2

BINARY SEARCH (with a pseudo code)

(15 points) Consider the list of ten numbers that follows this problem statement. First sort the list. Then use a binary search to search it. Your search algorithm should indicate whether the number is in the list and should also note its position. Sample output might look like this:

234 is in the list at position 7
78 is not in the list

Search the following list for 99, 183, 225, 642, and 999.

The list: 351, 499, 264, 506, 530, 219, 102, 183, 642, 512

A binary search works by comparing the target to be found with the middle entry in a sorted list. If they match, the search is over. Otherwise, if the target is less than the middle entry, the search can be confined to the first half of the list. Similary, if the target is greater than the middle entry, the search is confined to the second half of the list. In either case the same logic is then applied to the middle entry in the new list. Consider the following example where

Low = 1
High = 7
Middle = (Low + High)/2 (Using integer arithmetic)

(OVER)

#2 continued

```
Target, the number to be found, is 89
First search of List() for 89
12
14
26
45  <== (Middle = 4), Target > List(Middle)
57
67
89

Second search of List() for 89
12
14
26
45
-------
57
67  <== (Middle = 6), Target > List(Middle)
89

Third search of List() for 89
12
14
26
45
57
67
-------
89  <== (Middle = 7), Target = List(Middle)
```

A pseudocode version of a correct binary search algorithm might take this form:

```
Found = false
Low = 1
High = N
While (not Found) and (Low <= High) Do
   Middle = (Low + High)/2
   If Target < List(Middle) Then
      High = Middle - 1
   Else
      If Target > List(Middle) Then
         Low = Middle + 1
      Else
         Found = true
         Position = Middle
      Endif
   Endif
Endwhile
```