

**NAME: C Sahasra (192472078)**

**COURSE NAME : DATA STRUCTURES FOR MODERN COMPUTING SYSTEMS**

**COURSE CODE : CSA0302**

**WRITE A C PROGRAM TO PERFORM INSERTION OF AN ELEMENT IN DOUBLE LINKED LIST.**

1. **AT BEGINNING**
2. **AT END**
3. **AT ANY SPECIFIC POSITION**

**C PROGRAMMING CODE:**

#include <stdio.h>

#include <stdlib.h>

typedef struct Node {

int data;

struct Node\* prev;

struct Node\* next;

} Node;

Node\* head = NULL;

Node\* createNode(int value) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->data = value;

newNode->prev = NULL;

newNode->next = NULL;

return newNode;

}

void insertAtBeginning(int value) {

Node\* newNode = createNode(value);

if (head == NULL) {

head = newNode;

return;

}

newNode->next = head;

head->prev = newNode;

head = newNode;

}

void insertAtEnd(int value) {

Node\* newNode = createNode(value);

if (head == NULL) {

head = newNode;

return;

}

Node\* temp = head;

while (temp->next != NULL) temp = temp->next;

temp->next = newNode;

newNode->prev = temp;

}

void insertAtPosition(int value, int pos) {

if (pos <= 0) return;

if (pos == 1) {

insertAtBeginning(value);

return;

}

Node\* newNode = createNode(value);

Node\* temp = head;

for (int i = 1; i < pos - 1 && temp != NULL; i++)

temp = temp->next;

if (temp == NULL) {

free(newNode);

return;

}

newNode->next = temp->next;

newNode->prev = temp;

if (temp->next != NULL)

temp->next->prev = newNode;

temp->next = newNode;

}

void display() {

Node\* temp = head;

while (temp != NULL) {

printf("%d <-> ", temp->data);

temp = temp->next;

}

printf("NULL\n");

}

int main() {

insertAtBeginning(20);

printf("After insertion at beginning: ");

display();

insertAtEnd(40);

printf("After insertion at end: ");

display();

insertAtPosition(30, 2);

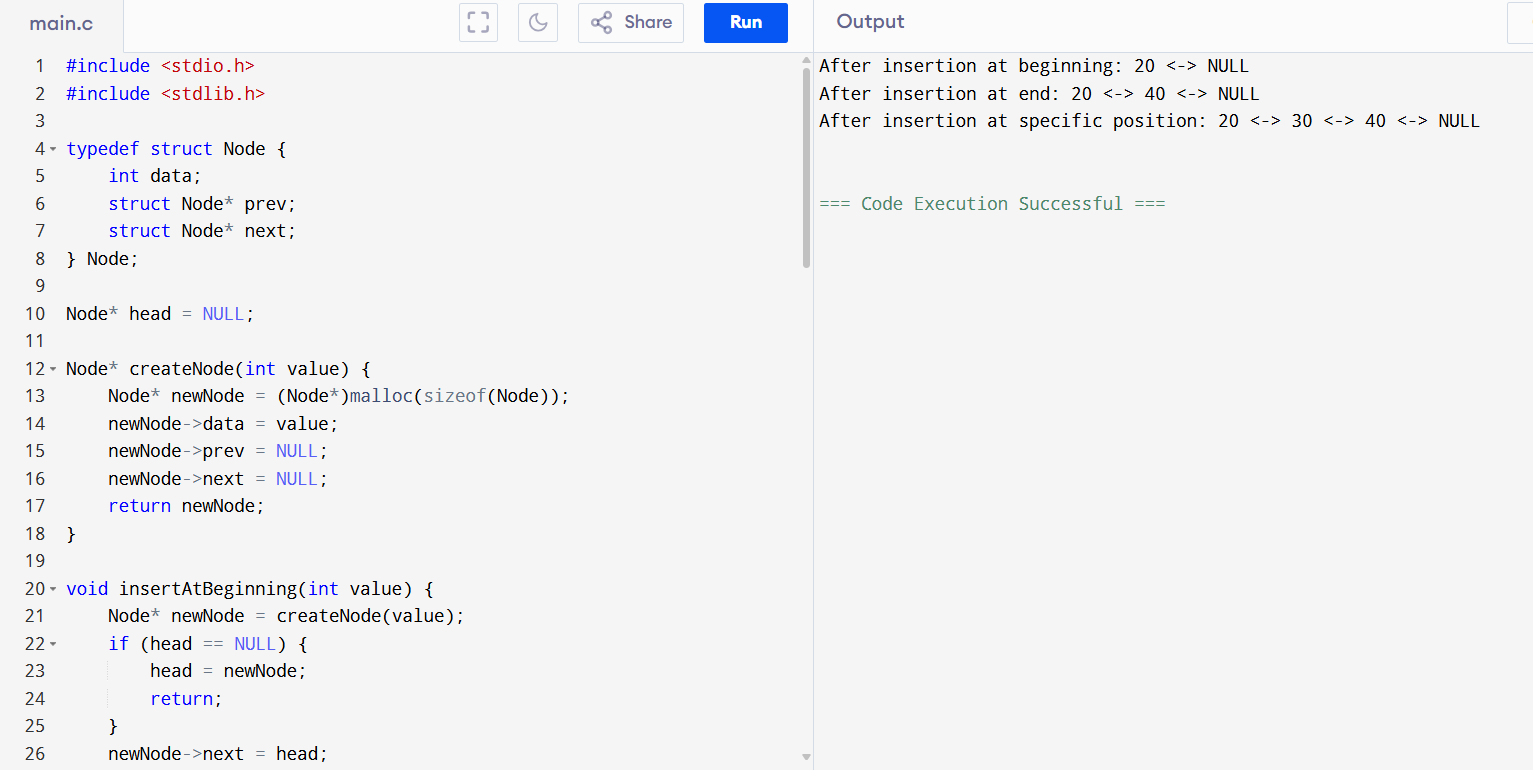
printf("After insertion at specific position: ");

display();

return 0;

}

**OUTPUT:**

****