Practice Problems 5

Topic: Linked Lists

ID 2023000000033

1. Write a code to create a doubly linked list containing five nodes and display the values. Take input from the user to insert the values. Nodes are: m=7 n =5 o =13 p =10 q =11
2. Delete a node with given value from that doubly linked list. Users will input the value to be deleted.
3. Delete a node with a given location from that doubly linked list. Users will input the location to be deleted.

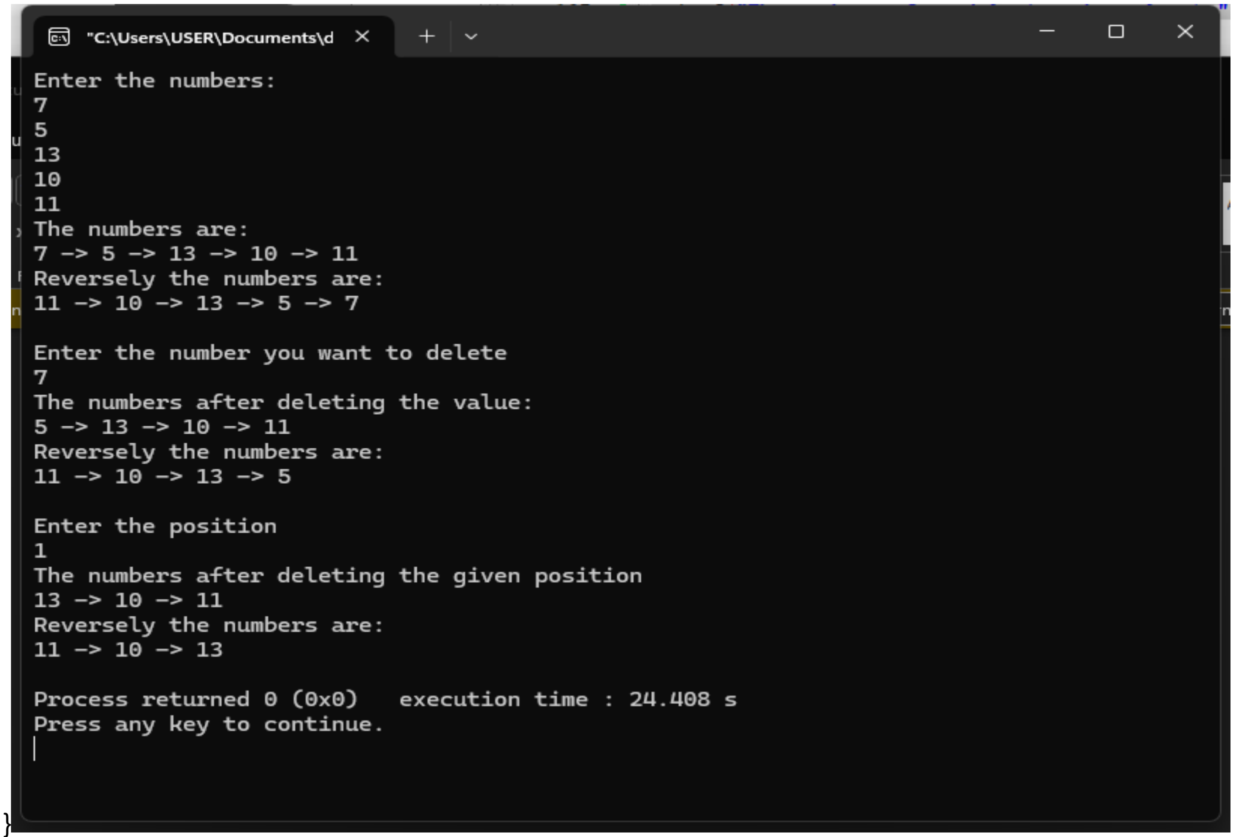
Solution:

```
 #include<stdio.h>
#include<stdlib.h>
struct node{
int data;
struct node* next;
struct node* prev;
};
     int main(){
 struct node* head =NULL;  struct node* last =NULL;  struct node* m =NULL;  struct node* n =NULL;
 struct node* o =NULL; struct node* p =NULL; struct node* q =NULL;
m=malloc (sizeof(struct node)); n=malloc (sizeof(struct node)); o=malloc (sizeof(struct node));
p=malloc (sizeof(struct node)); q=malloc (sizeof(struct node));
head=m; m->next=n;  n->next=o;  o->next=p;  p->next=q; q->next=NULL;
     q->prev=p;  p->prev=o;  o->prev=n;  n->prev=m; m->prev=NULL; last=q;
struct node* tmp=head;
printf("Enter the numbers:\n");
while(tmp!=NULL){
scanf("%d",&tmp->data);
tmp=tmp->next; }
void prnt(struct node** last){
    struct node* tmp2=*last;
    printf("Reversely the numbers are:\n");
while(tmp2!=NULL){
    printf("%d -> ",tmp2->data);
    if(tmp2->prev->prev==NULL){
       tmp2=tmp2->prev;
       printf("%d\n",tmp2->data);   }
    tmp2=tmp2->prev; } }
void printlist(struct node** head){
struct node* tmp1=*head;
```

```
while(tmp1!=NULL){
printf("%d -> ",tmp1->data);
if(tmp1->next->next==NULL){
tmp1=tmp1->next;
printf("%d\n",tmp1->data); }
tmp1=tmp1->next; } }
printf("The numbers are:\n");
printlist(&head);
prnt(&last);
int value;                  //Deletion of a node with a given value
printf("\nEnter the number you want to delete\n");
scanf("%d",&value);
struct node* temp=head;
struct node* prevs=NULL;
if(head==NULL){
printf("Empty list\n"); }
else if(head->data==value){
head=temp->next;
temp->next->prev=NULL;
free(temp); }
else{
while(temp!=NULL && temp->data!=value){
prevs=temp;
temp=temp->next; }
if(temp==NULL){
printf("Value not found\n"); }
else{
prevs->next=temp->next;
if(prevs->next!=NULL){
temp->next->prev=prevs;
free(temp); }
if(prevs->next==NULL){
last=prevs;
free(temp);
} } }
printf("The numbers after deleting the value:\n");
printlist(&head);
prnt(&last);

int pos;          //deletion with a given position
printf("\nEnter the position\n");
scanf("%d",&pos);
struct node* curr=head;
if(head==NULL){
```

```
printf("Empty list\n"); }
else if(pos==1){
head=curr->next;
free(curr);
head->prev=NULL; }
else{
   for(int i=1;curr!=NULL && i<pos-1;i++){
      curr=curr->next;    }
   if(curr->next==NULL || curr==NULL){
      printf("Position out of bounds\n");  }
   else{
   struct node* nextnode=curr->next->next;
   free(curr->next);
   curr->next=nextnode;
   if(nextnode!=NULL){
      nextnode->prev=curr;    }
   if(nextnode==NULL){
      last=curr;
   } } }
printf("The numbers after deleting the given position\n");
printlist(&head);
prnt(&last);
return 0;
```



```
Enter the numbers:
7
5
13
10
11
The numbers are:
7 -> 5 -> 13 -> 10 -> 11
Reversely the numbers are:
11 -> 10 -> 13 -> 5 -> 7

Enter the number you want to delete
7
The numbers after deleting the value:
5 -> 13 -> 10 -> 11
Reversely the numbers are:
11 -> 10 -> 13 -> 5

Enter the position
1
The numbers after deleting the given position
13 -> 10 -> 11
Reversely the numbers are:
11 -> 10 -> 13

Process returned 0 (0x0)   execution time : 24.408 s
Press any key to continue.
```

```
}
```