|  |  |  |
| --- | --- | --- |
| American University of Sharjah  College of Engineering  Department of Computer Science & Engineering  P. O. Box 26666, Sharjah, UAE |  | **Instructors:** Dr. Gerassimos Barlas  **Lab Instructor:** Praveena Kolli  **Office:** EB2-126  **Phone**: 971-6-5152352  **e-mail**: pkolli@aus.edu  **Semester**: Spring 2017 |

**CMP 305L - Data Structures and Algorithms Lab**

**Lab. Assignment 2**

**Objectives:**

* Understand Recursion
* Develop functions using recursion

**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}.

#include <iostream>

using namespace std;

double series(int n)

{

if (n == 1)

{

return 1;

}

else

{

return (1.0 / (double)n) + series(n-1);

}

}

void main()

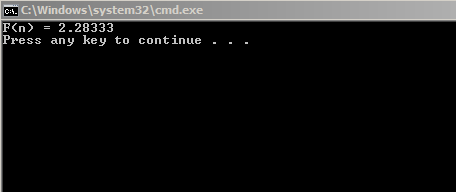
{

double m;

m = series(5.0);

cout << "F(n) = " << m << endl;

}



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).

#include <iostream>

using namespace std;

double series(int n)

{

if (n == 0)

{

return 0;

}

if (n == 1)

{

return 1;

}

else

{

int j;

if (n % 2 == 0)

{

j = 1;

j=j\*(-1);

return (1.0 / (double)n) + (j\*(series(n - 1)));

}

else if (n < 0)

{

n = (-1)\*n;

return (1.0 / (double)n) + (series(n - 1));

}

else

{

return (1.0 / (double)n) + (series(n - 1));

}

}

}

void main()

{

double m;

m = series(4.0);

cout << "F(n) = " << m << endl;

}

**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

**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 takes an integer ***n*** as its argument and prints all the *Fibonacci* numbers right from *F(0)* through *F(n ).*  Recall that the *Fibonacci* sequence is deﬁned by the relation

*F (n) = F (n−1) + F (n−2),* where *F (0) = 0 and F (1) = 1.*

*Sample input/output:*

Enter the range of the Fibonacci series: 15

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

Press any key to continue . **. .**