1. **Algorithm to add two integer numbers**

* Start
* Input an integer number A
* If non-integer number is entered, prompt user to input integer number
* Input an integer number B
* If non-integer number is entered, prompt user to input integer number
* Define an output integer sum to take the sum of the two numbers
* Sum=A+B
* Print sum
* Stop

1. **Algorithm to find the factorial of a number**

* Start
* Input an integer number Num
* Check that user input integer number which we are required to find its factorial
* Declare an integer variable-Factorial and give it an initial value of 1, to account for product operation
* 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
* Print out factorial
* Stop

**3. Algorithm for Fibonacci Number Series**

* Start
* Declare real variable and assigned first Fibonacci number
* Declare integer variable and assigned N-Fibonacci series
* Check that user entered integer variable of N-Fibonacci series
* Declare integer variable and initialized it to zero for effective addition operation
* Declare an array or list of real variables and resized it to accommodate N-Fibonacci Series
* Initiate a loop
  + Loop through from the first Fibonacci number to N-Fibonacci Series
    - Add initialized zero value to the first Fibonacci number
    - Give the first Fibonacci number to the initialized zero value
    - Give the newly added Fibonacci to the First Fibonacci number
    - Stored the generated Fibonacci number to the array or list
* End loop
* Print out the generated Fibonacci series
* Stop

**4. Algorithm to Sort an Array using Bubble Sort**

* Start
* Declare and input array of real number to be sorted
* Declare and input integer number of array size N
* Initiate a loop for cycle of swap
  + Initialized condition of false swap
  + Loop through from first element to N-array size
    - Compare the 1st and second 2nd and swap if 1st element is greater than 2nd element. Same comparison should be done between 2nd and 3rd elements and so on
    - Swap by replacing 1st element with 2nd element and replacing 2nd element with 1st element and so on till (N-1) times.
  + End loop
* Loop and perform same operations in all subsequent cycle until no swapped again
* Print out the sorted list or array
* Stop

**5. An Algorithm to convert a given number from one base to another**

* Start
* Input integer DecimalNumber and its base
* Input integer Base
* If non-integer DecimalNumber are entered, prompt the user to input integer DecimalNumber
* If non-integer Base are entered, prompt the user to input integer Base
* If a DecimalNumber, number in base 10 is to be converted to other base
  + Initiate a loop
  + Initialized a character of string to empty string
  + While decimal input number is greater than zero
    - StoredCharacters = Decimalnumber % Base. Ensure to repeatedly divides the decimal input number by Base and take the remainder using modular division.
    - StoredCharacters+= StoredCharacters
    - Decimalnumber/=base. To update the DecimalNumber after subsequent division
  + End While
  + Print out StoredCharacters
* If OtherNumberBase, number in any base is to be converted to base 10
  + Input integer OtherNumberbase and integer Base10
  + If non-integer OtherNumberbase are entered, prompt the user to input integer OtherNumberbase
  + If non-integer Base10 are entered, prompt the user to input integer Base10
  + If the inputted OtherNumberbase is greater than the Base10, prompt the user to enter base less than Base10
  + Convert the OtherNumberbase from integer variable to String characters
  + Get the number of characters N, in the OtherNumberbase
  + Initialized power of base to 1
  + Define an output integer SumNum and initialized it to zero
  + Initiate a loop
  + Loop from the assumed index of the first string character of OtherNumberBase to the index of the last string character of OtherNumberbase
    - SumNum+=OtherNumberbase\_index\*10 raised to power index
    - Power=power\*Base10
  + End Loop
  + Print out SumNum
* If is to convert a given number from a base to another base
  + Input Integer Number and NumberBase and Otherbase
  + Generate SumNum by Converting the Number to Base10
  + Generate StoredCharacters by converting the SumNum to OtherBase
  + Print out the StoredCharacters
* Stop