September 24, 2021

IMPLEMENTATION OF DOUBLY LINK LIST

—

```
/* creation of Doubly linked list & Operations On Doubly Linked List
        ->insertion
                ->at begning
                ->at end
                ->at specific position
                ->after specific position
        -> Deletion
                ->at begning
                ->at end
                ->at specific position
                ->after specific position
        ->find the length of list
        ->reverce the list
// Created By SMIT
*/

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


struct node{
        int data;
        struct node *next;
        struct node *prev;
};
struct node *head=0,*newnode,*temp,*tail;
void CreateDoubly();    //creates Doubly Link List
void DisplayDoubly();   //Display Doubly Link List
```

```
30. void InsAtBeg();                //Insert At the begning Of list
31. void InsAtPos();                //Insert At Given position in List
32. void InsAfterPos();             //Insert After Given Position In List
33. void InsAtEnd();                //Insert At the End Of the list
34. void DelAtBeg();                //Delete At the Begning Of the List
35. void DelAtPos();                //Delete At given Position in List
36. void DelAfterPos();             //Delete After Given Positon In list
37. void DelAtEnd();                //Delete At the End Of List
38. void DoublyLength();    //find the length of Doubly link list
39. void ReverseDoubly();   //Reverse The Link List
40.
41. //Main
42. int main(){
43.         int choice;
44.         //While Loop For Selecting Choice It will exit while user type 13
45.         while(choice!=13){
46.                 printf("\n\n*****Main Menu*****\n");
47.                 printf("\n1.  Create");
48.                 printf("\n2.  Display");
49.                 printf("\n3.  insert at Begning");
50.                 printf("\n4.  insert at position");
51.                 printf("\n5.  insert after position");
52.                 printf("\n6.  insert at End");
53.                 printf("\n7.  Delete At Begning");
54.                 printf("\n8.  Delete At Position");
55.                 printf("\n9.  Delete After Position");
56.                 printf("\n10. Delete At End");
57.                 printf("\n11. Find the length of list");
58.                 printf("\n12. Reverse Doubly Link List");
59.                 printf("\n13. exit\n");
60.                 printf("\nEnter Choice ");
61.                 scanf("%d",&choice);
62.
63.                 //Switch case according To the Selection Of user
64.                 switch(choice){
65.                         case 1:
66.                                 CreateDoubly();
67.                                 break;
68.                         case 2:
69.                                 DisplayDoubly();
70.                                 break;
71.                         case 3:
72.                                 InsAtBeg();
73.                                 break;
74.                         case 4:
```

```
                                InsAtPos();
                                break;
                        case 5:
                                InsAfterPos();
                                break;
                        case 6:
                                InsAtEnd();
                                break;
                        case 7:
                                DelAtBeg();
                                break;
                        case 8:
                                DelAtPos();
                                break;
                        case 9:
                                DelAfterPos();
                                break;
                        case 10:
                                DelAtEnd();
                                break;
                        case 11:
                                DoublyLength();
                                break;
                        case 12:
                                ReverseDoubly();
                                break;
                        case 13:
                                exit(0);
                                break;
                        default:
                                printf("Error Please Try Again");
                                break;
                }

        }
        return 0;
}

//This Function will Create Doubly Link List
void CreateDoubly(){
        int choice=1;
        while(choice){
        newnode=(struct node *)malloc(sizeof(struct node));
        printf("Enter Data:");
        scanf("%d",&newnode->data);
```

```
120.         newnode->next=0;
121.         newnode->prev=0;
122.         if(head==0){
123.                 head=temp=newnode;
124.         }
125.         else{
126.                 temp->next=newnode;
127.                 newnode->prev=temp;
128.                 temp=newnode;
