

## DS LAB

Rakeshittha.6

IBM17CS074

7/12/2020

```
#include <stdlib.h>
#include <string.h>
struct node
{
    int sem;
    struct node *next;
};
struct node *head = NULL;
struct node *head2 = NULL;
int c=0;
void Inset()
{
    struct node *newnode;
    struct node *temp;
    int s;
    printf("Enter integer: ");
    scanf("%d", &s);
    newnode = (struct node*) malloc(sizeof
    (struct node));
    newnode->sem = s;
    if(head == NULL)
    {
        newnode->next = NULL;
        head = newnode;
        printf("first node of linked list created\n");
        c++;
    }
    else
    {
        temp = head;
```

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

```
void Insert 2()
{
    struct node *newnode;
    struct node *temp;
    int x, y;
    printf("Enter elements to create list 2 \n");
    do
    {
        printf("Enter integers: \n");
        scanf("%d", &x);
        newnode = (struct node *) malloc
        (sizeof (struct node));
        newnode -> x = x;
        if (head 2 == NULL)
        {
            newnode -> next = NULL;
            head 2 = newnode;
            printf("first node of linked list created \n");
            C++;
        }
        else
        {
            temp = head 2;
```

```

while (temp -> next != NULL)
{
    temp = temp -> next;
}
temp -> next = newnode;
newnode -> next = NULL;
C++;
printf("Node created \n");
}
printf("do you want to continue adding:
0 or 1 \n");
scanf("%d", &y);
} while (y != 0);
}

void bubblesort()
{
    int swapped, i;
    struct node *pt1;
    struct node *pt2 = NULL;
    if (head == NULL)
        return;
    do
    {
        swapped = 0;
        pt1 = head;
        while (pt1 -> next != pt2)
        {
            if (pt1 -> sem > pt1 -> next -> sem)
            {
                int temp = pt1 -> sem;
                pt1 -> sem = pt1 -> next -> sem;
                pt1 -> next -> sem = temp;
                swapped = 1;
            }
        }
    }

```



```
void reverse()
```

```
{
    struct node* prev = NULL;
    struct node* current = head;
    struct node* next = NULL;
    while (current != NULL) {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    head = prev;
}
```

```
void concat()
```

```
{
    struct node* ptr;
    if (head == NULL) {
        head = head2;
    }
    if (head2 == NULL) {
        head2 = head;
    }
    ptr = head;
    while (ptr->next != NULL)
        ptr = ptr->next;
    ptr->next = head2;
}
```

```
void display()
```

```
{
    struct node* ptr;
    ptr = head;
    int i = 1;
    if (ptr == NULL)
```

```
{
printf ("Linked list is empty! \n");
}
else
{
while (ptr != NULL)
{
printf ("%d", ptr->sem);
i++;
ptr = ptr->next;
}
}
```

```
void display2()
{
struct node * ptr;
ptr = head2;
int i = 1;
if (ptr == NULL)
{
printf ("Linked list is empty! \n");
}
else
{
while (ptr != NULL)
{
printf ("%d", ptr->sem);
printf ("\n");
i++;
ptr = ptr->next;
}
}
```

```
int main()
{
```

```
int choice, pos;
do
{
    printf("\n1. Insert node\n2. sort
node\n3. reverse node\n4. concat
2 lists\n5. exit\n");
    printf("\nEnter your choice :");
    scanf("%d", &choice);
    switch(choice)
    {
        case 1:
            Insert();
            break;

        case 2:
            bubblesort();
            display1();
            break;
        case 3:
            reverse();
            display1();
            break;
        case '4':
            Insert2();
            concat();
            display1();
            break;
        case '5':
            break;
        default:
            printf("Wrong choice!\n");
            break;
    }
} while (choice != 5);
return 0;
```