Question 1 (9)

BINARYSEARCH(sortedList, LowerNumber, UpperNumber, midpoint)

‘’’takes a sorted list of integers. Two integers that specify the interval we are searching for. And an integer calculated by the starting index of the list + (the size of the list – the starting index of the list)/2. It will be used to index the midpoint. The function will return a Boolean value.‘’’

While valueFound = False and endOfList = False

if sortedList[midpoint] < UpperNumber and sortedList[midpoint] > LowerNumber

valueFound <- True

else if sortedList[midpoint] = UpperNumber or sortedList[midpoint] = LowerNumber

valueFound <-True

else if sortedList[midpoint] > UpperNumber

end <- midpoint – 1

if start = end

endOfList <- True

else

midpoint <- start + ( end – start)/2

round down midpoint

//else if sortedList[midpoint] < LowerNumber

else

start <- midpoint + 1

if start = end

endOfList

else

midpoint <- start + (end - start)/2

round down midpoint

return valueFound

// to do this with recursion remove the part of the list that is no longer searched until interval found or no more list.

1,2,3,4,6 7,8,9, 10

5+(5-5)/2

See if mid point is higher than higher bound, if it is, repeat process for lower half

If midpoint is lower than lower bound, repeat process for upper half

Return true if there are any numbers in that interval, false if there are not.

5-6