Recursion Exercises

Note: Do as many exercises as you can

Written exercises

1. Give the sequence of argument values that result when the following program is invoked for each of the integers 1 through 9.

```
 int puzzle(int n)

  {if (n == 1) return 1;

     if (n % 2 == 0)

       return puzzle(n/2);

        else

         return puzzle(3*n+1);

   }
```

2. Use the following function puzzle(..) to answer problems 2.1 - 2.3.

```
int puzzle(int base, int limit)

 { //base and limit are nonnegative numbers

    if(base > limit)

      return(-1);

        else if(base == limit)

          return 1;

           else

            return(base * puzzle(base + 1, limit));

 }
```

2.1 Identify the base case(s) of function puzzle(..)

2.2 Identify the recursive case(s) of function puzzle(..)

2.3 Show what would be displayed by the following calls.

a. System.out.print(puzzle(14,10));

b. System.out.print(puzzle(4,7));

c. System.out.print(puzzle(0,0));

3. Show the output that will be displayed by the call show(123);

```
 void show(int n)
  { if(n>0) show(n/10);
    System.out.print((n%10));
  }
```

4. Show the output that will be displayed by the call show(134);

```
 void show(int n)
  { System.out.print(n%10);
     if(n>0) show(n/10);
  }
```

5. Show the output that will be displayed by the call show(145);

```
void show( int n)
 { System.out.print((n%10));
    if(n>0) show(n/10);
    System.out.print((n%10));
  }
```

6. Complete the Java code to recursively evaluate the sum: sum = 1 + 1/2 + 1/3 +...+1/n

```
double sum(int n)                  // n>=1

{

if (___________)

return _____________;

            return ______ + sum(______);

}
```

7. Consider the following recursive function.

```
int mystery(int a, int b)
 { if (b == 0)
    return 0;
     else if (b % 2 == 0)
      return mystery(a+a, b/2);
       else
        return (mystery(a+a, b/2) +
a);
 }
```

a. What values of a and b are directly handled by the stopping (base) case?

b. What are the values of mystery(2, 25) and mystery(3, 11)? Given positive integers a and b, describe what value mystery(a, b) computes. Answer the same question, but replace + with *.

c. For the call mystery(3, 7), how many calls to mystery will be made, including the original call?

8. Consider the following recursive function.

```
void ex237(int n)
  { if (n <= 0) return;
    StdOut.println(n);
    ex237(n-2);
    ex237(n-3);
    StdOut.println(n);
  }
```

Give the sequence of integers printed by a call to ex237(6).

9. Consider the following recursive function:

```
String ex238(int n)
{
   if (n <= 0) return "";
    return (ex238(n-3) + n + ex238(n-2) + n);
}
```

Give the value of ex238(6).

10. Explain what the following function fib does:

```
int []term = new int [1000];

int fib(int n)

{

    if (n <= 1)

        return n;

    if (term[n] != 0)

        return term[n];

    else

    {

        term[n] = fib(n - 1) + fib(n - 2);

        return term[n];

    }

}
```

Practical exercises

Write Java programs to perfom the following tasks:

1. Write a recursive function that computes the sum of all numbers from 1 to n, where n is given as parameter.

//return the sum 1+ 2+ 3+ ...+ n

int sum(int n)

2. Write a recursive function that finds and returns the minimum element in an array, where the array and its size are given as parameters.

//return the minimum element in a[]

int findmin(int a[], int n)

3. Write a recursive function that computes and returns the sum of all elements in an array, where the array and its size are given as parameters.

//return the sum of all elements in a[]

int findsum(int a[], int n)

4. Write a recursive function that determines whether an array is a palindrome, where the array and its size are given as parameters.

//returns 1 if a[] is a palindrome, 0 otherwise. The string a is palindrome if it is the same as its reverse.

int ispalindrome(char a[], int n)

5. Write a recursive function that searches for a target in a sorted array using binay search, where the array, its size and the target are given as parameters.

6. **Stirling numbers**: A stirling number of the first kind is defined as follows

- $s(0,0) = 1$
- $s(n,0) = 0$, for all $n > 0$
- $s(n+1,k) = s(n,k-1) - n*s(n,k)$, for all $n \geq 0$ and $k>0$

  Write a recursive routine to calculate stirling numbers of the first kind.

7. **Tree height.** Given a labeled binary tree (represented by a pointer to a TreeNode) calculate its height.

8**. Tree size.** Given a labeled binary tree (represented by a pointer to a TreeNode) calculate its size.