class LongNumber //Name: Terry C. Wong //Student ID #: 907564702

{

// private field digit[] array to hold

// upto MAX\_SIZE digits

private int digit[];

private final int MAX\_SIZE = 50;

// default constructor

public LongNumber()

{

digit = new int [MAX\_SIZE];

resetArray();

} // end default constructor

// constructor:

// copy input number (in array) into digit[] array

// check for error: too long or invalid digit

public LongNumber(int longDigitArray[])

{

digit = new int [longDigitArray.length];

setLongNumber(longDigitArray); //use setLongNumber method to store given parameter

// need to implement!

} // end constructor

// constructor:

// copy input number (in string ) into digit[] array

// check for error: too long or invalid char

public LongNumber(String longNumStr)

{

digit = new int [longNumStr.length()];

setLongNumber(longNumStr); //use setLongNumber method to store given parameter

// need to implement!

}

// replace digit[] array by input number (in array)

// check for error: too long or invalid digit

public void setLongNumber(int longDigitArray[])

{

try //throw an exception if parameter is null or the # of elements in the given array is greater than MAX\_SIZE

{

if (longDigitArray == null || longDigitArray.length > MAX\_SIZE)

throw new Exception("Exception: Size too big or invalid character!");

else

digit = longDigitArray;

}

catch ( Exception e) //catch the exception with printed error msg and set proper digit[] to null

{

System.out.println( e.getMessage());

System.out.println("You have entered invalid values. Please enter the correct values.");

digit = null;

}

// need to implement!

} // end setLongNumber

// replace digit[] array by input number (in string)

// check for error: too long or invalid char

public void setLongNumber(String longNumStr)

{

try //throw an exception if parameter is null or the length of the given string is greater than MAX\_SIZE

{

if (longNumStr == null || longNumStr.length() > MAX\_SIZE)

throw new Exception("Exception: Size too big or invalid character!");

else

{

String tmp = null;

digit = new int [longNumStr.length()];

for (int i = longNumStr.length()-1; i >= 0; i--) //use a for loop to store information in integer array starting from the very end

{

tmp = longNumStr.substring(i,i+1); //use the substring method to read elements one by one

digit[i] = Integer.parseInt(tmp); //use parseInt method to convert string to int and store it in digit[]

}

}

}

catch (Exception e) //catch the exception with printed error msg and set proper digit[] to null

{

System.out.println( e. getMessage());

System.out.println("You have entered invalid values. Please enter the correct values.");

digit = null;

// need to implement!

}

}

// static method:

// create a LongNumber object from input number (in array)

// return the object

//

// check for error: too long or invalid char

public static LongNumber createLongNumber(int longDigitArray[])

{

LongNumber tmp = new LongNumber(longDigitArray); //call the constructor to create the LongNumber object

// need to implement!

// should return proper LongNumber object

return tmp;

} // end createLongNumber

// static method:

// create a LongNumber object from input number (in string)

// return the object

//

// check for error: too long or invalid digit

public static LongNumber createLongNumber(String longNumString)

{

LongNumber tmp = new LongNumber(longNumString); //call the constructor to create the LongNumber object

// need to implement!

// should return proper LongNumber object

return tmp;

} // end createLongNumber

// add this LongNumber with rhs LongNumber

// return a result LongNumber object

//

// check for error: overflow

// return null digit[] in result if there is an error

public LongNumber addLongNumber(LongNumber rhs)

{

LongNumber Result = new LongNumber(); //create a new instance of LongNumber to store the result

try //throw an exception if either digit[] or parameter is null or the length of the given object's digit[] length is greater than MAX\_SIZE

{

if (digit == null || rhs == null || rhs.digit.length > MAX\_SIZE)

throw new Exception("Exception: Size too big or invalid character!");

else

{

long input = Long.parseLong(this.toString()); //use Long.parseLong to convert digit[] in this LongNumber to int

input += Long.parseLong(rhs.toString()); //add the value from this LongNumber to the value from the given object

String sum = Long.toString(input); //store the sum int in a string variable

Result.setLongNumber(sum); //store value in the digit[] of the object being returned by calling the setLongNumber method

}

}

catch (Exception e) //catch the exception with printed error msg and set proper digit[] to null

{

System.out.println( e. getMessage());

System.out.println("You have entered invalid values. Please enter the correct values.");

Result.setLongNumber("null"); // need to implement!

}

return Result;

}

// multiply this LongNumber with rhs LongNumber

// return a result LongNumber object

//

// check for error: overflow

// return null digit[] in result if there is an error

public LongNumber multiplyLongNumber(LongNumber rhs)

{

LongNumber Result = new LongNumber(); //create a new instance of LongNumber to store the result

try //throw an exception if either digit[] or parameter is null or the length of the given object's digit[] length is greater than MAX\_SIZE

{

if (digit == null || rhs == null || rhs.digit.length > MAX\_SIZE)

throw new Exception("Exception: Size too big or invalid character!");

else

{

long input = Long.parseLong(this.toString()); //use Long.parseLong to convert digit[] in this LongNumber to int

input \*= Long.parseLong(rhs.toString()); //multiply the value from this LongNumber to the value from the given object

String product = Long.toString(input); //store the product int in a string variable

Result.setLongNumber(product); //store value in the digit[] of the object being returned by calling the setLongNumber method

}

}

catch (Exception e) //catch the exception with printed error msg and set proper digit[] to null

{

System.out.println( e. getMessage());

System.out.println("You have entered invalid values. Please enter the correct values.");

Result.setLongNumber("null"); // need to implement!

