SUPERIOR UNIVERSITY

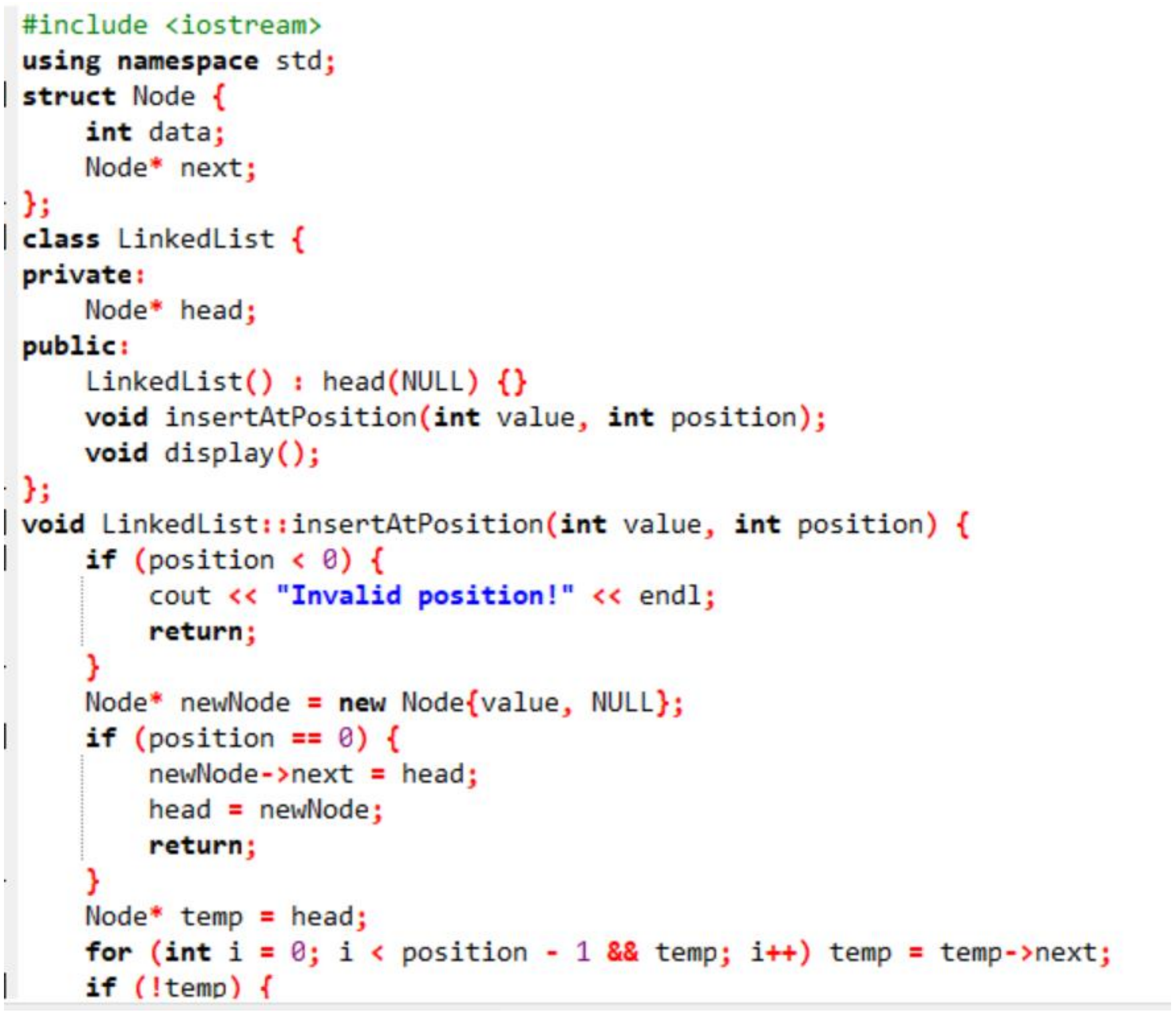
NAME :ZOHAIB KHAN

ROLL_ NO : 022

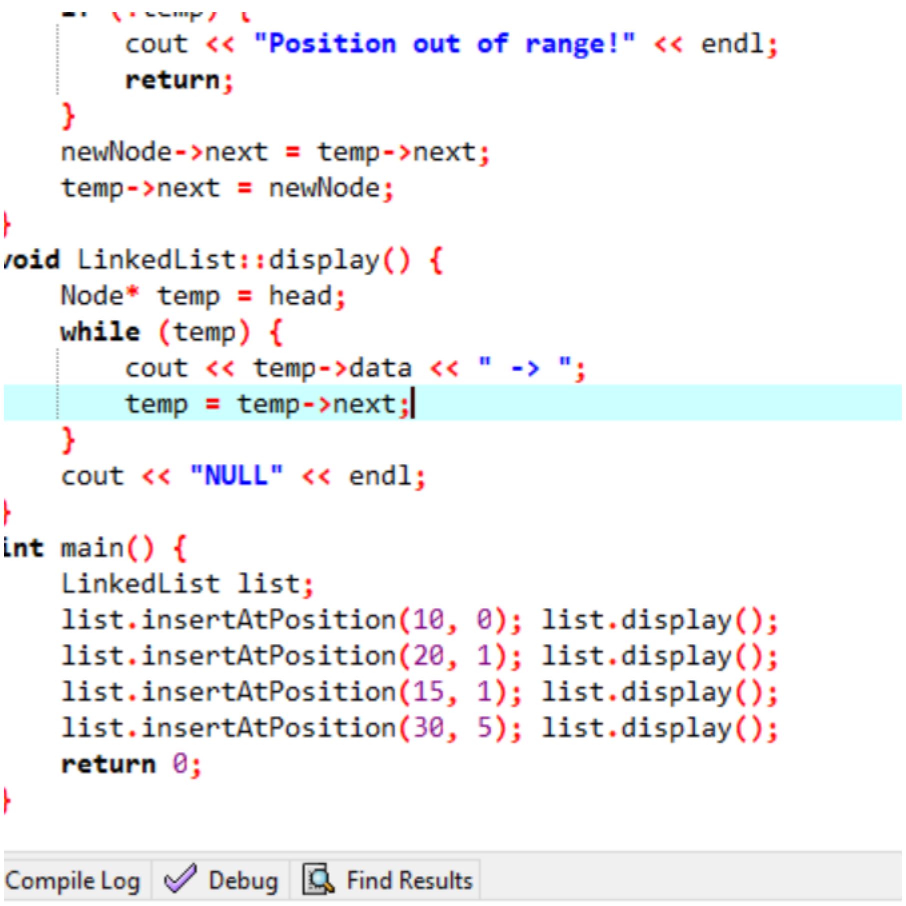
SUBJECT: DSA LAB

LAB NO 4

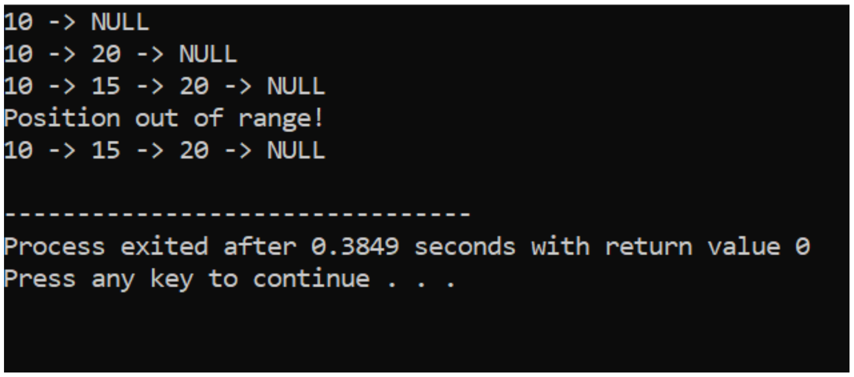
Task: Write a function to insert a node at a specific position in a singly linked list, ensuring valid position handling



```cpp
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node* next;
};
class LinkedList {
private:
    Node* head;
public:
    LinkedList() : head(NULL) {}
    void insertAtPosition(int value, int position);
    void display();
};
void LinkedList::insertAtPosition(int value, int position) {
    if (position < 0) {
        cout << "Invalid position!" << endl;
        return;
    }
    Node* newNode = new Node{value, NULL};
    if (position == 0) {
        newNode->next = head;
        head = newNode;
        return;
    }
    Node* temp = head;
    for (int i = 0; i < position - 1 && temp; i++) temp = temp->next;
    if (!temp) {
```



```
        cout << "Position out of range!" << endl;
        return;
    }
    newNode->next = temp->next;
    temp->next = newNode;
}
void LinkedList::display() {
    Node* temp = head;
    while (temp) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}
int main() {
    LinkedList list;
    list.insertAtPosition(10, 0); list.display();
    list.insertAtPosition(20, 1); list.display();
    list.insertAtPosition(15, 1); list.display();
    list.insertAtPosition(30, 5); list.display();
    return 0;
}
```



```
10 -> NULL
10 -> 20 -> NULL
10 -> 15 -> 20 -> NULL
Position out of range!
10 -> 15 -> 20 -> NULL

--------------------------------
Process exited after 0.3849 seconds with return value 0
Press any key to continue . . .
```

1. insertAtPosition(int value, int position): Inserts a node at a specific position, handling invalid positions.
2. display(): Traverses and prints the linked list.
3. Node* newNode = new Node{value, nullptr}: Dynamically allocates memory for a new node.