NAME : _____

CLASS : _____

DS UNIT-2 TEST-3

10 Questions

DATE : _____

1. Which of the following operations is performed more efficiently by doubly linked list than by linear linked list?

A Traversing the list to process each node

B Inserting a node after the node with a given location

C Searching an unsorted list for a given item

D Deleting a node whose location is given

2. The minimum number of fields with each node of doubly linked list is
A) in a normal case
B) in an optimal way

A 4, 4

B 2 ,3

C 1 ,2

D 3, 2

?

3. A doubly linked list is declared as
struct Node {
int Value;
struct Node *Fwd;*
struct Node Bwd;);
Where Fwd and Bwd represent forward and backward link to the adjacent elements of the list. Which of the following segments of code deletes the node pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last node of the list?

A. X->Bwd.Fwd = X->Fwd ; X.Fwd->Bwd = X->Bwd ;

B. X->Bwd->Fwd = X->Fwd; X->Fwd->Bwd = X->Bwd ;

C. X.Bwd->Fwd = X.Bwd ; X->Fwd.Bwd = X.Bwd ;

D. X->Bwd->Fwd = X->Bwd ; X->Fwd->Bwd = X->Fwd;

4. Consider a singly linked list of the form where F is a pointer to the first element in the linked list and L is the pointer to the last element in the list. The time of which of the following operations depends on the length of the list?

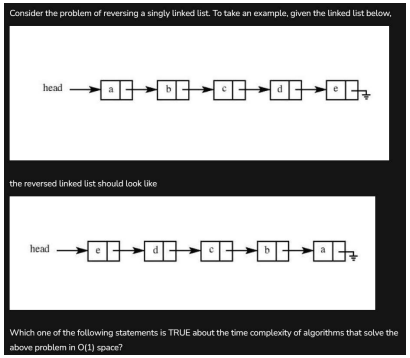
A. Add an element after the last element of the list

B. Delete the last element of the list

C. Delete the first element of the list

D. Interchange the first two elements of the list

5.



See the image and answer the question

A It is not possible to reverse a singly linked list in O(1) space.

B The best algorithm for the problem takes theta(n^2)
theta time in the worst case

C The best algorithm for the problem takes
theta(n)
theta time in the worst case

D The best algorithm for the problem takes
theta(n logn)
theta time in the worst case

6. Correct Program to find Middle of a Linked list ?

A

```
class Node:
  def init(self, k):
    self.data = k
    self.next = None
def printList(head):
  curr = head
  while curr != None:
    print(curr.data)
    curr = curr.next
  print()
  def printMiddle(ptr):
  if head == None:
    return
  count = 0
  curr = head
  while curr :
    curr = curr.next
    count+=1
   curr = head
  for i in range (count//2):
    curr = curr.next
  print(curr.data)
head = Node(10)
head.next = Node(10)
head.next.next = Node(20)
printList(head)
printMiddle(head)
```

B

```
class Node:
  def init(self, k):
    self.data = k
    self.next = None
def printList(head):
  curr = head
  while curr != None:
    print(curr.data)
    curr = curr.next
  print()
def printMiddle(ptr):
  if head == None:
    return
  count = 0
  curr = head
  while curr :
    curr != curr.next
    count+=1
   curr = head
  for i in range (count//2):
    curr = curr.next
  print(curr.data)
  head = Node(10)
head.next = Node(10)
head.next.next = Node(20)
printList(head)
printMiddle(head)
```

7. **Which of the following problems can be solved using 2 pointers on linked list?**

A Finding intersection of two linked lists

B Detecting cycle in a linked list

C Finding middle element of a linked list

8. **Which of the following is optimal to find an element at kth position at the linked list?**

A Single Linked List

B Double Linked List

C Circular Linked List

D Array implementation of Linked List

9. **The type of pointer used to point to the address of the next element in a linked list?**

A pointer to node

B pointer to character

C all of the above

D pointer to integer

10. **A linked list in which none of the nodes contains a NULL pointer is?**

A Circular Double Linked List

B Double Linked List

C Single Linked List

D Circular Single Linked List

Answer Key

1. d	2. d	3. b	4. b
5. c	6. a	7. b, a, c	8. d
9. a	10. d, a		