

## 1. write a program to swap two numbers in java ?

```
import java.util.Scanner;

public class SwapNumbers
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();
        System.out.println("Before swapping: ");
        System.out.println("First number: " + num1);
        System.out.println("Second number: " + num2);
        // Swap the numbers
        int temp=num1;
        num1=num2;
        num2=temp;
        System.out.println("After swapping: ");
        System.out.println("First number: " + num1);
        System.out.println("Second number: " + num2);
    }
}
```

OUTPUT:

### RESULT

```
Enter the first number: Enter the second number: Before swapping:
First number: 5
Second number: 10
After swapping:
First number: 10
Second number: 5
```

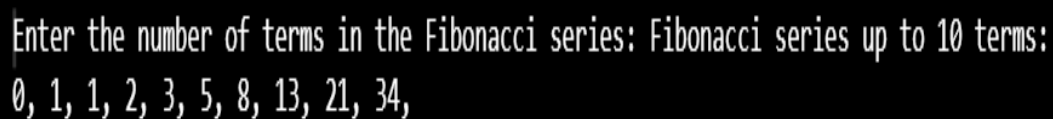
## 2. Write a program to print all the elements of the Fibonacci series.

```
import java.util.Scanner;

public class FibonacciSeries
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of terms in the Fibonacci series: ");
        int n = scanner.nextInt();
        System.out.println("Fibonacci series up to " + n + " terms:");
        for (int i = 0; i < n; i++)
        {
            System.out.print(fibonacci(i) + ", ");
        }
    }

    public static int fibonacci(int n)
    {
        if (n <= 1)
        {
            return n;
        }
        else
        {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }
}
```

OUTPUT:

A screenshot of a terminal window with a black background and white text. The text shows the program's output: "Enter the number of terms in the Fibonacci series: Fibonacci series up to 10 terms: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,". The input "10" is not visible, but the output sequence corresponds to the first 10 terms of the Fibonacci series starting from 0.

```
Enter the number of terms in the Fibonacci series: Fibonacci series up to 10 terms:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

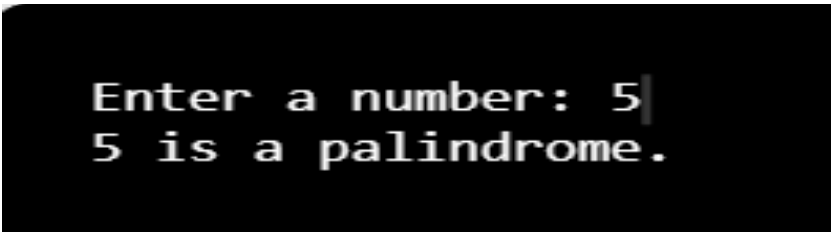
### 3. Check if a given number is palindrome or not.

```
import java.util.Scanner;

public class PalindromeCheck
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number: ");
        int number = scanner.nextInt();
        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 originalNumber = number;
        int reversedNumber = 0;
        while (number > 0)
        {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit; number /= 10;
        }
        return originalNumber == reversedNumber;
    }
}
```

OUTPUT:

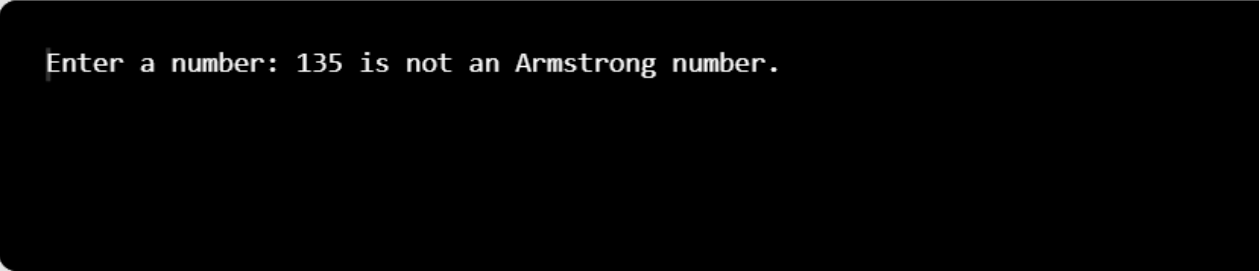
A screenshot of a terminal window with a black background and white text. The text shows the program's output for the input 5.

