

```

1: // C program to detect loop in a linked list
2: #include <stdbool.h>
3: #include <stdio.h>
4: #include <stdlib.h>
5:
6: /* Link list node */
7: typedef struct Node {
8:     int data;
9:     struct Node* next;
10:    int flag;
11: } Node;
12:
13: void push(struct Node** head_ref, int new_data)
14: {
15:     /* allocate node */
16:     struct Node* new_node = (Node*)malloc(sizeof(Node));
17:
18:     /* put in the data */
19:     new_node->data = new_data;
20:
21:     new_node->flag = 0;
22:
23:     /* link the old list off the new node */
24:     new_node->next = (*head_ref);
25:
26:     /* move the head to point to the new node */
27:     (*head_ref) = new_node;
28: }
29:
30: // Returns true if there is a loop in linked list
31: // else returns false.
32: bool detectLoop(struct Node* h)
33: {
34:     while (h != NULL) {
35:         // If this node is already traverse
36:         // it means there is a cycle
37:         // (Because you we encountering the
38:         // node for the second time).
39:         if (h->flag == 1)

```

```

40:         return true;
41:
42:         // If we are seeing the node for
43:         // the first time, mark its flag as 1
44:         h->flag = 1;
45:
46:         h = h->next;
47:     }
48:
49:     return false;
50: }
51:
52: /* Driver program to test above function*/
53: int main()
54: {
55:     /* Start with the empty list */
56:     struct Node* head = NULL;
57:
58:     push(&head, 20);
59:     push(&head, 4);
60:     push(&head, 15);
61:     push(&head, 10);
62:
63:     /* Create a loop for testing */
64:     head->next->next->next->next = head;
65:
66:     if (detectLoop(head))
67:         printf("Loop found");
68:     else
69:         printf("No Loop");
70:
71:     return 0;
72: }
73:
74: // This code is contributed by Aditya Kumar (adityakumar129)
75:

```