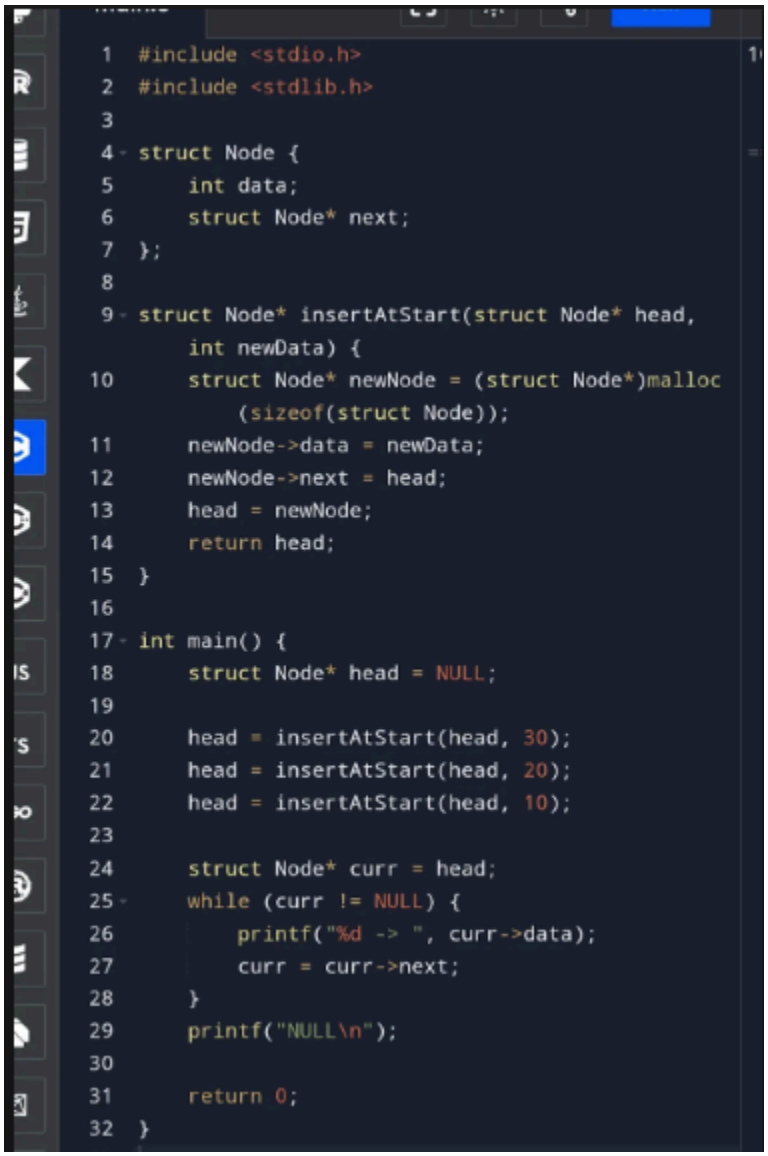
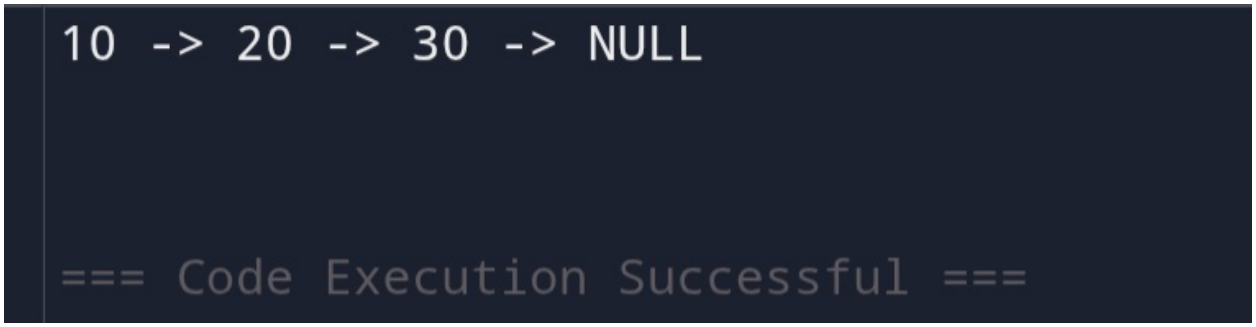
1RUA25BCA0096 Sanjana Pal

