|  |  |  |
| --- | --- | --- |
| American University of Sharjah  College of Engineering  Department of Computer Science & Engineering  P. O. Box 26666, Sharjah, UAE |  | **Instructors:** Dr. Michel Pasquier  **Lab Instructor:** Praveena Kolli  **Office:** EB2-12  **Phone**: 971-6-5152352  **e-mail**: pkolli@aus.edu  **Semester:** Spring 2021 |

**CMP305L - Data Structures and Algorithms Lab**

**Lab. Assignment 8-Recursion**

**Objectives:**

* Understand Recursion
* Develop functions using recursion
* Implement recursive traversal of linked lists

***Note:***

***Lab:*** Exercises 1,2,3,4 and 5 (10 marks)

***Bonus*:** Exercise 6 (1 mark)

**Exercise 1**

1. Develop and test a recursive function that calculates the harmonic series:

F(n) =  1 + {1 \over 2} + {1 \over 3} + {1 \over 4} + {1 \over 5} + \cdots = \sum_{n=1}^\infty {1 \over n}.

1. Develop and test a recursive function that calculates the alternate series:

F(n) =  1 - {1 \over 2} + {1 \over 3} - {1 \over 4} + {1 \over 5} - \cdots =\sum_{n=1}^\infty (-1)^{n+1} {1 \over n}=\ln(2).

**Exercise 2**

Develop and test the following *recursive function* that takes an *integer* (0 to 9) and prints the output as shown in sample the input/output.

void IntegerPalindrome(int value);

*Sample Input/Output:*

Enter an integer: 0

0123456789876543210

Enter an integer: 5

567898765

**Hint:** Remember lines of code after the recursive call are executed in reverse order of the calls.

**Exercise 3**

Develop and test a recursive *function* to check if positive integer ***n*** is a prime. An integer ***n*** is a prime if is divisible only by ***1*** and ***itself*** and *not* by any integer in the range from 2 to *sqrt(n)* (both inclusive).

***Note:*** 0 and 1 are not prime numbers. The *sqrt* function provided by <cmath> library returns a double and you must consider only the integral part of it.

**Exercise 4**

Develop and test a recursive function that converts the given decimal number into equivalent binary number. No data structure should be used.

**Exercise 5**

Write the following recursion function on a singly linked list developed in Lab3.

A function compare that takes two lists and returns true if both the lists have same values, if not return false.

**Bonus:**

**Exercise 6:**

Write the following recursion function on a singly linked list developed in Lab3.

Write a single function, findMinMax that traversals the given linked list once to find both the minimum and maximum and returns a pair that contains min and max of the list.

template <typename Object>

pair<Object,Object> findMinMax (SingleNode<Object>\*lst,Object min,Object max)

Reference: <https://www.cplusplus.com/reference/utility/pair/pair/>