SINGLY LINKED LIST

CODE:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int info;
struct node *next;
};
struct node *head=NULL,*ptr,*newptr;
struct node* getnode()
{
struct node* np;
np=(struct node*)malloc(sizeof(struct node));
printf("Enter the data:");
scanf("%d",&np->info);
np->next =NULL;
return np;
};
void create()
{
struct node *last;
char ch;
do
{
newptr= getnode();
if(head== NULL)
head=newptr;
else
{
last->next=newptr;
}
last=newptr;
printf("do you want to add moer(y/n)");
scanf(" %c",&ch);
}
while(ch=='Y' || ch=='y');
}
```

```
void display()
{
ptr=head;
printf("the elemlens are \n");
while(ptr !=NULL)
{
printf("%d\t",ptr->info);
ptr=ptr->next;
}
}
void insert_at_specific(){
int selectKey;
if(head !=NULL){
printf("Select a key:");
scanf("%d",&selectKey);
ptr = head;
while (ptr!=NULL && ptr->info!=selectKey ) {
ptr=ptr->next;
}
if(ptr==NULL){
printf("node with %d as key doesn't exits",selectKey);

}else {
newptr = getnode();
newptr->next = ptr->next;
ptr->next = newptr;
}
}
}
void insert_at_begning(){
struct node *temp;
printf("\n\t\tEnter the key to insert:");
scanf("%d",&temp->info);
if (head==NULL) {
temp->next=NULL;
head=temp;
}else {
temp->next = head;
head = temp;
```

```
}
printf("\n\t\t%d is inserted at begining\n",temp->info);
}
void insert_at_last(){

struct node *temp,*last;
printf("\n\t\tEnter the info:");
scanf("%d",&temp->info);
temp->next=NULL;
if(head == NULL){
head = temp;
}else {
last = head;
while (last->next!=NULL) {
last=last->next;
}
last->next =temp;
}
}
void insert()
{
int choice;
printf("\nWhere do you want to insert: \n1. At begning\t 2.At Specified Possiton \t 3. At
End \n");
scanf("%d",&choice);
switch (choice) {
case 1:
insert_at_begning();
break;
case 2:
insert_at_specific();
break;
case 3:
insert_at_last();
break;
default:
printf("\nNo options available .");
}
```

```
}

void deleteFromFront(){
if(head == NULL){
printf("List is empty");
}else {
ptr = head;
printf("%d is deleted !", ptr->info);
head = head->next;
free(ptr);
}

}
void deleteFromRandom(){
struct node *prevptr= NULL;
int key;
if(head == NULL){
printf("List is empty !");
}else {
printf("enter the key to delete:");
scanf("%d",&key);
ptr = head;
while (ptr!=NULL && ptr->info!=key) {
prevptr = ptr;
ptr=ptr->next;
}

if(ptr==NULL){
printf("node with key :%d doesn't exist.",key);
return;
}
printf("%d is deleted",ptr->info);
if(prevptr)
prevptr->next = ptr->next;
else
head=ptr->next;
free(ptr);
}
```

```
}
void deleteFromLast(){
struct node *prevptr = NULL;
if (head == NULL) {
printf("List is empty");
}else {
ptr = head;
while (ptr->next!=NULL) {
prevptr = ptr;
ptr = ptr->next;
}
printf("%d is deleted !",ptr->info);
if(!prevptr){
head = NULL;
}else {
prevptr->next = NULL;
}
free(ptr);
}

}
void delete()
{
int ch;
printf("\n\t1. Delete front \t 2. Delete from random \t 3. Delete from last \n Select
Option:");
scanf("%d",&ch);
switch (ch) {
case 1:
deleteFromFront();
break;
case 2:
deleteFromRandom();
break;
case 3:
deleteFromLast();
break;
default:
printf("No such option available:");
```

```
break;
}
}
int main()
{
int choice;
system("cls");
create();
while(1){
printf("\nLinked list \n1 insert\n2 delete\n3 display \n4 exit");
scanf("%d",&choice);
switch (choice)
{
case 1: insert();
break;
case 2: delete();
break;
case 3: display();
break;
case 4: exit(0);
break;
default:printf("invalid data");
}
}
return 0;
}
```

