****** SINGLY LINKED LIST ******

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define TRUE 1
#define FALSE 0
typedef struct
SLL
{ int data;
struct SLL
*next;
}node; node *create(); void main() { int choice,val; char ans; node
*head; void display(node *); node *search(node *,int); void
insert(node *); void delete(node **); node *get_prev(node *,int);
head=NULL; clrscr(); do { printf("\n Singly Linked List\n");
printf("\n
1.create\n2.Display\n3.search\n4.insert\n5.delete\n6.quit\n");
printf("\n Enter ur choice:"); scanf("%d",&choice); switch(choice)
{
case 1: head=create(); break; case
2:display(head); break; case
3:printf("Enter the element to
search"); scanf("%d",&val);
search(head,val); break; case
4:insert(head); break; case
5:delete(&head); break; case 6:exit(0);
efault:clrscr(); printf("Invalid
choice, try again"); getch(); } }
```

```
while(choice!=6); } node *create() {
node *temp,*new,*head; int val,flag;
char ans='y'; node *get_node();
temp=NULL; flag=TRUE; do {
printf("\nEmter the element:");
scanf("%d",&val); new=get_node();
if(new==NULL)
printf("\nMemory is not allocated"); new-
>data=val; if(flag) {head=new; temp=head;
flag=FALSE; } else { temp->next=new;
temp=new; } printf("\n Do u want to enter
more elements?"); ans=getche();
}while(ans=='y'); printf("\nThe singly
linked list is created\n"); getch();
return head; } node *get_node() { node
*temp; temp=(node *)malloc(sizeof(node));
temp->next=NULL; return temp; } void
display(node *head)
{ node *temp; temp=head;
if(temp==NULL) {
printf("\nThe listis
empty\n"); getch();
return; }
while(temp!=NULL) {
printf("%d->",temp->data);
temp=temp->next; }
printf("NULL"); getch(); }
node* search(node
*head,int key)
```

```
{ node *temp; int found;
temp=head; if(temp==NULL)
{ printf("\nThe listis
empty\n"); getch();
return NULL; }
found=FALSE;
while(temp!=NULL &&
!found)
{ if(temp->data!=key)
temp=temp->next; else
found=TRUE; } if(found) {
printf("\nThe element is
present\n"); getch(); return
temp;
} else { printf("\nThe element is
not found\n"); getch(); return
NULL;
} } void
insert(node
*head)
{ node *temp,*new; int val; temp=head;
if(temp==NULL) { printf("\nInsertion is not
possible\n"); getch(); return; } printf("\n
Enter the element after which to insert:");
scanf("%d",&val); temp=search(head,val);
if(temp!=NULL) { printf("Enter the element to
insert:"); scanf("%d",&val);
new=(node*)malloc(sizeof(node)); if(new==NULL)
printf("memory is not allocated\n"); new-
>data=val; new->next=NULL; new->next=temp->next;
```

```
temp->next=new; printf("\n The element is
inserted");
getch(); } } node*
get_prev(node *head,int val)
{ node *temp,*prev;
int flag; temp=head;
if(temp==NULL)
return NULL;
flag=FALSE;
prev=NULL;
while(temp!=NULL &&
!flag)
{ if(temp-
>data!=val) {
prev=temp;
temp=temp->next; }
else flag=TRUE; }
if(flag) return
prev; else return
NULL; } void
delete(node
**head)
{ node
*temp,*prev
; int key;
temp=*head;
if(temp==NU
LL)
{
```

```
printf("\nThe list is empty\n"); getch();
return; } printf("\nEnter the element u
want to delete:"); scanf("%d",&key);
temp=search(*head,key); if(temp!=NULL) {
prev=get_prev(*head,key); if(prev!=NULL) {
prev->next=temp->next; free(temp); } else
{
*head=temp->next; free(temp); }
printf("\nThe element is
deleted\n"); getch(); }
}
```

OUTPUT:

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delete

6.quit

Enter ur choice:1

Enter the element:12

Do u want to enter more elements?y

Enter the element:11

Do u want to enter more elements?n

The singly linked list is created

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delete

6.quit

Enter ur choice:2

12->11->NULL

Singly Linked List

1.Create

2.Display

3.search

4.insert

5.delete

6.quit

Enter ur choice:3

Enter the element to search 11

The element is present

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delet

e 6.quit

Enter ur choice:4

Enter the element after which to insert:12

The element is present

Enter the element to insert:10

The element is inserted

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delet

e 6.quit

Enter ur choice:2

12->10->11->NULL

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delet

e 6.quit

Enter ur choice:5

Enter the element u want to delete:10

The element is

present The element

is deleted

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delete

6.quit

Enter ur choice:2

12->11->NULL

Singly Linked List

1.create

2.Display

3.search

4.insert

5.delete

6.quit

Enter ur choice:6

RESULT:

Thus the singly linked list is implemented and insert, delete and search operations are performed And verified.