**Program No. 5**

Objective:-WAP for various operation performed on Singly Linked list

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct node\*head=NULL;

struct node

{

int data;

struct node\*link;

};

struct node\*Aakash\_createnewnode()

{

struct node \*newNode;

newNode=(struct node\*) malloc (sizeof(struct node));

printf("\nEnter new node=\n");

scanf("%d",&newNode->data);

newNode->link=NULL;

return newNode;

}

void createsinglylinkedlistbyaddingnodeatfront()

{

struct node\*newNode;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

}

else

{

newNode=Aakash\_createnewnode();

newNode->link=head;

head=newNode;

}

}

void createsinglylinkedlistbyaddingnodeatend()

{

struct node\*newNode,\*ptr;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

}

else

{

newNode=Aakash\_createnewnode();

ptr=head;

while(ptr->link!=NULL)

{

ptr=ptr->link;

}

ptr->link=newNode;

}

}

void createsinglylinkedlistbyaddingnodeatanyposition(int key)

{

struct node\*newNode,\*ptr;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

}

else

{

ptr=head;

while(ptr->data!=key)

{

ptr=ptr->link;

}

newNode=Aakash\_createnewnode();

newNode->link=ptr->link;

ptr->link=newNode;

}

}

void deletenode()

{

struct node\* ptr;

if (head==NULL)

{

printf("No! there is no node in the list");

}

else if(head->link==NULL)

{

printf("There is only one node");

ptr=head;

free(ptr);

head=NULL;

}

else

{

ptr=head;

head=head->link;

free(ptr);

}

}

void deletenodeatend()

{

struct node\* ptr,\*myptr;

if (head==NULL)

{

printf("No! there is no node in the list");

}

else if(head->link==NULL)

{

printf("There is only one node");

ptr=head;

free(ptr);

head=NULL;

}

else

{

ptr=head;

while(ptr->link!=NULL)

{

myptr=ptr;

ptr=ptr->link;

}

free(ptr);

myptr->link=NULL;

}

}

void deletenodeatanyposition(struct node \*\*head, int key)

{

struct node\* ptr = \*head, \*myptr;

if (ptr != NULL && ptr->data == key)

{

\*head = ptr->link;

free(ptr);

return;

}

while (ptr != NULL && ptr->data != key)

{

myptr = ptr;

ptr = ptr->link;

}

if (ptr == NULL)

return;

myptr->link = ptr->link;

free(ptr);

}

void printLinkedlist(struct node \* ptr)

{

while(ptr!=NULL)

{

printf("%d-->",ptr->data);

ptr=ptr->link;

}

}

void main()

{

int n;

char choice;struct node \*ptr;int key;

printf("Enter the value between (1-6)=\n");

printf("\n\nFor Creating Singly Linked List By Adding Node At Front Press 1:");

printf("\n\nFor Creating Singly Linked List By Adding Node At End Press 2:");

printf("\n\nFor Creating Singly Linked List By Adding Node At Any Position Press 3:");

printf("\n\nFor Deleting Any Node From Singly Linked List By Front Press 4:");

printf("\n\nFor Deleting Any Node From Singly Linked List By End Press 5:");

printf("\n\nFor Deleting Any Node From Singly Linked List By Any Position Press 6:\n\n");

scanf("%d",&n);

printf("\n");

switch(n)

{

case 1:

{

do

{

createsinglylinkedlistbyaddingnodeatfront();

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Linked List is in order of=\n\n");

printLinkedlist(head);

getch();

break;

}

case 2:

{

do

{

createsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(ptr);

getch();

break;

}

case 3:

{

do

{

createsinglylinkedlistbyaddingnodeatanyposition(key);

printf("Enter the value of key=\n\n");

scanf("%d",&key);

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Linked List is in order of=\n\n");

printLinkedlist(head);

getch();

break;

}

case 4:

{

do

{

createsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(ptr);

deletenode();

puts("\nLinked List after Deletion:\n\n");

printLinkedlist(head);

getch();

break;

}

case 5:

{

do

{

createsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(ptr);

deletenodeatend();

puts("\nLinked List after Deletion:\n\n");

printLinkedlist(head);

getch();

break;

}

case 6:

{

do

{

createsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(ptr);

printf("\nEnter the key=\n");

scanf("%d",&key);

deletenodeatanyposition(&head, key);

puts("\nLinked List after Deletion of key:\n\n");

printLinkedlist(head);

getch();

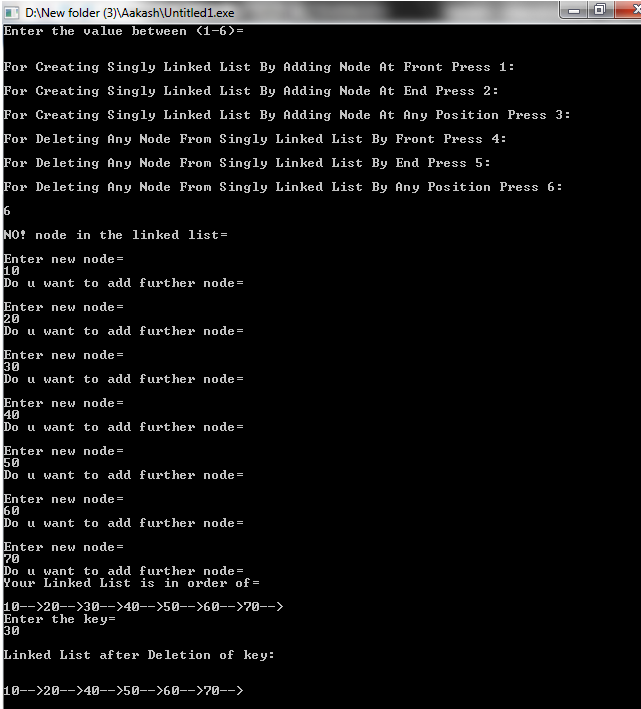
break;

}

}

}

Output:-



Operation performed on Singly Linked list