DSA LAB
Experiment number 06

Name: Aamir Ansari **Batch:** A **Roll no.** 01

Aim: Implementation of singly linked list

Theory:

A singly linked list is a type of linked list that is unidirectional, that is, it can be traversed in only one direction from head to the last node (tail).
Each element in a linked list is called a **node**. A single node contains data and a pointer to the next node which helps in maintaining the structure of the list.

Algorithms:

1.CREATE LIST

```
Step 1:[INITIALIZE] n /*number of nodes to be entered*?
Step 2: [INITIALIZE] node new_node[n]
Step 3:SET START2=new_node[0]
Step 4: Repeat step 5&6 while i<n
Step 5:SET newnode[i]->data
Step 6:SET newnode[i]->next = newnode[i+1]
Step 7: EXIT
```

2.INSERT

a)At a position:

```
Step 1: [INITIALIZE] new , p
Step 2:SET new->data=data;
Step 3: IF position==1
Step 4:         SET new->next=START
Step 5:         SET START=new[END OF IF]
Step 6:SET p=START
Step 7:Repeat step 8 & 9 while i<position -1
Step 8:         SET p=p->next
Step 9:         SET i++
Step 10:IF p==NULL
        PRINT "There are less elements"
Step 11:ELSE
        SET new->next=p->next
        SET p->next=new
Step 12:EXIT
```

b)After a given Value:

```
Step 1: [INITIALIZE] New_Node ,ptr ,preptr
Step 2: SET New_Node->data= data
Step 3: SET ptr=START
Step 4: SET preptr=ptr
Step 5: Repeat step 6&7  while preptr->data != val
Step 6:         SET preptr=ptr
Step 7:         SET ptr=ptr->next
Step 8: SET preptr->next=New_Node
Step 9: SET New_Node->next=ptr
Step 10:EXIT
```

c)Before a given value:

```
Step 1: [INITIALIZE] New_Node ,ptr ,preptr
Step 2: SET New_Node->data= data
Step 3: SET ptr=START
Step 4: SET preptr=ptr
Step 5: Repeat step 6&7  while ptr->data != val
Step 6:         SET preptr=ptr
Step 7:         SET ptr=ptr->next
Step 8: SET preptr->next=New_Node
Step 9: SET New_Node->next=ptr
Step 10:EXIT
```

d)At the beginning

```
Step 1: [INITIALIZE] New_node
Step 2: IF START == NULL
                SET START = New_node
                SET START->next = NULL
                [END OF IF]
                Goto Step 6
Step 3: SET New_Node->DATA = VAL
Step 4: SET New_Node->next = START
Step 5: SET START = New_Node
Step 6: EXIT
```

e)At the end

```
Step 1: [INITIALIZE] New_node,last
Step 2:SET New_Node->data=val
Step 3: IF START == NULL
                SET START = New_node
                SET START->next = NULL
                [END OF IF]
                Goto Step 6
Step 4: SET last = START;
Step 5:Repeat Step 6 while last->next != NULL
Step 6: SET last = last->next; [END OF LOOP]
Step 7:SET last->next = New_Node;
Step 8:SET New_Node->next = NULL;
Step 9: EXIT
```

3.DELETION

a)Value at a particular Position

```
Step 1: [INITIALIZE] ptr , preptr
Step 2: IF START == NULL
```

```
                PRINT "Linked list is already empty"
                Goto Step 13
Step 3:IF START->next == NULL
             free(START)
             START = NULL
             Goto step 10
Step 4: SET  count = 1
Step 5:SET ptr= START
Step 6: SET preptr = ptr
Step 7:Repeat step 8 to 10 while count < position
Step 8:         SET preptr = ptr
Step 9:         SET ptr = ptr->next
Step 10:        SET count++
Step 11: IF count == 1
             START = ptr->next
             ptr->next = NULL
             free(ptr)
Step 12:ELSE
             preptr->next = ptr->next
             ptr->next = NULL
             free(ptr)
Step 13:EXIT
```

b)Value at the beginning

```
Step 1: [INITIALIZE] New_node
Step 2: IF START == NULL
                PRINT "Linked list is already empty"
                Goto Step 6
Step 3: SET New_Node=START
Step 4: SET START = START->next
Step 5: free(New_Node)
Step 6: EXIT
```

c)At the end

```
Step 1: [INITIALIZE] ptr , preptr
Step 2: IF START == NULL
                PRINT "Linked list is already empty"
                Goto Step 6
Step 3:IF START->next == NULL
             free(START)
             START = NULL
             Goto step 10
Step 4:SET ptr= START
Step 5: Repeat step 6&7  while ptr->next != NULL
Step 6:         preptr = ptr
Step 7:         ptr = ptr->next
Step 8: SET preptr->next = NULL
Step 9: free(ptr)
Step 10:EXIT
```

d)After a particular value

```
Step 1: [INITIALIZE] temp ,ptr ,preptr
Step 2: SET ptr=START
Step 3: SET preptr=ptr
Step 4: Repeat step 5&6  while preptr->data != val
Step 5:         SET preptr=ptr
Step 6:         SET ptr=ptr->next
Step 7: SET temp=ptr
Step 8: SET preptr->next=temp->next
Step 9: free(temp)
Step 10:EXIT
```

e)Before a particular value

```
Step 1: [INITIALIZE] new_Node ,ptr ,preptr
Step 2: SET ptr=START
Step 3: SET preptr=ptr
Step 4: Repeat step 5&6  while ptr->data != val
Step 5:         SET preptr=ptr
Step 6:         SET ptr=ptr->next
Step 7: SET preptr->next=new_Node
Step 8:SET new_Node->next=ptr
Step 9:EXIT
```

