```c
// C program to detect loop in a linked list
#include <stdio.h>
#include <stdlib.h>

/* Link list node */
struct Node {
    int data;
    struct Node* next;
};

void push(struct Node** head_ref, int new_data)
{
    /* allocate node */
    struct Node* new_node
        = (struct Node*)malloc(sizeof(struct Node));

    /* put in the data */
    new_node->data = new_data;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref) = new_node;
}

int detectLoop(struct Node* list)
{
    struct Node *slow_p = list, *fast_p = list;

    while (slow_p && fast_p && fast_p->next) {
        slow_p = slow_p->next;
        fast_p = fast_p->next->next;
        if (slow_p == fast_p) {
            return 1;
        }
    }
    return 0;
}
```

```
40:
41: /* Driver program to test above function*/
42: int main()
43: {
44:     /* Start with the empty list */
45:     struct Node* head = NULL;
46:
47:     push(&head, 20);
48:     push(&head, 4);
49:     push(&head, 15);
50:     push(&head, 10);
51:
52:     /* Create a loop for testing */
53:     head->next->next->next->next = head;
54:
55:     if (detectLoop(head))
56:         printf("Loop found");
57:     else
58:         printf("No Loop");
59:     return 0;
60: }
61:
```