

LAB PROGRAM - 7.

```
#include <stdio.h>
#include <stdlib.h>
void create1();
void create2();
void sort();
void reverse();
void concatenate();
void display();
struct node
{
    int data;
    struct node *next;
};
struct node *head = NULL;
struct node *head2 = NULL;
int c;

int main()
{
    int choice;
    do
    {
        printf("\n1. Create  \n2. Sort linked list  \n3. Reverse linked list  \n4. Concatenate  \n5. Display  \n6. Exit ");
        printf("\nEnter your choice: \n");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: create1(); break;
            case 2: sort(); break;
            case 3: reverse(); break;
```

```
case 4: create2();  
        concatenate(); break;  
case 5: display(); break;  
case 6: printf("EXITING-----\n");  
        break;
```

```
}  
while(choice != 6);  
return 0;
```

```
void create1()
```

```
{  
    struct node *newnode;  
    struct node *temp;  
    int s;  
    printf("Enter integer : ");  
    scanf("%d", &s);  
    newnode = (struct node *) malloc(sizeof(struct node));  
    newnode->data = s;  
    if (head == NULL)
```

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

```
    printf("First node created\n");
```

```
    c++;
```

```
}  
else
```

```
{  
    printf("First node created to linked list\n");
```

```
    temp = head;
```

```
    while (temp->next != NULL)
```

```
{
```

```
        temp = temp->next;
```

```
}
```

```
temp->next = newnode;
```

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

```
c++;
```



```

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

```

```

void reverse()
{

```

```

    struct node *prev = NULL; *current = head;
    *next = NULL;

```

```

    while (current != NULL)
    {

```

```

        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }

```

```

    head = prev;

```

```

    printf("Reversed Successfully\n");
}

```

```

void display()
{

```

```

    struct node *ptr = NULL;

```

```

    ptr = head;

```

```

    if (ptr == NULL)
    {

```

```

        printf("Nothing to print\n");
    }
    else
    {

```

```

    }

```

```

    printf("Contents of linked list : \n");

```

```

    while (ptr != NULL)
    {

```

```

        printf("%d\t", ptr->data);

```

```

        ptr = ptr->next;
    }
}

```

```

}

```

```

printf("\n");
}

```

```
void concatenate()
```

```
{
    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 sort()
```

```
{
    int swap, i;
    struct node *ptr1;
    struct node *ptr2 = NULL;
    if (head == NULL)
        return;
    do
    {
        swap = 0;
        ptr1 = head;
        while (ptr1->next != ptr1)
        {
            if (ptr1->data > ptr1->next->data)
            {
                int temp = ptr1->data;
                ptr1->data = ptr1->next->data;
                ptr1->next->data = temp;
                swap = 1;
            }
            ptr1 = ptr1->next;
        }
        ptr2 = ptr1;
    }
}
```



```

    }
    while (swap);
}

```

```

void create2()
{

```

```

    struct node *newnode, *temp;

```

```

    int x, y;

```

```

    printf("\nEnter elements to second linked list:");
    do

```

```

    {

```

```

        printf("\nEnter Integer: ");

```

```

        scanf("%d", &x);

```

```

        newnode = (struct node*) malloc(sizeof(struct node));

```

```

        newnode->data = x;

```

```

        if (head2 == NULL)

```

```

        {

```

```

            newnode->next = NULL;

```

```

            head2 = newnode;

```

```

            printf("\nfirst node created\n");

```

```

            C++;

```

```

        }

```

```

    else

```

```

    {

```

```

        temp = head2;

```

```

        while (temp->next != NULL)

```

```

        {
            temp = temp->next;

```

```

        }
        temp->next = newnode;

```

```

        newnode->next = NULL;

```

```

        C++;

```

```

        printf("\nNode created\n");
    }

```

```

    printf("\nDo you want to continue adding? 0 or 1");

```

```

    scanf("%d", &y);

```

```

    while (y != 0);
}

```