

Loops Program

#1.) WAP to enter a number and calculate and print the sum of each digit present in the number using while loop only.

```
import java.util.Scanner;

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

        int sum = 0;

        while (number != 0) {
            sum += number % 10;
            number /= 10;
        }

        System.out.println("The sum of the digits is: " + sum);

        scanner.close();
    }
}
```

#2.) WAP to input a number to print the entered number is reverse order using while loop only.

```
import java.util.Scanner;

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

        int reverse = 0;

        while (number != 0) {
            int digit = number % 10;
            reverse = reverse * 10 + digit;
            number /= 10;
        }

        System.out.println("The reversed number is: " + reverse);

        scanner.close();
    }
}
```

#3.) WAP to enter a number. Check and print whether the entered number is automorphic number or not.

An automorphic number is a number whose square ends with the number itself. For example, 5 is an automorphic number because $5^2=25$ and 25 ends with 5.

```
import java.util.Scanner;

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

        int square = number * number;
        int temp = number;
        boolean isAutomorphic = true;

        while (temp > 0) {
            if (temp % 10 != square % 10) {
                isAutomorphic = false;
                break;
            }
            temp /= 10;
            square /= 10;
        }

        if (isAutomorphic) {
            System.out.println(number + " is an automorphic number.");
        } else {
            System.out.println(number + " is not an automorphic
number.");
        }

        scanner.close();
    }
}
```

#4.) WAP to enter a number. Print factors of the entered number.

```
import java.util.Scanner;

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

        System.out.println("The factors of " + number + " are:");
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                System.out.println(i);
            }
        }

        scanner.close();
    }
}
```

#4.) WAP to enter a number. Check and print whether the entered number is Armstrong number or not.

An Armstrong number (also known as a narcissistic number or pluperfect number) is a number that is equal to the sum of its own digits each raised to the power of the number of digits. For example, 153 is an Armstrong number because $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$.

```
import java.util.Scanner;

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

        // Calculate the number of digits
        int numDigits = String.valueOf(number).length();

        // Calculate the sum of the digits raised to the power of
numDigits
        int temp = number;
        int sum = 0;
        while (temp != 0) {
            int digit = temp % 10;
            sum += Math.pow(digit, numDigits);
            temp /= 10;
        }

        // Check if the sum is equal to the original number
        if (sum == number) {
            System.out.println(number + " is an Armstrong number.");
        } else {
            System.out.println(number + " is not an Armstrong
number.");
        }

        scanner.close();
    }
}
```

Pattern Program

Q.1) :-- 1 2 3 4 5

1 2 3 4 1

1 2 3 1 2

1 2 1 2 3

1 1 2 3 4

```
public class Main {  
    public static void main(String[] args) {  
        int n = 5; // size of the pattern  
  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n; j++) {  
                int num = j + 1;  
                if (num > n - i) {  
                    num = j - (n - i) + 1;  
                }  
                System.out.print(num + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Q.2) A B C D E
B C D E A
C D E A B
D E A B C
E A B C D

Note:-- Bhai Teacher ko yahi dikha dena actually vo output aayi hi nahi sakta, is liye us output se closely output ye aa raha hai. Iska code niche hai. 🙌

```
public class PatternProgram {  
    public static void main(String[] args) {  
        int rows = 5; // Number of rows in the pattern  
  
        // Outer loop for rows  
        for (int i = 0; i < rows; i++) {  
            char ch = (char) ('A' + i); // Starting  
character for each row  
  
            // Inner loop for columns  
            for (int j = 0; j < rows; j++) {  
                // Print characters  
                System.out.print(ch + " ");  
                // Update character for the next column  
                ch = (char) ((ch < 'E') ? ch + 1 : 'A');  
            }  
  
            // Move to the next line after each row  
            System.out.println();  
        }  
    }  
}
```