "\" are not anagrams.");
        }
    }




}
```

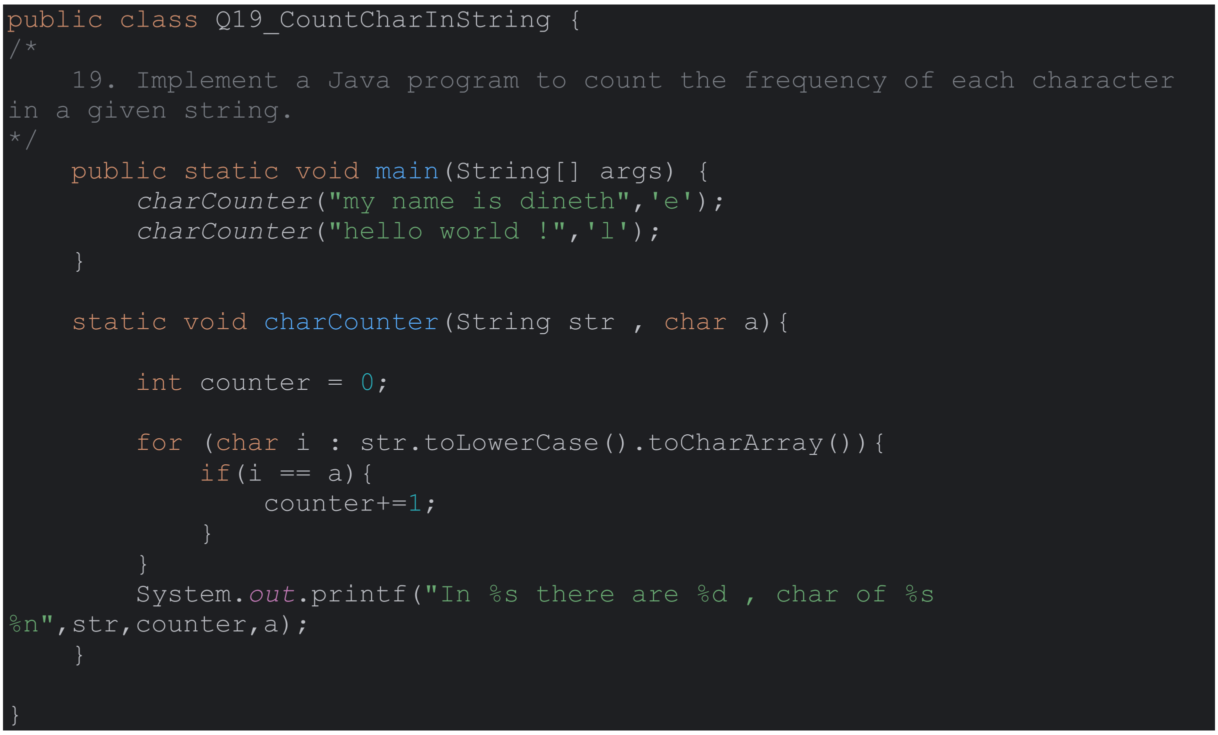
Q17 - Length of a string without using length () method

```
public class Q17_LengthOfStringWithoutLengthMethod {
/*
    17. Create a Java program to find the length of a string without using
the length() method.
*/
    static void findLength(String str) {
        int length = 0;

        for (char c : str.toCharArray()) {
            /*System.out.print(c +" ");*/
            length++;
        }
         System.out.println("Length of the string: " + length);     }

    public static void main(String[] args) {
        String str = "Dineth ";
        findLength(str);
    }
}
```

Q18 - Remove all white spaces from a given String.

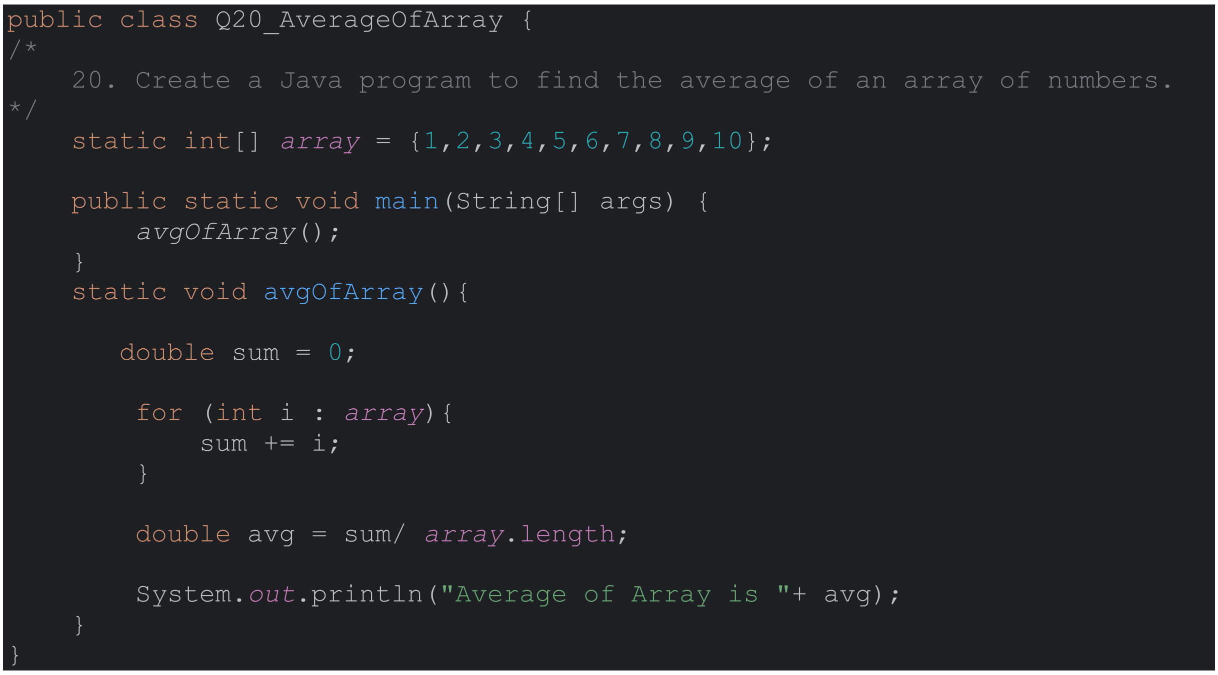
```
public class Q18_RemoveSpacesInString {
/*
    18. Write a Java program to remove all white spaces from a given
string.
*/
    public static void main(String[] args) {
        removeSpaces("my name is dineth dilhara");
        removeSpaces("remove spaces ");
    }
    static void removeSpaces(String str){
        String str1;

        str1 = str.replaceAll("\\s", "");

        System.out.println(str1);
    }
}
```

Q19 - count the frequency of each character in a string.



```
public class Q19_CountCharInString {
/*
    19. Implement a Java program to count the frequency of each character
in a given string.
*/
    public static void main(String[] args) {
        charCounter("my name is dineth",'e');
        charCounter("hello world !",'l');
    }

    static void charCounter(String str , char a){

        int counter = 0;

        for (char i : str.toLowerCase().toCharArray()){
            if(i == a){
                counter+=1;
            }
        }
        System.out.printf("In %s there are %d , char of %s
%n",str,counter,a);
    }

}
```

Q20 - find the average of an array of numbers.



```
public class Q20_AverageOfArray {
/*
    20. Create a Java program to find the average of an array of numbers.
*/
    static int[] array = {1,2,3,4,5,6,7,8,9,10};

    public static void main(String[] args) {
        avgOfArray();
    }
    static void avgOfArray(){

       double sum = 0;

        for (int i : array){
            sum += i;
        }

        double avg = sum/ array.length;

        System.out.println("Average of Array is "+ avg);
    }
}
```

Q21 - check a given number is a perfect number or not.

