Insert a node at a specific position in a linked list

```
#include <bits/stdc++.h>

using namespace std;

class SinglyLinkedListNode {
    public:
        int data;
        SinglyLinkedListNode *next;

        SinglyLinkedListNode(int node data) {
            this->data = node data;
            this->next = nullptr;
        }
};

class SinglyLinkedList {
    public:
        SinglyLinkedListNode *head;
        SinglyLinkedListNode *tail;

        SinglyLinkedList() {
            this->head = nullptr;
            this->tail = nullptr;
        }

        void insert_node(int node_data) {
            SinglyLinkedListNode* node = new
SinglyLinkedListNode(node_data);

            if (!this->head) {
                this->head = node;
            } else {
                this->tail->next = node;
            }

            this->tail = node;
        }
};

void print_singly_linked_list(SinglyLinkedListNode* node, string
sep, ofstream& fout) {
    while (node) {
        fout << node->data;
```

```
        node = node->next;

        if (node) {
            fout << sep;
        }
    }
}

void free_singly_linked_list(SinglyLinkedListNode* node) {
    while (node) {
        SinglyLinkedListNode* temp = node;
        node = node->next;

        free(temp);
    }
}

/*
 * Complete the 'insertNodeAtPosition' function below.
 *
 * The function is expected to return an
INTEGER_SINGLY_LINKED_LIST.
 * The function accepts following parameters:
 *  1. INTEGER_SINGLY_LINKED_LIST llist
 *  2. INTEGER data
 *  3. INTEGER position
 */

/*
 * For your reference:
 *
 * SinglyLinkedListNode {
 *     int data;
 *     SinglyLinkedListNode* next;
 * };
 *
 */

SinglyLinkedListNode* insertNodeAtPosition(SinglyLinkedListNode*
llist, int data, int position) {
    SinglyLinkedListNode* newNode=new SinglyLinkedListNode(data);
    if(position==0){
        newNode->next=llist;
        return newNode;
    }
    SinglyLinkedListNode* temp=llist;
```

```
    for(int i=0;i<position-1;i++){
        temp=temp->next;
    }
    newNode->next=temp->next;
    temp->next=newNode;
    return llist;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    SinglyLinkedList* llist = new SinglyLinkedList();

    int llist_count;
    cin >> llist_count;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    for (int i = 0; i < llist_count; i++) {
        int llist_item;
        cin >> llist_item;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        llist->insert_node(llist_item);
    }

    int data;
    cin >> data;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    int position;
    cin >> position;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    SinglyLinkedListNode* llist_head =
insertNodeAtPosition(llist->head, data, position);

    print singly linked list(llist head, " ", fout);
    fout << "\n";

    free singly linked list(llist head);

    fout.close();

    return 0;
}
```