```c
#include <stdio.h>
#include <stdlib.h>
void create();
void reverse();
void display();
struct node
{
   int data;
   struct node *next;
};

struct node *head=NULL;
int main(int argc, char **argv)
{
    printf("Create a List\n");
    create();
    printf("Reversed list\n");
    reverse();

    display();
}

void create()
{
   struct node *newnode,*temp;
    int item;
    char choice;
    do
    {
newnode =(struct node *) malloc (sizeof(struct node));
printf("Enter the data : ");
scanf("%d",&item);
newnode->data=item;
newnode->next=NULL;
if (head==NULL)
```

myfirst

ere | reverse.c | mergesort.c | concatination.c | *search.c | *stack.c | *queue.c

```
33      newnode->data=item;
34      newnode->next=NULL;
35      if (head==NULL)
36       {
37
38        head=newnode;
39
40       }
41       else
42      {
43      temp=head;
44         while(temp->next!=NULL)
45         {
46                    temp=temp->next;
47          }
48        temp->next=newnode;
49        newnode->next=NULL;
50
51    }
52       printf("Do u want to add element\n");
53       fflush(stdin);
54       scanf("%c",&choice);
55      }while(choice=='y' ||choice=='Y');
56  }
57
58  void reverse()
59  {
60      struct node *prev=NULL,*current=head, *next=NULL;
61      while(current!=NULL)
62      {
63          next=current->next;
64          current->next=prev;
65          prev=current;
66          current=next;
67      }
68      head=prev;
```

```
}

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;
}

void display()
{
    struct node *ptr=NULL;
    ptr=head;

    if(ptr==NULL)
    {
        printf("Nothing to print\n");
    }
    else
    {
        while(ptr!=NULL)
        {
        printf("%d ",ptr->data);
        ptr=ptr->next;
        }
    }

}
```