```
public class Q21_PerfectNumberChecker {
/*
    21. Write a Java program to check whether a given number is a perfect
number or not.
*/
    public static boolean isPerfectNumber(int num) {
        if (num <= 1) {
            return false;
        }

        int sum = 1;

        for (int i = 2; i * i <= num; i++) {
            if (num % i == 0) {
                sum += i;
                if (i != num / i) {
                    sum += num / i;
                }
            }
        }

        return sum == num;
    }

    public static void main(String[] args) {
        int num = 28;

        boolean result = isPerfectNumber(num);

        if (result) {
            System.out.println(num + " is a perfect number.");
        } else {
            System.out.println(num + " is not a perfect number.");
        }
    }
}
```

Q22 - check given string is valid palindrome or not (considering case)

```
public class Q22_PalindromeCheckerConsiderCase {
/*
    22. Implement a Java program to check whether a given string is a valid
palindrome or not (considering case and punctuation).
*/
    public static void main(String[] args) {

        palindromeChecker("dineth");
        palindromeChecker("dinethhtenid");
    }
    static void palindromeChecker(String str){

        StringBuilder revStr = new StringBuilder(str);

        if(str.equals(String.valueOf(revStr.reverse()))){
            System.out.printf("%s is a palindrome word %n",str);
        }else {
            System.out.printf("%s is not a palindrome word %n",str);
        }
    }
}
```

Q23 - Fibonacci series up to n terms

```
public class Q23_Fibonacci {
/*
    23. Create a Java program to find the Fibonacci series up to n terms.
*/
    public static void main(String[] args) {
        fibo(6);
    }
    static void fibo(int n){
        int f1 = 1;
        int f2 = 1;

        int NF;

        System.out.print(f1 + ",");
        System.out.print(f2 + ",");

        for (int i = 1; i <= (n - 2); i++) {
            NF = f1 + f2;
            System.out.print(NF + ",");
            f1 = f2;
            f2 = NF;
        }
        System.out.println("\b");
    }
}
```

Q24 – reverse a sentence word by word.

```
public class Q24_ReverseStringWordByWord {
/*
    24. Write a Java program to reverse a sentence word by word.
*/
    public static String reverseSentence(String sentence) {

        String[] words = sentence.split(" ");

        StringBuilder reversedSentence = new StringBuilder();

        for (int i = words.length - 1; i >= 0; i--) {
            reversedSentence.append(words[i]).append(" ");
        }

        return reversedSentence.toString().trim();
    }

    public static void main(String[] args) {

        String reversedSentence = reverseSentence("My Name Is Dineth");

        System.out.println("Reversed sentence: " + reversedSentence);
    }
}
```

Q25 - find the second largest element in an array.

```
public class Q25_SecondLargestInArray {
/*25. Implement a Java program to find the second-largest element in an
array.*/
    static int[] array = {1,2,3,4,5,6,7,8,9,10};

    public static void main(String[] args) {
        findSecondLargestNum();
    }
    static void findSecondLargestNum() {
        int largest = array[0];
        int secondLargest = array[0];

        for (int num : array) {
            if (num > largest) {
                secondLargest = largest;
                largest = num;
            } else if (num > secondLargest && num != largest) {
                secondLargest = num;
            }
        }
        if (secondLargest == largest) {
            System.out.println("There is no second largest number in the
array.");
        } else {
            System.out.println("The second largest number in the array is:
" + secondLargest);
        }
    }
}
```

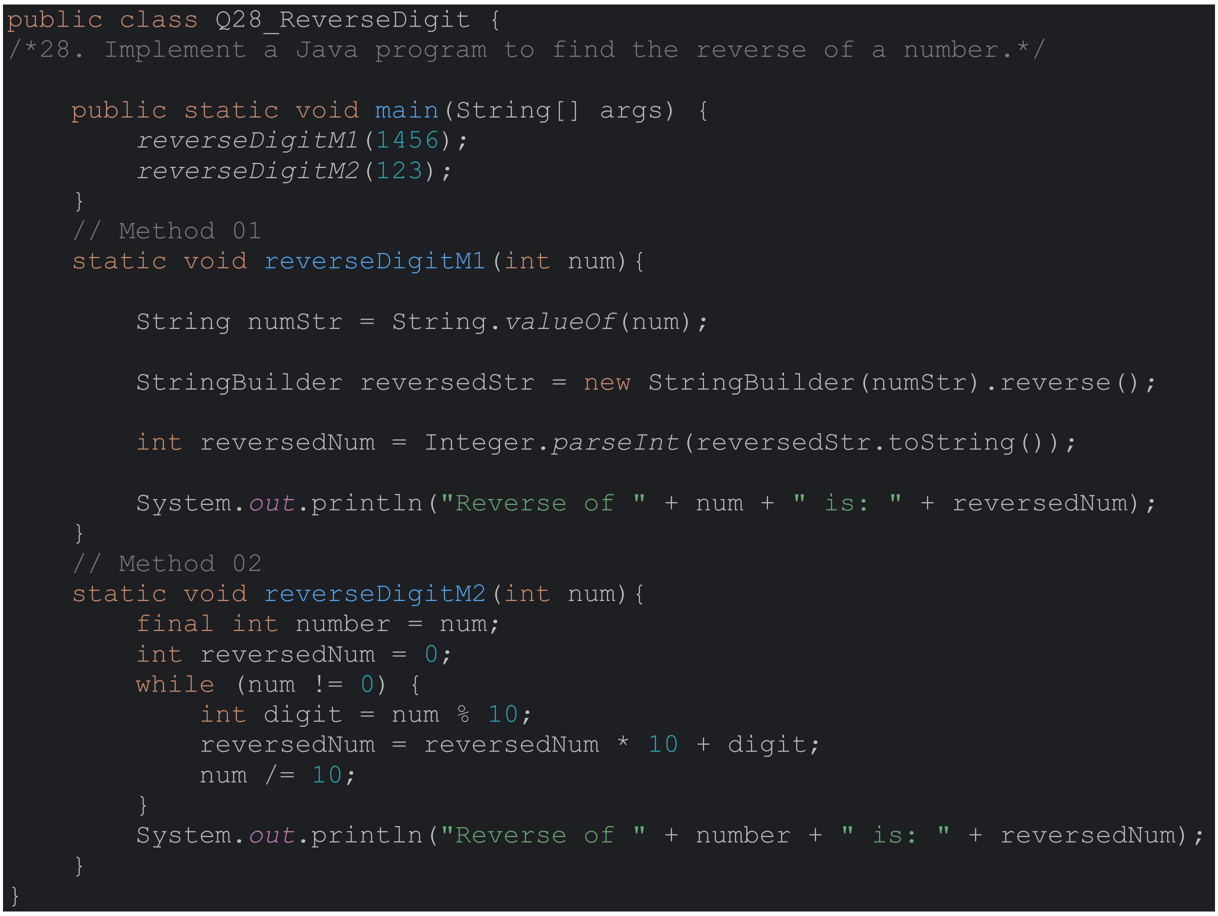
Q26 - Check given number is Armstrong number or not.

```
public class Q26_ArmstrongNumberChecker {

/*
    26. Create a Java program to check whether a given number is an
Armstrong number or not.
*/
    public static void main(String[] args) {
        armStrongChecker(16);
    }
    static void armStrongChecker(int i){
            double sumOfDigits =0;
            int num = i;
            int len = String.valueOf(num).length();

            while (num > 0) {
                int digit = num % 10;
                sumOfDigits += Math.pow(digit, len);
                num /= 10;
            }

            if (i==sumOfDigits){
                System.out.printf("%d is  an armstrong number",i);
            }else {
                System.out.printf("%d is not an armstrong number",i);
            }
    }
}
```

Q27 - check whether a given char is lowercase/uppercase.

```
public class Q27_CharCaseChecker {
/*27. Write a Java program to check whether a given char is a lowercase or
uppercase*/
    public static void main(String[] args) {
        caseChecker('A');
        caseChecker('v');
    }
    static void caseChecker(char Char){

        if (Char >= 'a' && Char <= 'z') {
            System.out.println("The entered character is lowercase.");
        } else if (Char >= 'A' && Char <= 'Z') {
            System.out.println("The entered character is uppercase.");
        } else {
            System.out.println("The entered character is not a valid
alphabet character.");
        }
    }
}
```

Q28 - find the reverse of a number.



