18/1/24 Week-5

Singly linked list.

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
  int data;
  struct node *next;
};
void display();
void insert_begin();
void insert_end();
void insert_pos();
void delete_begin();
void delete_end();
void delete_pos();
struct node *head=NULL;
void display()
{
  printf("Elements are:\n");
  struct node *ptr;
  if(head==NULL)
  {
    printf("List is empty\n");
    return;
  }
else
```

```
{
 ptr=head;
 while(ptr!=NULL)
 {
 printf("%d \n", ptr->data);
 ptr= ptr -> next;
 }
}
void insert()
{
 struct node *temp, *ptr;
 temp=(struct node *)malloc(sizeof(struct node));
 printf("Enter the value to be inserted \n");
 scanf("%d", &temp->data);
 temp->next=NULL;
 if(head==NULL)
 {
   head=temp;
 }
 else
 {
 ptr=head;
 while(ptr->next!=NULL)
 {
   ptr=ptr->next;
 }
  ptr->next=temp;
 }
}
```

```
void delete_begin()
{
struct node *ptr;
if(head == NULL)
{
 printf("List is empty \n");
 return;
}
else
{
ptr = head;
head = head ->next;
printf("the element deleted from the list - %d \n",
    ptr -> data);
free(ptr);
}
}
void delete_end()
{
 struct node *ptr, *temp;
 if(head == NULL)
 {
   printf("List is empty \n");
   return;
 }
else
 {
 ptr = head;
 while(ptr -> next != NULL)
  {
  temp = ptr;
  ptr = ptr -> next;
```

```
    }
    temp->next = NULL;
    printf("The element deleted - %d", ptr->data);
    free(ptr);
    }
}
void delete_pos()
{
    int i, pos;
    struct node *ptr, *temp;
    if (head==NULL)
    {
        printf("List is empty \n");
        return;
    }
    else
    {
        printf("Enter the position to be deleted: \n");
        scanf("%d", &pos);
        if (pos==0)
        {
            ptr = head;
            head = head->next;
            printf("The item deleted from the list is
                    %d", ptr->data);
            free(ptr);
        }
        else
        {
            ptr=head;
            for(i=0; i<pos; i++)
            {
```

```
temp = ptr;
ptr = ptr -> next;
if (ptr == NULL)
{
printf("position not found \n");
return;
}
}
temp -> next = ptr -> next;
printf("The value deleted is -%d", ptr -> data);
free(ptr);
}
}
}
```

18/11/24

```
void main()
{
int choice;
while(1)
{
printf("\n 1. to insert \n 2. to display \n
3. to delete beginning \n 4. to delete at
end \n 5. delete at a given position \n
6. to exit \n");
printf("Enter your choice \n");
scanf("%d", &choice);
switch(choice)
{
case 1:
```

```
        insert();
        break;
    case 2:
        display();
        break;
case 3:
        delete_begin();
        break;
case 4:
        delete_end();
        break;
case 5:
        delete_pos();
        break;
case 6:
        exit(0);
        break;
default:
    printf("Enter the correct choice \n");
    break;
  }
 }
}
```

output

1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. delete at position
6. to exit.

Enter your choice:1
Enter the value to be inserted 12.
1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. todelete at position
6. to exit.
Enter your choice:1
Enter the value to be inserted 13.
1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. delete at position
6. to exit.
Enter your choice:1
Enter the value to be inserted 14.
1. to insert.
2. to display
3. to delete beginning
4. to delete at end
5. delete at position
6. to exit
Enter your choice:2.
Elements are :12, 13, 14, 15.
1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. delete at position
6. exit.

Enter your choice: 3.
The element deleted from list - 12
1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. delete at a given ~~given~~ position
6. to exit.
Enter your choice: 4
The element deleted from list - 15
1. to insert
2. to display
3. to delete beginning
4. to delete at end
5. delete at a given position
6. to exit
Enter your choice: 5
Enter your position: 0
The element deleted from list - 13.