

Department of Electrical Engineering
Assignment
Date: 25/09/2020

Course Details

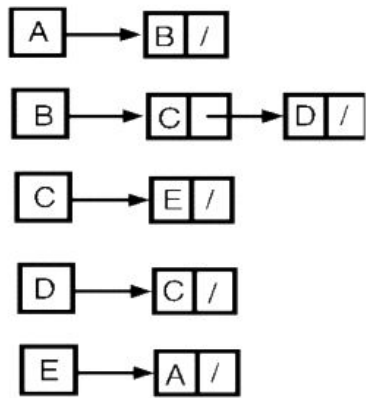
Course Title: <u>Data Structure and Algorithm</u>	Module: _____
Instructor: _____	Total Marks: <u>50</u>

Student Details

Name: _____ **Student ID:** _____

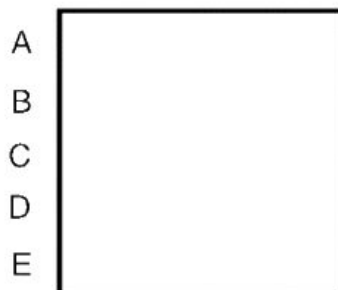
Note: Plagiarism of more than 20% will result in negative marking.
Similar answers of students will result in cancellation of the answer for all parties.

Q1.	Explain the following operations in a single link list. 1. Insert an Element 2. Delete an Element	CLO 1 Marks 10
Q2.	Write a program to insert a new element in the given unsorted array at K th position	CLO 2 Marks 12
Q3.	Explain Quick sort with the help of suitable example Write an algorithm for insertion in sorted linked list	CLO 1 Marks 13
Q4.	The adjacency list representation of a graph with five vertices A, B, C, D, E is given below.	CLO Marks 15



/ = null symbol

Part (a). Draw the adjacency matrix (complete the partial drawing below), and



Part (b). Draw the graph.

GOOD LUCK

Name: Hazrat Bilal

ID: 14622

Course Title: Data Structure & Algorithms

Date: 25/09/2020

Sem: Electrical Engineering

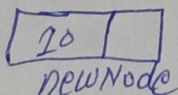
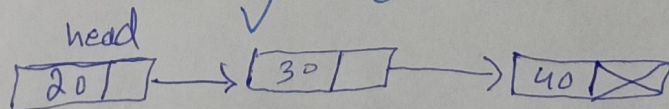
ii — ii — ii — ii — ii — ii

Q16- Explain the following operations in a single link list.

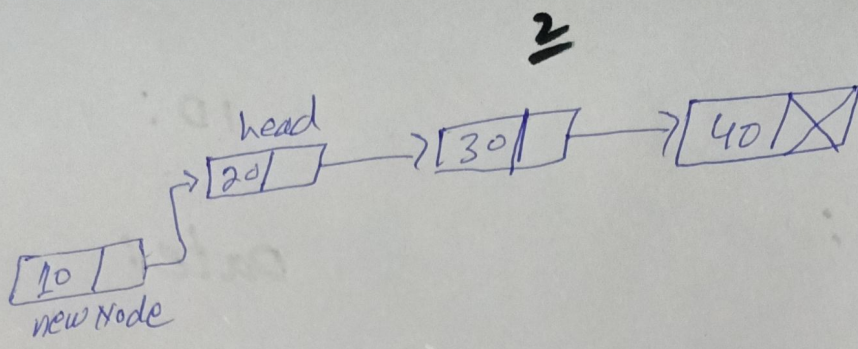
Ans:- insert an Element :-

insertion of Element in single link list is explain in three steps which is given below.

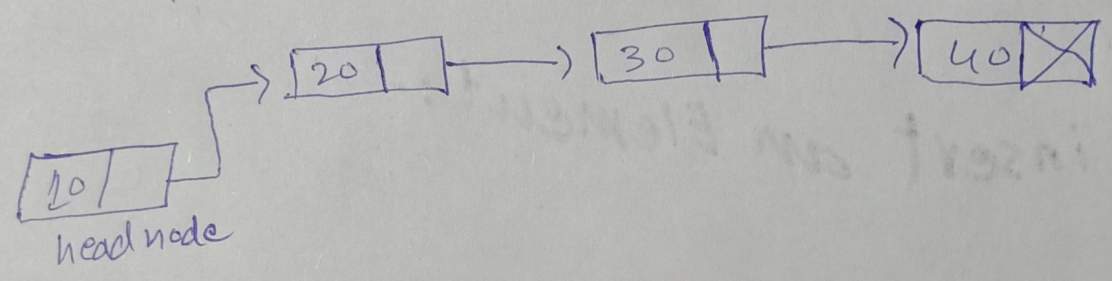
1):- create a new node, say newNode points to the newly created node.



2):- Link the newly created node with the head nodes i.e. the newNode will now point to head node.



3) make the new node as the head node, i.e. Now head node will point to new Node.