```java
public class Q28_ReverseDigit {
/*28. Implement a Java program to find the reverse of a number.*/

    public static void main(String[] args) {
        reverseDigitM1(1456);
        reverseDigitM2(123);
    }
    // Method 01
    static void reverseDigitM1(int num){

        String numStr = String.valueOf(num);

        StringBuilder reversedStr = new StringBuilder(numStr).reverse();

        int reversedNum = Integer.parseInt(reversedStr.toString());

        System.out.println("Reverse of " + num + " is: " + reversedNum);
    }
    // Method 02
    static void reverseDigitM2(int num){
        final int number = num;
        int reversedNum = 0;
        while (num != 0) {
            int digit = num % 10;
            reversedNum = reversedNum * 10 + digit;
            num /= 10;
        }
        System.out.println("Reverse of " + number + " is: " + reversedNum);
    }
}
```

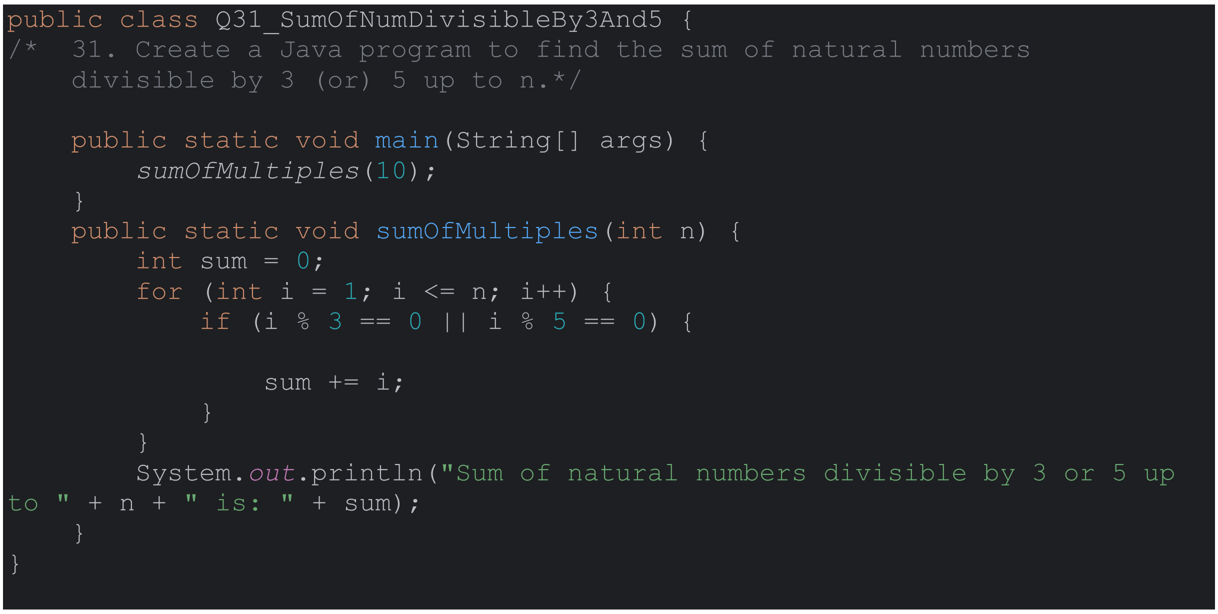
Q29 - find the sum of digits of a given number.

```
public class Q29_SumOfDigits {

    public static void main(String[] args) {
        sumOfDigits(123);
        sumOfDigits(111);
    }
    static void sumOfDigits(int num){
        final int number = num;
        int sumOfDigits = 0;

        while (num > 0) {
            int digit = num % 10;
            sumOfDigits += digit;
            num /= 10;
        }
        System.out.printf("Sum of digits of %d is %d
%n",number,sumOfDigits);
    }
}
```

Q30 - Print the first Armstrong number between 1-500

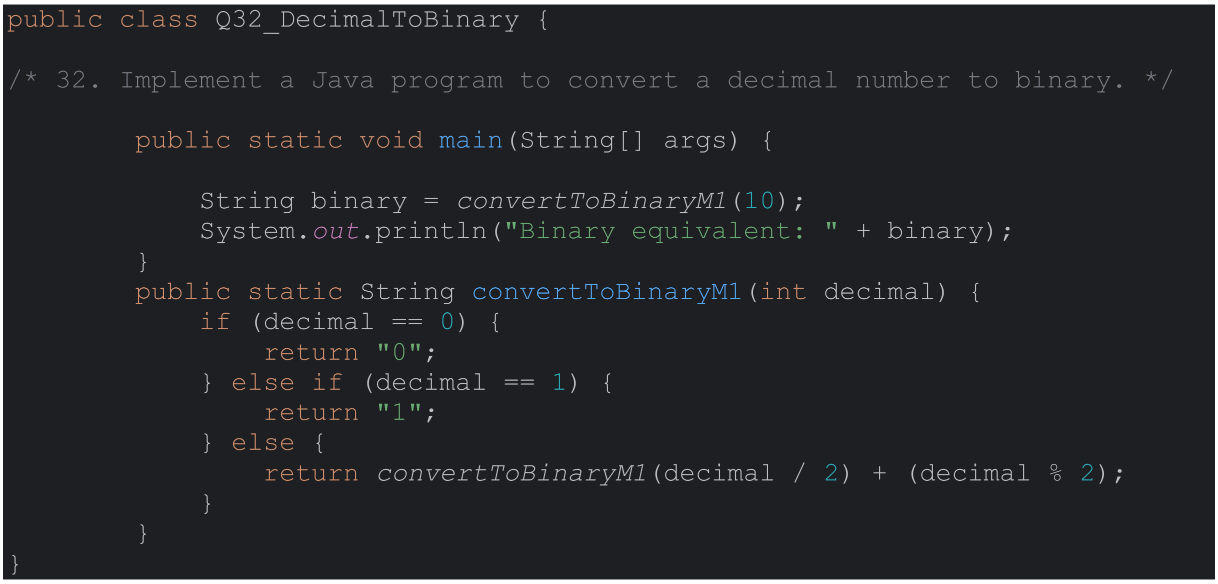
```
public class Q30_AmrstorngInFirst500 {
/*30. print first armstrong number between 1-500.*/

    public static void main(String[] args) {
        for (int i = 1; i <= 500; i++) {
            double sumOfDigits = 0;
            int num = i;
            int len = String.valueOf(num).length();
            while (num > 0) {
                int digit = num % 10;
                sumOfDigits += Math.pow(digit, len);
                num /= 10;
            }

