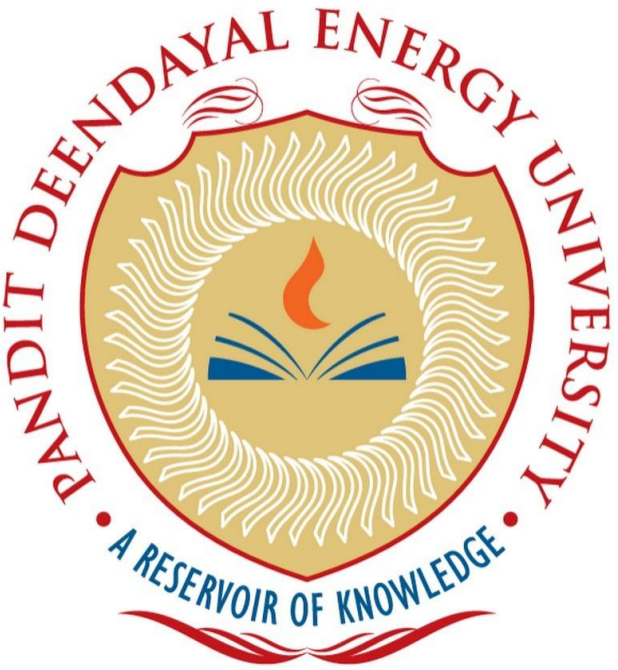
Name – Jay Lapani
Roll No. - 21BCP150
Div - 3 , G - 5



Design & Analysis of Algorithm

Assignment - 3

Adding Two Numbers Represented by Linked Lists

- **Algorithm to add two linked lists :-**

- Reverse the given linked lists l1 and l2.
- Convert the numbers represented by the two linked lists into integers num1 and num2.
- Add the two numbers as sum = num1+num2.
- Convert the above-calculated sum back to a linked list using our to_linkedlist() function which will one-by-one take the digits from the end of the number passed and create a linked list using them. And finally, return it.
- Return the resultant linked list ‘ans’ containing the sum.

- **Program :-**

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
   int data;
   struct Node *next;
};
// Function to create a new node with the given data
struct Node *newNode(int data)
{
   struct Node *node = (struct
                  Node *)malloc(sizeof(struct Node));
   node->data = data;
```

```
    node->next = NULL;
    return node;
}
// Function to reverse a linked list struct Node* reverseList(struct Node* head) {
struct Node *prev = NULL;
struct Node *current = head;
struct Node *next;
while (current != NULL)
{
    next = current->next;
    current->next = prev;
    prev = current;
    current = next;
}
return prev;
}
// Function to add two linked lists
struct Node *addTwoLists(struct Node *first, struct Node *
                                    second)
{
    first = reverseList(first);
    second = reverseList(second);
    struct Node *result = NULL;
    struct Node *temp = NULL;
    int carry = 0, sum;
    while (first != NULL || second != NULL)
    {
        sum = carry + (first ? first->data : 0) + (second ? second->data : 0);
        carry = (sum >= 10) ? 1 : 0;
        sum = sum % 10;
        temp = newNode(sum);
```

```
        if (result == NULL)
        {
            result = temp;
        }
        else
        {
            temp->next = result;
            result = temp;
        }
        if (first)
        {
            first = first->next;
        }
        if (second)
        {
            second = second->next;
        }
    }
    if (carry > 0)
    {
        temp->next = newNode(carry);
    }
    return reverseList(result);
}
// Function to print the linked list void printList(struct Node* head) {     while (head != NULL) {
printf("%d", head->data);         head = head->next;
}
printf("\n");
}
int main()
{
```

```
    struct Node *first = newNode(2);
    first->next = newNode(4);
    first->next->next = newNode(3);
    first->next->next->next = newNode(2);
    struct Node *second = newNode(5);
    second->next = newNode(6);
    second->next->next = newNode(4);
    second->next->next->next = newNode(9);
    struct Node *result = addTwoLists(first, second);
    printList(result);
    return 0;
}
```

- **Output :-**