National University of Computer and Emerging Sciences, Lahore Campus

	Course:	Data Structure	Course Code:	
	Program:	BSCS	Semester:	4th
			Section:	4A
	Name:	Sama	Time:	20 mins
	Registration #:	L19-1333	Assessment	Quiz 2

8/10

Q1: Reverse a linked list from position m to n. Note: $1 \le m \le n \le$ length of list.

Example:

Input: 1->2->3->4->5->NULL, m = 2, n = 4
Output: 1->4->3->2->5->NULL

a. Write down an algorithm for the aforementioned problem (Use any combination of your own imagination)
b. Write down code in C++ to accomplish the aforementioned task

3/

) Make two linklists, assume that we have functions of insertathead() and insertattail() - From m to n, store in auxillary created list using insertathead() it will store 4 3 2 in aux list - Then using insertattail() insert elements from original linklist till (m-1) Then from first auxillary list till null and then from original list from n+1. Assume print() and length() are available.

```
d reverse (linklist obj, int m, int n)
{
    if (m<n and n<=length())   // length() is a function.
    {
        int a=m, b=n;
        linklist aux1;
        linklist aux2;
        node *p = obj.head;   // assuming node is class with data and next
        int c=0;
        while (c != m)
        {
            p = p->next;
            c ?
        }
        while (c != n)
        {
            aux1.insertathead(p->data);   // storing 4 3 2
            p = p->next;
            c ?
        }
```

```
int m1=0;                    node *q= obj.head;
While (m1 != m-1)
{
    aux2.insertattail(q->data);
}
// Now m1 = m        node * n = aux1.head;
While ( n != NULL)
{   aux2.insertattail (n -> data);
    n = n->next;
}
node * newnode = obj.head;
int d=0;
While ( d != n )
{ newnode = newnode->next;
}
While (newnode != NULL)
{
    aux2.insertattail(newnode->data)
}
aux2.prin();
```

5/

m
2

n
4

1 2 3 4 5

if (m < n && n < length())

```
/assume print()
}
else
{
    cout << "values of
    m and n are not
    right";
}
```

Store in a new link list

while(m != n)

~~insert at tail~~

if (c == m)

insertatHead(p);

m++;

link

insert in a new link list

insert till m-1 the first original list
insert at tail

when m

insert the created linklist till n

Then insert the original

after reaching n

4 3 2