            if (i == sumOfDigits) {
                System.out.println("this is a ars num " + i);
            }
        }
    }
}
```

Q31 - sum of natural numbers divisible by 3 or 5 up to n.



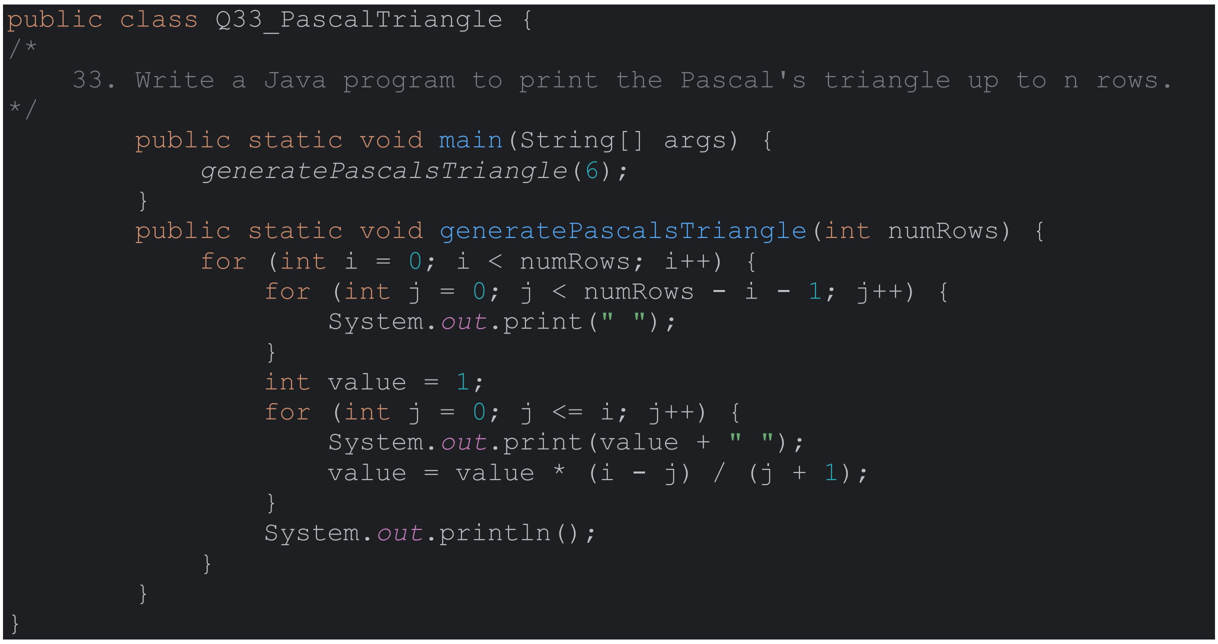
```java
public class Q31_SumOfNumDivisibleBy3And5 {
/*  31. Create a Java program to find the sum of natural numbers
    divisible by 3 (or) 5 up to n.*/

    public static void main(String[] args) {
        sumOfMultiples(10);
    }
    public static void sumOfMultiples(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            if (i % 3 == 0 || i % 5 == 0) {

                sum += i;
            }
        }
        System.out.println("Sum of natural numbers divisible by 3 or 5 up
to " + n + " is: " + sum);
    }
}
```

Q32 - convert a decimal number to binary.



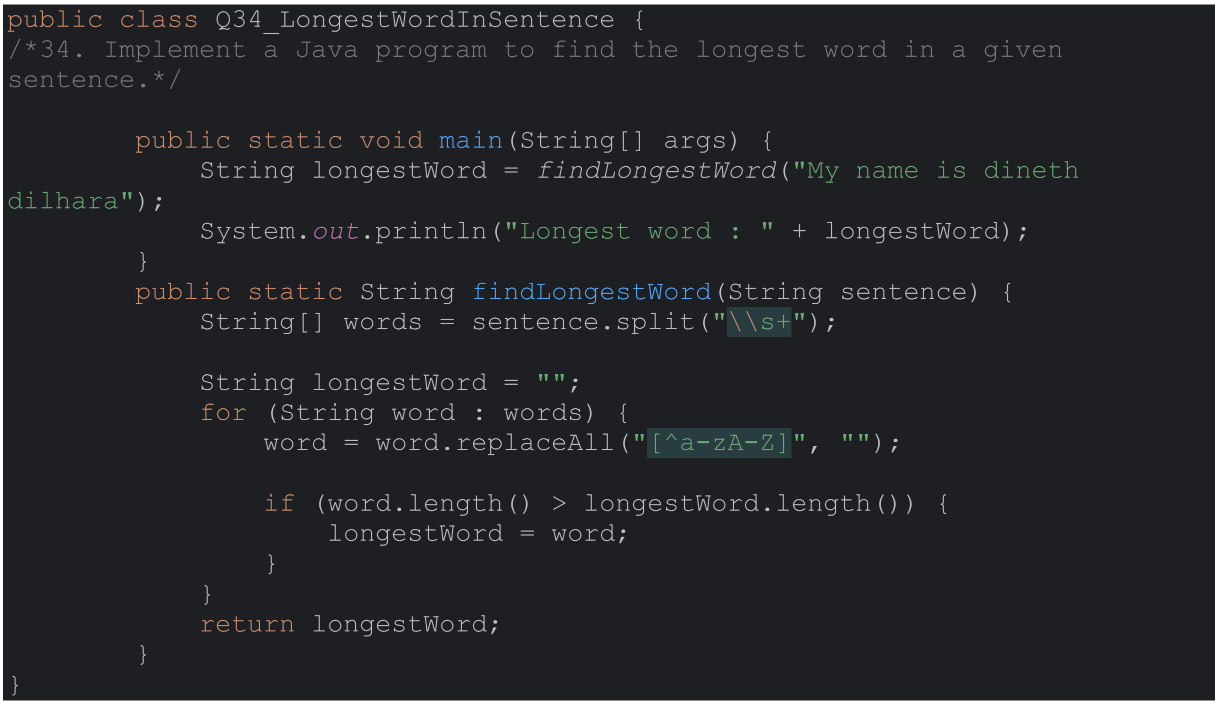
```java
public class Q32_DecimalToBinary {

/* 32. Implement a Java program to convert a decimal number to binary. */

        public static void main(String[] args) {

            String binary = convertToBinaryM1(10);
            System.out.println("Binary equivalent: " + binary);
        }
        public static String convertToBinaryM1(int decimal) {
            if (decimal == 0) {
                return "0";
            } else if (decimal == 1) {
                return "1";
            } else {
                return convertToBinaryM1(decimal / 2) + (decimal % 2);
            }
        }
}
```

Q33 - Pascal's triangle up to n rows



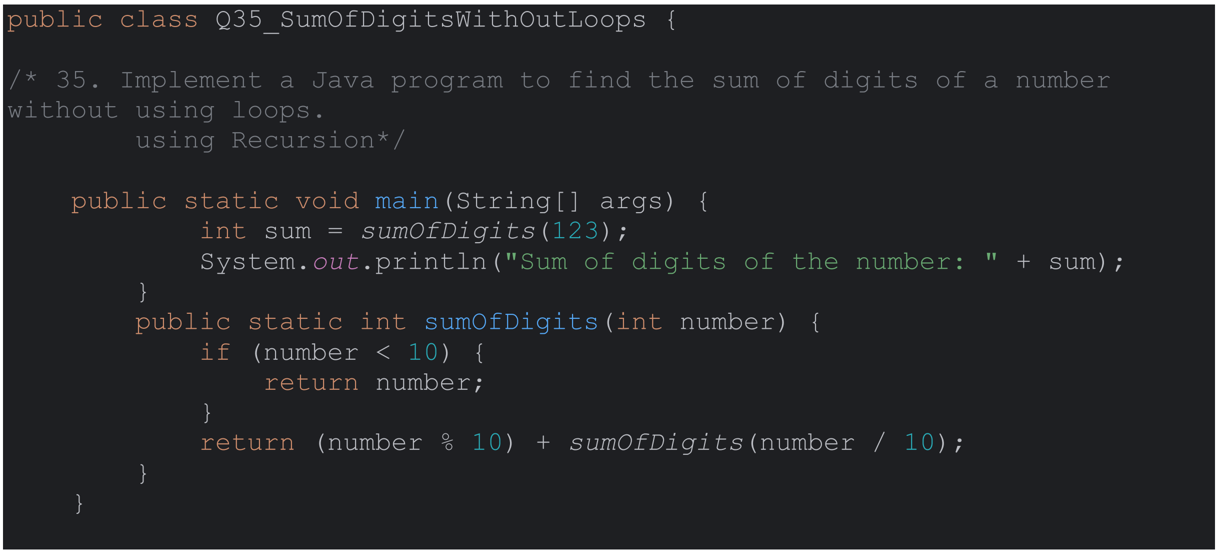
```
public class Q33_PascalTriangle {
/*
    33. Write a Java program to print the Pascal's triangle up to n rows.
*/
        public static void main(String[] args) {
            generatePascalsTriangle(6);
        }
        public static void generatePascalsTriangle(int numRows) {
            for (int i = 0; i < numRows; i++) {
                for (int j = 0; j < numRows - i - 1; j++) {
                    System.out.print(" ");
                }
                int value = 1;
                for (int j = 0; j <= i; j++) {
                    System.out.print(value + " ");
                    value = value * (i - j) / (j + 1);
                }
                System.out.println();
            }
        }
}
```

Q34 - find the longest word in sentence.



