Date __ / __ /

Linked List

```
#include <stdio.h>
#include <conio.h>
struct node{
   int info;
   struct node *link; };
typedef struct node *NODE;
NODE getnode(){
NODE x;
x =(NODE) malloc(sizeof (struct node));
if (x == NULL){
  printf(" mem full \n");
  exit(0); } return x; }
void freenode(NODE x){
  free (x); }
NODE insert-front(NODE first, int item){
NODE temp;
temp = getnode(); temp -> info = item; temp -> link = NULL;
if (first == NULL) return temp;
temp -> link = first; first = temp; return first; }
NODE delete-front(NODE first){
NODE temp;
if(first == NULL){
printf(" Empty \n"); return first; } temp = first; temp = temp -> link;
printf("deleted %d \n", first -> info); free (first); return temp; }
NODE insert-rear(NODE first, int item){
NODE temp, cur; temp = getnode(); temp -> info = item; tem->link=NULL;
if (first == NULL){ return temp; }
cur = first;
while (cur -> link! =NULL)
cur = cur -> link;
cur -> link = temp; return first; }
```

```
NODE delete_rear(NODE first){
NODE cur, prev;
 if(first==NULL){
printf("empty\n"); return first;}
if(first->link==NULL){
printf("deleted %d\n", first->info);
free(first); return NULL;}
prev=NULL; cur=first;
while(cur->link!=NULL){
prev=cur; cur=cur->link;}
printf("deleted %d", cur->info);
 free(cur); prev->link=NULL; return first;}
NODE order_list(int item, NODE first){
NODE temp, prev, cur; temp=getnode(); temp->info=item;
temp->link=NULL;
 if(first==NULL)return temp;
 if(item<first->info){
   temp->link=first; return temp;}
prev=NULL; cur=first;
 while(cur!=NULL && item>cur->info){
 prev=cur; cur=cur->link;} prev->link=temp;
 temp->link=cur; return first;}
 void display(NODE first){
 NODE temp; if(first==NULL) printf("empty");
 for(temp=first; temp!=NULL; temp=temp->link){
     printf("%d\n", temp->info); }}
void main(){
  int item, choice, key; NODE first=NULL;
for(;;){ printf("1.Insert\n 2.Del\n 3.order\n 4.Display\n5.Exit"
scanf("%d", &choice); switch(choice){
 Case 1: printf("Enter item"); scan("%d",&item);
       first=insert_front(first,item); break;
Case 2; first = delete_front(first); break;
Case 3: printf(" Enter item\n");
        scanf("%d", &item);
        first = order_list(item, first); break;
case 4: display (first); break;
default: exit(0); break; }} getch(); }

~~To concat & reverse~~ NODE concat(NODE first, NODE second){
~~#include<stdio.h>~~      NODE cur;
~~#include<conio.h>~~      if (first == NULL) return second;
~~struct node~~            if (second == NULL) return first;
cur = first;
while(cur->link != NULL) cur = cur->link;
cur->link = second; return first; }
NODE reverse(NODE first){
NODE cur, temp; cur = temp; cur = NULL;
while(first != NULL){ temp = first;
first = first->link; temp->link = cur; } return cur; }
```