```
Enter the data:1
do you want to add moer(y/n)y
Enter the data:2
do you want to add moer(y/n)n

Linked list
1 insert
2 delete
3 display
4 exit1

Where do you want to insert:
1. At begning    2.At Specified Possiton         3. At End
1

                Enter the key to insert:2

                2 is inserted at begining

Linked list
1 insert
2 delete
3 display
4 exit3
the elemlens are
2       1       2
Linked list
1 insert
2 delete
3 display
4 exit
```

Circular Linked List

Code :

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
struct node
{
int info;
struct node *next;
};
struct node *head=NULL,*ptr,*newptr,*prevptr;
struct node* getnode()
{
struct node* np;
np=(struct node*)malloc(sizeof(struct node));
printf("Enter the data:");
scanf("%d",&np->info);
np->next =NULL;
return np;
};
void create(){
struct node *temp;
char ch;
do {
newptr = getnode();
if(head == NULL){
head = newptr;
}else {
temp->next = newptr;
}
newptr->next = head;
temp = newptr;
printf("Woul you like to add more data:(Y/N)");
scanf(" %c",&ch);
}while (ch == 'y' || ch=='Y');
}
```

```
void display(){
if(head!=NULL){
ptr = head;
printf("Datas in the circular list are:\n");
do {
printf("%d \t",ptr->info);
ptr = ptr->next;
}while (ptr!=head);
}else {
printf("Circular link list is empty !");
}
}

void insertInCList(){
int key;
newptr = getnode();
if (head != NULL) {
printf("Enter the key after which you want to put:");
scanf("%d",&key);

ptr = head;
do{
ptr = ptr->next;
}while (ptr->info !=key && ptr!=head);

if(ptr->info!=key){
printf("Node with key does not exist .");

}else {
printf("%d is inserted !",newptr->info);
newptr->next = ptr->next;
ptr->next = newptr;
}
}
else {
head = newptr;
newptr->next = head;
printf("%d is inserted",newptr->info);
}
```

```
}

void deleteFromCList(){
ptr = head;
int key;
if(head != NULL)
{
printf("Enter the key to delete:");
scanf("%d",&key);
do{
prevptr = ptr;
ptr=ptr->next;
}while(ptr->info!=key && ptr != head);
if(ptr->info!=key)
{
printf("node with info does not exist.");
}else {
if(ptr==head){
if(ptr->next == head)
{
head=NULL;
}else {
head = ptr->next;
prevptr->next = ptr->next;
}
}else {
prevptr->next = ptr->next;
}
printf("%d is deleted.", ptr->info);
free(ptr);

}
}else {
printf("Link List is empty !");
}
}
int main(){
create();
int c;
```

```c
do{
printf("\n1.INSERT \t 2. DISPLAY \t 3. DELETE \t 4. EXIT \n Enter from options: ");
scanf("%d",&c);
switch (c) {
case 1:
insertInCList();
break;
case 2:
display();
break;
case 3:
deleteFromCList();
break;
case 4:
exit(0);
default:
printf("No such option available !");
}
}while(1);
return 0;
}
```

```
Enter the data:1
Woul you like to add more data:(Y/N)y
Enter the data:2
Woul you like to add more data:(Y/N)y
Enter the data:3
Woul you like to add more data:(Y/N)y
Enter the data:4
Woul you like to add more data:(Y/N)n

1.INSERT         2. DISPLAY      3. DELETE       4. EXIT
 Enter from options: 2
Datas in the circular list are:
1       2       3       4
1.INSERT         2. DISPLAY      3. DELETE       4. EXIT
 Enter from options: 3
Enter the key to delete:2
2 is deleted.
1.INSERT         2. DISPLAY      3. DELETE       4. EXIT
 Enter from options: 2
Datas in the circular list are:
1       3       4
1.INSERT         2. DISPLAY      3. DELETE       4. EXIT
 Enter from options:
```