4.UPDATE

a)Value at a given Position

```
Step 1: IF START == NULL
                PRINT "Linked list is already empty"
                Goto Step 7
Step 2: SET count = 1
Step 3: Repeat step 4&5 while count != position
Step 4:         ptr = ptr->next
Step 5:         count++
Step 6:SET ptr->data = data
Step 7: EXIT
```

b)Value at the beginning

```
Step 1: IF START == NULL
                PRINT "Linked list is already empty"
                Goto Step 3
Step 2: SET START->data = data
Step 3:EXIT
```

c)At the end

```
Step 1: IF START == NULL
                PRINT "Linked list is already empty"
                Goto Step 3
Step 2: [INITIALIZE] node ptr=START
Step 3:Repeat step 4 while ptr->next != NULL
Step 4:         ptr = ptr->next
Step 5:SET ptr->data = data
Step 6:EXIT
```

d)After a particular value

```
Step 1: IF START == NULL
```

```
                 PRINT "Linked list is already empty"
                 Goto Step 3
Step 2: [INITIALIZE] node ptr=START , postptr
Step 3:Repeat step 4 while ptr->next != val
Step 4:          ptr = ptr->next
Step 5: IF ptr->next == NULL
                 PRINT "There is no element after this!"
Step 6:ELSE
                 SET postptr = ptr->next
                 SET postptr->data = data
Step 7: EXIT
```

e)Before a particular value

```
Step 1: IF START == NULL
                 PRINT "Linked list is already empty"
                 Goto Step 3
Step 2: [INITIALIZE] node ptr=START , preptr
Step 3:Repeat step 4 while ptr->next != val
Step 4:          ptr = ptr->next
Step 5: IF ptr->next == NULL
                 PRINT "There is no element after this!"
Step 6:ELSE
                 SET preptr = ptr
                 SET ptr = ptr->next
                 SET count++
Step 7: SET preptr->data = data;
Step 8:EXIT
```

5.SEARCH

```
Step 1: SET PTR = START
Step 2: Repeat Step 3 while PTR != NULL
Step 3: IF VAL = PTR->DATA
                         PRINT ‘ELEMENT FOUND’
                         Go To Step 5
                 ELSE
                         SET PTR = PTR->NEXT
                 [END OF IF]
Step 4: PRINT ‘ELEMENT  NOT FOUND’
Step 5: EXIT
```

6.REVERSE

```
Step 1:[INITIALIZE] prev, ptr, next
Step 2:SET prev=NULL
Step 3:SET ptr=START
Step 4: Repeat step 5 to 8 while ptr!=NULL
Step 5:          SET  next=ptr->next
Step 6:           SET ptr->next=prev
Step 7:          SET prev=ptr
Step 8:          SET ptr=next
Step 9:SET START=prev
Step 10:EXIT
```

7.COUNT NODES

```
Step 1: INITIALIZE count = 0,node current = START
Step 2:Repeat step3&4 while current != NULL
Step 3:        SET count++
Step 4:        SET current = current->next
Step 5 :RETURN count
Step 6:EXIT
```

8.DISPLAY

```
Step 1:Repeat step 2&3 while START != NULL
Step 2:        PRINT (START->data)
Step 3:        SET START = START->next
Step 4: EXIT
```

9.MERGE

```
Step 1: [INITIALIZE ] ptr , ptr2
Step 2: SET ptr = START
Step 3:Repeat step 4 while ptr->next != NULL
Step 4:        SET ptr = ptr->next
Step 5: SET ptr->next = START2
Step 6:[INITIALIZE] node traverse , min , temp
Step 7:Repeat step 8&9 while START->next
Step 8:        SET min = START
Step 9:        SET traverse = START->next
Step 10:Repeat step 11&12 while traverse is true
Step 11:               IF min->data > traverse->data
                              SET min = traverse
Step 12:               SET traverse = traverse->next
Step 13: SET temp = START->data
Step 14: SET START->data = min->data
Step 15: SET min->data = temp
Step 16: SET START = START->next
Step 17:EXIT
```

10.SORT

```
Step 1:[INITIALIZE] node traverse , min , temp
Step 2:Repeat step 3&4 while START->next
Step 3:        SET min = START
Step 4:        SET traverse = START->next
Step 5:Repeat step 6&7 while traverse is true
Step 6:        IF min->data > traverse->data
                       SET min = traverse
Step 7:        SET traverse = traverse->next
Step 8: SET temp = START->data
Step 9: SET START->data = min->data
Step 10: SET min->data = temp
Step 11: SET START = START->next
Step 12:EXIT
```

11.CONCATENATE TWO SLLS.

```
Step 1: [INITIALIZE ] ptr , ptr2
Step 2: SET ptr = START
Step 3:Repeat step 4 while ptr->next != NULL
Step 4:         SET ptr = ptr->next
Step 5: SET ptr->next = START2
Step 6:EXIT
```