Practical No:- 2 (a)

1. (a) Write a program to create a single linked list and display the node elements in reverse order.

Practical Implementation:-

Code:-

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

// Create a class Node to enter values and address in the list

class Node {

public:

int data;

Node\* next;

};

// Function to reverse the linked list

void reverseLL(Node\*\* head)

{

// Create a stack "s" of Node type

stack<Node\*> s;

Node\* temp = \*head;

while (temp->next != NULL) {

// Push all the nodes in to stack

s.push(temp);

temp = temp->next;

}

\*head = temp;

while (!s.empty()) {

// Store the top value of stack in list

temp->next = s.top();

// Pop the value from stack

s.pop();

// update the next pointer in the list

temp = temp->next;

}

temp->next = NULL;

}

// Function to Display the elements in List

void printlist(Node\* temp)

{

while (temp != NULL) {

cout << temp->data << " ";

temp = temp->next;

}

}

// Program to insert back of the linked list

void insert\_back(Node\*\* head, int value)

{

// we have used insertion at back method to enter values

// in the list.(eg: head->1->2->3->4->Null)

Node\* temp = new Node();

temp->data = value;

temp->next = NULL;

// If \*head equals to NULL

if (\*head == NULL) {

\*head = temp;

return;

}

else {

Node\* last\_node = \*head;

while (last\_node->next != NULL)

last\_node = last\_node->next;

last\_node->next = temp;

return;

}

}

// Driver Code

int main()

{

Node\* head = NULL;

insert\_back(&head, 1);

insert\_back(&head, 2);

insert\_back(&head, 3);

insert\_back(&head, 4);

cout << "Given linked list\n";

printlist(head);

reverseLL(&head);

cout << "\nReversed linked list\n";

printlist(head);

return 0;

}

Output:-

