Lab Sheet 7

Lab Tasks:

A. Implement insertion, deletion, traversing operation in singly linked list.

Your code of singly linked list should contain the following operations -

i. Insertion in the beginning (head)
ii. Insertion at the end
iii. Insert at a specific index.
iv. Delete from the head.
v. Delete from the end
vi. Delete from any specific location.
vii. Count the total number of nodes.
viii. Search a node by value.
ix. Printing the linked list in the reverse order (bonus task).

Also you code should contain a display function that will print the linked list.
You can use menu system for doing the task like you used in stack and queue.

Your sample code can look like this:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;
void beginsert()
{
    struct node *ptr;
    int item;
    ptr = malloc(sizeof(struct node *));
    printf("\nEnter value\n");
    scanf("%d",&item);
    ptr->data = item;
    ptr->next = head;
    head = ptr;
    printf("\nNode inserted");


}

void display()
{
    struct node *ptr;
    ptr = head;
```

```
    printf("\nprinting values . . . . .\n");
    while (ptr!=NULL)
    {
        printf("%d-> ",ptr->data);
        ptr = ptr -> next;
    }

}

void main ()
{
    int choice =0;
    while(choice != 3)
    {
        printf("\n\n*********Main Menu*********\n");
        printf("\nChoose one option from the following list ...\n");
        printf("\n===============================================\n");
        printf("\n1.Insert in beginning\n2.Show\n3.Exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
        case 1:
            beginsert();
            break;
        case 2:
            display();
            break;
        case 3:
            exit(0);
            break;
        default:
            printf("Please enter valid choice..");
        }
    }
}
```

Reading materials:

Primary idea about singly linked list:
https://www.geeksforgeeks.org/data-structures/linked-list/singly-linked-list/