Indexing Sequences

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

def Print( s ) :

    # Print the items of the sequence 's', one per line  [ see Handout #4 ]

    for i in range( len( s ) ) :
        print( s[ i ] )

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

def PrintReversed( s ) :

    # Print the items of the sequence 's', one per line, in reverse order

    lastindex = len( s ) - 1

    for i in range( len( s ) ) :
        print( s[ lastindex - i ] )

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

def IsSorted( s ) :

    # Is sequence 's' sorted in ascending order?  [ see Handout #7 ]

    for i in range( len( s ) - 1 ) :
        if s[ i + 1 ] < s[ i ] :
            return False

    return True

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

def AddLists( numbers1, numbers2 ) :

    # The list formed by adding corresponding items
    # in numeric lists 'numbers1' and 'numbers2';
    # assume that both lists have the same number of items

    return [ numbers1[ i ] + numbers2[ i ] for i in range( len( numbers1 ) ) ]

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

def Rotate( lst, n ) :

    # A copy of list 'lst' rotated leftwards by 'n' items;
    # assume that  0 <= n <= len( lst ) - 1

    rotate = [ ]

    for i in range( n, len( lst ) ) :
        rotate += [ lst[ i ] ]

    for i in range( 0, n ) :
        rotate += [ lst[ i ] ]

    return rotate

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

```
>>> Print( "bread" )
b
r
e
a
d


>>> PrintReversed( "bread" )
d
a
e
r
b


>>> IsSorted( "abcdefg" )
True

>>> IsSorted( "aaaaaaa" )
True

>>> IsSorted( "abcedfg" )
False


>>> AddLists( [ 3, 1, 5 ], [ 2, 8, 2 ] )
[5, 9, 7]

>>> AddLists( [ ], [ ] )
[]


>>> Rotate( [ 5, 2, 7, 4, 1, 8 ], 3 )
[4, 1, 8, 5, 2, 7]

>>> Rotate( [ 5, 2, 7, 4, 1, 8 ], 0 )
[5, 2, 7, 4, 1, 8]

>>> Rotate( [ 5, 2, 7, 4, 1, 8 ], 5 )
[8, 5, 2, 7, 4, 1]
```