August 22, 2021

HOW TO IMPLEMENT LINKED LIST IN C

—

```
/* creation of singly linked list & Operations On Singly Linked List
        ->insertion
                ->at begning
                ->at end
                ->at specificposition
        -> Deletion
                ->at begning
                ->at end
                ->at specificposition
        ->find the length of list
        ->reverce the list
// Created By SMIT
*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
```

```
17.
18.
19. //Global Declaractions
20. int count=0;
21. struct node{
22.         int data;
23.         struct node *next;
24. };
25. struct node *head=0,*temp;      //Global Pointers
26. void CreateList();      //To Create Link LIst
27. void InsAtBeg();        //To Insert Data At Beginig
28. void InsAtPos();        //This Function will Insert Data At Given Position
29. void InsAtEnd();        //To Insert Data At The End Of List
30. void DelAtBeg();        //To delete first Element/Node Of Link list
31. void DelAtPos();        //To delete Spasific Node Of Link LIst
32. void DelAtEnd();        //To delete Last Node Of List
33. void FindLength();      //To find the length Of Link list
34. void ReverseList();     //To reverse The Link List
35. void Display();         //To Display Link LIst
36.
37. int main(){
38.         int choice;
39.         while(choice!=11){              //This Loop Will Run Untill You Give 11 As an Input
40.                 printf("\n**************MAIN MENU**************\n");
41.                 //choice Must Beatween 1 To 11
42.                 printf("1.Create a list\n");
43.                 printf("2.insert at begning\n");
44.                 printf("3.insert at  n(th) position\n");
45.                 printf("4.insert at end\n");
46.                 printf("5.delete from begning\n");
47.                 printf("6.delete from n(th) position\n");
```

```
48.                 printf("7.delete from end\n");
49.                 printf("8.Find the length\n");
50.                 printf("9.Reverse the Linked list\n");
51.                 printf("10.Display Linked List\n");
52.                 printf("11.exit\n");
53.                 scanf("%d",&choice);
54.
55.                 //To Ran Spacific Function According To choice
56.         switch(choice){
57.                 case 1:
58.                         CreateList();
59.                         break;
60.                 case 2:
61.                         InsAtBeg();
62.                         break;
63.                 case 3:
64.                         InsAtPos();
65.                         break;
66.                 case 4:
67.                         InsAtEnd();
68.                         break;
69.                 case 5:
70.                         DelAtBeg();
71.                         break;
72.                 case 6:
73.                         DelAtPos();
74.                         break;
75.                 case 7:
76.                         DelAtEnd();
77.                         break;
78.                 case 8:
```

```
                        FindLength();
                        break;
                case 9:
                        ReverseList();
                        break;
                case 10:
                        Display();
                        break;
                case 11:
                        exit(0);
                        break;
                default:
                        printf("\n\n\nEnter valid choice\n\n\n");
        }
}
        return 0;
// Created By SMIT
}

//This Function Will Create A Linked List
void CreateList(){
struct node *newnode;
int choice=1;
while(choice){
newnode=(struct node*)malloc(sizeof(struct node));
printf("Enter Data ");
scanf("%d",&newnode->data);
newnode->next=0;
if(head==0){
head=temp=newnode;
}
```

```
110. else{
111.         temp->next=newnode;
112.         temp=newnode;
113. }
114. printf("\nPress 0 to exit Press 1 To Add ");
115. scanf("%d",&choice);
116. count++;
117. }
118. printf("\n\n\n%d nodes created successfully\n\n\n",count);
119. }
120.
121. //This Function Will Display The List
122. void Display(){
123.         if(head==0){
124.                 printf("\n\n\nI Think You should Create A list First\n\n\n");
125.         }
126.         else{
127.                 temp=head;
128.                 printf("\n\n\nThe List is");
129.                 while(temp!=0){
130.                         printf("\n%d\n",temp->data);
131.                         temp=temp->next;
132.                 }
133.         }
134. }
135.
