Assignment 5: Singly Linked List

Name: Aarya Gawade
UEC No.: UEC2023122
Batch: A2

Code:

```c
#include <stdio.h>
#include <stdlib.h>

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

void display(struct node *curr);
void display_rev(struct node *curr);

int main()
{
    struct node *head = NULL, *curr, *p, *q, *r;
    int ch, c2, c3, fnd = 0, pos = 0, sr, n;
    char c1;
    curr = (struct node *)malloc(sizeof(struct node));
    printf("Enter data: ");
    scanf("%d", &curr->number);
    head = curr;
    do
    {
        printf("Do you want to continue (y/n)? ");
        scanf(" %c", &c1);

        if (c1 == 'y')
```

```
        {
            curr->next = (struct node *)malloc(sizeof(struct node));
            curr = curr->next;
            printf("Enter data: ");
            scanf("%d", &curr->number);
        }

    } while (c1 == 'y' || c1 == 'Y');

    curr->next = NULL;
    display(head);

    do
    {
        printf("\n1. Search\t2. Insert\t3. Delete\t4. Revert\t5. Display
reverse\t6. exit\n");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            // search
            printf("Enter key to be searched: ");
            scanf("%d", &sr);
            curr = head;
            while (fnd == 0 && curr != NULL)
            {
                if (curr->number == sr)
                {
                    fnd = 1;
                    break;
                }
                else
                {
                    pos++;
                    curr = curr->next;
                }
            }
            if (fnd == 1)
                printf("%d is found at position: %d", sr, pos + 1);
            else
```

```
                printf("Not found");

            break;

        case 2:
            // insert
            p = (struct node *)malloc(sizeof(struct node));
            printf("Enter data: ");
            scanf("%d", &p->number);
            p->next = NULL;
            printf("Insert at 1. Beginning 2. End 3. Anywhere in
middle\n");
            scanf("%d", &c2);

            // at head
            if (c2 == 1)
            {
                p->next = head;
                head = p;
            }
            // at end
            if (c2 == 2)
            {
                curr = head;
                while (curr->next != NULL)
                {
                    curr = curr->next;
                }
                curr->next = p;
            }
            // anywhere in the middle
            if (c2 == 3)
            {
                printf("Enter the node number after which you have to
insert new node: ");
                scanf("%d", &n);
                curr = head;
                while (curr->number != n)
                {
                    curr = curr->next;
```

```
                }
                p->next = curr->next;
                curr->next = p;
            }
            display(head);
            break;

        case 3:
            printf("Delete");
        }

    } while (ch != 6);

    return 0;
}

void display(struct node *curr)
{
    if (curr == NULL)
        printf("Empty list");
    else
    {
        while (curr != NULL)
        {
            printf("%d\t", curr->number);
            curr = curr->next;
        }
    }
}

void display_rev(struct node *curr)
{
    if (curr != NULL)
    {
        display_rev(curr->next);
        printf("%d", curr->number);
    }
}
```

Output:

```
D:\OneDrive\Dokumen\Clg_work>cd "d:\OneDrive\Dokumen\Clg_work\Assignments\" && gcc 5sll.c -o 5sll && "d:\OneDrive\Dokumen\Clg_work\Assignments\"5sll
Enter data: 1
Do you want to continue (y/n)? y
Enter data: 2
Do you want to continue (y/n)? y
Enter data: 3
Do you want to continue (y/n)? y
Enter data: 4
Do you want to continue (y/n)? n
1       2       3       4
1. Search       2. Insert       3. Delete       4. Revert       5. Display reverse      6. exit
1
Enter key to be searched: 2
2 is found at position: 2
1. Search       2. Insert       3. Delete       4. Revert       5. Display reverse      6. exit
2
Enter data: 5
Insert at 1. Beginning 2. End 3. Anywhere in middle
3
Enter the node number after which you have to insert new node: 2
1       2       5       3       4
1. Search       2. Insert       3. Delete       4. Revert       5. Display reverse      6. exit
5
4       3       5       2       1
```