=>

```
#include <stdio.h>
struct node{
        int data;
        struct node * next
};

struct node * head = NULL, * newnode, *
temp ;

void create (){
        int i, n;
        printf ("Enter the number of elem
        scanf ("%d", &n);
        for (i=0; i<n; i++){
            newnode = (struct node *)malloc
              (sizeof (struct node));
            printf ("Enter the %d element
              i+1) ;
            newnode -> next = NULL;

            if (head == NULL)
                  temp = head = newnode;

            else {
                temp -> next = newnode;
                temp = newnode;
            }
}       }
```

```
void display(){
        temp = head;
        printf("The elements are:\n");
        while (temp != NULL){
                printf("%d\n", temp -> data);
                temp = temp -> next;
        }
}

void insertbeg(){
        newnode = (struct node *) malloc (sizeof
                        (struct node));
        printf("Enter the new element:\n");
        scanf("%d", &newnode -> data);
        newnode -> next = head;
        head = newnode;
}

void insertend(){
        newnode = (struct node *) malloc (size of
                        (struct node));
        printf("Enter the new element:\n");
        scanf("%d", &newnode -> data);
        newnode -> next = NULL;
        temp = head;
        While (temp -> next != NULL)
                temp = temp -> next;
        temp -> next = newnode;
}
```

```
void insertpos(){
    int pos, i=0;
    newnode=(struct node*) malloc(
        sizeof(struct node));
    printf("Enter the position:\n");
    scanf("%d", &pos);

    if(pos<0)
        printf("Invalid position:\n");

    else{
        temp=head;
        while(i<pos-1){
            temp=temp->next;
            i++;
        }
        printf("Enter the new element:\n");
        scanf("%d", &newnode->data);
        newnode->next=temp->next;
        temp->next=newnode;
    }
}

void main(){
    int choice;
    while(1)
    {
        printf("Enter operation:\n
```

```
1. Create \n 2. Display \n 3. insert at beginig \n 4. Inscat at end \n 5. Insert at a pasition \n
6. End program \n");
scanf ("%d", &choice);
if (choice == -1)
        printf ("Operation completed! \n");
        break;
}
else
{
    switch (choice)
    {
        case 1:
            create ();
            break;
        case 2:
            display ();
            break;
        case 3:
            insertbeg ();
            break;
        case 4:
            insertend ();
            break;
        case 5:
            insertpos ();
            break;
        case 6:
            exit (0);
? ? } }
```

```
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
1
Enter the number of elements :
3
Enter the 1 element:
12
Enter the 2 element:
34
Enter the 3 element:
56
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
2
The elements are:
12
34
56
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
3
Enter the new element :
15
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
2
The elements are:
15
12
34
56
```

```
2
The elements are:
15
12
34
56
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
4
Enter the new element :
45
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
2
The elements are:
15
12
34
56
45
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
5
Enter the position :
3
Enter the new element:
58
Enter operation :
 1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at a position
 6.END PROGRAM
2
The elements are:
```

2. WAP to Implement Singly linked list with following operation:

a) Create a linked list.
b) Deletion of first element, specific element and last element in the list.
c) Display the contents of the list.

=>

```
#include <stdio.h>
struct node{
        int data;
        struct node *next;
};

struct node *head = NULL, *newnode, *temp;

void create(){
        int i, n;
        printf("Enter the number of elements");
        scanf("%d", &n);
        for(i=0; i<n; i++){
                newnode = (struct node *) malloc
                        (sizeof(struct node));
                printf("Enter the %d element", i+1);
                scanf("%d", &newnode->data);
                newnode->next = NULL;
```

```
        if (head == NULL)
            temp = head = newnode;
        else {
            temp -> next = newnode;
            temp = newnode;
        }
    }
}

void display (){
    temp = head;
    printf("The elements are : \n");
    while (temp != NULL){
        printf("%d \n", temp -> data
        temp = temp -> next;
    }
}

void deletebeg (){
    temp = head;
    if (head == NULL)
        printf ("list is empty");
    else {
        head = temp -> next;
        free (temp);
    }
}
```

```
void deleteend(){
    struct node* prenode;
    temp=head;
    while(temp->next!=NULL)
    {
        prenode=temp;
        temp=temp->next;
    }

    if(temp==head)
        head=NULL;
    else
        prenode->next=0;
    free(temp);
}

void deletepos(){
    struct node* nextnode;
    int pos, i=1;
    temp=head;
    printf("Enter position\n");
    scanf("%d",&pos);
    while(i<pos){
        temp=temp->next;
        i++;
    }
    nextnode=temp->next;
    temp->next=nextnode->next;
    free(nextnode);
```

```
void main()
{
    int choice;
    while(1)
    {
        printf("Enter operation: \n1.
        create\n 2. Display\n 3. delete
        begining \n 4. delete at end \n
        5. delete at a position. \n 6. ex
    scanf("%d", &choice);
    if (choice == -1)
    {
        printf("operation completed");
        break;
    }

    else{
      switch(choice)
      {
        case 1: create();
                break;

        case 2: display()
                break;

        case 3: delete beg()
                break;
```


```
case 4 : delete end()
         break;

case 5 : deletepos()
         break;

case 5 : end(0);
```

}
}
}
}
}

Event
NLP
22/4/24

```
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
1
enter the number of elements:
5
Enter the element 1:
12
Enter the element 2:
34
Enter the element 3:
56
Enter the element 4:
67
Enter the element 5:
54
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
2
The elements are:
12
34
56
67
54
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
5
enter the position:
3
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
```

```
2
The elements are:
12
34
67
54
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
3
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
2
The elements are:
34
67
54
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
4
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
6.End program
2
The elements are:
34
67
Enter operation:
1.create
2.display
3.delete at beginnning
4.delete at end
5.delete at position
```