1. Compare and contrast the float and Decimal classes' benefits and drawbacks.

```
The float and Decimal classes in Python are used to represent and perform calculations with decimal numbers.

Float class:

Benifits:
    Efficient for general-purpose floating-point calculations.
    Supports a wide range of mathematical operations and functions.
Drawbacks:
    Not suitable for precise financial or decimal-based calculations.




Decimal class:

Benifits:
    Decimal numbers provide arbitrary precision, allowing for precise calculations with decimal fractions
    without rounding errors.
Drawbacks:
    The range of Decimal numbers is limited by system memory, so extremely large or small numbers may exceed
    the available memory.

```

2. Decimal('1.200') and Decimal('1.2') are two objects to consider. In what sense are these the sameo bject? Are these just two ways of representing the exact same value, or do they correspond to different internal states?

```
When you create a Decimal object with the string '1.200' or '1.2', both objects will have the same value of 1.2.

when comparing Decimal objects for equality, the comparison is based on the numeric value and not the internal
representation.
```

3. What happens if the equality of Decimal('1.200') and Decimal('1.2') is checked?

```
if == is used it will return True
```

4. Why is it preferable to start a Decimal object with a string rather than a floating-point value?

```
Floating-point values in Python are stored using binary representation, which can
lead to rounding errors
While using the string values ot will give the exact results
```

5. In an arithmetic phrase, how simple is it to combine Decimal objects with integers?

In [1]:

```
from decimal import Decimal

k=Decimal("2.5")

print(3+k)
```

```
5.5
```

```
combine Decimal objects with integers is same as normal arithmatic operations

```

6. Can Decimal objects and floating-point values be combined easily?

```
combination of Decimal objects and the floating point is not supportive
```

7. Using the Fraction class but not the Decimal class, give an example of a quantity that can be expressed with absolute precision.

```
Yes,The fraction class produce the result without any precisssion loss.

```

8. Describe a quantity that can be accurately expressed by the Decimal or Fraction classes but not by a floating-point value.

```
# Non_terminative decimal
```

Q9.Consider the following two fraction objects: Fraction(1, 2) and Fraction(1, 2). (5, 10). Is the internal state of these two objects the same? Why do you think that is? ¶

```
No, the internal state of the Fraction(1, 2) and Fraction(1, 2). (5, 10) objects
is not the same.


Fraction(1, 2), the fraction is in its simplest form, where the numerator 1 and
denominator 2
have no common factors other than 1.

Fraction(1, 2). (5, 10) represents a fraction where the numerator 5 and
denominator 10 can
be simplified by dividing both by their greatest common divisor, which is 5. After
simplification, the
fraction becomes Fraction(1, 2).
```

Q10. How do the Fraction class and the integer type (int) relate to each other? Containment or inheritance?

```
The Fraction class and the integer type  do not have a direct inheritance
relationship.
```

In [ ]:

```

```