

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

```
Scanner input = new Scanner(System.in)
```

1. Write a Program to find if the pair of numbers entered by the user is "amicable"? Steps to find amicable numbers:
 - Read or initialize two numbers (N1, N2) from the user.
 - Find the divisor (divOfN1, divOfN2) of both numbers.
 - Find the sum of divisors (sumdiv1, sumdiv2) of both numbers.
 - Check if the sum of divisors of one number is equal to the other number or not.
 - If equal, the pair of numbers is amicable.
 - Else, not amicable.

```
System.out.println("Enter first number: ");
```

```
int N1 = input.nextInt();
```

```
System.out.println(N1);
```

```
System.out.println("Enter second number: ");
```

```
int N2 = input.nextInt();
```

```
System.out.println(N2);
```

```
int sumdiv1 = 0;
```

```
int sumdiv2 = 0;
```

```
for (int i = 1; i < N1; i++) {  
    if (N1%i == 0) {  
        sumdiv1 += i;  
    }  
}
```

```
for (int i = 1; i < N2; i++) {  
    if (N2%i == 0) {  
        sumdiv2 += i;  
    }  
}
```

```
if (sumdiv1 == N2 && sumdiv2 == N1) {  
    System.out.println("The numbers are amicable.");  
} else {  
    System.out.println("The numbers are not amicable.");  
}
```

```
input.close();
```

```
Enter first number:
```

```
220
```

```
Enter second number:
```

```
284
```

```
The numbers are amicable.
```

1. Write a program to find if the number entered (accept the input from user at run time) by the user is a Magic Number.

```
Scanner input = new Scanner(System.in);
```

```
System.out.println("Enter number: ");
```

```
int num = input.nextInt();
```

```
public static int findSum(int num)
{
    int sum;
```

```
    for (sum = 0; num > 0; sum = sum + num % 10, num = num / 10);
```

```
    return sum;
}
```

```
while (num % 10 != num) {
    num = findSum(num);
}
```

```
if (num == 1) {
    System.out.println("Magic : True");
} else {
    System.out.println("Magic : False");
}
```

```
input.close();
```

```
Enter number:
```

```
Magic : True
```

1. Write a Java program to check whether the given input strings have no repetition of letters. The input strings may be word (or) phrase (or) sentence.

```
Scanner input = new Scanner(System.in);
```

```
System.out.println("Enter string: ");
```

```
String str = input.nextLine();
```

```
System.out.println(str);
```

```
int len = str.length();
```

```
boolean flag = true;
```

```
List list = new ArrayList<>();
```

```

for (int i = 0; i < len; i++) {
    if (list.contains(str.charAt(i))) {
        flag = false;
        break;
    } else {
        list.add(str.charAt(i));
    }
}

```

```
System.out.println("No repetitions : " + flag);
```

```
input.close()
```

Enter string:

rhythm

No repetitions : false

1. Design a class 'Complex' with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.

```
import java.lang.Math;
```

```

public static class Complex {
    private double real;
    private double imaginary;

    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }

```

```

    public double getReal() {
        return this.real;
    }

```

```

    public double getImaginary() {
        return this.imaginary;
    }

```

```
}
```

```

public void add(Complex a, Complex b) {
    double real = a.getReal() + b.getReal();
    double imaginary = a.getImaginary() + b.getImaginary();
    System.out.println("Sum: " + real + " + " + imaginary + "i");
}

```

```

public void subtract(Complex a, Complex b) {
    double real = a.getReal() - b.getReal();

```

```

        double imaginary = a.getImaginary() - b.getImaginary();
        System.out.println("Difference: " + real + " + " + imaginary +
        "i");
    }

    public void divide(Complex a, Complex b) {
        double real = (a.getReal() * b.getReal() + a.getImaginary() *
        b.getImaginary()) / (Math.pow(b.getReal(), 2) +
        Math.pow(b.getImaginary(), 2));
        double imaginary = (a.getImaginary() * b.getReal() - a.getReal() *
        b.getImaginary()) / (Math.pow(b.getReal(), 2) +
        Math.pow(b.getImaginary(), 2));

        System.out.println("Quotient: " + real + " + " + imaginary + "i");
    }

    public void multiply(Complex a, Complex b) {
        double real = a.getReal() * b.getReal() + a.getImaginary() *
        b.getImaginary();
        double imaginary = a.getImaginary() * b.getReal() + a.getReal() *
        b.getImaginary();

        System.out.println("Product: " + real + " + " + imaginary + "i");
    }

    Complex a = new Complex(1, 2);
    Complex b = new Complex(3, 4);

    System.out.println(a.getReal());
    System.out.println(a.getImaginary());

    System.out.println(b.getReal());
    System.out.println(b.getImaginary());

    add(a, b);
    subtract(a, b);
    divide(a, b);
    multiply(a, b);

    1.0
    2.0
    3.0
    4.0
    Sum: 4.0 + 6.0i
    Difference: -2.0 + -2.0i
    Quotient: 0.44 + 0.08i
    Product: 11.0 + 10.0i

```

1. Design and develop a context for implementing an interface for Vehicles. Consider the example of vehicles like bicycle, car and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply breaks. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way.

```
interface vehicle {
    public void GearChange();
    public void SpeedUp();
    public void ApplyBrakes();
}

class Bicycle implements vehicle {
    private int gear;
    private int speed;

    public Bicycle(int gear, int speed) {
        this.gear = gear;
        this.speed = speed;
    }

    public void GearChange() {
        if (this.gear < 18) {
            this.gear++;
        }
    }

    public void SpeedUp() {
        if (this.speed < 50) {
            this.speed++;
        }
    }

    public void ApplyBrakes() {
        if (this.speed > 0) {
            this.speed--;
        }
    }

    public void printStates() {
        System.out.println("Gear: " + this.gear);
        System.out.println("Speed: " + this.speed);
    }
}

class Bike implements vehicle {
    private int gear;
    private int speed;

    public Bike(int gear, int speed) {
```

```

        this.gear = gear;
        this.speed = speed;
    }

    public void GearChange() {
        if (this.gear < 18) {
            this.gear++;
        }
    }

    public void SpeedUp() {
        if (this.speed < 50) {
            this.speed++;
        }
    }

    public void ApplyBrakes() {
        if (this.speed > 0) {
            this.speed--;
        }
    }

    public void printStates() {
        System.out.println("Gear: " + this.gear);
        System.out.println("Speed: " + this.speed);
    }
}

class Car implements vehicle {
    private int gear;
    private int speed;

    public Car(int gear, int speed) {
        this.gear = gear;
        this.speed = speed;
    }

    public void GearChange() {
        if (this.gear < 18) {
            this.gear++;
        }
    }

    public void SpeedUp() {
        if (this.speed < 50) {
            this.speed++;
        }
    }

    public void ApplyBrakes() {

```

```
        if (this.speed > 0) {  
            this.speed--;  
        }  
    }  
  
    public void printStates() {  
        System.out.println("Gear: " + this.gear);  
        System.out.println("Speed: " + this.speed);  
    }  
}
```