5. Lab 5: Recursion

5.1. Objectives

Apply Recursion in pracitical mathematical problems.

5.2. Problems

5.3. Problem 1: Use the following function puzzle(..) to answer problems 1 - 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);
}
```

1. (10 points) Identify the base case(s) of function puzzle(..)
2. (10 points) Identify the recursive case(s) of function puzzle(..)
3. (10 points) 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));

5.4. Problem 2: Complete the Java code to recursively evaluate the sum: sum = 1 + 1/2 + 1/3 +...+1/n, n > 1.

```
double sum(int n)          // n>=1
{
  if (_____________)
          return _____________;


  return ______ + sum(______);
```

5.5. Problem 4: 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)
int findsum(int a[], int n)
```

5.6. Problem 6: Write a method that receives two integers and returns the largest common divisor. The formula to calculate the Largest common divisor is shown below:

$$gcd(p, q) = \begin{cases} p & \text{if } q = 0 \\ gcd(q, p \% q) & \text{otherwise} \end{cases}$$

5.7. Problem 8: Write a recursive function to generate all subsets of a given set.

5.8. Problem 10: Use recursion to generate a Sierpinski triangle fractal

https://en.wikipedia.org/wiki/Sierpi%C5%84ski_triangle