```
Enter a number: 5
5 is a palindrome.
```

**4. Write a program to find whether a number is an Armstrong number or not.**

```
import java.util.Scanner;
public class ArmstrongNumberCheck
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: "); int number = scanner.nextInt();
        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 = number;
            int result = 0;
            int power = String.valueOf(number).length();
            while (number > 0) {
                int digit = number % 10;
                result += Math.pow(digit, power);
                number /= 10;
            }
            return originalNumber == result;
        }
    }
}
```

**OUTPUT:**

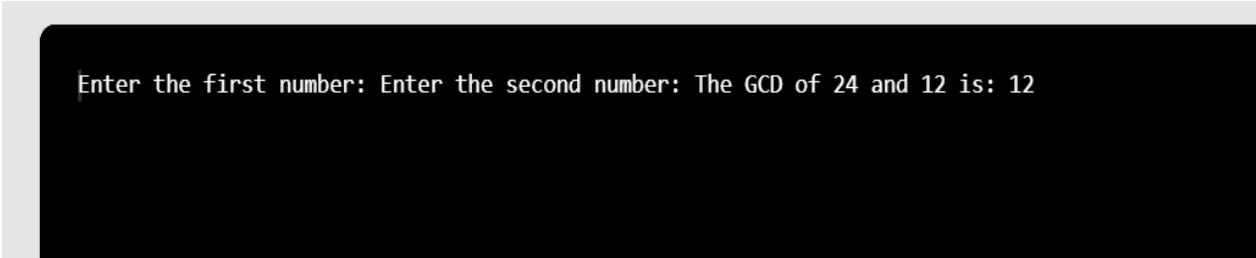
A screenshot of a terminal window with a black background and white text. The text shows the program's output for the input 135.

```
Enter a number: 135 is not an Armstrong number.
```

## 5. Find the GCD of two numbers

```
import java.util.Scanner;
public class GCDCalculator
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();
        int gcd = findGCD(num1, num2);
        System.out.println("The GCD of " + num1 + " and " + num2 + " is: " + gcd);
    }
    public static int findGCD(int a, int b)
    {
        while (b != 0)
        {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

### OUTPUT:

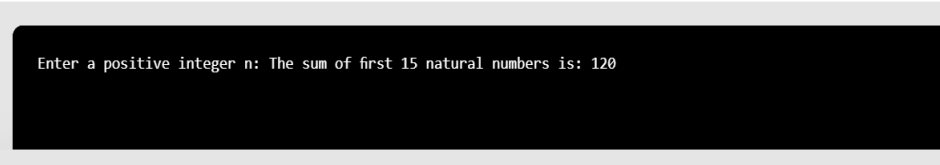
A screenshot of a terminal window with a black background and white text. The text shows the output of the program: "Enter the first number: Enter the second number: The GCD of 24 and 12 is: 12".

```
Enter the first number: Enter the second number: The GCD of 24 and 12 is: 12
```

6. **Write a program to find the sum of n natural numbers.**

```
import java.util.Scanner;
public class SumOfNaturalNumbers
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer n: ");
        int n = scanner.nextInt();
        int sum = 0;
        for (int i = 1; i <= n; i++)
        {
            sum += i;
        }
        System.out.println("The sum of first " + n + " natural numbers is: " + sum);
    }
}
```

OUTPUT:

A screenshot of a terminal window with a black background and white text. The text shows the program's output: "Enter a positive integer n: The sum of first 15 natural numbers is: 120".

Enter a positive integer n: The sum of first 15 natural numbers is: 120

7. **Write a program to find the lcm of two numbers.**

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

```
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the first number: ");
int num1 = scanner.nextInt();
System.out.print("Enter the second number: ");
int num2 = scanner.nextInt();
int lcm = findLCM(num1, num2);
System.out.println("The LCM of " + num1 + " and " + num2 + " is: " + lcm);
}
public static int findLCM(int a, int b)
{
    int max = Math.max(a, b);
    while (true) {
        if (max % a == 0 && max % b == 0)
        {
            return max;
        }
        max++;
    }
}
}
```

OUTPUT:

A screenshot of a Java program's output. The text "Enter the first number: Enter the second number: The LCM of 12 and 24 is: 24" is displayed on a black background with a light gray border. The text is in a monospaced font, and the numbers 12 and 24 are highlighted in yellow.

```
Enter the first number: Enter the second number: The LCM of 12 and 24 is: 24
```

