**<http://www.java2novice.com/java-interview-programs/string-reverse-recursive/>**

Write a program to reverse a string using recursive methods.

You should not use any string reverse methods to do this.

public class StringRecursiveReversal {

    String reverse = "";

    public String reverseString(String str){

        if(str.length() == 1){

            return str;

        } else {

            reverse += str.charAt(str.length()-1)

                    +reverseString(str.substring(0,str.length()-1));

            return reverse;

        }

    }

    public static void main(String a[]){

        StringRecursiveReversal srr = new StringRecursiveReversal();

        System.out.println("Result: "+srr.reverseString("Java2novice"));

    }

}

Write a program to reverse a number using numeric operations. Below example shows how to reverse a number using numeric operations.

public class NumberReverse {

    public int reverseNumber(int number){

        int reverse = 0;

        while(number != 0){

            reverse = (reverse\*10)+(number%10);

            number = number/10;

        }

        return reverse;

    }

    public static void main(String a[]){

        NumberReverse nr = new NumberReverse();

        System.out.println("Result: "+nr.reverseNumber(17868));

    }

}

While your original number is nonzero, take your result, multiply it by 10, and add the remainder from dividing the original by 10.

For example, say your original number is 12345. Start with a result of 0.

Multiply result by 10 and add 5, giving you 5. (original is now 1234.)

Multiply result by 10 and add 4, giving you 54. (original is now 123.)

Multiply result by 10 and add 3, giving you 543. (original = 12.)

Multiply result blah blah 5432. (original = 1.)

Multiply, add, bam. 54321. And 1 / 10, in int math, is zero. We're done.

Your mission, should you choose to accept it, is to implement this in Java. :) (Hint: division and remainder are separate operations in Java. % is the remainder operator, and / is the division operator. Take the remainder separately, then divide the original by 10.)

**public** **class** Test {

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

**try** {

// Specify the path of file

File src=**new** File("C:/Testleaf Selenium Library/sample.xlsx");

// load file

FileInputStream fis=**new** FileInputStream(src);

// Load workbook

XSSFWorkbook wb=**new** XSSFWorkbook(fis);

// Load sheet- Here we are loading first sheetonly

XSSFSheet sh1= wb.getSheetAt(0);

// getRow() specify which row we want to read.

// and getCell() specify which column to read.

// getStringCellValue() specify that we are reading String data.

System.***out***.println(sh1.getRow(0).getCell(0).getStringCellValue());

System.***out***.println(sh1.getRow(0).getCell(1).getStringCellValue());

System.***out***.println(sh1.getRow(1).getCell(0).getStringCellValue());

System.***out***.println(sh1.getRow(1).getCell(1).getStringCellValue());

System.***out***.println(sh1.getRow(2).getCell(0).getStringCellValue());

System.***out***.println(sh1.getRow(2).getCell(1).getStringCellValue());

} **catch** (Exception e) {

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

}

}

}

how are square numbers related to triangular numbers?