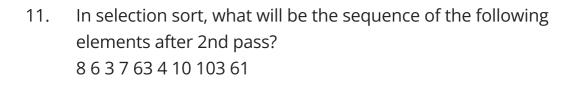
DS Unit-1 Test-3 14 Questions		NAME: CLASS: DATE:						
					1.	Which of the following is not a stable sorting algorithm?		
A	Bubble sort	B Selection sort						
С	Quick sort	D None of these						
2.	Which of the following is/are in-place sorting algorithm(s)?							
Α	Selection sort	B Bubble sort						
С	Quick sort	D Merge sort						
3.	The worst-case time complexity of a quick sort algorithm?							
А	O(nlogn)	B O(logn)						
С	O(n^2)	D O(n^2logn)						
4.	The no.of comparisons in first pass in bubble sort to sort n numbers is?							
Α	n	B n(n-1)/2						
С	n-1	D n^2						
5.	The maximum no.of swappings in first pass of selection sort to sort n numbers is?							
Α	1	B n(n-1)/2						
С	n-1	D n^2						

	В	2			
3		_			
)	D	4			
If the list is almost sorted, then which of the following algorithm gives the better performance?					
Bubble sort	В	Selection sort			
Quick sort	D	All of these			
If the list is almost sorted, then which of the following algorithm is/are not suitable?					
Bubble sort	В	Selection sort			
Quick sort	D	All of these			
n bubble sort, what will be th after 7 passes? 20 15 14 45 11 18 78 95	e sequ	ience of the following elements			
n bubble sort, what will be th after 1st pass? 2 1 4 45 11 18 78 95	e sequ	ience of the following elements			
	Bubble sort Quick sort Ithe list is almost sorted, to algorithm is/are not suitable Bubble sort Quick sort Quick sort In bubble sort, what will be the fter 7 passes? Ithe 15 14 45 11 18 78 95 In bubble sort, what will be the fter 1st pass?	f the list is almost sorted, then walgorithm gives the better performable sort Bubble sort Guick sort The list is almost sorted, then walgorithm is/are not suitable? Bubble sort Bubble sort Bubble sort Duick sort D			

In quick sort, based on key or pivot element position, the list is

6.



12. The no.of passes in selection sort to sort **n** numbers is?

A n

B n-1

C n^2

D logn

13. The no.of swapings in selection sort to sort **n** numbers is?

A n

B n-1

C n^2

D logn

14. Write C code to swap two integers **a** and **b** by using a temporary variable **t**?

Answer Key

1. b

2. a, b, c

3. c

4. c

5. a

6. b

7. a

8. b, c

9.11 14 15 18

101 2 4 11 18

118 6 3 7 61 4

12. b

20 45 78 95

45 78 95

10 63 103

13. b

14. t=a;a=b;b=t;