```
public class Q34_LongestWordInSentence {
/*34. Implement a Java program to find the longest word in a given
sentence.*/

        public static void main(String[] args) {
            String longestWord = findLongestWord("My name is dineth
dilhara");
            System.out.println("Longest word : " + longestWord);
        }
        public static String findLongestWord(String sentence) {
            String[] words = sentence.split("\\s+");

            String longestWord = "";
            for (String word : words) {
                word = word.replaceAll("[^a-zA-Z]", "");

                if (word.length() > longestWord.length()) {
                    longestWord = word;
                }
            }
            return longestWord;
        }
}
```

Q35 - sum of digits of a number without using loops.

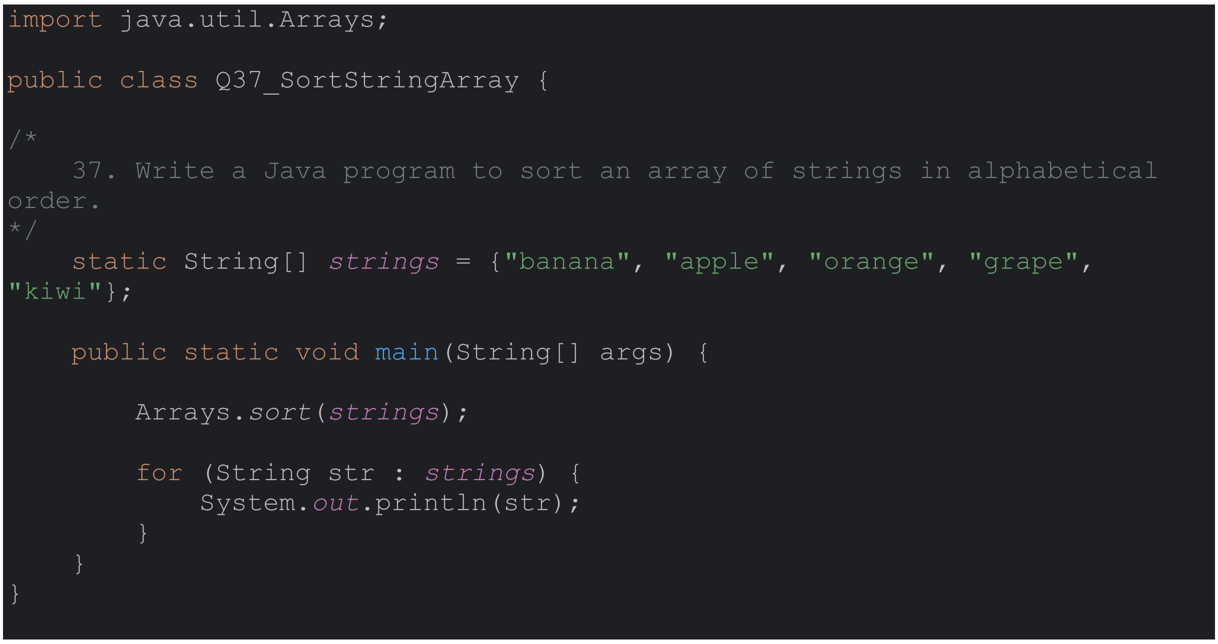


```java
public class Q35_SumOfDigitsWithOutLoops {

/* 35. Implement a Java program to find the sum of digits of a number
without using loops.
        using Recursion*/

    public static void main(String[] args) {
            int sum = sumOfDigits(123);
            System.out.println("Sum of digits of the number: " + sum);
        }
        public static int sumOfDigits(int number) {
            if (number < 10) {
                return number;
            }
            return (number % 10) + sumOfDigits(number / 10);
        }
    }
```

Q36 - check input is a valid email address or not.

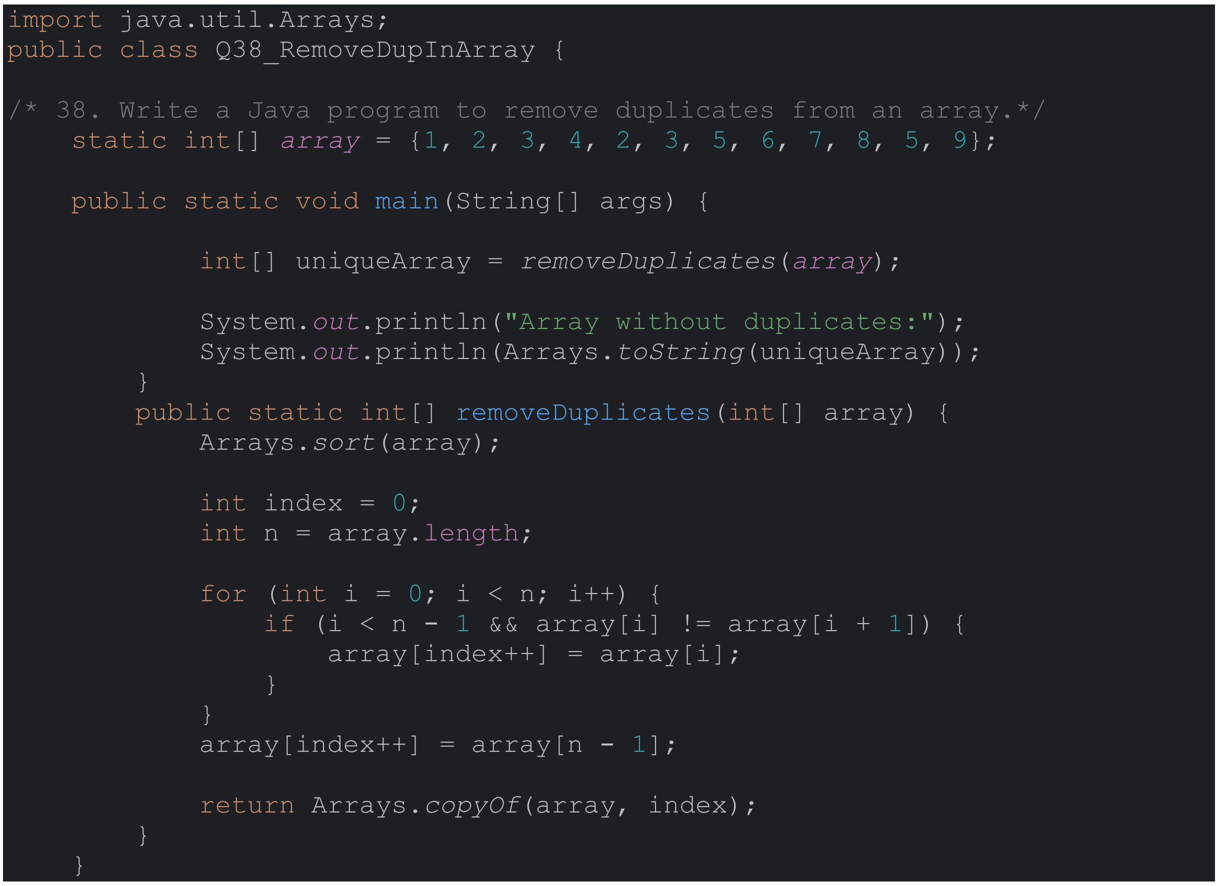
```
public class Q36_ValidEmail {

/*36. Implement a Java program to check whether a given string is a valid
email address or not.*/

        public static void main(String[] args) {
            //Method 01
            isValidEmailM1("dineth@yahoo.lk");
            isValidEmailM1("dinethdilhara");

            System.out.println();

            //Method 02
            isValidEmailM2("dineth@yahoo.lk");
            isValidEmailM2("dinethdilhara");
        }

