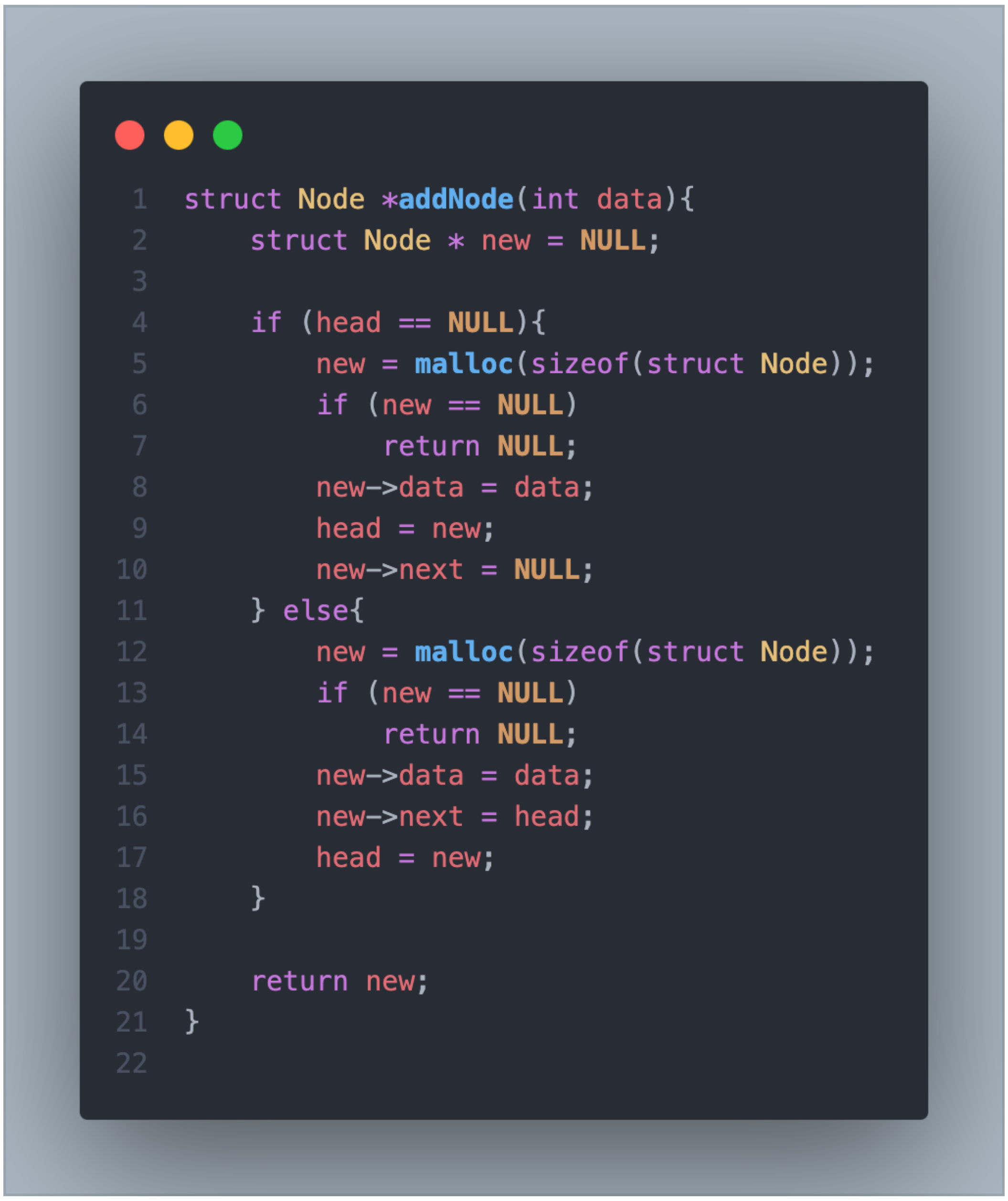
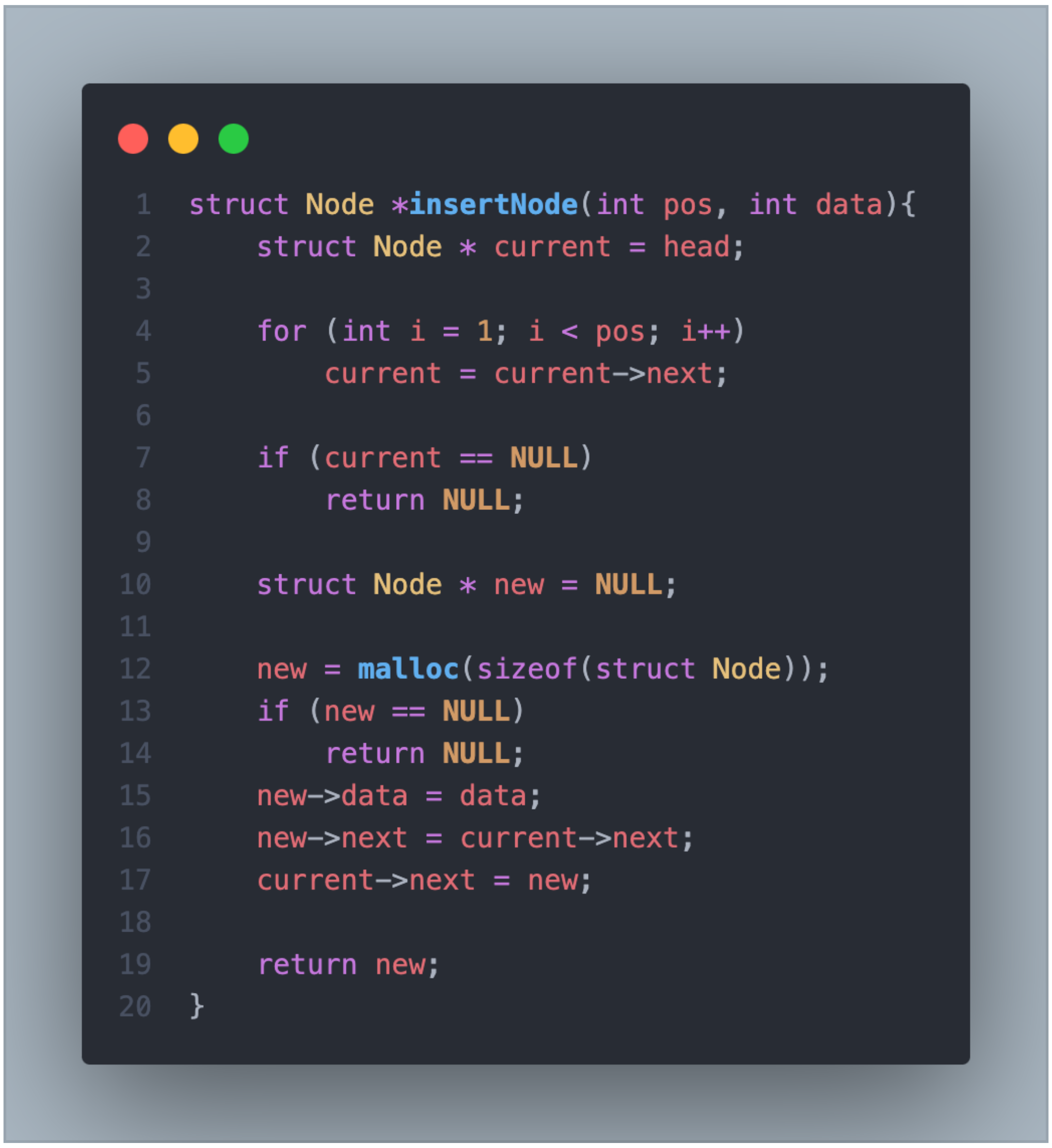
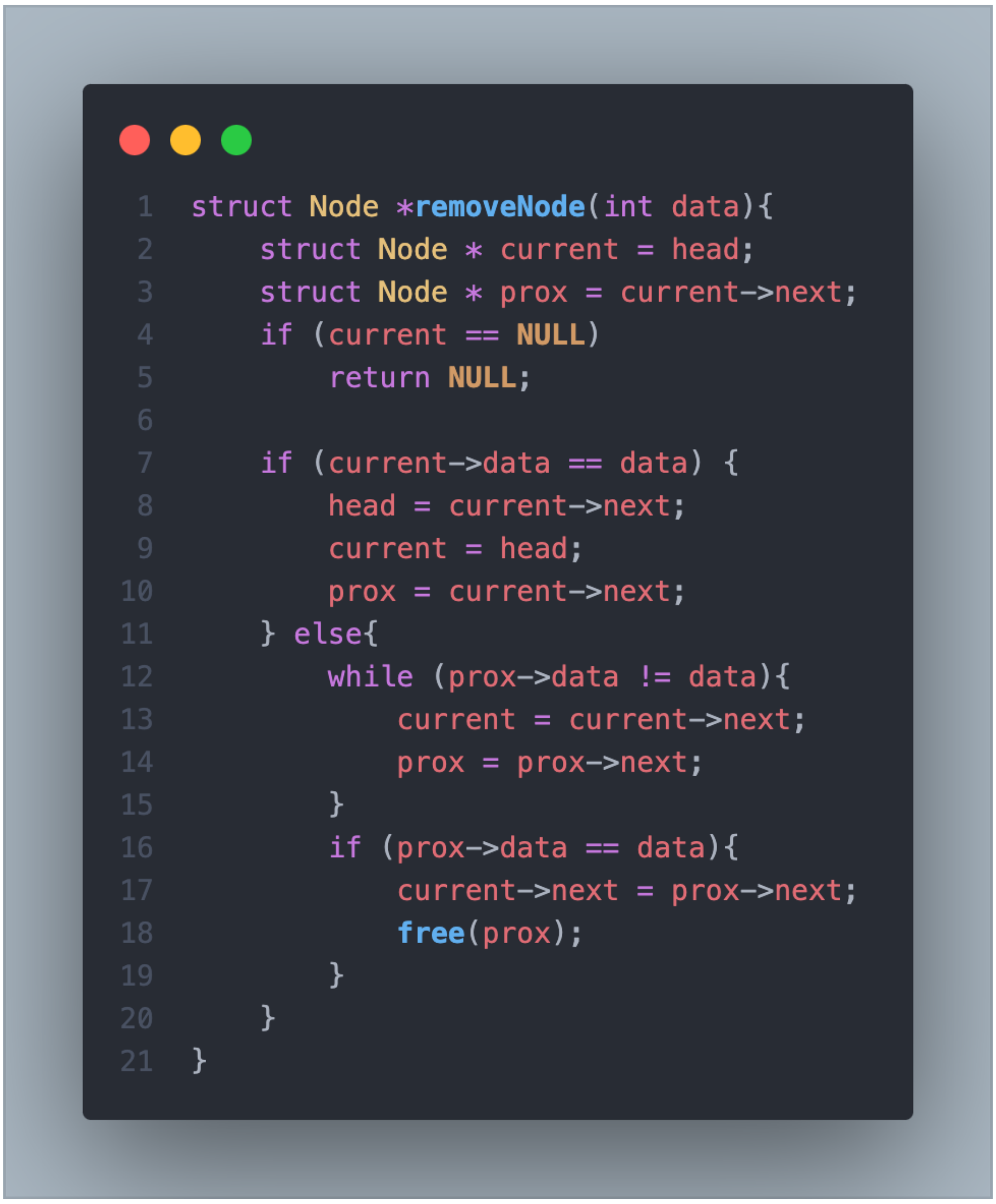
OPERAZIONI CON LE LISTE



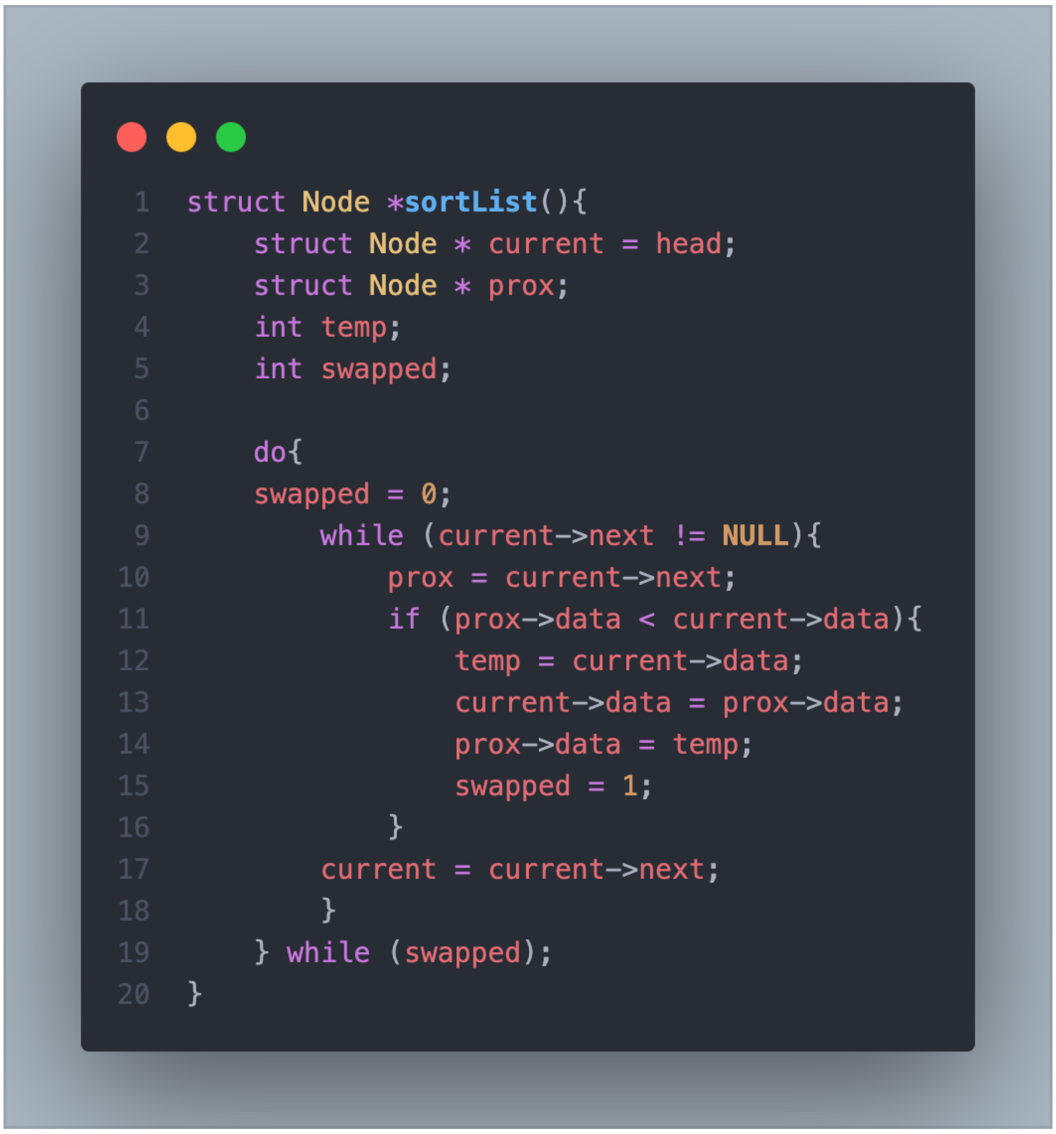
```c
struct Node *addNode(int data){
    struct Node * new = NULL;

    if (head == NULL){
        new = malloc(sizeof(struct Node));
        if (new == NULL)
            return NULL;
        new->data = data;
        head = new;
        new->next = NULL;
    } else{
        new = malloc(sizeof(struct Node));
        if (new == NULL)
