```c
// Iterative C program to reverse a linked list
#include <stdio.h>
#include <stdlib.h>

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

/* Function to reverse the linked list */
static void reverse(struct Node** head_ref)
{
    struct Node* prev = NULL;
    struct Node* current = *head_ref;
    struct Node* next = NULL;
    while (current != NULL) {
        // Store next
        next = current->next;

        // Reverse current node's pointer
        current->next = prev;

        // Move pointers one position ahead.
        prev = current;
        current = next;
    }
    *head_ref = prev;
}

/* Function to push a node */
void push(struct Node** head_ref, int new_data)
{
    struct Node* new_node
        = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}
```

```
40:
41: /* Function to print linked list */
42: void printList(struct Node* head)
43: {
44:     struct Node* temp = head;
45:     while (temp != NULL) {
46:         printf("%d ", temp->data);
47:         temp = temp->next;
48:     }
49: }
50:
51: /* Driver code*/
52: int main()
53: {
54:     /* Start with the empty list */
55:     struct Node* head = NULL;
56:
57:     push(&head, 20);
58:     push(&head, 4);
59:     push(&head, 15);
60:     push(&head, 85);
61:
62:     printf("Given linked list\n");
63:     printList(head);
64:     reverse(&head);
65:     printf("\nReversed Linked list \n");
66:     printList(head);
67:     getchar();
68: }
69:
```