    //Method 01
        public static void isValidEmailM1(String email) {

            boolean isValid = email.contains("@") && email.contains(".");

            if (isValid) {
                System.out.println("The email address \"" + email + "\" is
valid.");
            } else {
                System.out.println("The email address \"" + email + "\" is
not valid.");
            }

        }

    // Method 02
        public static void isValidEmailM2(String email) {
            String regex = "^[a-zA-Z0-9_+&*-]+(?:\\.[a-zA-Z0-9_+&*-
]+)*@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$";

            boolean isValid = email.matches(regex);
            if (isValid) {
                System.out.println("The email address \"" + email + "\" is
valid.");
            } else {
                System.out.println("The email address \"" + email + "\" is
not valid.");
            }
        }
}
```

Q37 - sort an array of strings in alphabetical order.



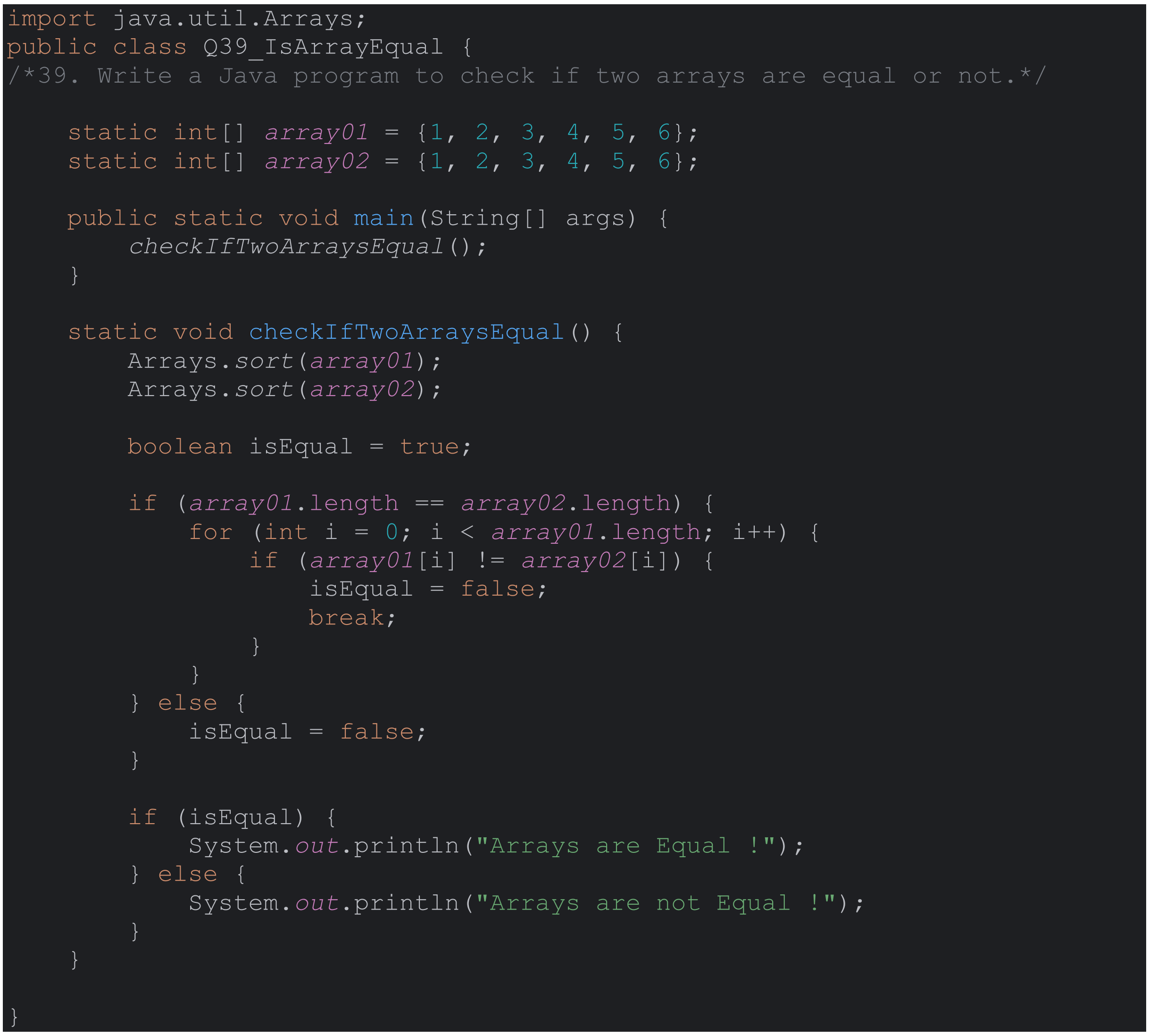
```java
import java.util.Arrays;

public class Q37_SortStringArray {

/*
    37. Write a Java program to sort an array of strings in alphabetical
order.
*/
    static String[] strings = {"banana", "apple", "orange", "grape",
"kiwi"};

    public static void main(String[] args) {

        Arrays.sort(strings);

        for (String str : strings) {
            System.out.println(str);
        }
    }
}
```

Q38 - remove duplicates from an array.



```java
import java.util.Arrays;
public class Q38_RemoveDupInArray {

/* 38. Write a Java program to remove duplicates from an array.*/
    static int[] array = {1, 2, 3, 4, 2, 3, 5, 6, 7, 8, 5, 9};

    public static void main(String[] args) {

            int[] uniqueArray = removeDuplicates(array);

            System.out.println("Array without duplicates:");
            System.out.println(Arrays.toString(uniqueArray));
        }
        public static int[] removeDuplicates(int[] array) {
            Arrays.sort(array);

            int index = 0;
            int n = array.length;

            for (int i = 0; i < n; i++) {
                if (i < n - 1 && array[i] != array[i + 1]) {
                    array[index++] = array[i];
                }
            }
            array[index++] = array[n - 1];

            return Arrays.copyOf(array, index);
        }
    }
```

Q39 - Check if two arrays are equal or not.



```java
import java.util.Arrays;
public class Q39_IsArrayEqual {
/*39. Write a Java program to check if two arrays are equal or not.*/

    static int[] array01 = {1, 2, 3, 4, 5, 6};
    static int[] array02 = {1, 2, 3, 4, 5, 6};

    public static void main(String[] args) {
        checkIfTwoArraysEqual();
    }

    static void checkIfTwoArraysEqual() {
        Arrays.sort(array01);
        Arrays.sort(array02);

        boolean isEqual = true;

        if (array01.length == array02.length) {
            for (int i = 0; i < array01.length; i++) {
                if (array01[i] != array02[i]) {
                    isEqual = false;
                    break;
                }
            }
        } else {
            isEqual = false;
        }

        if (isEqual) {
            System.out.println("Arrays are Equal !");
        } else {
            System.out.println("Arrays are not Equal !");
        }
    }