Insertion at start:
Input:



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

struct Node {
    int data;
    struct Node* next;
};

struct Node* insertAtStart(struct Node* head,
    int newData) {
    struct Node* newNode = (struct Node*)malloc
        (sizeof(struct Node));
    newNode->data = newData;
    newNode->next = head;
    head = newNode;
    return head;
}

int main() {
    struct Node* head = NULL;

    head = insertAtStart(head, 30);
    head = insertAtStart(head, 20);
    head = insertAtStart(head, 10);

    struct Node* curr = head;
    while (curr != NULL) {
        printf("%d -> ", curr->data);
        curr = curr->next;
    }
    printf("NULL\n");

    return 0;
}
```

Output:



```
10 -> 20 -> 30 -> NULL


=== Code Execution Successful ===
```

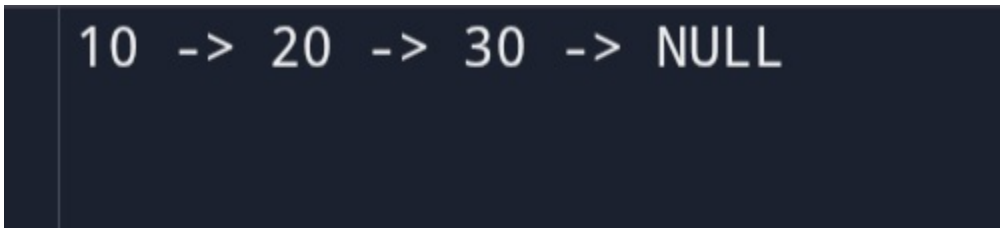
Insertion at end:

1RUA25BCA0096 Sanjana Pal

Input:

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

struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

void insertAtEnd(int newData) {
    struct Node* newNode = (struct Node*)malloc
        (sizeof(struct Node));
    newNode->data = newData;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        return;
    }

    struct Node* temp = head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}

int main() {
    insertAtEnd(10);
    insertAtEnd(20);
    insertAtEnd(30);

    struct Node* curr = head;
    while (curr != NULL) {
        printf("%d -> ", curr->data);
        curr = curr->next;
    }
    printf("NULL\n");

    return 0;
}
```

