Aim:

Write a C program to reverse elements of a single linked list.

Source Code:

reverseElements.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
struct node
{
    int data;
    struct node *next;
}*first;
int main()
{
    int n,c;
    printf("Enter the total number of nodes: ");
    scanf("%d",&n);
    createNodes(n);
    printf("Data in the list\n");
    printlist();
    printf("Press 1 to reverse the order of singly linked list\n");
    scanf("%d",&c);
    if(c==1)
    {
        reverselist();
    }
    printf("Data in the list\n");
    printlist();
    return 0;
}
void createNodes(int n)
{
    struct node *newnode,*temp;
    int data,i;
    if(n<=0)
    {
        printf("list is empty\n");
        return;
    }
    first=(struct node *)malloc(sizeof(struct node));
    if(first==NULL)
    {
        printf("Unable to allocate memory\n");
    }
    else
    {
        printf("Enter the data of node 1: ");
        scanf("%d",&data);
        first->data=data;
        first->next=NULL;
```

```c
        temp=first;
        for(i=2;i<=n;i++)
        {
            newnode=(struct node *)malloc(sizeof(struct node));
            if(newnode==NULL)
            {
                printf("Unable to alocate memory");
                break;
            }
            else
            {
                printf("Enter the data of node %d: ",i);
                scanf("%d",&data);
                newnode->data=data;
                newnode->next=NULL;
                temp->next=newnode;
                temp=temp->next;
            }
        }

    }
}
void reverselist()
{
    struct node *pre,*cur;
    if(first!=NULL)
    {
        pre=first;
        cur=first->next;
        first=first->next;
        pre->next=NULL;
        while(first!=NULL)
        {
            first=first->next;
            cur->next=pre;
            pre=cur;
            cur=first;
        }
        first=pre;
    }
}
void printlist()
{
    struct node *temp;
    if(first==NULL)
    {
        printf("List is empty\n");
    }
    else
    {
        temp=first;
        while(temp!=NULL)
        {
            printf("Data = %d\n",temp->data);
            temp=temp->next;
        }
```

```
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the total number of nodes: 5
Enter the data of node 1: 26
Enter the data of node 2: 394
Enter the data of node 3: 145
Enter the data of node 4: 624
Enter the data of node 5: 731
Data in the list 1
Data = 26 1
Data = 394 1
Data = 145 1
Data = 624 1
Data = 731 1
Press 1 to reverse the order of singly linked list 1
Data in the list
Data = 731
Data = 624
Data = 145
Data = 394
Data = 26

Test Case - 2
User Output
Enter the total number of nodes: 8
Enter the data of node 1: 21
Enter the data of node 2: 94
Enter the data of node 3: 214
Enter the data of node 4: 24
Enter the data of node 5: 45
Enter the data of node 6: 694
Enter the data of node 7: 321
Enter the data of node 8: 356
Data in the list 1
Data = 21 1
Data = 94 1
Data = 214 1
Data = 24 1
Data = 45 1
Data = 694 1
Data = 321 1
Data = 356 1
Press 1 to reverse the order of singly linked list 1
Data in the list
Data = 356
Data = 321

| Data = 694 |
| Data = 45 |
| Data = 24 |
| Data = 214 |
| Data = 94 |
| Data = 21 |