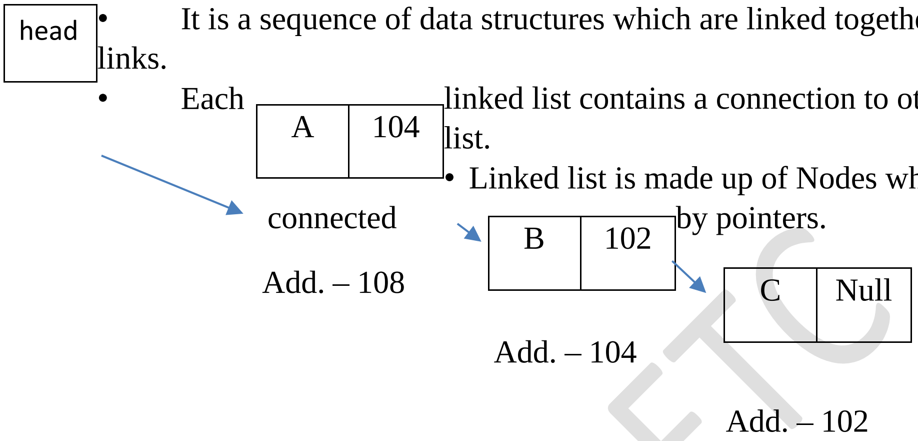
	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2021-2022	CLASS: SE-5
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2021-22/	Starting date: 22/11/2021
	Roll No:22108	Submission date: 30/11/2021
Title:	**Linked List Operations**	
Problem statement	Create a singly linked list with options: a. Insert (at front, at end, in the middle) b. Delete (at front, at end, in the middle) c. Display d. Display Reverse e. Revert the SLL	
Prerequisites:	Basics of C programming	
	Decision making and loop controls	
	Choice based program	
	Linked List	
Objectives:	Learn to create and display a link list	
	Implement various operation on array to understand its effect on data.	
Theory:		

DS_Writeups_ETC

	Linked List – • It is a sequence of data structures which are linked together via links. • Each linked list contains a connection to other list. • Linked list is made up of Nodes which are connected by pointers.  head A 104 Add. – 108 B 102 Add. – 104 C Null Add. – 102
ERROR and REMEDY	-

Code	#include <stdio.h>
	#include <stdlib.h>

```
struct Node
{
        int data;
struct Node *next;
};

struct Node* getNode()
{
        struct Node *temp;      temp = (struct Node
*) malloc(sizeof(temp));
        printf("\n Enter the data for the Node - ");
scanf("%d", &temp->data);    temp->next =
NULL;
        return temp;
}

void display_data(struct Node *h)
{
        while(h != NULL)
        {
```

DS_Writeups_ETC

```
                printf(" %d -> ", h->data);
                h = h->next;
        }
        printf(" NULL ");
}

struct Node * insert_front(struct Node* head)
{
        struct Node *t;
        t = getNode();
        t->next = head;
        head = t;
        return head;
}

struct Node * insert_middle(struct Node* head, int pos)
{
        struct Node *t, *p;
        int i;

        t = getNode();
        p = head;
        for(i = 0; i < pos-2 && p->next != NULL ; i++)
                p = p->next;
        t->next = p->next;
        p->next = t;

        return head;
}
struct Node * insert_last(struct Node* head)
{
        struct Node *t, *p;      t
= getNode();  p = head;
        while(p->next !=
NULL)                   p = p-
>next;
        p->next = t;

        return head;
```

DS_Writeups_ETC

```
}

struct Node * delete_front(struct Node* head)
{
        struct Node *t;
        t = head;
        head = head-
>next;          free(t);
        return head;
}

struct Node * delete_middle(struct Node* head, int pos)
{
        struct Node *t,
*p;     int i;    p = head;
        t = head;
        for(i = 0; i < pos-1 && p->next != NULL; i++)
        {
                t = p;
                p = p->next;
        }
        t->next = p->next;
        free(p);

        return head;
}

struct Node * delete_last(struct Node* head)
{
        struct Node *t, *p;      p
= head;         t = head;
        while(p->next !=
NULL)
        {
                t = p;   p =
        p->next;
        }
        t->next = p-
        >next; free(p);
        return head;
}

struct Node * revert (struct Node * head)
{
```

