MOVIKA BHAGAVAKAR
movikabh@gmail.com

ASSIGNMENT-DAY-3

Question 1

Write a function “insert_any()” for inserting a node at any given position of the linked list. Assume position starts at 0.

```
// C++ program for insertion in a single linked
// list at a specified position
#include <bits/stdc++.h>
using namespace std;

// A linked list Node
struct Node {
    int data;
    struct Node* next;
};

// Size of linked list
int size = 0;

// function to create and return a Node
Node* getNode(int data)
{
    // allocating space
    Node* newNode = new Node();

    // inserting the required data
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// function to insert a Node at required position
void insert_any(Node** current, int pos, int data)
{
    // This condition to check whether the
    // position given is valid or not.
    if (pos < 1 || pos > size + 1)
        cout << "Invalid position!" << endl;
    else {
        // Keep looping until the pos is zero
        while (pos--) {

            if (pos == 0) {

                // adding Node at required position
                Node* temp = getNode(data);

                // Making the new Node to point to
                // the old Node at the same position
                temp->next = *current;

                // Changing the pointer of the Node previous
                // to the old Node to point to the new Node
                *current = temp;
            }
            else
            // Assign double pointer variable to point to the
            // pointer pointing to the address of next Node
            current = &(*current)->next;
        }
        size++;
    }
}

// This function prints contents
// of the linked list
void printList(struct Node* head)
{
    while (head != NULL) {
        cout << " " << head->data;
        head = head->next;
    }
    cout << endl;
}

// Driver Code
int main()
```

```
{
        // Creating the list 3->5->8->10
        Node* head = NULL;
        head = getNode(3);
        head->next = getNode(5);
        head->next->next = getNode(8);
        head->next->next->next = getNode(10);

        size = 4;

        cout << "Linked list before insertion: ";
        printList(head);

        int data = 12, pos = 3;
        insert_any(&head, pos, data);
        cout << "Linked list after insertion of 12 at position 3: ";
        printList(head);

        // front of the linked list
        data = 1, pos = 1;
        insert_any(&head, pos, data);
        cout << "Linked list after insertion of 1 at position 1: ";
        printList(head);

        // insetion at end of the linked list
        data = 15, pos = 7;
        insert_any(&head, pos, data);
        cout << "Linked list after insertion of 15 at position 7: ";
        printList(head);

        return 0;
}
```

Question 2

Write a function “delete_beg()” for deleting a node from the beginning of the linked list.

```
// CPP program to remove first node of
// linked list.
#include <iostream>
```

ASSIGNMENT-DAY-3

```
using namespace std;

/* Link list node */
struct Node {
        int data;
        struct Node* next;
};

/* Function to remove the first node
of the linked list */
Node* delete_beg(struct Node* head)
{
        if (head == NULL)
                return NULL;

        // Move the head pointer to the next node
        Node* temp = head;
        head = head->next;

        delete temp;

        return head;
}

// Function to push node at head
void push(struct Node** head_ref, int new_data)
{
        struct Node* new_node = new Node;
        new_node->data = new_data;
        new_node->next = (*head_ref);
        (*head_ref) = new_node;
}

// Driver code
int main()
{
        /* Start with the empty list */
        Node* head = NULL;
        /* Use push() function to construct
        the below list 8 -> 23 -> 11 -> 29 -> 12 */
        push(&head, 12);
        push(&head, 29);
        push(&head, 11);
        push(&head, 23);
        push(&head, 8);

        head = delete_beg(head);
        for (Node* temp = head; temp != NULL; temp = temp->next)
                cout << temp->data << " ";

        return 0;
}
```

Question 3

Write a function “delete_end()” for deleting a node from the end of the linked list.

```
// CPP program to remove last node of
// linked list.
#include <iostream>
using namespace std;

/* Link list node */
struct Node {
        int data;
        struct Node* next;
};

/* Function to remove the last node
of the linked list */
Node* delete_end(struct Node* head)
{
        if (head == NULL)
                return NULL;

        if (head->next == NULL) {
                delete head;
```

```
		return NULL;
	}

	// Find the second last node
	Node* second_last = head;
	while (second_last->next->next != NULL)
		second_last = second_last->next;

	// Delete last node
	delete (second_last->next);

	// Change next of second last
	second_last->next = NULL;

	return head;
}

// Function to push node at head
void push(struct Node** head_ref, int new_data)
{
	struct Node* new_node = new Node;
	new_node->data = new_data;
	new_node->next = (*head_ref);
	(*head_ref) = new_node;
}

// Driver code
int main()
{
	/* Start with the empty list */
	Node* head = NULL;

	/* Use push() function to construct
	the below list 8 -> 23 -> 11 -> 29 -> 12 */
	push(&head, 12);
	push(&head, 29);
	push(&head, 11);
	push(&head, 23);
```

```
    push(&head, 8);


    head = delete_end(head);
    for (Node* temp = head; temp != NULL; temp = temp->next)
        cout << temp->data << " ";


    return 0;
}
```