

## Programming 1B Practice Test- ICE Task 2

### Question 1:

1. A
2. B
3. C
4. A, B, C, D
5. B
6. C
7. A
8. C
9. A, B, C
10. "Value : b4"

11. "Value : 1.0"  
"Value : 9.1"  
"Value : 2.3"  
"Value : 6.8"  
"Value : 3.5"

12. " 1: 0.5"  
" 2: 2"  
" 3: 4.5"  
" 4: 8"  
" 5: 12.5"

13. " Value : 1"  
" Value : 4"  
" Value : 7"  
" Value : 10"

14. a. Correct – Output: "2, 4, 6, 8, "

14.b Incorrect –The code will run however the string array is never given values and therefore will output as null

- 14.c Correct – Output: "0, 2, 4, 6, 8, 10, "

### Question 3:

3a) Bubblesort is a sorting algorithm that sorts and orders data in pairs. It works from left to right starting with positions 1 and 2, comparing them to find which is bigger. Once it identifies the bigger integer, if working in ascending order, it either swaps the integers or leaves them as is, so that the larger integer is on the right in position 2. Once that pair is sorted it then moves 1 position to compare positions 2 and 3, once again ordering it so that the bigger integer ends up on the right position, in this case position 3. It proceeds like this until it reaches the end of the array. Once it reaches the end it starts back at position 1 and runs through the array again. It does this repeatedly until all the data is sorted in order.

3b) It is the shortest time it takes the code to execute

3c) It is the time it takes to execute the code as well as how much memory the code uses when running

3d) It is an in-place sorting algorithm

3e)  $O(n)$

3f)  $O(n^2)$

3g)  $O(n^2)$

### Question 4:

4a) It is a comparison-based algorithm that works through an array from left to right that sorts data as it goes.

4b) Insertion Sort works by starting with the value in index(0) and categorizes it as sorted. It then looks at the next value and places it in the sorted list according to where it needs to be. If working in ascending order the next value will be placed before the previous value if smaller and after it if larger. If there are 2 values sorted and the new value is larger than one and smaller than the other, it will be placed in between the 2. This pattern repeats until all the data in the list is sorted.

4c) Time Complexity:

-Best Case: ( $O(n)$ )

-Average Case: ( $O(n^2)$ )

-Worst Case: ( $O(n^2)$ )

Space Complexity: ( $O(1)$ )

4d) When you want a shorter run time or less iterations of the code

4e)