DS_Writeups_ETC

```
        struct Node *c, *t, *r;
        t = NULL;
        c = head;
        r = NULL;

        if(head->next == NULL)
        return head;

        while(c != NULL)
        {
                t = c->next;
                c->next =
r;              r = c;
        c = t;
        }

        head = r;
        return head;
}

void reverse_list (struct Node * head)
{
        if(head != NULL)
        {
                revert(head->next);
                printf(" %d ", head->data);
        }
}

int main()
{
        int i, n, ch, pos,checker;
        struct Node *p, *head=NULL;

        printf("\nDo you want to create Singly Linked list ?\nAns [Y(1)/N(0)] :
");
        scanf("%d",&checker);
   if(checker == 1)
   {
```

DS_Writeups_ETC

```
        head = getNode();     printf("\n How many elements do

      you want to enter - ");     scanf("%d", &n);


        p = head;

        for(i = 0; i < n; i++)
        {
                p->next = getNode();
                p = p->next;
        }
  printf("\nFollowing are the opertations on linked list, enter your choice of
operation as integer corresponding to the serial no. of the choices below.... \n
1. Insert at front\n 2. Insert in middle \n 3. Insert at last\n 4. Delete from front\n
5. Delete from middle\n 6. Delete from last \n 7. Revert the list\n 8. Reverse
the list\n 9. Display the list\n 10. Exit");
            printf("\n Enter your choice - ");
            scanf("%d",&ch);
```

DS _ Writeups _ ETC

```
        switch(ch)
        {
            case 1:
                    printf("\n Insert in front.\n");
                    head = insert_front(head);
                    break;
            case 2:
                    printf("\n Insert in middle.\nEnter the
position - ");
                    scanf("%d", &pos);
       head = insert_middle(head, pos);
                    break;
            case 3:
                    printf("\nInsertion at last.\n");
                    head = insert_last(head);
                    break;
            case 4:
                    printf("\n\n Deleting the first.\n");
                    head = delete_front(head);
                    printf("\nDeleted successfully");
                    display_data(head);
                    break;
            case 5:
                    printf("\nDeleting in the middle.\nEnter
the position: \n");
                    scanf("%d", &pos);
                    head = delete_middle(head, pos);
                    printf("\nDeleted successfully");
                    break;
            case 6:
                    printf("\nDelete at end.\n");
                    head = delete_last(head);
```

DS_Writeups_ETC

```
                    printf("\n Deleted successfully");
                    break;
            case 7:

                    printf("\nReverting your list.");
                    head = revert(head);
                    display_data(head); break;
            case 8:
                    printf("\n\n Reversing your list - ");
                    reverse_list(head);
                    printf("\n\n Original list - ");
                    display_data(head);
                    break;
            case 9:
                    printf("\n\n Your List - \n");
                    display_data(head);
                    break;
            case 10:
                    return 0;
            default:
                    printf("\n Please enter a valid option");
        }


    return 0;
}
}
```

DS _ Writeups_ETC

CONCLUSION:

In this experiment, we implemented linked list and applied operations like insert, delete, reverse and revert linked list.

REFERENCES:

Seymour Lipschutz, Data Structure with C, Schaum’s Outlines, Tata McGrawHill

E Balgurusamy - Programming in ANSI C, Tata McGraw-Hill (Third Edition)

	Yashavant Kanetkar- Let Us C, BPB Publication, 8th Edition.

DS_Writeups_ETC

Continuous Assessment for DS AY 2021-22			
RPP (5)	SPO (5)	Total (10)	Signature:

			Assessed By: Mr. V. B. Vaijapurkar

Start date	**Submission date**	**Date:**
22/11/2021	**30/11/2021**	**Roll. No.22108**
***Regularity, Punctuality, performance** ***Submission, Presentation, orals**		

DS_Writeups_ETC