            return NULL;
        new->data = data;
        new->next = head;
        head = new;
    }

    return new;
}
```



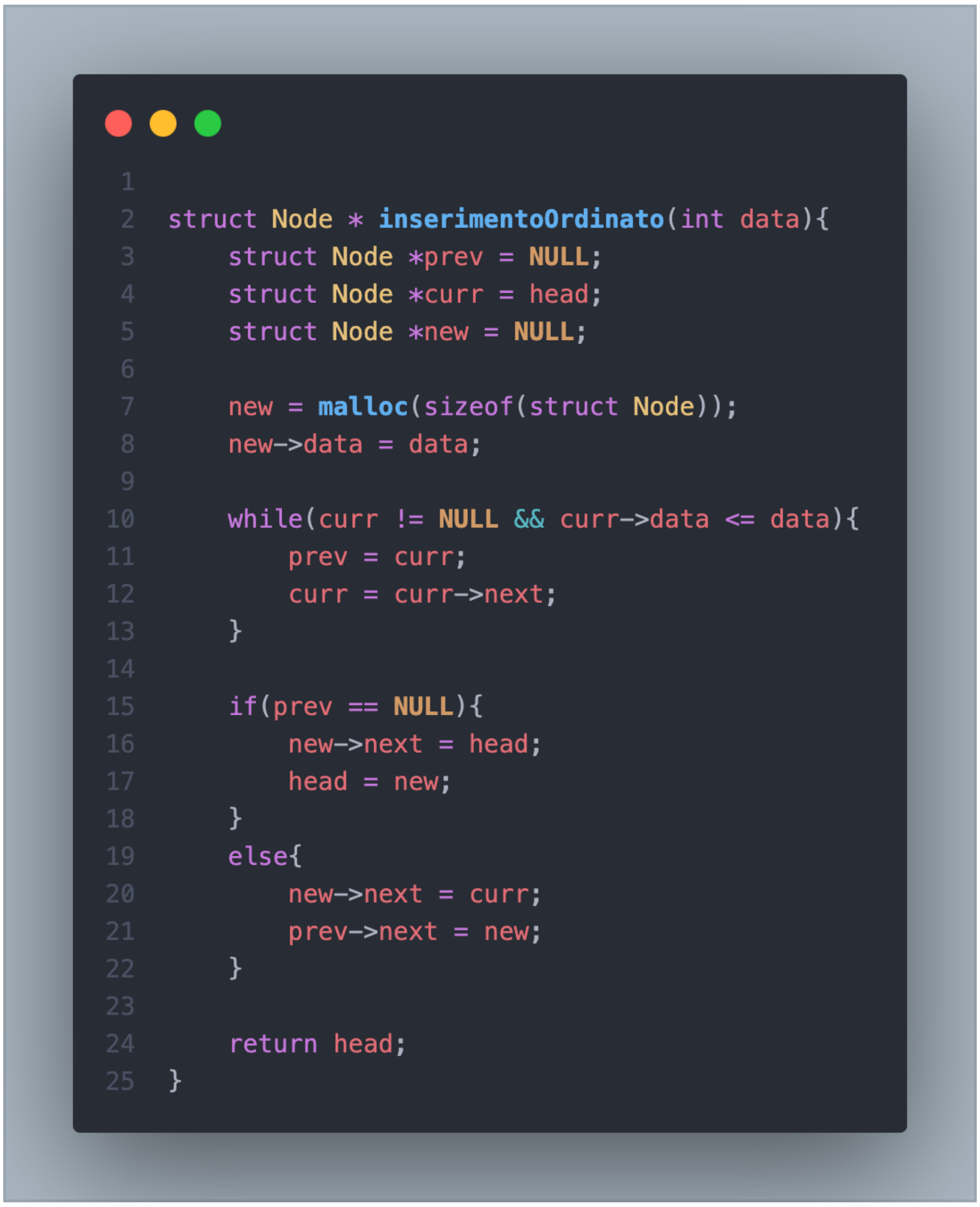
```c
struct Node *insertNode(int pos, int data){
    struct Node * current = head;

    for (int i = 1; i < pos; i++)
        current = current->next;

    if (current == NULL)
        return NULL;

    struct Node * new = NULL;

    new = malloc(sizeof(struct Node));
    if (new == NULL)
        return NULL;
    new->data = data;