}

// need to implement!

// should return proper LongNumber object

return Result;

} // end multiplyLongNumber

// convert digit[] number into string and return it

public String toString()

{

String tmp = new String("");

// special case when there was an error, digit[] is null

if (digit == null)

tmp="null";

else {

int size = digit.length;

int i=0;

// skip front 0 digits

while((i < size ) && (digit[i]==0))

i++;

// special case, all 0 in digit[]

if (i == size ) tmp += "0";

else {

for (int j = i; j < size; j++)

tmp += digit[j];

}

}

// return result string

return tmp;

} // end toString

//============================================================

// add more private methods here if you like to .....

private void resetArray()

{

for (int i = 0; i < MAX\_SIZE; i++)

digit[i] = 0;

}

//============================================================

// print msgs for main method

private static void printMsg(String msg)

{

System.out.println(msg);

System.out.println("==============================");

}

// use main method to create test cases

// these are test cases that i will use to test your program

public static void main(String[] args)

{

int num1[] = {0,0,0,0,0,0,0,0,0,1};

int num2[] = {1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9};

int num3[] = {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2};

int num4[] = {9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9};

String str1 = "1234567890123456789012345678901234567890";

String str2 = "0000123456789012345678901234567890123456";

String str3 = "98765432109876543210987654321098765432109876543210";

String str4 = "989";

String str5 = "3211";

int badNum[] = {2,2,2,2,2,20,2,2,2,2,2,2,2,2,2,2,2,2};

int longNum[] = {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2};

String badString = "12345678901234567890x12345678901234567890";

String longString = "987654321098765432109876543210987654321098765432108888";

// check constructors

LongNumber longNum0 = new LongNumber();

printMsg("Test#1: longNum="+longNum0);

LongNumber longNum1 = new LongNumber(num1);

printMsg("Test#2: longNum="+longNum1);

LongNumber longNum2 = new LongNumber(num2);

printMsg("Test#3: longNum="+longNum2);

LongNumber longNum3 = new LongNumber(longNum);

printMsg("Test#4: longNum="+longNum3);

LongNumber longNum4 = new LongNumber(badNum);

printMsg("Test#5: longNum="+longNum4);

LongNumber longNum5 = new LongNumber(str1);

printMsg("Test#6: longNum="+longNum5);

LongNumber longNum6 = new LongNumber(str2);

printMsg("Test#7: longNum="+longNum6);

LongNumber longNum7 = new LongNumber(longString);

printMsg("Test#8: longNum="+longNum7);

LongNumber longNum8 = new LongNumber(badString);

printMsg("Test#9: longNum="+longNum8);

longNum1.setLongNumber(str2);

printMsg("Test#10: longNum="+longNum1);

longNum1.setLongNumber(longString);

printMsg("Test#11: longNum="+longNum1);

// need to create new storage

longNum1=new LongNumber();

longNum1.setLongNumber(badString);

printMsg("Test#12: longNum="+longNum1);

longNum1=new LongNumber();

longNum1.setLongNumber(num2);

printMsg("Test#13: longNum="+longNum1);

longNum1.setLongNumber(longNum);

printMsg("Test#14: longNum="+longNum1);

longNum1=new LongNumber();

longNum1.setLongNumber(badNum);

printMsg("Test#15: longNum="+longNum1);

LongNumber longNum9 = LongNumber.createLongNumber(num3);

printMsg("Test#16: longNum="+longNum9);

LongNumber longNum10 = LongNumber.createLongNumber(badString);

printMsg("Test#17: longNum="+longNum10);

LongNumber longNum11 = LongNumber.createLongNumber(str2);

printMsg("Test#18: longNum="+longNum11);

LongNumber longNum12 = LongNumber.createLongNumber(longNum);

printMsg("Test#18: longNum="+longNum12);

longNum2= new LongNumber(num2);

longNum3= new LongNumber(num3);

longNum5= new LongNumber(num4);

longNum4 = longNum2.addLongNumber(longNum3);

printMsg("Test#19:"+longNum2 + " + " + longNum3 + " = " + longNum4);

longNum4 = longNum2.addLongNumber(longNum5);

printMsg("Test#20:"+longNum2 + " + " + longNum5 + " = " + longNum4);

longNum4 = longNum2.multiplyLongNumber(longNum3);

printMsg("Test#21:"+longNum2 + " \* " + longNum3 + " = " + longNum4);

longNum4 = longNum2.multiplyLongNumber(longNum5);

printMsg("Test#22:"+longNum2 + " \* " + longNum5 + " = " + longNum4);

longNum2.setLongNumber(str4);

longNum3.setLongNumber(str5);

longNum4 = longNum2.addLongNumber(longNum3);

printMsg("Test#23:"+longNum2 + " + " + longNum3 + " = " + longNum4);

longNum4 = longNum2.multiplyLongNumber(longNum3);

printMsg("Test#24:"+longNum2 + " \* " + longNum3 + " = " + longNum4);

} // end main

}