### 8. Calculate the sum of digits of a given number.

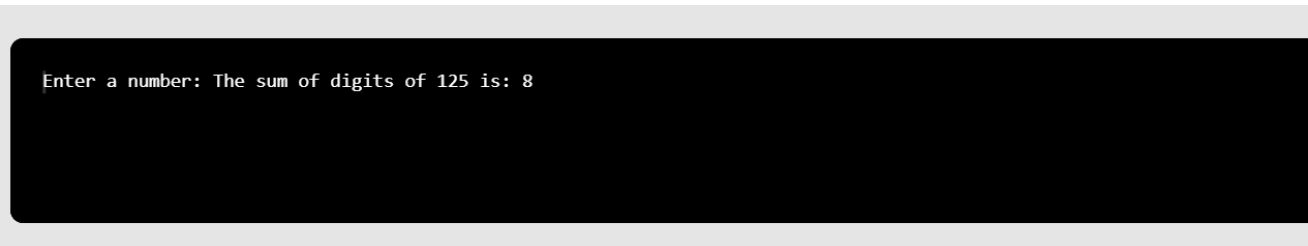
```
import java.util.Scanner;

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();
        int sum = calculateSumOfDigits(number);
        System.out.println("The sum of digits of " + number + " is: " + sum);
    }

    public static int calculateSumOfDigits(int num)
    {
        int sum = 0;
        while (num > 0) {
            sum += num %
            10; num /= 10;
        }
        return sum;
    }
}
```

OUTPUT:

A screenshot of a terminal window with a black background and white text. The text shows the program's output: "Enter a number: The sum of digits of 125 is: 8".

```
Enter a number: The sum of digits of 125 is: 8
```

### 9. Write a program to reverse a string.

```
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: ");
```

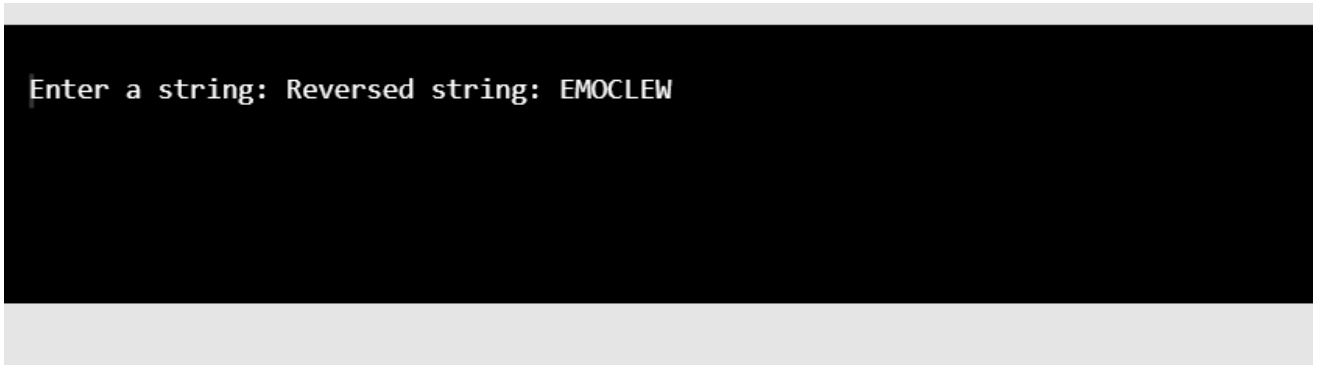


```

    String input = scanner.nextLine();
    String reversedString = reverseString(input);
    System.out.println("Reversed string: " + reversedString);
}
public static String reverseString(String str)
{
    StringBuffer stringBuffer = new
    StringBuffer(str);return
    stringBuffer.reverse().toString();
}
}

```

OUTPUT:



```

Enter a string: Reversed string: EMOCLEW

```

**10. Write a code to print all the first n prime numbers where n will be given as input.**

```

import java.util.Scanner;
public class FirstNPrimes
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of n: ");
        int n = scanner.nextInt();
        System.out.println("The first " + n + " prime numbers are:");
        printFirstNPrimes(n);
    }
    public static void printFirstNPrimes(int n)
    {
        int count = 0;
        int num = 2;

```

```

while (count < n)
{
    if (isPrime(num))
    {
        System.out.print(num + " ");
        count++;
    }
    num++;
}

public 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;
}
}

```

OUTPUT:

```

Enter the value of n: The first 25 prime numbers are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

```