

Zayd Ahmed
1BM21CS254

Program to Perform Sorting,Reversing,Concatenation and displaying in Linked list

Input:-

```
// program to perform operations of linked list
// sorting
// reversal
// concatenation

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

struct NODE
{
    int data;
    struct NODE *link;
};

typedef struct NODE Node;
Node *Start, *new, *curr;

void create()
{
    int ch;
    Start = (Node *)malloc(sizeof(Node));
    printf("enter the element\n");
    scanf("%d", &Start->data);
    curr = Start;
    printf("do you want to enter more elements(1/0)\n");
    scanf("%d", &ch);
    while (ch == 1)
    {
        new = (Node *)malloc(sizeof(Node));
        printf("enter the element\n");
        scanf("%d", &new->data);
        curr->link = new;
        curr = new;
    }
}
```

```

        printf("do you want to enter more elements(1/0)\n");
        scanf("%d", &ch);
    }
    curr->link = NULL;
}

void sort()
{
    if (Start == NULL)
    {
        printf("linked list does not exist\n");
        return;
    }

    else
    {
        int t, cnt = 0;
        int arr[100];
        Node *temp1, *temp2;
        temp1 = Start;
        while (temp1 != NULL)
        {
            arr[cnt] = temp1->data;
            temp1 = temp1->link;
            cnt++;
        }
        for (int i = 0; i < cnt; i++)
        {
            for (int j = 0; j < cnt - i; j++)
            {
                if ((arr[j]) < (arr[j + 1]))
                {
                    t = arr[j + 1];
                    arr[j + 1] = arr[j];
                    arr[j] = t;
                }
            }
        }
        temp2 = Start;
        for (int i = 0; i < cnt; i++)
        {
            temp2->data = arr[i];
            temp2 = temp2->link;
        }
        return;
    }
}

void reverse()
{
    if (Start == NULL)
    {
        printf("linked list does not exist\n");
        return;
    }
}

```

```

    }
    else
    {
        int t, cnt = 0;
        int arr[100];
        Node *temp1, *temp2;
        temp1 = Start;
        while (temp1 != NULL)
        {
            arr[cnt] = temp1->data;
            temp1 = temp1->link;
            cnt++;
        }
        for (int i = 0; i < cnt; i++)
        {
            for (int j = 0; j < cnt - 1 - i; j++)
            {
                t = arr[j + 1];
                arr[j + 1] = arr[j];
                arr[j] = t;
            }
        }
        temp2 = Start;
        for (int i = 0; i < cnt; i++)
        {
            temp2->data = arr[i];
            temp2 = temp2->link;
        }
        return;
    }
}

void concat()
{
    printf("enter the linked list to concatenate\n");
    int ch;
    Node* Start1 = (Node *)malloc(sizeof(Node));
    printf("enter the element\n");
    scanf("%d", &Start1->data);
    Node* currl = Start1;
    printf("do you want to enter more elements(1/0)\n");
    scanf("%d", &ch);
    while (ch == 1)
    {
        Node* new1 = (Node *)malloc(sizeof(Node));
        printf("enter the element\n");
        scanf("%d", &new1->data);
        currl->link = new1;
        currl = new1;
        printf("do you want to enter more elements(1/0)\n");
        scanf("%d", &ch);
    }
    currl->link = NULL;
}

```

```

        curr->link=Start1;
        printf("Successfully concatenated\n");
        return;
    }

void display()
{
    Node *temp;
    temp = Start;
    do
    {
        printf("%d\n", temp->data);
        temp = temp->link;
    } while (temp != NULL);
}

int main()
{
    int choice;
    do
    {
        printf("1.create\n2.sort\n3.reverse\n4.concatenate\n5.display\n6.exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                create();
                break;
            case 2:
                sort();
                break;
            case 3:
                reverse();
                break;
            case 4:
                concat();
                break;
            case 5:
                display();
                break;
            case 6:
                exit(0);
            default:
                printf("enter a valid choice\n");
                break;
        }
    } while (choice != 6);
    return 0;
}

```

Output:-

```

1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
1
enter the element
10
do you want to enter more elements(1/0)
1
enter the element
20
do you want to enter more elements(1/0)
1
enter the element
30
do you want to enter more elements(1/0)
0
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
5
10
20
30
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
3
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
5
30
20
10
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
4
enter the linked list to concatenate
enter the element
90
do you want to enter more elements(1/0)
1
enter the element
80
do you want to enter more elements(1/0)
0
Successfully concatenated
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
5
30
20
10
90
80
1.create
2.sort
3.reverse
4.concatenate
5.display
6.exit
6
6

...Program finished with exit code 0
Press ENTER to exit console.

```