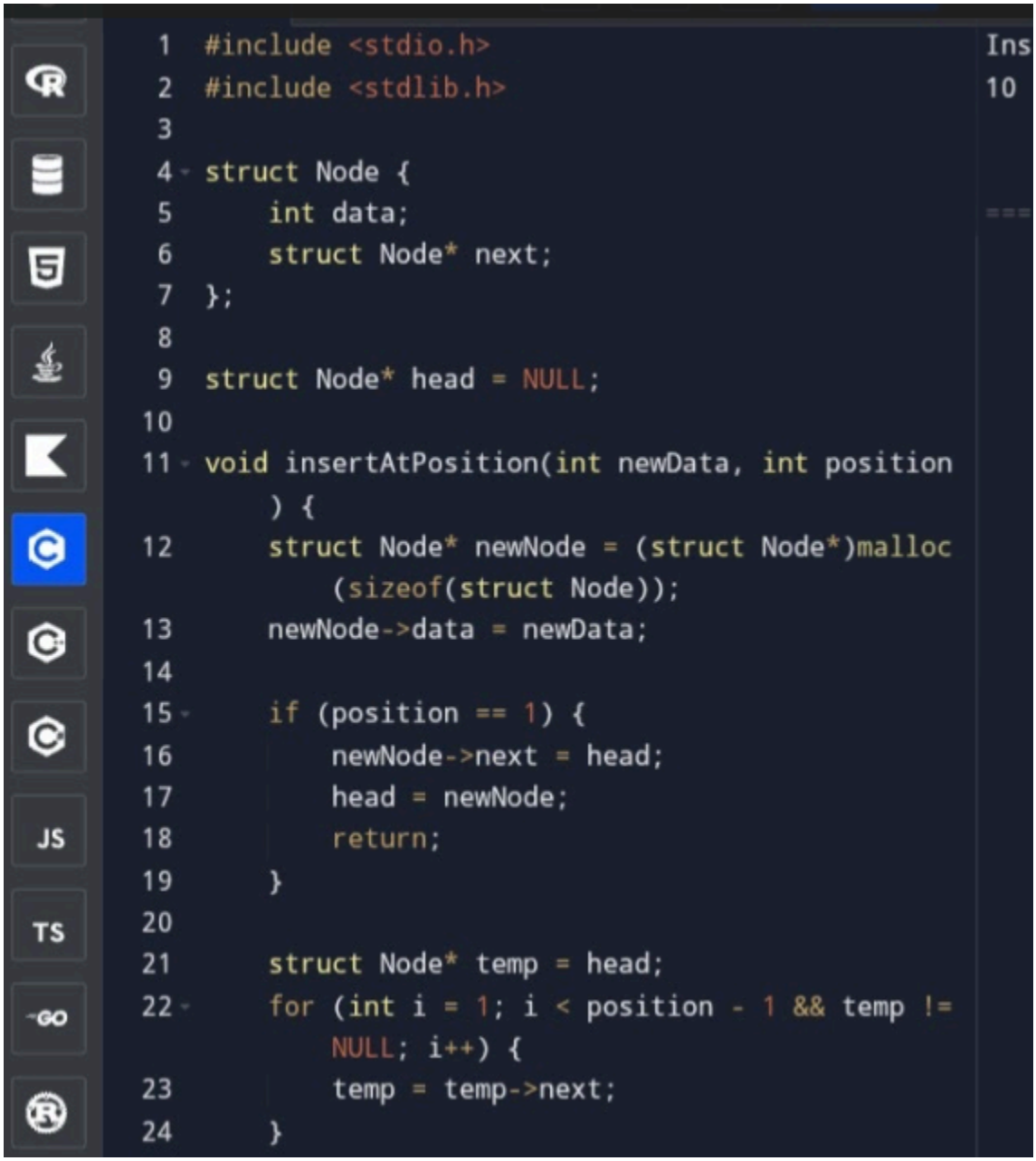
Output:



