```c
#include<stdio.h>
#include<stdlib.h>
typedef struct node{
 int value;
 struct node *prev;
 struct node *next;
} Node;
Node *insertleft(Node *head, int data, int key)
{
 Node *new,*ptr;
 new = malloc(sizeof(Node));
 new->value = data;
 new->prev = NULL;
 new->next = NULL;
 ptr = head;
 if(head==NULL)
 {
 return new;
 }
 while(ptr!=NULL)
 {
 if(ptr->value==key)
 {
 break;
 }
 ptr=ptr->next;
 }
 if(ptr->value==key)
 {
 new->prev = ptr->prev;
 (ptr->prev)->next = new;
 new->next = ptr;
 ptr->prev = new;
 return head;
 }
 printf("no values");
 return head;
}
Node *deleteval(Node *head,int key)
{
 Node *ptr;
 if(head==NULL)
 {
```

```c
printf("list empty");
return NULL;
}
ptr=head;
while(ptr!=NULL&&ptr->value!=key)
{
ptr=ptr->next;
}
if(ptr->value==key)
{
(ptr->next)->prev=ptr->prev;
(ptr->prev)->next=ptr->next;
free(ptr);
return head;
}
printf("no value");
return head;
}
int main()
{
Node *head = malloc(sizeof(Node));
head->value = 8;
head->prev = NULL;
head->next = NULL;
Node *current = malloc(sizeof(Node));
current->value = 10;
current->prev = head;
current->next = NULL;
head->next = current;
Node *current2 = malloc(sizeof(Node));
current2->value = 14;
current2->prev = current;
current2->next = NULL;
current->next = current2;
insertleft(head, 15, 14);
Node *ptr1 = head;
while (ptr1 != NULL)
{
printf("%d\n", ptr1->value);
ptr1 = ptr1->next;
}
deleteval(head,8);
Node *ptr = head;
```

```
while (ptr != NULL)
{
printf("%d", ptr->value);
ptr = ptr->next;
}
}
```

```
PS C:\Users\kadab\OneDrive\Desktop\DS> gcc ten.c
PS C:\Users\kadab\OneDrive\Desktop\DS> .\a.exe
8
10
15
14
PS C:\Users\kadab\OneDrive\Desktop\DS>
```

{

struct   node +

int n;

19/12/24 · printf (" Enter
scanf (" %.d"

newnode = (stru