136. //To Insert Node At Begning
137. void InsAtBeg(){
138.         if(head==0){
139.                 printf("\n\n\nPlease Create a List First\n\n\n");
140.         }
```

```
        else{
        struct node *newnode;
        newnode=(struct node *)malloc(sizeof(struct node));
        printf("Enter data");
        scanf("%d",&newnode->data);
        newnode->next=head;
        head=newnode;
        }
}

//To Insert Node At spacific Positon
void InsAtPos(){
        if(head==0){
                printf("\n\n\nPlease Create a List First\n\n\n");
        }
        else{
                int pos,i=1;
                printf("Enter Position");
                scanf("%d",&pos);
                if(pos>count){
                        printf("Invalid Position");
                }
                else{
                        struct node *newnode;
                        temp=head;
                        while(i<pos){
                                temp=temp->next;
                                i++;
                        }
                        newnode=(struct node*)malloc(sizeof(struct node));
                        printf("Enter Data");
```

```
172.                        scanf("%d",&newnode->data);
173.                        newnode->next=temp->next;
174.                        temp->next=newnode;
175.                }
176.        }
177. }
178.
179. //To Insert Node At The End Of List
180. void InsAtEnd(){
181.        if(head==0){
182.                printf("\n\n\nPlease Create a List First\n\n\n");
183.        }
184.        else{
185.                struct node *newnode;
186.                temp=head;
187.                while(temp->next!=0){
188.                        temp=temp->next;
189.                }
190.                newnode=(struct node*)malloc(sizeof(struct node));
191.                printf("Enter Data");
192.                scanf("%d",&newnode->data);
193.                newnode->next=0;
194.                temp->next=newnode;
195.        }
196. }
197.
198. //To Delete First Node
199. void DelAtBeg(){
200.        if(head==0){
201.                printf("\n\n\nPlease Create a List First\n\n\n");
202.        }
```

```
        else{
                temp=head;
                head=temp->next;
                free(temp);
                printf("\n\n\nDeleted successfully..\n\n\n");
        }
}
// Created By Smit
//To Delete Node AT Spacific Position
void DelAtPos(){
        int pos,i=1;
        if(head==0){
                printf("\n\n\nPlease Create A list First\n\n\n");
        }
        else{
                struct node *deletethis;
                temp=head;
                printf("Enter Position ");
                scanf("%d",&pos);
                if(pos>count){
                        printf("\n\nInvalid Position\n\n");
                }
                else{
                        while(i<pos-1){
                                temp=temp->next;
                                i++;
                        }
                        deletethis=temp->next;
                        temp->next=deletethis->next;
                        free(deletethis);
                        printf("\n\n\nDeleted successfully...\n\n\n");
```

```
234.                 }
235.         }
236. }
237.
238. //To Delete Last Node
239. void DelAtEnd(){
240.         if(head==0){
241.                 printf("\n\n\nPlease Create A List First\n\n\n");
242.         }
243.         else{
244.                 struct node *prev;
245.                 temp=head;
246.                 while(temp->next!=0){
247.                         prev=temp;
248.                         temp=temp->next;
249.                 }
250.                 free(temp);
251.                 prev->next=0;
252.                 printf("\n\n\nDeleted Successfully...\n\n\n");
253.         }
254. }
255.
256. //To Find The Lenth Of List
257. void FindLength(){
258.         if(head==0){
259.                 printf("\n\n\nYou can't find length Without Creating List\n\n\n");
260.         }
261.         else{
262.         temp=head;
263.         count=0;
264.         while(temp!=0)  {
```

```
                temp=temp->next;
                count++;
        }
        printf("\n\n\nThe Length Of Linked List Is %d \n\n\n",count);
        }
}

//To Reverse The List
void ReverseList(){
        if(head==0){
                printf("\n\n\nThere Is No List To Reverse\n\n\n");
        }
        else{
                struct node *previousnode,*currentnode,*nextnode;
                previousnode=0;
                currentnode=nextnode=head;
                while(nextnode!=0){
                nextnode=nextnode->next;
                currentnode->next=previousnode;
                previousnode=currentnode;
                currentnode=nextnode;
                }
                head=previousnode;
                printf("\n\n\nList Reversed Successfully..\n\n\n");
        }
}
// Created By SMIT
```