Exercise 4-1: Visualization of bubble sort

- Modify sort_bubble.c and create a program sort_bubble1.c that visualizes bubble sort.
- Also, count and display the number of comparisons and the number of replacements.
  - ex)
    - "*" Before the reference element
    - ">" Before the element to be compared
    - [Comparison count] and [Replacement count] are displayed at the beginning of the array.

```
[ 1][ 0] *8 >2 7 4 5 6 9 0 1 3
[ 2][ 1] 2 *8 >7 4 5 6 9 0 1 3
[ 3][ 2] 2 7 *8 >4 5 6 9 0 1 3
[ 4][ 3] 2 7 4 *8 >5 6 9 0 1 3
[ 5][ 4] 2 7 4 5 *8 >6 9 0 1 3
[ 6][ 5] 2 7 4 5 6 *8 >9 0 1 3
[ 7][ 5] 2 7 4 5 6 8 *9 >0 1 3
[ 8][ 6] 2 7 4 5 6 8 0 *9 >1 3
[ 9][ 7] 2 7 4 5 6 8 0 1 *9 >3

      ....


[38][24] 2 0 *4 >1 3 5 6 7 8 9
[39][25] 2 0 1 *4 >3 5 6 7 8 9
[40][26] *2 >0 1 3 4 5 6 7 8 9
[41][27] 0 *2 >1 3 4 5 6 7 8 9
[42][28] 0 1 *2 >3 4 5 6 7 8 9
[43][28] *0 >1 2 3 4 5 6 7 8 9
[44][28] 0 *1 >2 3 4 5 6 7 8 9
[45][28] *0 >1 2 3 4 5 6 7 8 9
```

- The origin of "bubble" is from the appearance that the larger values move (emerge) to the edge in order.
- Bubble sort always makes n (n -1) / 2 comparisons