}
```

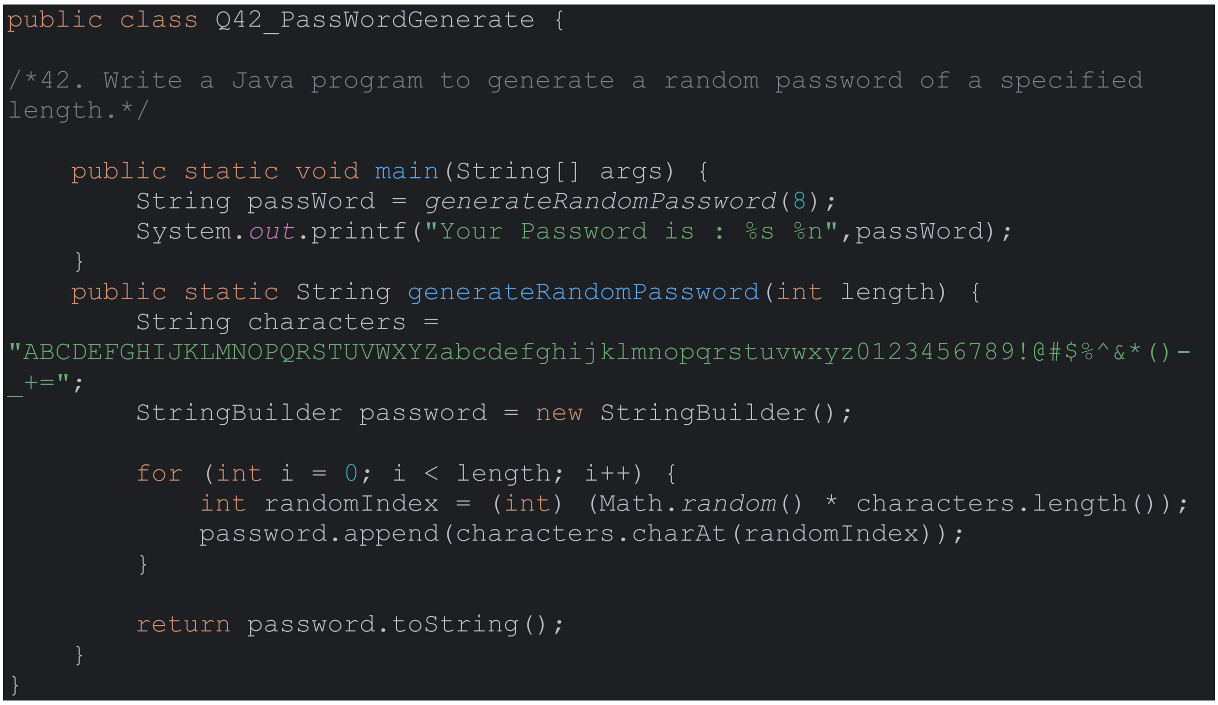
Q40 - Find a given element's position in the array.

```
public class Q40_FindElementInArray {
/*
    40. Find a given element's position in array
*/
    static int[] array = {1,2,3,4,5,6};

    public static void main(String[] args) {
        findElement(3);
        findElement(9);
    }

    static void findElement(int numToFind){

        boolean found = false;
        int position = -1;

        for (int i = 0; i < array.length; i++) {
            if (array[i] == numToFind) {
                found = true;
                position = i;
                break;
            }
        }

        if (found) {
            System.out.println("Element " + numToFind + " found at position
" + position);
        } else {
            System.out.println("Element " + numToFind + " not found in the
array");
        }
    }
}
```

Q41 - simulate a dice roll using random.

```
import java.util.Random;

public class Q41_DiceRollSimulate {
/*41. Write a Java program to simulate a diceroll.*/
    public static void main(String[] args) {
        diceRoll();
    }
    static void diceRoll(){

        Random random = new Random();

        int result = random.nextInt(1, 7);

        System.out.println("The result of the dice roll is: " + result);
    }
}
```

Q42 - generate a random password of a specified length.



```java
public class Q42_PassWordGenerate {

/*42. Write a Java program to generate a random password of a specified
length.*/

    public static void main(String[] args) {
        String passWord = generateRandomPassword(8);
        System.out.printf("Your Password is : %s %n",passWord);
    }
    public static String generateRandomPassword(int length) {
        String characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-
_+=";
        StringBuilder password = new StringBuilder();

        for (int i = 0; i < length; i++) {
            int randomIndex = (int) (Math.random() * characters.length());
            password.append(characters.charAt(randomIndex));
        }

        return password.toString();
    }
}
```

Q43 - Validate Integer input between 1-20.

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

public class Q43_ValidateIntegerInput {

/*43. Validate Integer input between 1-20 .*/

    public static void main(String[] args) {
        validateInput();
    }

    static void validateInput() {
        Scanner scanner = new Scanner(System.in);
        boolean isValid = false;
        int num = -1;

        while (!isValid) {
            try {
                System.out.print("Enter a number between 1 and 20:");
                num = scanner.nextInt();

                if (num > 0 && num <= 20) {
                    isValid = true;
                } else {
                    System.out.println("Number must be between 1 and 20.");
                }
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter an 
integer.");
                scanner.next();
            }
        }
        System.out.printf("input is : %d %n",num);
    }
}
```

Q44 - area and circumference of a circle

```
import java.util.Scanner;
public class Q44_AreaAndCircumferenceOfCircle {
/* 44. Write a Java program to find the area and circumference of a circle
given its radius. */

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

        System.out.print("Enter the radius of the circle: ");
        double radius = scanner.nextDouble();

        double area = calculateArea(radius);
        double circumference = calculateCircumference(radius);

        System.out.println("Area of the circle: " + area);
        System.out.println("Circumference of the circle: " +
circumference);

        scanner.close();
    }
    public static double calculateArea(double radius) {
        return Math.PI * radius * radius;
    }
    public static double calculateCircumference(double radius) {
        return 2 * Math.PI * radius;
    }
}
```

Q45 - convert temperature from Celsius to Fahrenheit

```
public class Q45_TempConverter {
/*45. Write a Java program to convert temperature from Celsius to
Fahrenheit using a method.*/

    public static void main(String[] args) {
        double fahrenheit = convertCelsiusToFahrenheit(36);
        System.out.println("Temperature in Fahrenheit: " + fahrenheit);
    }

    public static double convertCelsiusToFahrenheit(double celsius) {
        // Formula: (Celsius * 9/5) + 32
        return (celsius * 9 / 5) + 32;
    }
}
```

Q46 - Age calculater

```
import java.time.LocalDate;
import java.time.Period;
import java.util.Scanner;

public class Q46_AgeCalculater {

/*46. Write a Java program to calculate a person's age in years, months,
and days based on
        their birthdate and the current date.*/

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

        System.out.print("Enter your birth date (yyyy-mm-dd): ");
        String birthDateString = scanner.next();

        LocalDate birthDate = LocalDate.parse(birthDateString);

        LocalDate currentDate = LocalDate.now();

        Period period = Period.between(birthDate, currentDate);

        int years = period.getYears();
        int months = period.getMonths();
        int days = period.getDays();

        System.out.println("You are " + years + " years, " + months + "
months, and " + days + " days old.");

        scanner.close();
    }
}
```

