

**Program 1:**

```
import java.util.Scanner;
public class SeriesGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        int n = scanner.nextInt();
        printSeries(a, b, n);
    }
    private static void printSeries(int a, int b, int n) {
        int result = a;
        for (int i = 0; i < n; i++) {
            result += Math.pow(2, i) * b;
            System.out.print(result + " ");
        }
    }
}
```

**Program 2:**

```
import java.util.Scanner;
public class ReverseNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();
        int reversedNumber = reverseNumber(number);
        System.out.println("Reverse: " + reversedNumber);
    }
    private static int reverseNumber(int num) {
        int reversed = 0;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        return reversed;
    }
}
```

**Program 3:**

```
import java.util.Scanner;
public class DecimalToBinary {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");
        int x = scanner.nextInt();
        String binary = decimalToBinary(x);
        System.out.println("Binary representation: " + binary);
    }
    private static String decimalToBinary(int decimal) {
        if (decimal <= 0) {
            return "Invalid input. Please enter a positive integer.";
        }
        StringBuilder binary = new StringBuilder();
        while (decimal > 0) {
            int remainder = decimal % 2;
            binary.insert(0, remainder);
            decimal /= 2;
        }
        return binary.toString();
    }
}
```

**Program 4:**

```
import java.util.Scanner;
public class BusTicketBooking {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the From Stage Number: ");
        int fromStage = scanner.nextInt();
        System.out.print("Enter the To Stage Number: ");
        int toStage = scanner.nextInt();
        System.out.print("Enter the number of Adult passengers: ");
        int adultPassengers = scanner.nextInt();
        System.out.print("Enter the number of Children passengers: ");
        int childPassengers = scanner.nextInt();
        int numStages = Math.abs(toStage - fromStage);
        int adultCost = 10 * adultPassengers * numStages;
        int childCost = 5 * childPassengers * numStages;
        int totalTicketCost = adultCost + childCost;
        double discount = 0.0;
        if (adultPassengers >= 5) {
            discount = 0.20;
        } else if (adultPassengers == 4) {
            discount = 0.15;
        } else if (adultPassengers == 3) {
            discount = 0.10;
        } else if (adultPassengers == 2) {
            discount = 0.05;
        }
        double discountedTicketCost = totalTicketCost - (discount * totalTicketCost);
        double serviceCharge = 0.05 * discountedTicketCost;
        double totalCost = discountedTicketCost + serviceCharge;
        System.out.println("\nTicket Details:");
        System.out.println("Number of Stages: " + numStages);
        System.out.println("Adult Cost: Rs." + adultCost);
        System.out.println("Child Cost: Rs." + childCost);
        System.out.println("Total Ticket Cost: Rs." + totalTicketCost);
        System.out.println("Discount Applied: " + (discount * 100) + "%");
        System.out.println("Discounted Ticket Cost: Rs." + discountedTicketCost);
        System.out.println("Service Charge: Rs." + serviceCharge);
        System.out.println("Total Cost: Rs." + totalCost);
    }
}
```

**Program 5:**

```
import java.util.Scanner;
public class NearestPrime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer X: ");
        int X = scanner.nextInt();
        int nearestPrime = findNearestPrime(X);
        System.out.println("Nearest Prime Number: " + nearestPrime);
    }
    private static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
    private static int findNearestPrime(int X) {
        if (X <= 1) {
            return 2;
        }
        int lower = X - 1;
        int upper = X + 1;

        while (true) {
            if (isPrime(lower)) {
                return lower;
            } else if (isPrime(upper)) {
                return upper;
            } else {
                lower--;
                upper++;
            }
        }
    }
}
```

**Program 6:**

```
import java.util.Scanner;
public class PrimeDigitSum {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.print("Enter a positive integer (or enter 0 to exit): ");
            int num = scanner.nextInt();

            if (num == 0) {
                System.out.println("Exiting the program.");
                break;
            }
            int result = primeDigitSum(num);
            boolean isResultPrime = isPrime(result);
            System.out.println("Sum of prime digits: " + result);
            System.out.println("Is the sum prime? " + isResultPrime);
        }
        scanner.close();
    }
    public static int primeDigitSum(int X) {
        int sum = 0;
        while (X > 0) {
            int digit = X % 10;
            if (isPrime(digit)) {
                sum += digit;
            }
            X /= 10;
        }
        return sum;
    }
    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;
    }
}
```

**Program 7:**

```
import java.util.Scanner;
public class NearestArmstrong {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer X: ");
        int X = scanner.nextInt();
        if (X <= 0) {
            System.out.println("Please enter a positive integer.");
        } else {
            int nearestArmstrong = findNearestArmstrong(X);
            System.out.println("The nearest Armstrong number to " + X + " is: " + nearestArmstrong);
        }
        scanner.close();
    }
    public static boolean isArmstrong(int num) {
        int originalNum = num;
        int sum = 0;
        int numberOfDigits = (int) Math.log10(num) + 1;
        while (num > 0) {
            int digit = num % 10;
            sum += Math.pow(digit, numberOfDigits);
            num /= 10;
        }
        return originalNum == sum;
    }
    public static int findNearestArmstrong(int X) {
        int lower = X - 1;
        int upper = X + 1;
        while (true) {
            if (isArmstrong(lower)) {
                return lower;
            } else if (isArmstrong(upper)) {
                return upper;
            }
            lower--;
            upper++;
        }
    }
}
```

**Program 8:**

```
import java.util.Scanner;
public class FibPrime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the lower bound X: ");
        int X = scanner.nextInt();
        System.out.print("Enter the upper bound Y: ");
        int Y = scanner.nextInt();
        if (X <= 0 || Y <= 0 || X > Y) {
            System.out.println("Please enter valid positive integers with X <= Y.");
        } else {
            printFibonacciPrimes(X, Y);
        }
        scanner.close();
    }
    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;
    }
    public static boolean isFibonacci(int num) {
        int a = 0, b = 1;
        while (a <= num) {
            if (a == num) {
                return true;
            }
            int temp = a;
            a = b;
            b = temp + b;
        }
        return false;
    }
    public static void printFibonacciPrimes(int X, int Y) {
        boolean found = false;
        for (int i = X; i <= Y; i++) {
            if (isFibonacci(i) && isPrime(i)) {
                System.out.print(i + " ");
            }
        }
    }
}
```

```
        found = true;
    }
}
if (!found) {
    System.out.println("0");
}
}
}
```



**Program 9:**

```
import java.util.Scanner;
public class FibPrime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer X: ");
        int X = scanner.nextInt();

        if (X <= 0) {
            System.out.println("Please enter a positive integer.");
        } else {
            int result = findFibonacciPrime(X);
            System.out.println("Result: " + result);
        }
        scanner.close();
    }
    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;
    }
    public static boolean isFibonacci(int num) {
        int a = 0, b = 1;
        while (a <= num) {
            if (a == num) {
                return true;
            }
            int temp = a;
            a = b;
            b = temp + b;
        }
        return false;
    }
    public static int findFibonacciPrime(int X) {
        int a = 0, b = 1;
        while (true) {
            int c = a + b;
            if (c >= X) {
```

```
        if (isPrime(c)) {  
            return c;  
        }  
    }  
    a = b;  
    b = c;  
}  
}
```