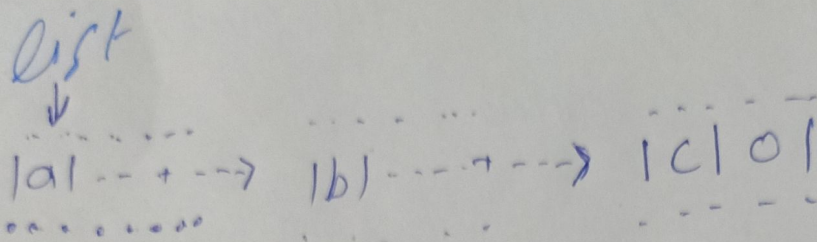
2) :- Delete an Element from single link Qist :-

1) Removing from the beginning of Qist :-

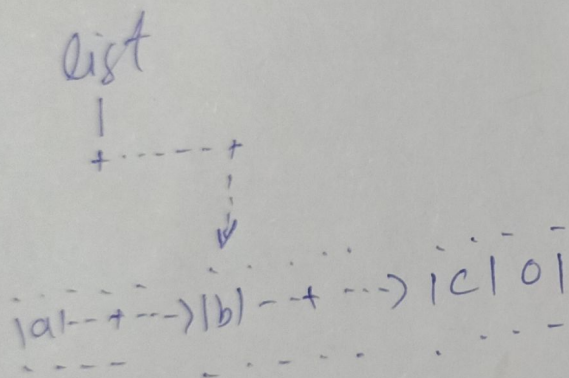
When removing the node at the beginning of the Qist, there is no re-linking of nodes to be performed, since the first node has no preceding node.

3

For example:- removing node with a:

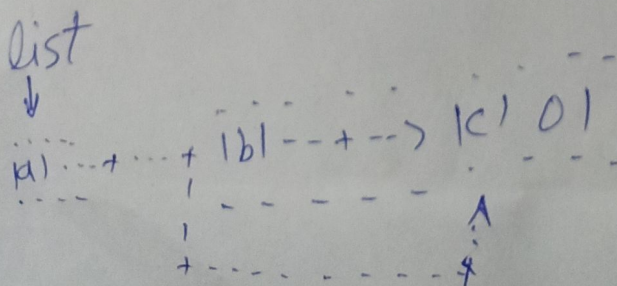


However, we must fix the pointer to the beginning of the list:



2) Removing from the middle of list:

Removing a node from the middle requires that the preceding node skips over the node being removed. For example node with b:

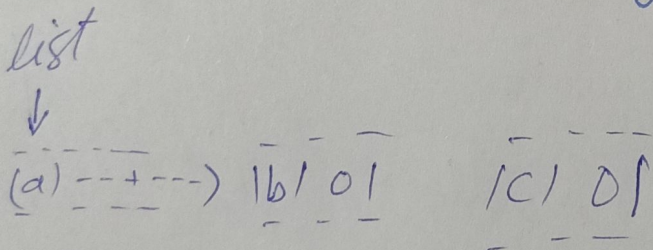


This means that we need some way to refer to the node before the one we want to remove.

3) Removing from the end of the list:

Removing the nodes from the end requires that the preceding node becomes the new end of list

For example: removing the node with c



Note the the last two cases can be combined by saying that "The preceding the one to be removed must point where the one to be removed does".

Q No 3 a)

Ans:- Quick Sort:-

Quick Sort is a divide and conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

There are many different versions of quick sort that pick pivot in different ways.

- 1) Always pick first element as pivot
- 2) Always pick last element as pivot.
- 3) Pick a random element as pivot.
- 4) Pick median as pivot.

Q NO 3 B)

ANS:- Algorithm for insertion in sorted linked list:-

a: array of items

n: size of list

Start

1. declare variable - i, key, j

2. loop: $i = 1$ to $n - 1$

2.1 $key = a[i];$

2.2 $j = i - 1;$

2.3 loop: ($j \geq 0$ && $arr[j] > key$)

2.3.1 $arr[j+1] = arr[j];$

2.3.2 $j = j - 1;$

end loop

2.4 $arr[j+1] = key;$

end loop

STOP.

Q No 2: -

Ans: -

Program code:

```
#include <stdio.h>

int main()
{
    int array[100], position, c, n, value;
    printf("Enter number of elements in
    array\n");
    scanf("%d", &n);
    printf("Enter %d element\n", n);
    for (c=0; c<n; c++)
        scanf("%d", &position, array[c]);
    printf("Enter the location where you
    wish to insert an element\n");
    scanf("%d", &position);
```

```
printf("Enter the 8value to insert\n");
```

```
scanf("%d", &value);
```

```
for (c = n - 1; c >= position - 1; c--)
```

```
array[c + 1] = array[c];
```

```
array[position - 1] = value;
```

```
printf("Resultant array is\n");
```

```
for (c = 0; c <= n; c++)
```

```
printf("%d\n", array[c]);
```

```
return
```

```
return 0;
```

```
}
```