Question 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass # |  |  |  |  |  |
| 0 | Renee | Brien | Vincent | Doris | Scarlett |
| 1 | Brien | Renee | Vincent | Doris | Scarlett |
| 2 | Brien | Doris | Vincent | Renee | Scarlett |
| 3 | Brien | Doris | Renee | Vincent | Scarlett |
| 4 | Brien | Doris | Renee | Scarlett | Vincent |
| Finished | Brien | Doris | Renee | Scarlett | Vincent |

Question 3

a) Check if the smallest value in the unsorted subarray is equal to the element one cell to the right of the sorted subarray. If the values are equal, then do not swap the elements.

b) It is probably more efficient to swap an element with itself than to check whether the elements are the same. Especially for non-primitive data types such as Strings, swapping an element with itself takes less operations in comparison, and the programmer doesn’t have to think or type as much either.

Question 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pass # |  |  |  |  |  |  |  |
| 0 | 6 | 2 | 8 | 3 | 1 | 7 | 4 |
| 1 | 6 | 2 | 8 | 3 | 1 | 7 | 4 |
| 2 | 2 | 6 | 8 | 3 | 1 | 7 | 4 |
| 3 | 2 | 6 | 8 | 3 | 1 | 7 | 4 |
| 4 | 2 | 3 | 6 | 8 | 1 | 7 | 4 |
| 5 | 1 | 2 | 3 | 6 | 8 | 7 | 4 |
| 6 | 1 | 2 | 3 | 6 | 7 | 8 | 4 |
| 7 | 1 | 2 | 3 | 4 | 6 | 7 | 8 |
| Finished | 1 | 2 | 3 | 4 | 6 | 7 | 8 |

Question 7

An unstable insertion sort differs from a stable insertion sort because the unstable one makes an “equal to or greater than” comparison when swapping, whereas the stable one makes a “greater than” comparison only.

For example, for the unsorted integer array [**3**, **1**, **4**, **5**, **4**], the two algorithms will treat the **4**s differently (marked in green and yellow):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass # | Unstable Insertion Sort | | | | |
| 0 | **3** | **1** | **4** | **5** | **4** |
| 1 | **1** | **3** | **4** | **5** | **4** |
| 2 | **1** | **3** | **4** | **5** | **4** |
| 3 | **1** | **3** | **4** | **5** | **4** |
| 4 | **1** | **3** | **4** | **4** | **5** |
| Done | **1** | **3** | **4** | **4** | **5** |

After the orange **4** swaps with **5**, the comparison between **4** and **4** is made, resulting in a swap. Even though the **4** was originally in the correct place, the sort still changed its location.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass # | Stable Insertion Sort | | | | |
| 0 | **3** | **1** | **4** | **5** | **4** |
| 1 | **1** | **3** | **4** | **5** | **4** |
| 2 | **1** | **3** | **4** | **5** | **4** |
| 3 | **1** | **3** | **4** | **5** | **4** |
| 4 | **1** | **3** | **4** | **4** | **5** |
| Done | **1** | **3** | **4** | **4** | **5** |

After the orange **4** swaps with **5**, the comparison between **4** and **4** is made, resulting in no swap. The **4** was remains in its original position and in the correct place.

Question 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pass # |  |  |  |  |  |
| 0 | 3 | 8 | 3 | 2 | 7 |
| 1 | 3 | 3 | 2 | 7 | 8 |
| 2 | 3 | 2 | 3 | 7 | 8 |
| 3 | 2 | 3 | 3 | 7 | 8 |
| 4 | 2 | 3 | 3 | 7 | 8 |
| Finished | 2 | 3 | 3 | 7 | 8 |

Question 9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pass # |  |  |  |  |  |  |  |
| 0 | 2 | 9 | 4 | 6 | 1 | 7 | 8 |
| 1 | 2 | 4 | 6 | 1 | 7 | 8 | 9 |
| 2 | 1 | 2 | 4 | 6 | 7 | 8 | 9 |

Question 11