Experiment 1.3

Student Name:Rajiv Paul **UID:20BCS1812**
Branch: CSE **Section/Group:6B**
Semester: 3rd **Date of Performance: 09/09/2021**
Subject Name:Java Program Lab **Subject Code:20CSP-219**

Q1. Write a program to print the total number of occurrences of a given item in the linked list.

1) **Aim/Overview of the practical:**

To write a program to print the total number of occurrences of a given item in the linked list.

2) Software required:

Vs Code

3) Source Code:

```
#include <iostream>
using namespace std;
class node
{
public:
int data;
node *next;
node(int data)
{
this->data = data;
next = NULL;
}
};
void insert_end(node *head, int data)
{
node *n = new node(data);
if (head == NULL)
{
head = n;
return;
}
node *temp = head;
while (temp->next != NULL)
temp = temp->next;
temp->next = n;
}
```

```
void occurance(node *root, int data)
{
int ans = 0;
node *temp = root;
while (temp)
{
if (temp->data == data)
ans++;
temp = temp->next;
}
cout << "occurance of " << data << " is " << ans << " times." << endl;
}
int main()
{
int data;
cout << "Enter the number: ";
cin >> data;
node *head = new node(1);
```

```
insert_end(head, 2);
insert_end(head, 3);
insert_end(head, 4);
insert_end(head, 9);
insert_end(head, 4);
insert_end(head, 2);
insert_end(head, 1);
insert_end(head, 2);
insert_end(head, 3);
insert_end(head, 9);
insert_end(head, 6);
insert_end(head, 5);
insert_end(head, 6);
insert_end(head, 9);
insert_end(head, 9);
insert_end(head, 9);
insert_end(head, 4);
insert_end(head, 4);
insert_end(head, 9);
insert_end(head, 3);
insert_end(head, 6);
insert_end(head, 1);
occurance(head, data);
return 0;
}
```

4. Output:

```
Enter the number: 9
occurance of 9 is 6 times.
```

Q2. Write a program to multiply every element of the linked list with 10.

1) Aim/Overview of the practical:

To write a program to multiply every element of the linked list with 10.

2) Software required:

Vs Code

3) Source Code:

```
#include <iostream>
using namespace std;
class node
{
public:
int data;
node *next;
node(int val)
{
this->data = val;
next = NULL;
}
};
void insert_end(node *head, int val)
{
node *n = new node(val);
if (head == NULL)
{
head = n;
return;
}
```

```
node *temp = head;
while (temp->next != NULL)
temp = temp->next;
temp->next = n;
}
void display(node *head)
{
node *temp = head;
while (temp != NULL)
{
cout << temp->data << "->";
temp = temp->next;
}
cout << "NULL\n";
}
void multiply(node *&root)
{
node *temp = root;
while (temp)
{
temp->data *= 10;
temp = temp->next;
}
}
```

```
int main()
{
node *head = new node(1);
insert_end(head, 2);
insert_end(head, 3);
insert_end(head, 4);
insert_end(head, 5);
insert_end(head, 6);
insert_end(head, 7);
insert_end(head, 8);
insert_end(head, 9);
multiply(head);
display(head);
return 0;
}
```

4. Output: