**CSCI 3033 OLA103**

Due: 10/12 (Monday), midnight (11:59)

**Problem**: **HugeIntegerClass**

Create a class HugeInteger which uses a 40-element array of digits to store integers as large as 40 digits each. Provide methods ***Input, toString, Add, and Subtract***. For comparing HugeInteger objects, provide the following methods: ***IsEqualTo, IsNotEqualTo, IsGreaterThan, IsLessThan, IsGreaterThanOrEqualTo and IsLessThanOrEqualTo***. Each of these is a method that returns true if the relationship holds between the two HugeInteger objects and returns false if the relationship does not hold. Provide method ***IsZero***. In the Input method, use the string method toCharArray to convert the input string into an array of characters, then iterate through these characters to create your HugeInteger. If you feel ambitious, provide methods Multiply, Divide, and Remainder (Optional for extra credit 5 % toward the next exam) .

// Driver Class

**import** java.util.\*;

**public** **class** HugeIntegerTest

{

**public** **static** **void** main(String[] args)

{

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

//String str1;

//String str2;

HugeInteger integer1 = **new** HugeInteger();

HugeInteger integer2 = **new** HugeInteger();

System.***out***.println("Enter first HugeInteger: ");

integer1.Input(input.nextLine());

System.***out***.println("Enter second HugeInteger: ");

integer2.Input(input.nextLine());

System.***out***.println("HugeInteger1 " + integer1);

System.***out***.println("HugeInteger2 " + integer2);

HugeInteger result;

// add two HugeIntegers

result = integer1.Add(integer2);

System.***out***.println ("Add result" + result);

// subtract two HugeIntegers

result = integer1.Subtract(integer2);

System.***out***.println ("Subtract result" + result);

//result = integer1.multiply(integer2);

//System.out.println ("Subtract result" + result);

// compare two HugeIntegers

System.***out***.println ( "HugeInteger 1 is zero: " + integer1.IsZero());

System.***out***.println ( "HugeInteger 2 is zero: " + integer2.IsZero());

System.***out***.println (

"HugeInteger 1 is equal to HugeInteger 2: " + integer1.IsEqualTo(integer2));

System.***out***.println (

"HugeInteger 1 is not equal to HugeInteger 2:" + integer1.IsNotEqualTo(integer2));

System.***out***.println (

"HugeInteger 1 is greater than HugeInteger 2: " + integer1.IsGreaterThan(integer2));

System.***out***.println (

"HugeInteger 1 is less than HugeInteger 2:" + integer1.IsLessThan(integer2));

System.***out***.println (

"HugeInteger 1 is greater than or equal to HugeInteger 2: " + integer1.IsGreaterThanOrEqualTo(integer2));

System.***out***.println ( "HugeInteger 1 is less than or equal to HugeInteger 2: " + integer1.IsLessThanOrEqualTo(integer2));

}

}

Sample Run 1:

Enter first HugeInteger:

1234567890123456789012345678901234567890

Enter second HugeInteger:

0987654321098765432109876543210987654321

HugeInteger1 +1234567890123456789012345678901234567890

HugeInteger2 +987654321098765432109876543210987654321

Add result+2222222211222222221122222222112222222211

Subtract result+246913569024691356902469135690246913569

HugeInteger 1 is zero: false

HugeInteger 2 is zero: false

HugeInteger 1 is equal to HugeInteger 2: false

HugeInteger 1 is not equal to HugeInteger 2:true

HugeInteger 1 is greater than HugeInteger 2: true

HugeInteger 1 is less than HugeInteger 2:false

HugeInteger 1 is greater than or equal to HugeInteger 2: true

HugeInteger 1 is less than or equal to HugeInteger 2: false

Sample Run 2:

Enter first HugeInteger:

65479876253763637782636782636

Enter second HugeInteger:

989

HugeInteger1 +65479876253763637782636782636

HugeInteger2 +989

Add result+65479876253763637782636783625

Subtract result+65479876253763637782636781647

HugeInteger 1 is zero: false

HugeInteger 2 is zero: false

HugeInteger 1 is equal to HugeInteger 2: false

HugeInteger 1 is not equal to HugeInteger 2:true

HugeInteger 1 is greater than HugeInteger 2: true

HugeInteger 1 is less than HugeInteger 2:false

HugeInteger 1 is greater than or equal to HugeInteger 2: true

HugeInteger 1 is less than or equal to HugeInteger 2: false

How to submit the program

Submit your java files (both the<yourLastName>\_ HugeInteger.java and <yourLastName>HugeIntegerTest.java to D2L dropbox “OLA103”

1. Name your driver program <yourlastname>\_HugeIntegerTest. For example, if my last name were Carroll, then for me, it would be Carroll\_HugeIntegerTest.java.
2. Compile and execute your project and make sure your assignment works appropriately.
3. When you are ready to submit your assignment,
4. Your program must compile and run.
5. Submit the assignment files (both the class and the test class) to the proper dropbox on D2L.

**public** **class** HugeInteger

{

**private** **final** **int** DIGITS = 40;

**private** **int**[] integer;// array containing the integer

**private** **boolean** positive; // whether the integer is positive

// parameterless constructor

**public** HugeInteger()

{

//

}

// Convert a string to HugeInteger

**public** **void** Input(String inputstring)

{

//

}

// add two HugeIntegers

**public** HugeInteger Add(HugeInteger addValue)

{

//

}

// add two positive HugeIntegers

**private** HugeInteger AddPositives(HugeInteger addValue)

{

//

}

// subtract two HugeIntegers

**public** HugeInteger Subtract(HugeInteger subtractValue)

{

**//**

}

// subtract two positive HugeIntegers

**private** HugeInteger SubtractPositives(HugeInteger subtractValue)

{

//

}

// find first non-zero position of HugeInteger

**private** **int** FindFirstNonZeroPosition()

{

//

}

// get string representation of HugeInteger

**public** String toString()

{

//

}

// test if two HugeIntegers are equal

**public** **boolean** IsEqualTo(HugeInteger compareValue)

{

//

}

// test if two HugeIntegers are not equal

**public** **boolean** IsNotEqualTo(HugeInteger compareValue)

{

//

}

// test if one HugeInteger is greater than another

**public** **boolean** IsGreaterThan(HugeInteger compareValue)

{

//

}

// test if one HugeInteger is less than another

**public** **boolean** IsLessThan(HugeInteger compareValue)

{

//

}

// test if one HugeInteger is greater than or equal to another

**public** **boolean** IsGreaterThanOrEqualTo(HugeInteger compareValue)

{

**//**

}

// test if one HugeInteger is less than or equal to another

**public** **boolean** IsLessThanOrEqualTo(HugeInteger compareValue)

{

**//**

}

// test if one HugeInteger is zero

**public** **boolean** IsZero()

{

//

}

//Optional

**public** HugeInteger multiply(HugeInteger multiplyValue)

{

//

}

}