    new->next = current->next;
    current->next = new;

    return new;
}
```



```c
struct Node *removeNode(int data){
    struct Node * current = head;
    struct Node * prox = current->next;
    if (current == NULL)
        return NULL;

    if (current->data == data) {
        head = current->next;
        current = head;
        prox = current->next;
    } else{
        while (prox->data != data){
            current = current->next;
            prox = prox->next;
        }
        if (prox->data == data){
            current->next = prox->next;
            free(prox);
        }
    }
}
```



```c
struct Node *sortList(){
    struct Node * current = head;
    struct Node * prox;
    int temp;
    int swapped;

    do{
    swapped = 0;
        while (current->next != NULL){
            prox = current->next;
            if (prox->data < current->data){
                temp = current->data;
                current->data = prox->data;
                prox->data = temp;
                swapped = 1;
            }
        current = current->next;
        }
    } while (swapped);
}
```



```c

struct Node * inserimentoOrdinato(int data){
    struct Node *prev = NULL;
    struct Node *curr = head;
    struct Node *new = NULL;

    new = malloc(sizeof(struct Node));
    new->data = data;

    while(curr != NULL && curr->data <= data){
        prev = curr;
        curr = curr->next;
    }

    if(prev == NULL){
        new->next = head;
        head = new;
    }
    else{
        new->next = curr;
        prev->next = new;
    }

    return head;
}
```