Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

delfun(int ele) : void

myfirst

link list insertion.c | link list deletion.c | *Untitled3

```c
#include <stdio.h>
#include <stdlib.h>
void create();
void display();
void delfun(int);
void delend();
void delfront();

struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;
int main()
{
    int choice,ele;
    char ch;
    int n=1;
    do
    {

    printf("\n1. Create \n2. Display  \n3. Delete \n 4. deiete_at_end \n 5. delete_at_front \n
    printf("\nEnter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: create(); break;
        case 2: display();break;
        case 3: printf("Enter the element to be deleted\n");
```

Scanned by CamScanner

ere X | link list insertion.c X | link list deletion.c X | *Untitled3 X

```
    switch(choice)
    {
        case 1: create(); break;
        case 2: display();break;
        case 3: printf("Enter the element to be deleted\n");
                scanf("%d",&ele);
                delfun(ele); break;
        case 4: delend(); break;
        case 5: delfront(); break;


    }
    }while(choice != 6);
}
 void create()
{
    struct node *newnode,*temp;
     int item;
     newnode =(struct node *) malloc (sizeof(struct node));
     printf("Enter the data : ");
     scanf("%d",&item);
     newnode->data=item;
     if (head==NULL)
      {
        newnode->next=NULL;
       head=newnode;
       printf("Node created\n");
      }
      else
     {
     temp=head;
        while(temp->next!=NULL)
        {
                temp=temp->next;
        }
     temp->next=newnode;
```

ners

Scanned by CamScanner

```
        {
                temp=temp->next;
         }
        temp->next=newnode;
        newnode->next=NULL;
         printf("Node created\n");
    }
}

 void display()
{
     struct node *ptr=NULL;
     ptr=head;

     if(ptr==NULL)
     {
         printf("Nothing to print\n");
     }
     else
     {
         while(ptr!=NULL)
        {
         printf("%d ",ptr->data);
         ptr=ptr->next;
        }
    }

}

void delfun(int ele)
{
    struct node *temp,*del=NULL;

     if (head == NULL)
     {
         printf("Empty List  Can't delete\n");return;
```

Scanned by CamScanner

link list insertion.c X | link list deletion.c X | *Untitled3 X

```
    if (head == NULL)
    {
        printf("Empty List. Can't delete\n");return;
    }
    temp=head;
    if(head->data==ele)
    {
        head=head->next;
        return;
     }
   while (temp->next!=NULL)
   {
       if(temp->next->data==ele)
       {
            del=temp->next;
           if(del->next==NULL)
           temp->next=NULL;
           else
            temp->next=del->next;
       }

       else
          temp=temp->next;
   }
   if(del==NULL)
   {
       printf("Element not found in the list\n");return;
   }


}

void delend()
 {
     struct node * temp;
```

Scanned by CamScanner

here | link list insertion.c | link list deletion.c | *Untitled3

```
123  void delend()
124  {
125      struct node * temp;
126      if (head==NULL)
127      {
128          printf("no elements in the list\n");
129          return;
130      }
131      temp=head;
132      while (temp->next->next!=NULL)
133      {
134          temp=temp->next;
135      }
136      printf("deleted element %d",temp->next->data);
137      temp->next=NULL;
138  }
139
140  void delfront()
141  {
142      struct node*temp;
143      int ele;
144      if (head==NULL)
145      {
146          printf("empty list\n");
147          return;
148      }
149      temp=head;
150      ele=head->data;
151      printf("deleted element is %d\n",ele);
152      head=temp->next;
153  }
154
155
156
157
```

Scanned by CamScanner