129.                 tail=newnode;
130.         }
131.         printf("Press 0 to exit Press 1 To Add ");
132.         scanf("%d",&choice);
133.         }
134. }
135. //This Function Will Display Doubly Link List
136. void DisplayDoubly(){
137.         temp=head;
138.         while(temp!=0){
139.                 printf("\n%d",temp->data);
140.                 temp=temp->next;
141.         }
142. }
143.
144. //This Function will Insert in doubly link list at begingng
145. void InsAtBeg(){
146.         temp=head;
147.         newnode=(struct node *)malloc(sizeof(struct node));
148.         printf("Enter Data: ");
149.         scanf("%d",&newnode->data);
150.         temp->prev=newnode;
151.         newnode->next=temp;
152.         newnode->prev=0;
153.         head=newnode;
154.         printf("\nNode Created Successfully");
155. }
156. //This Function will Insert in doubly link list at Given position
157. void InsAtPos(){
158.         temp=head;
159.         int pos,i;
160.         printf("Enter position ");
161.         scanf("%d",&pos);
162.         for(i=1;i<pos-1;i++){
163.                 temp=temp->next;
164.         }
```

```
165.         newnode=(struct node *)malloc(sizeof(struct node));
166.         printf("Enter Data ");
167.         scanf("%d",&newnode->data);
168.         newnode->next=temp->next;
169.         newnode->prev=temp;
170.         temp->next->prev=newnode;
171.         temp->next=newnode;
172.         printf("\nNode Inserted successfully");
173. }
174. //This Function will Insert in doubly link list after Given position
175. void InsAfterPos(){
176.         temp=head;
177.         int i,pos;
178.         printf("Insert Position ");
179.         scanf("%d",&pos);
180.         for(i=1;i<pos;i++){
181.                 temp=temp->next;
182.         }
183.         newnode=(struct node *)malloc(sizeof(struct node));
184.         printf("Enter Data");
185.         scanf("%d",&newnode->data);
186.         newnode->next=temp->next;
187.         newnode->prev=temp;
188.         temp->next->prev=newnode;
189.         temp->next=newnode;
190.         printf("\nNode created Successfully");
191. }
192. //This Function will Insert in doubly link list at End
193. void InsAtEnd(){
194.         newnode=(struct node *)malloc(sizeof(struct node));
195.         printf("Enter data ");
196.         scanf("%d",&newnode->data);
197.         newnode->next=0;
198.         newnode->prev=tail;
199.         tail->next=newnode;
200.         tail=newnode;
201.         printf("\nNode Inserted Successfully");
202. }
203.
204. //This function will delete the node at begning
205. void DelAtBeg(){
206.         temp=head;
207.         head=temp->next;
208.         head->prev=0;
209.         free(temp);
```

```
210.         printf("\nNode Deleted Successfully");
211. }
212.
213. //This function will delete the node at Given Position
214. void DelAtPos(){
215.         temp=head;
216.         int i,pos;
217.         printf("Enter Position ");
218.         scanf("%d",&pos);
219.         for(i=1;i<pos;i++){
220.                 temp=temp->next;
221.         }
222.         temp->prev->next=temp->next;
223.         temp->next->prev=temp->prev;
224.         free(temp);
225.         printf("\nNode Deleted Successfully");
226. }
227.
228. //This function will delete the node after given position
229. void DelAfterPos(){
230.         temp=head;
231.         int i,pos;
232.         printf("Enter Position ");
233.         scanf("%d",&pos);
234.         for(i=1;i<pos+1;i++){
235.         temp=temp->next;
236.         }
237.         temp->prev->next=temp->next;
238.         temp->next->prev=temp->prev;
239.         free(temp);
240.         printf("\nNode Deleted Successfully");
241. }
242.
243. //This function will delete the node at End
244. void DelAtEnd(){
245.         tail=tail->prev;
246.         free(tail->next);
247.         tail->next=0;
248.         printf("\nNode Deleted Successfully");
249. }
250.
251. //This Function will find the length of List
252. void DoublyLength(){
253.         temp=head;
254.         int count=1;
```

```
255.         while(temp->next!=0){
256.                 temp=temp->next;
257.                 count++;
258.
259.         }
260.         printf("The length Of Doubly Link List is %d",count);
261. }
262.
263.
264. //This Function will reverse the List
265. void ReverseDoubly(){
266. struct node *nextnode;
267. temp=head;
268. while(temp!=0){
269.         nextnode=temp->next;
270.         temp->next=temp->prev;
271.         temp->prev=nextnode;
272.         temp=nextnode;
273.         }
274.         temp=head;
275.         head=tail;
276.         tail=temp;
277.         printf("Link Reversed Successfully");
278. }
279.
280. //that's it  This is all about Doubly link list
281. //your logic may be differ
```

Share

Labels: Implementation of Doubly link list

COMMENTS

Enter your comment...

POPULAR POSTS