



Assignment : 3

Q.1) Write function "insertAny()" for inserting node at any given position of linked list. Assume position starts at 0.

→ in C++

```
#include <bits/stdc++.h>
using namespace std;
struct Node {
    int data;
    struct Node* next;
} ;
int size = 0;
Node* getNode(int data)
{
    Node* newnode = new Node();
    newnode->data = data;
    newnode->next = NULL;
    return newnode;
}
void insertPos(Node** current, int pos, int data)
{
    if (pos < 1 || pos > size + 1)
        cout << "Invalid position!"
```

```
else {
    while (pos--) {
        if (pos == 0) {
            Node *temp = getNode(data);
            temp->next = *current;
            *current = temp;
        }
    }
    current = &(*current)->next;
}
size++;
}
```

```
void printList (struct Node *head)
```

```
{
    while (head != NULL) {
        cout << " " << head->data;
        head = head->next;
    }
    cout << endl;
}
```

```
int main ( )
```

```
{ 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;
insertPos(&head, pos, data);
cout << "Linked List after insertion of 12 at position 1 : ";
printList(head);
data = 15, pos = 7;
insertPos(&head, pos, data);
cout << "Linked List after insertion of 15 at position 7 : ";
printList(head);
return 0;
}

```

Q.3) Write function "delete_end()" for deleting a node from the end of linked list

→ in C++ :

```

#include<iostream.h>
using namespace std;
struct Node {
    int data;
    struct Node *next;
};
Node *removeLastNode(struct Node *head)
{

```

```

    if (head == NULL)
        return NULL;
    if (head->next == NULL)
    {
        delete head;
        return NULL;
    }
    Node *second_last = head;
    while (second_last->next->next != NULL)
    {
        second_last = second_last->next;
        delete (second_last->next = NULL);
    }
    return 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;
}

int main()
{
    Node *head = NULL;
    push (&head, 12);
    push (&head, 29);
    push (&head, 11);
    push (&head, 23);
    push (&head, 8);
    head = removeLastNode (head);
    for (Node *temp = head; temp != NULL;
        cout << temp->data << " "; cout return 0, 3
}

```

Q.8)

Write function delete_beg() for deleting node from beginning of linked list

→

```
#include < stdio.h >
```

```
#include < stdlib.h >
```

```
struct node
```

```
{
```

```
int data;
```

```
struct node * next;
```

```
}
```

```
void push(struct node ** head_ref, int new_data)
```

```
{
```

```
struct node * new_node = (struct node *) malloc
```

```
(sizeof(struct node));
```

```
new_node->data = new_data;
```

```
new_node->next = (*head_ref);
```

```
(*head_ref) = new_node;
```

```
}
```

```
void deletenode(struct node ** head_ref, int key)
```

```
{ struct node * temp = *head_ref, * prev;
```

```
if (temp == NULL || temp->data == key)
```

```
{
```

```
*head_ref = temp->next;
```

```
free(temp);
```

```
return;
```

```
}
```

```

        while (temp != NULL & & temp->data != key)
    {
        prev = temp;
        temp = temp->next;
    }
    if (temp == NULL) return;
    prev->next = temp->next;
    free(temp);
}

void printList(struct Node *node)
{
    while (node != NULL)
    {
        printf("%d", node->data);
        node = node->next;
    }
}

// Driver code
int main()
{
    Node *head = NULL;
    push(&head, 7);
    push(&head, 2);
    push(&head, 3);
    push(&head, 2);
    puts("Created Linked List:");
    printList(head);
    deleteNode(&head, 2);
    puts("In linked list after deletion of 2:");
    printList(head);
    return 0;
}

```

created Linked List :

2 3 1 7

Linked List after Deletion of 1 :

2 3 7