Date .....

Expt. No. .....

Page No. 25

Q WAP in C to write functions in linked list to find maximum, minimum, and counting of elements

Source Code

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

typedef struct nd {
        int data;
        struct nd *link;
} node;

node *start;
void display()
{
        node *p;
        if (start == NULL){
                printf ("List is empty!");
                exit(0);
        }
        p == start;
        while (p != NULL) {
                printf ("\n %d", p->data);
                p = p -> link;
        }
}
```

Teacher's Signature :

```
void addatend(){
        node *p, *temp;
        int val;
        temp = (node *) malloc (sizeof(node));
        printf("Enter the value...");
        scanf("%d",&val);
        temp->data = val;
        temp -> link = NULL;
        p = start;
        while(p-> link != NULL)
                p = p -> link;
        p->link = temp;
}

void count(){
        int c=0;
        node *p;
        p = start;
        while(p != NULL){
                c++;
                p = p -> link;
        }
        printf("\nTotal number of node is %d",c);
}
```

```
void evencount(){
    int c=0;
    node *p;
    p=start;
    while(p!=NULL){
        if(p->data%2==0)
            c++;
        p=p->link;
    }
    printf("\n Total even numbers is %d", c);
}

void oddcount(){
    int c=0;
    node *p;
    p=start;
    m=p->data
    while(p!=NULL){
        // if(p->data
        if(p->data%2 != 0)
            c++;
        p=p->link;
    }

void maximum(){
    int m;
    node *p;
    p=start;
```

```
        m = p -> data;
        while (p != NULL){
            if (p -> data > m)
                m = p -> data;
            p = p -> link;
        }
        printf (" The maximum no. is %d", m);
}

void minimum(){
        int m;
        node *p;
        p = start;
        m = p -> data;
        while (p != NULL){
            if (p -> data < m)
                m = p -> data;
            p = p -> link;
        }
        printf ("\n The minimum no. is %d", m);
}

void reverse(){
        node *p, *q, *r;
        p = start;
        q = NULL;
        while (p != NULL){
            r = q;
```

Teacher's Signature : .....

```
            q = p;
            p = p -> link;
            q -> link = r;
        }
        start = q;
}

void main() {

    int val, i;

    start = NULL;
    start = (node*) malloc (sizeof (node));
    printf ("Enter item");
    scanf ("%d", &val);
    start -> data = val;
    start -> link = NULL;

    printf ("Now add notes at the end ...\n");
    for (i = 1; i <= 5; i++)
            add at end();
    printf ("The list is...");
    display();
    count();
    evencount();
    oddcount();
    maximum();
    minimum();
    reverse();
```

Teacher's Signature :

```
    printf(" The reversed list is ...\n");
    display();

    return 0;
}
```

13/4