

1-2-24

Week 70

WAP to implement doubly link list with primitive operations

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

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

```
struct Node
```

```
{
```

```
    int data;
```

```
    struct Node *prev;
```

```
    struct Node *next;
```

```
};
```

```
struct Node *createNode(int data)
```

```
{
```

```
    struct Node *newNode = (struct Node *) malloc (sizeof (struct Node));
```

```
    if (newNode == NULL)
```

```
        printf("Memory allocation failed\n");
```

```
        exit(1);
```

```
    newNode->data = data;
```

```
    newNode->prev = NULL;
```

```
    newNode->next = NULL;
```

```
    return newNode;
```

```
}
```

```
void insertNode(struct Node *head, struct Node *forget, int data)
```

```
{
```

```
    struct Node *newNode = createNode(data);
```

```
    if (forget->prev != NULL)
```

```
{
```

```
        forget->prev->next = newNode;
```

```
        newNode->prev = forget->prev;
```

```
}
```

```
else
```

```
    head = newNode;
```

```
    newNode->next = forget;
```

```
    forget->prev = newNode;
```

```
}
```

```
void deleteNode (struct Node *head, int value)
{
```

```
    struct Node *current = head;
```

```
    while (current != NULL)
```

```
    {
```

```
        if (current->data == value)
```

```
        {
```

```
            if (current->prev != NULL)
```

```
                current->prev->next = current->next;
```

```
            else
```

```
                head = current->next;
```

```
            if (current->next != NULL)
```

```
                current->next->prev = current->prev;
```

```
            free (current);
```

```
            return;
```

```
        }
```

```
        current = current->next;
```

```
    }
```

```
    printf("node with value %d not found \n", value);
```

```
}
```

```
void display (struct Node *head)
```

```
{
```

```
    printf("doubly linked list: \n");
```

```
    while (head != NULL)
```

```
    {
```

```
        printf("%d <==> ", head->data);
```

```
        head = head->next;
```

```
    }
```

```
    printf("NULL \n");
```

```
}
```

NA
12/24

void main()

{

struct Node *head = NULL;

head = createNode(1);

head → next = createNode(2);

head → next → prev = head;

head → next → next = createNode(3);

head → next → next → prev = head → next;

display(head);

insertNode(head, head → next, 10);

printf("after insertion : \n");

display(head);

deleteNode(head, 2);

printf("after deletion : \n");

display(head);

}

Output:

doubly linked list:

1 ↔ 2 ↔ 3 ↔ NULL

after insertion:

doubly linked list:

1 ↔ 10 ↔ 2 ↔ 3 ↔ NULL

after deletion:

doubly linked list:

1 ↔ 10 ↔ 3 ↔ NULL