Lab 3
Recursion

Do two (2) of the following problems:

Problem1:
Develop a recursive algorithm to evaluate a string of PARENTHESES for proper format and balance.

Examples of balanced: () , (()) , (()()) , ()() , ((())())
Examples of improper: (,) , ()) , (() ,)(,)(,)(

Write a program that implements and demonstrates the algorithm.

Testing:
The program should repeatedly prompt the user for an input string of parentheses
The output should display intermediate results as you validate the string and finally put out a result of:

The parentheses are VALID | INVALID and BALANCED | NOTBALANCED

Problem2:
Add a function to your Array class that will permit the caller to search for a particular element in the array. The array must contain at least 16 slots and be populated in ascending order.

Call the new function ***findElement(x, low, high)***
where ***x*** is the element to be found, ***low*** is zero, and ***high*** is the greatest array slot number.

For example:

y = findElement(12, 0, 15)
requests to search for the integer content 12 in an array that has
16 ordered indexed slots (numbered 0 through 15)

Develop a recursive algorithm to perform the binary search.
Choose an even number of array slots for easier debugging.
Write (or update) a user interface that prompts the user and demonstrates the function.

Testing:
Demonstrate that it works for the structure and algorithm's boundary conditions.
Show the number of recursions needed to find (or fail to find) the selected target integer.
Show all inputs, outputs, return code and any informational messages.

Problem3:

Part I

Let "S" be the sum of the series defined by the following:

S(n) = 1/2 + 1/4 + 1/8 + 1/16 + 1/32 + . . . + 1/(2**n) ,for n = 1,2,3,...

(Note: **2******n** means the number **2** raised to the power of **n**.
so to calculate 1/16 your code would be 1 / pow(2, 4) *in C++*
so to calculate 1/16 your code would be 1 / Math.pow(2, 4) *in JAVA*

For example: **S(1) = 1/2**
S(2) = 1/4 + S(1)
S(3) = 1/8 + S(2)
S(4) = 1/16 + S(3)
...
...
S(100) = 1/(2100) + S(99)**

Develop and program a recursive algorithm to calculate the Sum of this series expansion for the **first 10, first 50** and **first 100** terms. This is your 3 inputs, the integers 10, 50 and 100.

Write a very small program that implements and demonstrates the algorithm.
The program should either prompt the user for an input integer as to how many terms in the expansion.
The output should look something like: The Sum of the series for "N" terms is "xxxxxxxxx".

Part II

Let "P" be the sum of the series defined by the following:

1/2 - 1/4 + 1/8 - 1/16 + 1/32 - . . . + 1/(2**n) ,for n = 1,2,3,...

Note that when "n" is an "odd" number the term is **positive**,
when "n" is an "even" number the term is **negative**.

Develop and program a recursive algorithm to calculate the Sum of the Series Expansion for the **first 10, first 50** and **first 100** terms.

Show what value does each series converge to.