TOPIC 9 EXERCISES

Tracing Exercises

1. Trace the selection sort algorithm on each of the following lists of items. Then do the same for the bubble sort algorithm.

```
(a)  n = 6   45  32  17  6  1  0
(b)  n = 5   3   -8   129   2   3
(c)  n = 7   -1  -76  508  -87  -34  124  -54
(d)  n = 4   2  1  18  5
(e)  n = 8   's'  'x'  'q'  'u'  'b'  'a'  'V'  'T'
(f)  n = 5   12.3  154.56  97.5  3.37  -5.32
(g)  n = 6   9  9  6  1  -3  2
(h)  n = 13  "WenEng" "Ayshea" "Hong" "Alex" "Dmitry" "Ari" "Yukie"
             "Malky" "Henry" "Vitaly" "Ezhar" "Boris" "Yuriy"
```

2. Given the following set of strings, sort them in ascending alphabetical order. Does it matter whether we use a selection or a bubble sort?

```
"a"     "b"     "an"     "a  "   "a "    " a"
" an"   "and"   "Bob"    "bob"   " Bob"  "an "
```

3. In the selectionSort() method, the inner **for** loop header looks like this:

```
for (int cand = pass + 1; cand < n; cand++)
```

What happens if this is changed to each of the following?

```
(a)  for (int cand = pass + 1; cand <= n; cand++)

(b)  for (int cand = pass; cand < n; cand++)

(c)  for (int cand = pass + 1; cand < n - 1; cand++)

(d)  for (int pass = pass + 1; pass < n; pass++)
```

If you are unsure of an answer, try running a program using each header on a set of data.

Analysis of the Sorting Algorithms

4. In both the selection and the bubble sort, we do not swap two elements if they are equal but only if the first is greater than the second.

(a) In the selection function, what happens if we replace the test

```
if (numb[pass] > numb[cand])          by

if (numb[pass] >= numb[cand])   ?
```

(b) In the bubble sort function, what happens if we replace the test

```
if (numb[pos] > numb[pos + 1])        by

if (numb[pos] >= numb[pos + 1])   ?
```

Run a program for each part on a simple example with a repeated value (e.g., 5 7 3 5). The results may surprise you.

Analysis of the Searching Algorithms

5. Assume that an array vals of integers holds the following numbers: 3 7 12 14 16 35 123. Here is the list of values whose positions (if any) in vals are to be located: 8 16 12 26 112 35 6 7 14 13.

(a) Trace the linear search algorithm in processing this set of data. Find the average number of comparisons required to locate an integer in the array and the number needed to show that a new value is not present.

(b) Repeat part (a) using the binary search algorithm.

(c) Compare your answers from parts (a) and (b). Is this a good test of the two algorithms? Explain.