#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node \*link;

};

typedef struct Node node;

node \*start = NULL;

void create();

void display();

void InsertStart();

void InsertPosition();

void InsertEnd();

void main() {

int ch;

while (1) {

printf("\n1. Create \n2. Display \n3. Insert at Beginning \n4. Insert at Position \n5. Insert at End \n6. Exit");

printf("\nEnter Your Choice: ");

scanf("%d", &ch);

switch (ch) {

case 1:

create();

break;

case 2:

display();

break;

case 3:

InsertStart();

break;

case 4:

InsertPosition();

break;

case 5:

InsertEnd();

break;

case 6:

exit(0);

default:

printf("Enter a Number between 1 and 6.\n");

}

}

}

void create() {

char ch;

node \*new1, \*curr;

do {

new1 = (node\*)malloc(sizeof(node));

printf("\nEnter value:\n");

scanf("%d", &new1->data);

if (start == NULL) {

start = new1;

curr = new1;

} else {

curr->link = new1;

curr = new1;

printf("Do You Want to Add an Element (Y/N)? ");

scanf(" %c", &ch);

} while (ch == 'y' || ch == 'Y');

curr->link = NULL;

}

void InsertStart() {

node \*new1 = (node\*)malloc(sizeof(node));

printf("\nEnter value:\n");

scanf("%d", &new1->data);

if (start == NULL) {

start = new1;

new1->link = NULL;

} else {

new1->link = start;

start = new1;

}

}

void InsertPosition() {

node \*new1 = (node\*)malloc(sizeof(node));

printf("\nEnter value:\n");

scanf("%d", &new1->data);

if (start == NULL) {

start = new1;

new1->link = NULL;

return;

}

int pos, i = 1;

node \*temp = start;

printf("\nEnter position:\n");

scanf("%d", &pos);

while (temp != NULL && i < pos - 1) {

temp = temp->link;

i++;

}

if (temp == NULL) {

printf("Position out of range.\n");

return;

}

new1->link = temp->link;

temp->link = new1;

}

void InsertEnd() {

node \*new1 = (node\*)malloc(sizeof(node));

printf("\nEnter value:\n");

scanf("%d", &new1->data);

if (start == NULL) {

start = new1;

new1->link = NULL;

return;

}

node \*temp = start;

while (temp->link != NULL) {

temp = temp->link;

}

temp->link = new1;

new1->link = NULL;

}

void display() {

if (start == NULL) {

printf("\nLinked List is Empty.\n");

return;

}

node \*temp = start;

printf("\nElements in Linked List: \n");

while (temp != NULL) {

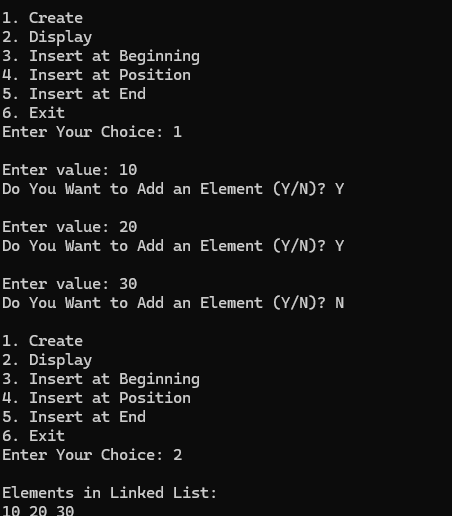
printf("%d ", temp->data);

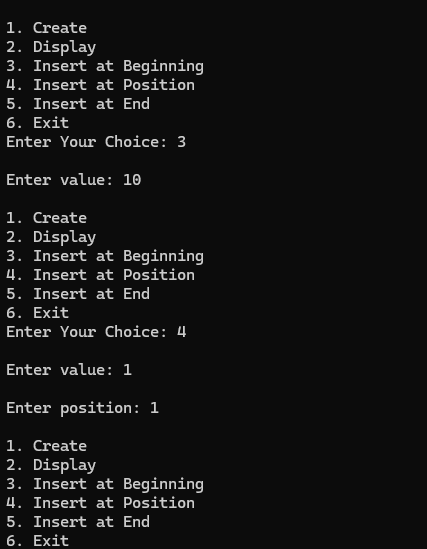
temp = temp->link;