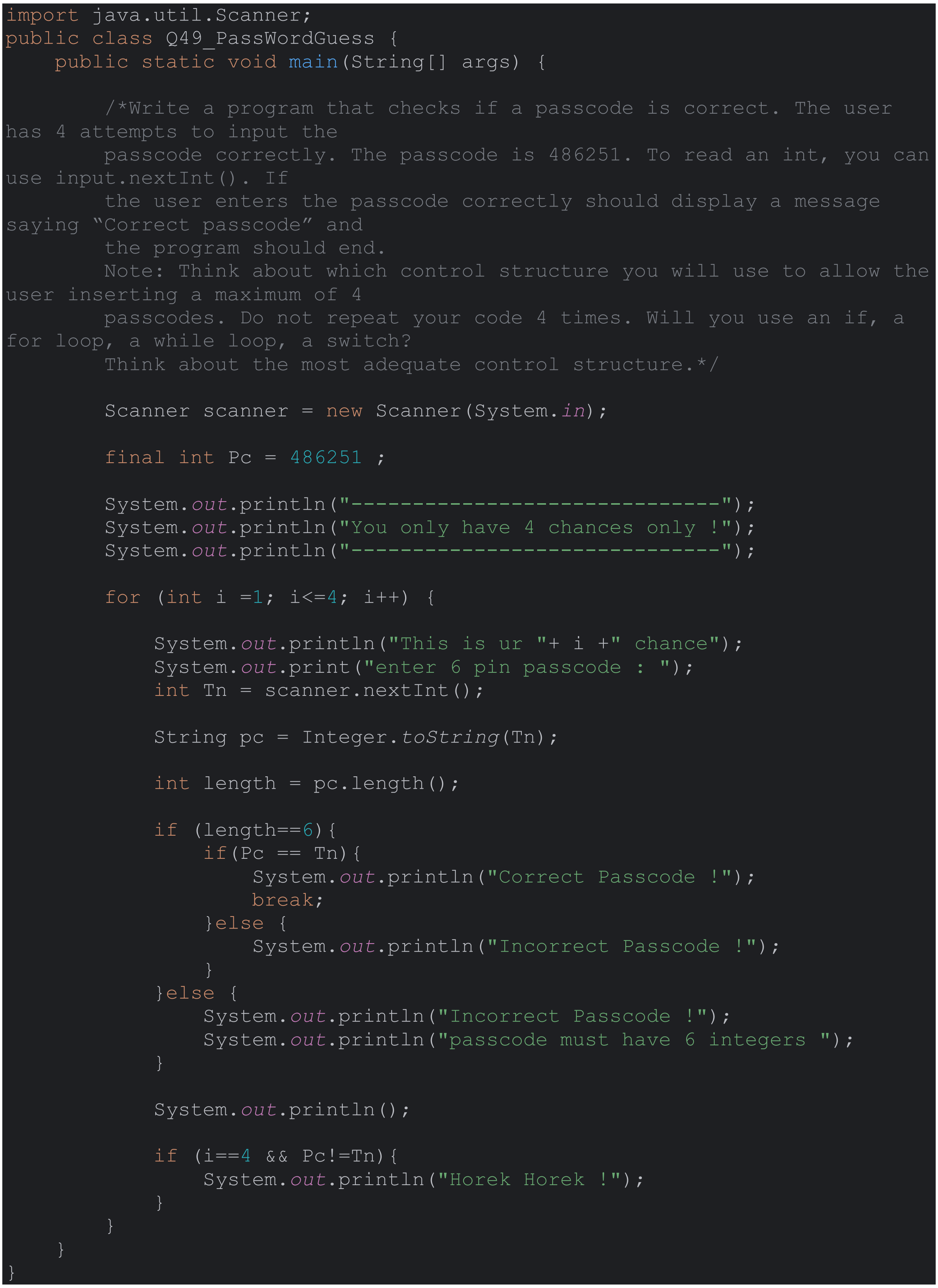
Q47 - calculate the power of a number using recursion.

```
public class Q47_PowerCalUsingRecursion {

/*47. Write a Java program to calculate the power of a number using
recursion.*/
    public static void main(String[] args) {

        double result = calculatePower(2, 3);
        System.out.println(result);
    }

    public static double calculatePower(double base, int exponent) {
        if (exponent == 0) {
            return 1;
        }
        if (exponent == 1) {
            return base;
        }
        if (exponent < 0) {
            return 1 / (base * calculatePower(base, -exponent - 1));
        }
        return base * calculatePower(base, exponent - 1);
    }
}
```

Q48 – Simple a random number game.

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

public class Q48_GuessRandomNumber {

    /* Write a program that generates a random number between 1 and 20 and
asks the user to guess
    the number. The user should be able to enter a new number if the number
is incorrect.
    Note: Use Random to generate a*/

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        Random random = new Random();
        int y = random.nextInt(20) + 1; // Random number between 1 and 20
        System.out.println("Random number generated: " + y);

        System.out.println("Enter your guess number between 1-20: ");
        int guessNumber = scanner.nextInt();

        while (y != guessNumber) {
            if (guessNumber < y) {
                System.out.println("Your guess is too low! Try again.");
            } else {
                System.out.println("Your guess is too high! Try again.");
            }
            System.out.println("Enter your guess number between 1-20: ");
            guessNumber = scanner.nextInt();
        }

        System.out.println("Congratulations! You guessed the correct
number.");
    }
}
```

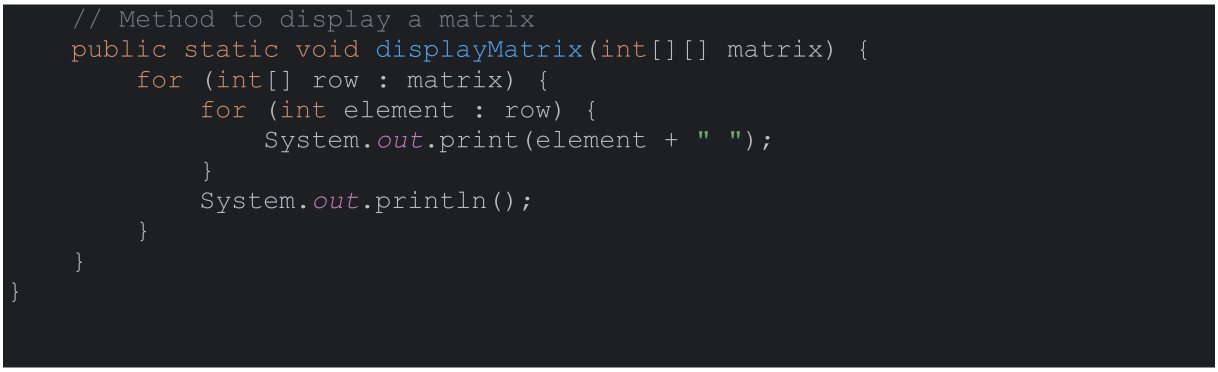
Q49 - guessing game with only 4 attempts.



```
import java.util.Scanner;
public class Q49_PassWordGuess {
    public static void main(String[] args) {

        /*Write a program that checks if a passcode is correct. The user
has 4 attempts to input the
        passcode correctly. The passcode is 486251. To read an int, you can
use input.nextInt(). If
        the user enters the passcode correctly should display a message
saying “Correct passcode” and
        the program should end.
        Note: Think about which control structure you will use to allow the
user inserting a maximum of 4
        passcodes. Do not repeat your code 4 times. Will you use an if, a
for loop, a while loop, a switch?
        Think about the most adequate control structure.*/

        Scanner scanner = new Scanner(System.in);

        final int Pc = 486251 ;

        System.out.println("------------------------------");
        System.out.println("You only have 4 chances only !");
        System.out.println("------------------------------");

        for (int i =1; i<=4; i++) {

            System.out.println("This is ur "+ i +" chance");
            System.out.print("enter 6 pin passcode : ");
            int Tn = scanner.nextInt();

            String pc = Integer.toString(Tn);

            int length = pc.length();

            if (length==6){
                if(Pc == Tn){
                    System.out.println("Correct Passcode !");
                    break;
                }else {
                    System.out.println("Incorrect Passcode !");
                }
            }else {
                System.out.println("Incorrect Passcode !");
                System.out.println("passcode must have 6 integers ");
            }

            System.out.println();

            if (i==4 && Pc!=Tn){
                System.out.println("Horek Horek !");
            }
        }
    }
}
```

Q50 - matrix addition

```java
import java.util.Scanner;

public class Q50_MatrixAddition {
    /* Write a Java program to perform matrix addition.  */

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

        System.out.print("Enter the number of rows in the matrices: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns in the matrices: ");
        int columns = scanner.nextInt();

        int[][] matrix1 = new int[rows][columns];
        int[][] matrix2 = new int[rows][columns];

        System.out.println("Enter the elements of the first matrix:");
        enterMatrixElements(scanner, matrix1);

        System.out.println("Enter the elements of the second matrix:");
        enterMatrixElements(scanner, matrix2);

        int[][] sumMatrix = addMatrices(matrix1, matrix2);

        System.out.println("Result of matrix addition:");
        displayMatrix(sumMatrix);

        scanner.close();
    }

    // Method to enter elements of a matrix
    public static void enterMatrixElements(Scanner scanner, int[][] matrix)
{
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                System.out.print("Enter element [" + (i + 1) + "][" + (j +
1) + "]: ");
                matrix[i][j] = scanner.nextInt();
            }
        }
    }
    // Method to perform matrix addition
    public static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {
        int rows = matrix1.length;
        int columns = matrix1[0].length;
        int[][] sumMatrix = new int[rows][columns];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        return sumMatrix;
    }
```



```java
    // Method to display a matrix
    public static void displayMatrix(int[][] matrix) {
        for (int[] row : matrix) {
            for (int element : row) {
                System.out.print(element + " ");
            }
            System.out.println();
        }
    }
}
```