## 1. initParts(string number)

- Initialize integer and fractional parts of the number.

- Loop through each character in 'number'.

- When a decimal point is encountered, set a flag to start processing the fractional part.

- Append characters to the integer or fractional part based on the flag.

## 2. BigReal operator+(BigReal &other)

- Addition of two BigReal numbers, taking into account their signs.

- If both numbers have the same sign:

  1. Add their integer parts.

  2. Add their fractional parts.

- If they have different signs:

  1. Find the number with the larger magnitude (absolute value).

  2. Subtract the smaller-magnitude number from the larger-magnitude number.

  3. The sign of the result is the same as the number with the larger magnitude.

### 3. BigReal operator-(BigReal &other)

- Subtraction of two BigReal numbers by inverting the sign of the second number and using addition.

- Negate the sign of the 'other' number.

- Use addition to perform subtraction with the negated 'other' number.

### 4. int compMagnitude(const BigReal &num) const

- Compare the magnitude of two BigReal numbers.

- Compare their integer parts:

  1. If the current number has a larger integer part, return 1.

  2. If the 'num' parameter has a larger integer part, return -1.

- If integer parts are equal:

  1. Compare their fractional parts digit by digit.

  2. Return 1 if the current number's fractional part is greater.

  3. Return -1 if the 'num' parameter's fractional part is greater.

  4. Return 0 if the fractional parts are equal.

### 5. bool operator<(const BigReal &other) const

- Check if the current BigReal number is less than the 'other' BigReal number:

1. If the sign of the current number is negative, return true.

2. If the signs are the same and the current number's magnitude is less than 'other,' return true.

3. Otherwise, return false.

### 6. bool operator>(const BigReal &other) const

- Check if the current BigReal number is greater than the 'other' BigReal number:

1. If it's not less than or equal to 'other,' return true.

2. Otherwise, return false.

### 7. bool operator==(const BigReal &other) const

- Check if the current BigReal number is equal to the 'other' BigReal number:

1. If their signs are the same and their magnitudes are equal, return true.

2. Otherwise, return false.

### 8. ostream& operator<<(ostream &os, const BigReal &num)

- Overload the stream insertion operator for output:

1. If the BigReal number has a negative sign, display '-'.

2. Display the BigReal number as a string.

| Mohamed mostafa<br>20220309 | Ahmed tamer<br>20220013 | Yousef Nasser<br>2022416 |
|---|---|---|
| Header<br>Main<br>Testing<br>Is positive<br>Size<br>Operator –<br>Constructor | Operator =<br>Operator +<br>Operator <<<br>Isvalidreal<br>Setnum | Compmagnitude<br>operator <<br>operator ><br>Operator ==<br>Constructor |