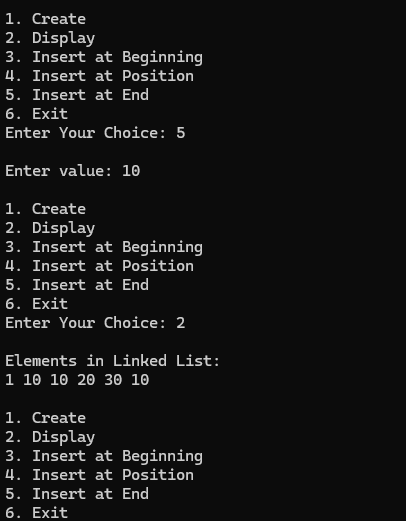
}

printf("\n");

}







#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node \*link;

};

typedef struct Node node;

node \*start = NULL;

void create();

void display();

void DeletefromStart();

void DeleteatPosition();

void DeleteatEnd();

void main() {

int ch;

while (1) {

printf("\n1. Create \n2. Display \n3. Delete from Beginning \n4. Delete at Position \n5. Delete at End \n6. Exit");

printf("\nEnter Your Choice: ");

scanf("%d", &ch);

switch (ch) {

case 1:

create();

break;

case 2:

display();

break;

case 3:

DeletefromStart();

break;

case 4:

DeleteatPosition();

break;

case 5:

DeleteatEnd();

break;

case 6:

exit(0);

default:

printf("Enter a Number between 1 and 9.\n");

}

}

}

void create() {

char ch;

node \*new1, \*curr;

do {

new1 = (node\*)malloc(sizeof(node));

printf("\n enter value:\n");

scanf("%d",&new1->data);

if (start==NULL)

{

start=new1;

curr=new1;

}

else {

curr->link = new1;

curr=new1;

}

printf("Do You Want to Add an Element (Y/N)? ");

scanf(" %c", &ch);

} while (ch == 'y' || ch == 'Y');

curr->link=NULL;

}

void display() {

if (start == NULL) {

printf("\nLinked List is Empty.\n");

return;

}

node \*temp = start;

printf("\nElements in Linked List: \n");

while (temp != NULL) {

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

temp = temp->link;

}

printf("\n");

}

void DeletefromStart() {

if (start == NULL) {

printf("\nLinked List is Empty.\n");

return;

}

node \*temp = start;

start = start->link;

free(temp);

printf("\nFirst element deleted successfully.\n");

}

void DeleteatPosition() {

int pos, i = 1;

if (start == NULL) {

printf("\nLinked List is Empty.\n");

return;

}

printf("\nEnter the position to delete: ");

scanf("%d", &pos);

node \*temp = start;

node \*prev = NULL;

if (pos == 1) {

start = temp->link;

free(temp);

printf("\nElement at position %d deleted successfully.\n", pos);

return;

}

while (temp != NULL && i < pos) {

prev = temp;

temp = temp->link;

i++;

}

if (temp == NULL) {

printf("\nPosition not found.\n");

return;

}

prev->link = temp->link;

free(temp);

printf("\nElement at position %d deleted successfully.\n", pos);

}

void DeleteatEnd() {

if (start == NULL) {

printf("\nLinked List is Empty.\n");

return;

}

node \*temp = start;

node \*prev = NULL;

if (start->link == NULL) {

start = NULL;

free(temp);

printf("\nLast element deleted successfully.\n");

return;

}

while (temp->link != NULL) {

prev = temp;

temp = temp->link;

}

prev->link = NULL;

free(temp);

printf("\nLast element deleted successfully.\n");

}

