Fixed point arithmetic

In this exercise you will familiarize yourself with fixed point arithmetic using binary numbers.

In the given C++ file, you will have class called `FixedPoint`, which has six arithmetic operators that need to be implemented: addition, subtraction, multiplication, left shift, and right shift. Additionally, it contains method `change_precision` for changing the precision of the fixed-point number. It should be noted that all these methods return a new object, and do not change the value of the existing object, i.e., all the objects of the class should be treated as constants.

For all the operations, assume that there is no overflow and keep the full precision of the decimal part.

The left and right shift operators are intended for changing the precision, i.e., the value should be changed only in case the new precision is not enough to hold the complete value, in which case the value should be rounded, for the right shift, for the left shift you can still assume that there is no overflow.

The file also contains a `test` function that contains several test cases, your implementation should be able to pass all these test cases, but as a part of the exercise you should add some of your own tests. It should be clearly indicated what the inputs and expected outputs are for each test case. The class contains a `to_string` method for converting the values into strings as base 10 decimal numbers.

To return fill the information to the comments in the beginning of the file and return to Moodle by Feb. 1 for the on-time bonus.

Implementing the division is an extra task.