1. Assignment Testing:

   Here you have to test if assignment operations of two(a=b) or more instances(a=b=c) of BigNum class is working or not.

   Check all possible combinations of Bignum instances created by all the constructors (and also check with different corresponding critical inputs).

   For example,

   ```
   Bignum test1;
   BigNum test2 = new BigNum(100);
    test2 = test1;
   ///now check if test2's digits contain 0 because test1 contains 0 (default constructor )
   ```

2. Operator overloading testing:
   a. +=,-=,*=,/=,%=
      For these overloading functions, declare tow Bignum instances initialized with critical inputs and then check the output.
      For example

      ```
      Bignum test1;
      BigNum test2;
       test1+=test2;

      ///now test1 should contain 0 because both test1 and test2 were created using the default constructor. Same goes for the other operators.
      ```

   b. ++,--
      Here the Bignum instance's number should be incremented or decremented by 1.

      For example
      ```
      BigNum test1;
       test1++;
      //now test1 should contain 1.
      ```

   c. +,-,*,/,%

      Here check if the operation is working accordingly.

      For example,
      ```
      BigNum test1;
      BigNum test2 = new BigNum("123");
      BigNum test3 = test1+test2;
      ```

//test3 should contain 123.

d. >.<,<=,=>,= =,!=
Implement these operators for BigNum. The rule is the same as the signs say.

For example,

```
BigNum test1 = new BigNum (“123”);
BigNum test2 = new BigNum(123);
 if(test1 == test2)
        Cout<<”test is passed”;
 else
        Cout<< “test has failed”;
```