**Lab 05**

**Recursion**

**Revised by Tran Thanh Tung**

1. **Introduction:**

You have learnt about Recursion. This lab will help you to apply Recursion in Java programming language.

1. **Objectives**

Apply Recursion in calculating mathematical expressions

1. **Problems:**

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

if ( base > limit )

return -1;

else if ( base == limit )

return 1;

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

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

1. Show what would be displayed by the following calls.
   1. System.out.print(puzzle(14,10));

-1

* 1. System.out.print(puzzle(4,7));

120

* 1. System.out.print(puzzle(0,0));

1

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

double sum(int n)

{

if (n==1)

return 1;

return (1/n)+ sum(n-1);

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)

{

if (n==1)

return 1;

return n+sum(n-1);

}

1. 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)

1. 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)

1. Write a method that receives two integers and returns the largest common divisor. Formula to calculate the Largest common divisor is shown below:

