```
/*
 * C Program to Reverse a Linked List using iterative
 */
#include<stdio.h>
#include<malloc.h>
/*
* A linked list node
*/
struct node
{
    int data;
    struct node* next;
};

//Globally initialized head pointer
struct node* head = NULL;

//function prototyping
struct node* create_node(int);
void insert_begin(int);
void reverse_list();
void print();
int main()
{
    /* Create some nodes and insert data into them */
   // insert_begin(10);
    //insert_begin(90);
    //insert_begin(31);
    //insert_begin(78);
```

```
    //insert_begin(99);
    printf("Linked List before reversed: \n");
    print();
    reverse_list();
    printf("\nLinked List after reversed: \n");
    print();
    return 0;
}

/*
* Creates a new node using the malloc function
*/
struct node* create_node(int data)
{
    struct node* new_node = (struct node*) malloc (sizeof(struct node));
    if (new_node == NULL)
    {
        printf("Memory can't be allocated for new node");
        return NULL;
    }
    else
    {
        new_node -> data = data;
        new_node -> next = NULL;
        return new_node;
    }
}

/*
```

```
* insert a new node at the beginning of the list
*/
void insert_begin(int data)
{
   struct node* new_node = create_node(data);
   if (new_node != NULL)
   {
      new_node -> next = head;
      head = new_node;
   }
}

/*
* reverse the linked list
*/
void reverse_list()
{
   if (head == NULL)
   {
      return;
   }
   struct node* temp = head;
   struct node* new_head = NULL;

   // create new nodes and insert them beginning
   while (temp != NULL)
   {
      struct node* new_node = create_node(temp->data);
      new_node->next = new_head;
```

```
        new_head = new_node;
        temp = temp->next;
    }

    // update the head with the new head
    head = new_head;
}

/*
* prints the linked list
*/
void print()
{
    struct node* temp = head;
    while (temp != NULL)
    {
        printf("%d --> ", temp->data);
        temp = temp->next;
    }
    printf("NULL \n");
}
```