**EXPERIMENT 2**

**Aim :**

Write a program to reverse a linked list using iterative and recursive method.

**Code :**

#include <stdio.h>

#include <stdlib.h>

typedef struct node

{

    int i;

    struct node \*next;

} node1;

void reverselinkedlist(node1 \*head)

{

    if (head->next == NULL)

    {

        printf("%d->", head->i);

    }

    else

    {

        reverselinkedlist(head->next);

        printf("%d->", head->i);

    }

}

int main()

{

    char hh;

    do

    { char ch;

        node1 \*head = NULL, \*tail = NULL;

        do

        {

    printf("\n Program to reverse a linked list (Iterative & Recursive)");

            printf("\n Creating linked list...");

            printf("\n Enter an integer : ");

            if (head == NULL)

            {

                head = malloc(sizeof(node1));

                scanf("%d", &head->i);

                head->next = NULL;

                tail = head;

            }

            else

            {

                node1 \*n1 = malloc(sizeof(node1));

                scanf("%d", &n1->i);

                n1->next = NULL;

                tail->next = n1;

                tail = n1;

            }

            printf(" Linked list is : ");

            node1 \*n2 = head;

            printf(" ");

            while (n2 != NULL)

            {

                printf("%d->", n2->i);

                n2 = n2->next;

            }

            printf("null");

            printf("\n Do you want to enter more elements ? (y/n) : ");

            fflush(stdin);

            scanf("%c", &ch);

        } while (ch == 'y' || ch == 'Y');

        char ch1;

        do

        { int choice;

            printf(" Entered Linked List is : ");

            node1 \*n2 = head;

            printf(" ");

            while (n2 != NULL)

            {

                printf("%d->", n2->i);

                n2 = n2->next;

            }

            printf("null");

            printf("\n 1. Reverse the linked list Iteratively");

            printf("\n 2. Reverse the linked list Recursively");

            printf("\n Enter your choice : ");

            scanf("%d", &choice);

            switch (choice)

            {

            case 1:

                printf("");

                node1 \*n2 = head;

                node1 \*c, \*chead = NULL;

                while (n2 != NULL)

                {

                    c = malloc(sizeof(node1));

                    c->i = n2->i;

                    if (chead == NULL)

                    {

                        c->next = NULL;

                        chead = c;

                    }

                    else

                    {

                        c->next = chead;

                        chead = c;

                    }

                    n2 = n2->next;

                }

                node1 \*x = chead;

                printf("\n Reversed linked list is (Iteratively) : ");

                while (x != NULL)

                {

                    printf("%d->", x->i);

                    x = x->next;

                }

                printf("null");

                break;

            case 2:

                printf("\n Reversed linked list is (Recursively) : ");

                reverselinkedlist(head);

                printf("null");

                break;

            default:

                printf(" Wrong choice!");

                break;

            }

  printf("\n Do you want to reverse using a different choice ? (y/n) : ");

            fflush(stdin);

            scanf("%c", &ch1);

        } while ((ch1 == 'Y') || (ch1 == 'y'));

        printf("\n Do you want to run this program again ? (y/n) : ");

        fflush(stdin);

        scanf("%c", &hh);

    } while ((hh == 'y') || (hh == 'Y'));

    return 0;

}

**Output Screenshot :**

