Linked list code

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;

};

void linkedListTraversal(struct Node \*ptr)

{

while (ptr != NULL)

{

printf("Element: %d\n", ptr->data);

ptr = ptr->next;

}

}

struct Node \* deleteAtIndex(struct Node \* head, int index){

struct Node \*p = head;

struct Node \*q = head->next;

for (int i = 0; i < index-1; i++)

{

p = p->next;

q = q->next;

}

p->next = q->next;

free(q);

return head;

}

struct Node \* deleteFirst(struct Node \* head){

struct Node \* ptr = head;

head = head->next;

free(ptr);

return head;

}

struct Node \* deleteAtLast(struct Node \* head){

struct Node \*p = head;

struct Node \*q = head->next;

while(q->next !=NULL)

{

p = p->next;

q = q->next;

}

p->next = NULL;

free(q);

return head;

}

int main()

{

struct Node \*head;

struct Node \*second;

struct Node \*third;

struct Node \*fourth;

// Allocate memory for nodes in the linked list in Heap

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

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

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

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

// Link first and second nodes

head->data = 7;

head->next = second;

// Link second and third nodes

second->data = 11;

second->next = third;

// Link third and fourth nodes

third->data = 41;

third->next = fourth;

// Terminate the list at the third node

fourth->data = 66;

fourth->next = NULL;

linkedListTraversal(head);

//char c='Y';

while(1){

printf("\n1.delete at first\n");

printf("\n2.delete at last\n");

printf("\n3.delete at the index\n");

printf("\n4.traverse\n");

printf("5. break\n");

printf("enter the option\n");

int f;

scanf("%d",&f);

if(f==1){

head=deleteFirst(head);

printf("list is\n");

linkedListTraversal(head);

}

else if(f==3){

int ind;

printf("enter the index\n");

head=deleteAtIndex(head,ind);

printf("list is\n");

linkedListTraversal(head);

}

else if(f==2){

head=deleteAtLast(head);

printf("list is\n");

linkedListTraversal(head);

}

else if(f==4){

linkedListTraversal(head);

}

else if(f==5){

break;

}

else{

printf("invalid output");

}

}

return 0;

}