**Linked List Menu Driven Program**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*root;

void append(void);

int length(void);

void addatbegin(void);

void delete(void);

void addafter(void);

void display(void);

void main(){

int ch,len;

while (1){

printf("\n===Linked List Menu===\n");

printf("1.Append Node\n");

printf("2.Length\n");

printf("3.Add at Begin\n");

printf("4.Delete a Node\n");

printf("5.Insertion of Node\n");

printf("6.Traverse the List\n");

printf("7.Exit\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch (ch){

case 1:

printf("Append.");

append();

break;

case 2:

printf("Length");

len = length();

printf("Length of the list is %d.",len);

break;

case 3:

printf("Add at begin");

addatbegin();

break;

case 4:

printf("Delete a node");

delete();

break;

case 5:

printf("Insertion");

addafter();

break;

case 6:

printf("Traverse");

display();

break;

case 7:

exit(1);

default:

printf("Enter valid option.\n");

}

}

}

void append()

{

struct node\* temp;

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

printf("Enter Node Data: ");

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

temp->next = NULL;

if(root == NULL)

{

root = temp;

}

else

{

struct node\* P;

P = root;

while (P->next != NULL)

{

P = P->next;

}

P->next= temp;

}

}

int length()

{

int count = 0;

struct node\* temp;

temp = root;

while(temp != NULL)

{

count++;

temp = temp->next;

}

return count;

}

void addatbegin()

{

struct node\* temp;

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

printf("Enter node data:");

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

temp->next = NULL;

if(root==NULL)

{

root = temp;

}

else

{

temp->next = root;

root = temp;

}

}

void delete()

{

struct node\* temp;

int loc;

printf("Enter location to delete: ");

scanf("%d", &loc);

if(loc> length())

{

printf("Invalid Location.");

}

else if (loc == 1)

{

temp = root;

root = temp->next;

temp->next = NULL;

free(temp);

}

else{

struct node\* P = root, \*q;

int i = 1;

while (i<loc-1)

{

P= P -> next;

i++;

}

q = P->next;

P->next = q->next;

q->next = NULL;

free(q);

}

}

void addafter()

{

struct node\* temp,\*P;

int loc, len,i=1;

printf("Enter Location: ");

scanf("%d",&loc);

len = length();

if (loc>len)

{

printf("Invalid location \n");

printf("Currently list is having %d nodes",len);

}

else

{

P = root;

while (i< loc)

{

P = P-> next;

i++;

}

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

printf("Enter node data:");

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

temp->next = NULL;

temp-> next = P->next;

P->next = temp;

}

}

void display()

{

struct node\* temp;

temp = root;

if (temp == NULL)

{

printf("No nodes in the list \n");

}

else

{

while (temp != NULL)

{

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

temp = temp->next;

}

}

}