DS LAB PROGRAM 7 WRITE UP

nithil Raj

lab program (7)

```
#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;
}
NODE insert_rear(NODE first, int item)
{
NODE temp, cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
cur=first;
while(cur->link!=NULL)
cur=cur->link;
cur->link=temp;
```

```
return first;
}
void display (NODE first);
{
NODE temp;
if (first == NULL) {
 printf ("List empty");
printf ("Contents of lists : \n");
for (temp = first; temp != NULL; temp =
 temp->link)
{
printf ("%d \n", temp->info);
 }
}
NODE concat (NODE first, NODE second)
{
NODE cur;
if (first == NULL)
 return second;
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 = NULL;
while (first != NULL)
```

```
{
temp= first;
first = first -> link;
temp -> link = cur;
cur=temp;
}
return cur;
}
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 = cur;
cur= cur -> link;
}
prev -> link = temp;
temp -> link = cur;
return first;
}
void main()
{
int item, choice, pos, i, n;
NODE first = NULL, a, b;
```

```
{
temp= first;
first = first -> link;
temp -> link = cur;
cur=temp;
}
return cur;
}
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 = cur;
cur= cur -> link;
}
prev -> link = temp;
temp -> link = cur;
return first;
}
void main()
{
int item, choice, pos, i, n;
NODE first = NULL, a, b;
```

```
for (;;)
{
printf ("1. insert front \n2. concat \n3
reverse \n4. display \n5: order list
\n6. exit \n");
printf ("enter your choice \n");
scanf ("%d", &choice);
switch (choice)
{
case 1 : printf ("enter the item\n");
         scanf ("%d", &item);
        first = insert_rear (first, item);
        break;
case 2 : printf ("enter the no of nodes
in 1 \n");
        scanf ("%d", &n);
        a = NULL;
        for (i=0; i<n; i++)
        {
        printf ("enter the item \n");
        scanf ("%d", &item);
        a = insert_rear (a, item);
    }
printf ("enter the no of nodes in 2\
  scanf ("%d", &n);
        b = NULL;
        for (i=0; i<n; i++)
        {
        printf ("enter the item \n");
        scanf ("%d", &item);
        b = insert_rear (b, item);
    }
```

```
a= concat (a,b);
display (a);
break;
case 3: first = reverse (first);
        display (first);
        break;
case 4: display (first);
        break;
case 5: printf (" enter the item
   be inserted in ordered list
   scanf ("%d", &item);
first = order_list (item, first);
  break;
default: exit (0);
}
}
}
```