CYPHERCRESCENT SOFTWARE ACADEMY ASSIGNMENT ON ALGORITHMS

BY ISAAC DESMOND

* ALGORITHM TO ADD TWO NUMBERS
* Start
* Input an integer number A
* If non-integer number is entered by user, ask user to repeat input.
* Input an integer number B
* If non-integer number is entered by user, ask user to repeat input.
* Define an output integer Sum to take the sum of the two numbers.
* Sum = A + B
* Print Sum
* Stop
* ALGORITHM TO FIND THE FACTORIAL OF A NUMBER
* Start
* Enter a positive integer number Num
* Declare an integer variable – Factorial and give it an initial value of 1.
* Check that user entered integer number and prompt them to try again if not.
* Initiate a loop
  + While Num is greater than zero
  + Multiply Factorial by Num and store result in Factorial
  + Subtract 1 from Num.
* End While Loop
* When Num is 0 then Factorial should be 1.
* Print out Factorial
* Stop
* ALGORITHM TO FIND THE FIBONACCI SEQUENCE
* Start
* Enter a positive integer number Num
* Check that Num is a positive integer and prompt the user to try again if it is not
* Declare an integer variable CurrentNum and give it a value of 0
* Declare an integer variable TempVal and give it a value of 0
* Declare an integer variable NextNum and give it a value of 1
* Print out CurrentNum
* Initiate a loop
  + While CurrentNum is not Greater than Num
  + Store the value of NextNum in TempVal
  + Add CurrentNum to NextNum and store the result in NextNum
  + Store the value of TempVal in CurrentNum
  + Print out CurrentNum
* End loop
* Stop
* ALGORITHM TO USE BUBBLE SORT
* Start
* Declare a variable Temp and give it a value of 0
* Ask the user to enter a positive integer Length
* Check that Length is a positive integer and prompt the user to try again if it is not
* Declare an array of integers with a length equal to Length and name it Nums
* Use a loop to gather the elements to be stored in the array
* Check that each element provided is an integer and prompt the user to try again if it is not.
* Initiate a loop to run as many times as the Length of the array minus 2
  + Initiate a loop to run as many times as the Length of the array minus 2
    - When the value of the element in the current index is greater than the one of the next index after it then;
      * The value of the element in the current index should be stored in Temp
      * The value of the element stored in the index after the current index should be stored in the current index.
      * The value of Temp should then be stored in the index after the current index.
  + End loop
* End loop
* Print out the sorted array using a loop
* Stop
* ALGORITHM TO CONVERT BETWEEN BASES
* Prompt the user to enter the base of the number they’d like to convert and store it in a variable: basePrime
* Check that the value entered is valid.
* Prompt the user to enter the base they’d like to convert to and store it in a variable: endBase
* Check that the value entered is valid
* Prompt the user to enter the Number they’d like to convert and store it in a variable: Num
* Check that the value entered is valid
* Declare a variable: No
* Declare a variable Decimal and another count giving them both a value of 0
* Declare an array of integers named ToDecimal and another array named ToBases
* Initiate a loop
  + While No is greater than 0
  + Store the remainder of No divided by 10 in the array ToDecimal
  + Add 1 to count
* End Loop
* Initiate a Loop to run through the array ToDecimal in reverse order from count to 1
  + Take the values stored in to Decimal multiplied by basePrime raised to the power of the elements index and add the result to Decimal
* End Loop
* Add 1 to Decimal
* Make count 0 again
* Initiate a Loop
  + While Decimal is greater than 0
  + Store the remainder of decimal divided by base in the array ToBases
  + Add 1 to count
* End loop
* Print out ToBases in reverse order that is from the index of count to 0
* Stop