**Design 01 Sort Algorithm – Selection Sort**

Begin Sub

For PassNumber = 0 to ListSize - 2 ‘loop from beginning to last bit one element.

‘find smallest element in unsorted range.

SmallestPos = PassNumber ‘initialise smallest as leftmost unsorted element.

‘Loop through rest of unsorted range to find smallest

For Position = PassNumber + 1 to List Size - 1 ‘loop from next unsorted to end.

If List(Position) < List(SmallestPos) Then

SmallestPos = Position

Next Position

‘if element as PassNumber isn’t smallest, move it to there.

If PassNumber <> SmallestPos Then

‘swap elements

Temp = List(SmallestPos)

List(SmallestPos) = List(PassNumber)

List(PassNumber) = Temp

End If

‘at this point, List from 0 to PassNumber is sorted

Next PassNumber

End Sub

**Design 02 Sort Algorithm – Quick Sort**

Begin Sub as Quicksort(List)

If length(array) > 1 Then ‘if the given sub-list is not empty (for recursion)

‘define variables from list given

Pivot = first element of array List

Start = first index of List

End = last index of List

While Start <= End

‘increment Start until we find an item larger than Pivot

While List(Start) < Pivot

Start += 1

End While

‘decrement End until we find an item smaller than Pivot

While List(End) > Pivot

End -= 1

End While

If Start <= End Then ‘if not in sorted

‘swap the two elements

Temp = List(Start)

List(Start) = List(End)

List(End) = Temp

‘adjust variables after swap

Start += 1

End -= 1

End If

End While

‘call this procedure again to sort ever-so-smaller lists (until completely sorted)

Quicksort(List from first index to End)

Quicksort(List from Start to last index

End If

End Sub