Linear Search and Binary Search : Iterative Versions

```
#-------------------------------------------------------------------------------

# Linear Search of a list with  n  items :
#
#    number of list items inspected ( worst case )  =  n  .

#-------------------------------------------------------------------------------

def LinearSearch( item, lst ) :

    # A position in 'lst' in which 'item' occurs, or 'None' if not present

    for pos in range( len( lst ) ) :
        if item == lst[ pos ] :
            return pos

    return None

#-------------------------------------------------------------------------------
```

```
#-------------------------------------------------------------------------------

>>> #       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
>>> L1 = [ 53, 76, 31, 12, 57, 85, 73, 49, 44, 60, 93, 18, 65, 96, 23 ]

>>> LinearSearch( 73, L1 )
6

>>> LinearSearch( 59, L1 )
None

#-------------------------------------------------------------------------------
```

```
#-------------------------------------------------------------------------------

# Binary Search of a list with  n  items :
#
#    number of list items inspected ( worst case )  =  log n [base 2]  (approx)

#-------------------------------------------------------------------------------

def BinarySearch( item, lst ) :

    # A position in 'lst' in which 'item' occurs, or 'None' if not present
    #
    # NOTE : 'lst' must already be sorted in ascending order

    lo = 0                  # INVARIANT : always searching between
    hi = len( lst ) - 1     #             lst[ lo ]  and  lst[ hi ]

    while lo <= hi :
        mid = ( lo + hi ) // 2
        if   item < lst[ mid ] :
            hi = mid - 1
        elif item > lst[ mid ] :
            lo = mid + 1
        else :
            return mid

    return None

#-------------------------------------------------------------------------------
```

```
#-------------------------------------------------------------------------------

>>> #       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
>>> L2 = [ 12, 18, 23, 31, 44, 49, 53, 57, 60, 65, 73, 76, 85, 93, 96 ]

>>> BinarySearch( 73, L2 )
10

>>> BinarySearch( 59, L2 )
None

#-------------------------------------------------------------------------------
```