

|  |
| --- |
| **LAB 4** of DSA LAB |

**Submitted by:**

**Name:** Muhammad shuraim

**Roll no:** SU92-S24-BSSEM-103

**Section:** 3A

**Submitted to**

Sir Rasikh Ali

**Question 1**

#include <iostream>

using namespace std;

class Node {

public:

int value;

Node\* next;

Node(int val) {

value = val;

next = nullptr;

}

};

class LinkedList {

public:

Node\* head;

LinkedList() {

head = nullptr;

}

void insert\_at\_any\_pos(int pos, int value) {

if (pos < 1) {

cout << "Invalid Position!" << endl;

return;

}

Node\* newnode = new Node(value);

if (pos == 1) {

newnode->next = head;

head = newnode;

return;

}

Node\* temp = head;

int count = 1;

while (temp != nullptr && count < pos - 1) {

temp = temp->next;

count++;

}

if (temp == nullptr) {

cout << "Invalid Position!" << endl;

delete newnode;

return;

}

newnode->next = temp->next;

temp->next = newnode;

}

void display() {

if (head == nullptr) {

cout << "List is empty!" << endl;

return;

}

Node\* temp = head;

while (temp != nullptr) {

cout << temp->value << " -> ";

temp = temp->next;

}

cout << "NULL" << endl;

}

};

int main() {

LinkedList list;

list.insert\_at\_any\_pos(1, 5);

list.insert\_at\_any\_pos(2, 10);

list.insert\_at\_any\_pos(3, 15);

list.insert\_at\_any\_pos(2, 20);

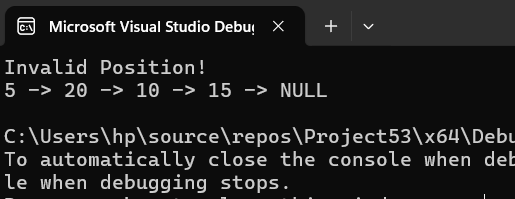
list.insert\_at\_any\_pos(10, 25);

list.display();

return 0;

}

**OUTPUT**



Explanation

* Class node represents a node with value and pointer next
* Class linked list contain
* A fucnction insert\_at\_any\_pos(int pos, int value)
* If(pos==1)insert at start
* Otherwise transverse to (pos-1) node and insert
* If pos>listsize then invalid
* display