CpE 221 - Data Structures and Algorithms Analysis

[NOTE: The Difficulty of the C++ Programs are not arrange.]

C++; P.S. - All of the programs uses the common functions:

cout, cin, system("PAUSE").

cout<<""; - printing

cin>>; - scan a data type

system("PAUSE"); - pausing the screen or get character or scanning char

(?) - Unfinished or uncertain programs.

// (0000000000) Oban Numbers,

// Example: 4, Four - Oban

// (stack, class function) <CpE221/Engl1>

// (0000000001) Fibonacci Series in classes C++. <CpE221/Mathematics>

// (class function)

// (0000000010) Conversion (Decimal to Binary, Octal, Hexadecimal) <CpE221/CpE321>

// (class function)

// (0000000011) Area of Rectangle <CpE221/Math0>

// Turbo C functions (getch)

// (0000000100) Decimal to Hexadecimal <CpE221/CpE321>\

// (class function)

// (0000000101) Converting decimal points to whole <CpE221/Mathematics>

// (class function)

// (0000000110) Number to Roman Numerals <CpE221/Mathematics>

// (class function)

// (0000000111) VACANT <CpE221/Mathematics>

// (0000001000) Checking if your equation is valid <CpE221/Mathematics>

// (class function)

// (0000001001) Enter Input then,

// [A] - Integer Ascending Sort

// [B] - Decimal Descending Sort <CpE221/Mathematics>

// (class function)

// (0000001010) Number (Whole or Decimal Places) to

// Roman Numerals <CpE221/Mathematics>

// (class function)

// (0000001011) Sorting Numbers <CpE221/Mathematics>

// (0000001100) Reversing Unilimited Inputs <CpE221/Mathematics>

// (dynamic array)

// (0000001101) Longest Word <CpE221/English>

// (class function, string)

// (0000001110) VACANT <CpE221/Mathematics>

// (0000001111) Queue (Enqueue/Dequeue) <CpE221/Mathematics>

// (class function)

// (0000010000) Stack (Push/Pop) <CpE221/Mathematics>

// (class function)

// (0000010001) VACANT <CpE221/Mathematics>

// (0000010010) Converting Uppercase to Lowercase vice versa <CpE221/Mathematics>

/// (0000010011) Wasteful number in Queue <CpE221/Mathematics>

// (class function)

// (0000010100) Add all digit and one digit from decimal places <CpE221/Mathematics>

// Ex: 1.2 = 3

// (0000010101.0001) Complpete function Singly-List <CpE221/Mathematics>

// (class function)

// (0000010101.0010) Selected function Singly-List <CpE221/Mathematics>

// (class function)

// (0000010110) Happy Number <CpE221/Mathematics>

// (class function)

// (0000010111) Input numbers then determine the maximum

// number and locate what position. <CpE221/Mathematics>

// (0000011000) Input numbers then determine the

// composite and determine the largest number of the

// composite and it's prime factors <CpE221/Mathematics>

// (0000011001) Double Link List <CpE221/Mathematics>

// (class function)

// (0000011010) VACANT <CpE221/Mathematics>

// (0000011011) VACANT <CpE221/Mathematics>

// (0000011100) Rounding off numbers <CpE221/Mathematics>

// (0000011101) Look and say! <CpE221/Mathematics>

// (class function)

// (0000011110) VACANT <CpE221/Mathematics>

// (0000011111) Determine words if anagram <CpE221/English>

// (class function, pointers)