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

// A Linked List Node
struct Node
{
    int data;
    struct Node* next;
};

// Helper function to print a given linked list
void printList(struct Node* head)
{
    struct Node* ptr = head;
    while (ptr)
    {
        printf("%d —> ", ptr->data);
        ptr = ptr->next;
    }

    printf("NULL");
}

// Helper function to insert a new node at the beginning of the linked list
void push(struct Node** head, int data)
```

```
{
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = *head;
    *head = newNode;
}

// Remove duplicates from a sorted list
void removeDuplicates(struct Node* head)
{
    // do nothing if the list is empty
    if (head == NULL) {
        return;
    }

    struct Node* current = head;

    // compare the current node with the next node
    while (current->next != NULL)
    {
        if (current->data == current->next->data)
        {
            struct Node* nextNext = current->next->next;
            free(current->next);
            current->next = nextNext;
        }
        else {
            current = current->next;    // only advance if no deletion
        }
    }
}

int main(void)
{
    // input keys
    int keys[] = {1, 2, 2, 2, 3, 4, 4, 5};
    int n = sizeof(keys)/sizeof(keys[0]);

    // points to the head node of the linked list
    struct Node* head = NULL;

    // construct a linked list
    for (int i = n-1; i >= 0; i--) {
        push(&head, keys[i]);
    }

    removeDuplicates(head);

    // print linked list
    printList(head);
    return 0;

}
```