

## C programming code to create a singly linked list data structure

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

typedef struct ListNode{

int data; // the element data
struct ListNode *next; // next link
}ListElem;

//-----

void Insert (int,int);
void Delete(int);//delete item from the list
void printall();//print out all items on the screen
int countitem();//return the number of items in the list

ListElem *pfirst;
ListElem *plast;

//insert item
void Insert (int val,int pos)
{

int t;
ListElem *item;//new element to be inserted
item=(ListElem *)malloc(sizeof(ListElem)); //allocate space
if(!item) {printf("Memory problem..."); exit(100);}

item->data=val;

//insert a new item to the empty list

if(pfirst==NULL && plast==NULL){
//The first and last item point to the new item when they are null--empty list.
item->next=NULL;
pfirst=item;
plast=item;

printf("Inserted:%d",item->data);printf("\n");

}

//insert a new item at the beginning of the list
else if(pos==1)
{
```

```

item->next=pfirst;
pfirst=item;

}
//insert a new item between items
else if(pos>1 && pos<=countitem()){

ListElem *ta;
ta=pfirst;
for(t=1;t<pos-1;t=t+1){ta=ta->next;} //move to the insertion point
item->next=ta->next;
ta->next=item;

}

//insert a new item at the end of the list
else if(pos==countitem()+1){

item->next=NULL; //The next link of the item is null.
plast->next=item;
plast=item;
printf("Inserted:%d",item->data);printf("\n");

}
//show message if position is not valid.
else printf("%s %d %s","Invalid position! Position must be between 1 and ",countitem()+2,"\n");

}

//Print out all items on the screen

void printall()
{

ListElem *i;
i=pfirst;
if(countitem()>0){
while(i!=NULL){
printf("%d\n",i->data);
i=i->next;
}
}
else printf("This is no item.\n");

}

//count the number of items in the list
int countitem()
{

```

```

ListElem *i;
int t=0;
i=pfirst;
while(i!=NULL){
t=t+1;
i=i->next;

}

return t;

}

//delete item
void Delete(int pos){
int i;
if(countitem(>0){ //make sure the list is not empty.
ListElem *temp,*del;

if(pos==1){//delete the first item
if(countitem()==1){ //The list contains only one item
pfirst=NULL;
plast=NULL;

}
else{ //The list contains more than one item
temp=pfirst;
pfirst=pfirst->next;
temp=NULL;
}

}

else if(pos>1 && pos<=countitem()){//delete middle item
temp=pfirst;
for(i=1;i<pos-1;i=i+1){temp=temp->next;} //move to the item staying before the target item to be
deleted
del=temp->next; //target item to be deleted
temp->next=del->next;
if(del->next==NULL)plast=temp; //delete last item
del=NULL;

}

else printf("%s\n","Invalid position!\n");

}

else printf("%s\n", "No item found\n");

}

```

```

void showmenu(){

printf("=====\n");
printf("Linked List Operations Menu\n");
printf("=====\n");
printf("1.Add a new item\n");
printf("2.Delete an item\n");
printf("3.Show number of items\n");
printf("4.Show all items\n");
printf("5.Exit\n");

}

void select(){
int val, pos, ch;
char yes='y';
while(yes=='y'){
printf("Enter your choice:");scanf("%d",&ch);
switch(ch){
case 1:
printf("Value:");scanf("%d",&val);
printf("Position:");scanf("%d",&pos);
Insert(val,pos);
break;
case 2:
printf("Position:");scanf("%d",&pos);
Delete(pos);
break;
case 3:
printf("Number of items:");printf("%d\n",countitem());
break;
case 4:
printf("All items:\n");
printall();
break;
case 5: exit(0);

default: "Invalid choice";

}
printf("Continue?y/n:");scanf("%s",&yes);
}

}

int main(){

showmenu();
select();
getch();
}

```

```
return 0;  
}
```

By: <http://www.worldbestlearningcenter.com>