EDU

EAST DELTA
UNIVERSITY

ASSIGNMENT : 01

Course title: Data Structure

Course Code: CSE 211

Submitted by,

Name: Mohammad Fahim

ID: 242002112

Section: 6

Department: Computer Science and Engineering

Submitted to,

Name: Mohammad Akbar Bin Shah

Designation: Lecturer, SoSET

Date Of Submission: 22-03-2025

Answer to the question no 1

Insert into a Sorted Singly Linked List:

```
begin
 newNode ← createNode (value)
 if head = null or value < head → data then
  newNode → next ← head
  head ← newNode
else
   temp ← head
   while temp → next ≠ null and temp → next → data < value do
         temp ← temp → next
   end while
   newNode → next ← temp → next
   temp → next ← newNode
 end if
end
```

Answer to the question no 2

Remove Duplicates from an Unsorted Linked List:

```
begin
  current ← head
  while current ≠ null and current → next ≠ null do
    temp ← current
    while temp → next ≠ null do
    if temp → next → data = current → data then
        duplicateNode ← temp → next
        temp → next ← temp → next → next
        delete duplicateNode
     else
        temp ← temp → next
     end if
  end while
  current ← current → next
 end while
end
```

Answer to the question no 3

Merge Two Sorted Linked List:

```
begin
    merge ← createNode()
    temp ← merge

    while head1 ≠ null and head2 ≠ null do.
        if head1 → data < head2 → data then
            temp → next ← head1
            head1 ← head1 → next
        else
            temp → next ← head2
            head2 ← head2 → next
        end if
        temp ← temp → next
    end while

    if head1 ≠ null then
        temp → next ← head1
    else
        temp → next ← head2
    end if

    return merge → next
end
```

Answer to the question no 4

Stack Using Linked List:

```
push (value):
begin
    newNode ← createNode (value)
    newNode → next ← top
    top ← newNode
end


pop ( ):
begin
    if top = null then
        return
    end if
    temp ← top
    top ← top → next
    delete temp
end
```

```
peek():
begin
    if top = null then
        return -1
    else
        return top→data
    end if
end
```

```
isEmpty():
begin
    return top = null
end
```

Answer to the question no 5

Sort a Stack Using Another Stack:

```
begin
    while s1 is not empty do
        temp ⟵ pop from s1
      while s2 is not empty and s2.top >temp do
          push (pop from s2) to s1
       end while
       push temp to s2
    end while
  while s2 is not empty do
       push (pop from s2) to s1
   end while
end
```