29-01-24

(a) single linked list - sort

```
-> struct node {

   void sort List (struct Node * * head)
   {
     struct Node * current, * nextNode;
     int temp;
     current = *head;

     while (current != NULL)
     {
       nextNode = current->next;
       while (nextNode != NULL)
       {
         if (current -> data > nextNode -> data)
         {
           temp = current -> data;
           current -> data = nextNode -> data;
           nextNod -> data = temp;
         }
       nextNode = nextNode -> next;
       }
     current = current -> next;
   }
}
```

& Single linked list- Concatenation

```
-> struct Node * Concatenate Lists
        (struct Node * List 1, struct Node* li
   {
     if (list1==NULL)
      {
        return list2;
      }

    struct Node * current=list1;
    while (current ->next != NULL)
     {
       current = current ->next;
     }
     current ->next= list 2;
     return list;
```

Single linked list - reverse lists.

```
-> void reverse list (Struct Node ** headRe
   {
   struct Node *prev, *current, *nextNode;
    prev = NULL;
    current = *headRef;

   while(current != NULL)
   {
   nextNode = current ->next;
   current ->next = Prev;
```

```
    current = nextNode;
  }
  *headRef = prev;
}.
```

ⓐ Stack implementation using single linked list.

O/P:
```
1. Insert in list 1   2. Insert in list 2.   3. Sort
4. Reverse   5. Concatenate   6. Print   0. Exit.

Enter choice : 1
 Enter data : 3 -> ~~4~~ 5 -> ~~8~~1 -> NULL

 Enter choice : 2.
 Enter data : 6 -> 7 -> 8 -> NULL

 Enter ~~dat~~ choice : 3.
  list 1 Sorted.
 Enter choice : 6
1->3->5 -> NULL
 Enter choice : 4
  list 1 reversed
  1->5->3->NULL

 Enter choice : 5
  concatenated
 Enter choice : 6
 3->5->1->6->7->8 -> NULL
```