

## Copilot

Sure, here is a simple example of a singly linked list in C:

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

// Define the Node data structure
typedef struct Node {
    int data;
    struct Node* next;
} Node;

// Function to insert a node at the end of the linked list
void insert(Node** head, int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;

    if (*head == NULL) {
        *head = newNode;
    } else {
        Node* last = *head;
        while (last->next != NULL) {
            last = last->next;
        }
        last->next = newNode;
    }
}

// Function to print the linked list
void printList(Node* node) {
    while (node != NULL) {
        printf("%d ", node->data);
        node = node->next;
    }
}

int main() {
    Node* head = NULL;

    insert(&head, 1);
    insert(&head, 2);
    insert(&head, 3);
    insert(&head, 4);
    insert(&head, 5);

    printList(head);

    return 0;
}
```

This code creates a linked list with nodes containing the values 1, 2, 3, 4, and 5, and then prints these values to the console. The insert function is used to add nodes to the end of the list, and the printList function is used to print the data in each node of the list. The Node structure represents a node in the linked list, containing an integer data and a next pointer to the next node in the list. The head pointer points to the first node in the list. If the list is empty, head is NULL.