Experiment 2: Two lists are to be maintained containing integer values. The values are not stored in continuous memory locations. Let one list be 7->1->5 and another list be 5->9->2. Create a third list such that the third list contains the sum of the first two list i.e 517+295 =812 and save it as 2->1>8.

Node\* addTwoNumbers(Node\* l1, Node\* l2) {

if(!l1)return NULL;

if(!l2)return NULL;

Node\* ptr = new Node(0);

Node\* prev = ptr;

int cary = 0;

while(l1 || l2){

int sum = (l1 != NULL ? l1->val : 0) + (l2 != NULL ? l2->val : 0) + cary;

cary = sum/10;

Node\* temp = new Node(sum%10);

prev->next = temp;

prev = prev->next;

if(l1)l1=l1->next;

if(l2)l2=l2->next;

}

if(cary){

Node\* temp = new Node(cary);

prev->next = temp;

}

return ptr->next;

}

Experiment 3: Names of persons are saved in a list. Create function to check if the names are palindrome or not.

CO attained CO1, CO3, CO5

#include <bits/stdc++.h>

using namespace std;

class Node {

public:

int data;

Node(int d) { data = d; }

Node\* ptr;

};

// Function to check if the linked list

// is palindrome or not

bool isPalin(Node\* head)

{

// Temp pointer

Node\* slow = head;

// Declare a stack

stack<int> s;

// Push all elements of the list

// to the stack

while (slow != NULL) {

s.push(slow->data);

// Move ahead

slow = slow->ptr;

}

// Iterate in the list again and

// check by popping from the stack

while (head != NULL) {

// Get the top most element

int i = s.top();

// Pop the element

s.pop();

// Check if data is not

// same as popped element

if (head->data != i) {

return false;

}

// Move ahead

head = head->ptr;

}

return true;

}

// Driver Code

int main()

{

// Addition of linked list

Node one = Node(1);

Node two = Node(2);

Node three = Node(3);

Node four = Node(2);

Node five = Node(1);

// Initialize the next pointer

// of every current pointer

five.ptr = NULL;

one.ptr = &two;

two.ptr = &three;

three.ptr = &four;

four.ptr = &five;

Node\* temp = &one;

// Call function to check palindrome or not

int result = isPalin(&one);

if (result == 1)

cout << "isPalindrome is true\n";

else

cout << "isPalindrome is false\n";

return 0;

}