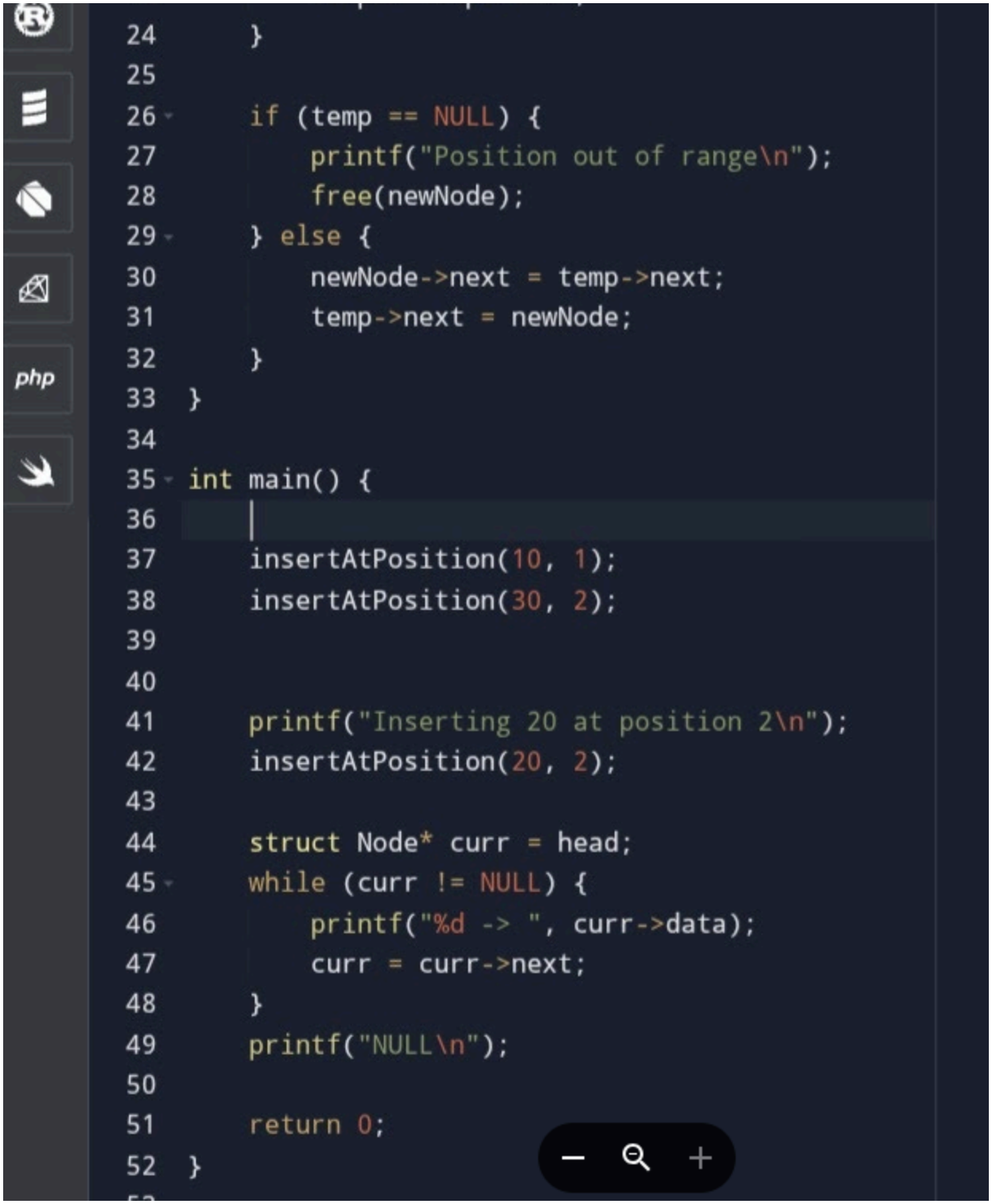
Insertion at position

1RUA25BCA0096 Sanjana Pal

Input:



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

struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

void insertAtPosition(int newData, int position
    ) {
    struct Node* newNode = (struct Node*)malloc
        (sizeof(struct Node));
    newNode->data = newData;

    if (position == 1) {
        newNode->next = head;
        head = newNode;
        return;
    }

    struct Node* temp = head;
    for (int i = 1; i < position - 1 && temp !=
        NULL; i++) {
        temp = temp->next;
    }
```



```
    }

    if (temp == NULL) {
        printf("Position out of range\n");
        free(newNode);
    } else {
        newNode->next = temp->next;
        temp->next = newNode;
    }
}

int main() {

    insertAtPosition(10, 1);
    insertAtPosition(30, 2);


    printf("Inserting 20 at position 2\n");
    insertAtPosition(20, 2);

    struct Node* curr = head;
    while (curr != NULL) {
        printf("%d -> ", curr->data);
        curr = curr->next;
    }
    printf("NULL\n");

    return 0;
}
```

Output:

1RUA25BCA0096 Sanjana Pal