***PROGRAM:***

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void insertAtBeginning(int);

void insertAtEnd(int);

void insertBetween(int,int,int);

void display();

void removeBeginning();

void removeEnd();

void removespecific(int);

struct Node

{

int data;

struct Node \*next;

};head =NULL;

void main()

{

int choice,value,choice1,loc1,loc2;

while(1)

{

mainMenu:printf("\n\n \*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\n1.insert\n2.Display\n3.Delete\n4.Exit\nEnter your choice");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("Enter the value to be insert :");

scanf("%d",&value);

while(1)

{

printf("where you want to insert : \n1.At Beginning \n2.At End \n3.Between\nEnter your choice:");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

insertAtBeginning(value);

break;

case 2:

insertAtEnd(value);

break;

case 3:

printf("Enter the values where you want to insert:");

scanf("%d%d",&loc1,&loc2);

insertBetween(value,loc1,loc2);

break;

default:

printf("\nWrong input!!!\n\n");

gotomainMenu;

}

gotosubMenuEnd;

}

subMenuEnd:

break;

case 2:

display();

break;

case 3:

printf("How do you want to delete : \n1.From Beginning\n2.From End \n3.specific\nEnter your choice:");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

removeBeginning();

break;

case 2:

removeEnd(value);

break;

case 3:

printf("Enter the value which you want to delete:");

scanf("%d",&loc2);

removeSpecific(loc2);

break;

default :

printf("\nWrong input!!!Try again!!!\n\n");

gotomainMenu;

}

break;

case 4:

exit(0);

break;

default :

printf("\nWrong input!!!Try again!!!\n\n");

}

}

}

void insertAtBeginning(int value)

{

struct Node \*newNode;

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

newNode->data=value;

if(head == NULL)

{

newNode->next = NULL;

head = newNode;

}

else

{

newNode->next=head;

head = newNode;

}

printf("\nOne mode inserted!!!\n");

}

void insertAtEnd(int value)

{

struct Node \*newNode;

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

newNode->data = value;

newNode->next = NULL;

if(head == NULL)

head =newNode;

else

{

struct Node \*temp = head;

while(temp->next != NULL)

temp = temp->next;

temp->next = newNode;

}

printf("\nOne node inserted!!!\n");

}

void insertBetween(int value, int loc1, int loc2)

{

struct Node \*newNode;

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

newNode->data = value;

if(head == NULL)

{

newNode->next=NULL;

head =newNode;

}

else

{

struct Node \*temp = head;

while(temp->data != loc1 && temp->data != loc2)

temp = temp->next;

newNode->next = temp->next;

temp->next = newNode;

}

printf("\nOne node inserted!!!\n");

}

void removeBeginning()

{

if(head=NULL)

printf("\n\nList is Empty!!!");

else

{

struct Node \*temp=head;

if(head==NULL)

{

head = NULL;

free(temp);

}

else

{

head = temp->next;

free(temp);

printf("\nOne node deleted!!!\n\n");

}

}

}

void removeEnd()

{

if(head == NULL)

{

printf("\nList is Empty!!!\n");

}

else

{

struct Node \*temp1 = head,\*temp2;

if(head == NULL)

head = NULL;

else

{

while(temp1->next != NULL)

{

temp2 = temp1;

temp1 = temp1->next;

}

temp2->next = NULL;

}

free(temp1);

printf("\nOne node deleted!!!\n\n");

}

}

void removeSpecific(int delvalue)

{

struct Node \*temp1=head,\*temp2;

while(temp1->data != delvalue)

{

if(temp1 -> next == NULL)

{

printf("\nGiven node not found in the list!!!");

gotofunctionEnd;

}

temp2 = temp1;

temp1 = temp1 -> next;

}

temp2 -> next = temp1 -> next;

free(temp1);

printf("\nOne node deleted!!!\n\n");

functionEnd:;

}

void display()

{

if(head == NULL)

{

printf("\nList is Empty\n");

}

else

{

struct Node \*temp = head;

printf("\n\nList elements are - \n");

while(temp->next != NULL)

{

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

temp = temp->next;

}

printf("%d--->NULL",temp->data);

}

}

***OUTPUT:***

******

