

```
myfirst
link list insertion.c X link list deletion.c X *Untitled3 X

#include <stdio.h>
#include <stdlib.h>
void create();
void display();
void insert_at_Begining();
void insert_at_End();
void insert_at_Middle();
struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;
int main(int argc, char **argv)
{
    int choice,ele;
    char ch;
    do
    {
        printf("\n1. Create \n2. Display \n3. Insert_at_begining \n4. Insert_at_end \n5. Insert_at_middle");
        printf("\nEnter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: create(); break;
            case 2: display();break;
            case 3: insert_at_Begining();
            break;
            case 4: insert_at_End();break;
            case 5: insert_at_Middle();break;
        }
    }while(choice != 0);
}
```

rs

de:Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X

rtion.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0



```
}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;
```

```
        newnode->next=NULL;
```

```
        printf("Node created\n");
```

```
    }
```

```
}
```

```
void display()
```



```
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 insert_at_Begining()
```

```
{  
    struct node *newnode;  
    int ele;  
    printf("Enter the element : ");  
    scanf("%d", &ele);  
  
    newnode=(struct node*)malloc(sizeof(struct node));  
    if(head == NULL)  
    {  
        printf("empty list");  
    }  
    else{
```



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

```
99     printf( "empty list");
100 }else{
101
102
103     newnode->data =ele;
104     newnode->next=head;
105     head=newnode;
106 }
107
108 }
109
110 void insert_at_End()
111 {
112     struct node *newnode, *temp;
113     int ele;
114     printf("enter the element : ");
115     scanf("%d", &ele);
116     newnode = (struct node*)malloc(sizeof(struct node));
117     if(head == NULL)
118     {
119         printf("empty list");
120     }
121     else{
122
123
124         newnode->data = ele;
125         newnode->next = NULL;
126         temp = head;
127         while(temp != NULL && temp->next != NULL)
128             temp = temp->next;
129         temp->next = newnode;
130     }
131 }
132
133 void insert_at_Middle()
134 {
135     struct node *newnode, *temp;
```

:Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppC

on.c

C/C++

Windows (CR+LF) WIND



