**ALGORITHM EXERCISES**

2. FACTORIAL OF A NUMBER

* Start
* Declare an integer number, num
* Declare an integer number, Factorial and initialize it as 1
* Enter a positive integer number, num
* Check that the user entered a positive integer number and prompt them to try again if not
* Initiate loop
  + While num is greater than zero
    - Multiply factorial by num and store the result in factorial
    - Subtract 1 from num
  + End while loop
* Print Factorial
* end

3. FIBONACCI SEQUENCE

Constraints

1. The Fibonacci number is the sum of the two positive integer numbers in the sequence before it
2. The kick off point for any Fibonacci number is F0 equals 0 and F1 equals 1
3. Therefore, finding the Fibonacci number starts from 2 upwards

Algorithm

* Start
* Declare an integer array variable Fn for the Fibonacci number
* Initialize Fn index 0 as 0 and index 1 as 1
* Declare an integer variable n that will hold the user input
* Input a positive integer number, n
* Check if n is greater than or equal to 2. If not, prompt the user to try again
* For integer i equals 2; i increment of 1; i is less than or equal to n
  + Add Fn index [i minus 1] to Fn index [i minus 2] and store the result in Fn index i
* End loop
* Print Fn
* End

4. BUBBLE SORT

* Declare an integer variable n to hold the number of values to be sorted
* Input the number of values n to be sorted
* Check that the user entered a positive integer number and prompt them to try again if not
* Declare an integer array variable called values to hold the values to be sorted
* Declare an integer variable value to hold each value the user inputs
* For integer i equals 0; i increment of 1; i is less than n
  + Print string enter each value to be sorted
  + Input each value
  + values index i equals value
* End loop
* For integer j equals 0; j increment of 1; j is less than n
  + For integer i equals 0; i increment of 1; i is less than n-1
    - If values index i is greater than values index i+1
      * values index i equals values index i+1
      * values index i+1 equals values index i
    - End if
  + End loop
* End loop
* For i equals 0; i increment of 1; i is less than n
  + Print values index i
* End loop

BASE CONVERSION

* Start
* Declare an integer variable n to hold the number to be converted
* Declare an integer variable Sbase to hold the source base
* Declare an integer variable Tbase to hold the target base
* Input the number n to be converted
* Input the source base Sbase
* Input the target base Tbase
* Check that the user entered positive integer numbers for n, Sbase and Tbase. Prompt them to try again if not
* If Sbase is equal to 10
  + Convert n to Tbase
* Else
  + Convert n to base 10
  + Convert n to Tbase
* End if
* End