```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
#include <stdlib.h>

struct node {
        int info;
        struct node *next;
};

struct node *start = NULL;

struct node *create(struct node *start);
struct node *display(struct node *start);
void reverse(struct node *start);

int main() {
        clrscr();
        start = create(start);
        start = display(start);
        printf("\n");
        printf("Reverse \t");
        reverse(start);
        getch();
        return 0;
}

struct node *create(struct node *start) {
        struct node *new_node = NULL, *temp = NULL;
        int val;
        printf("Enter -1 value to exit list.\n");
        printf("Enter the value : \n");
        scanf("%d", &val);
        while (val != -1) {
                new_node = (struct node *)malloc(sizeof(struct node));
                new_node->info = val;
                if (start == NULL) {
                        start = new_node;
                        new_node->next = NULL;
                } else {
                        temp = start;
                        while (temp->next != NULL) {
                                temp = temp->next;
                        }
```

```
                                        temp->next = new_node;
                                        new_node->next = NULL;
                        }
                        printf("Enter the value : \n");
                        scanf("%d", &val);
                }
                printf("List is successfully created.\n");
                return start;
}

struct node *display(struct node *start) {
                struct node *temp = NULL;
                temp = start;
                printf("List is :\n");
                while (temp != NULL) {
                        printf("%d \t", temp->info);
                        temp = temp->next;
                }
                return start;
}

void reverse(struct node *start) {
                struct node *prev = NULL;
                struct node *current = start;
                struct node *next_node;
                while (current != NULL) {
                        next_node = current->next;
                        current->next = prev;
                        prev = current;
                        current = next_node;
                }
                start = prev;
                start = display(start);
}
```