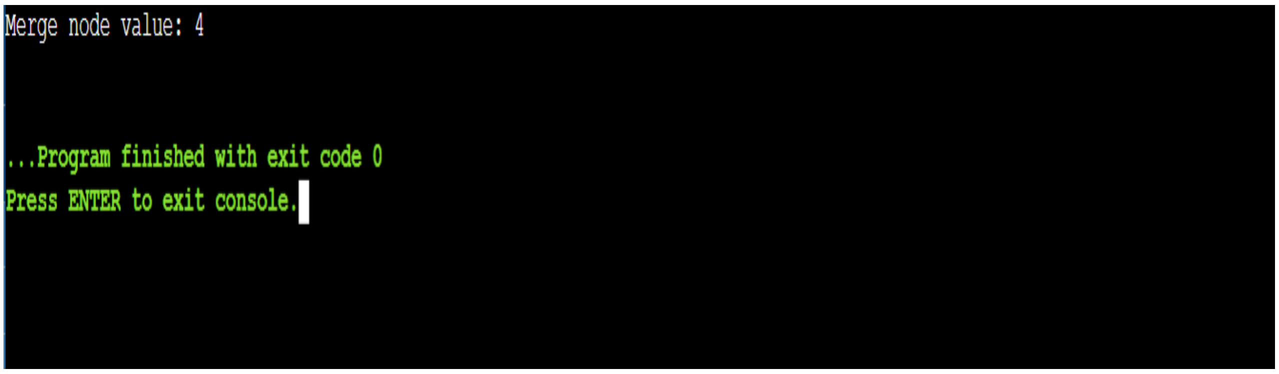
```c
#include <stdio.h>
#include <stdlib.h>

// Definition for singly-linked list.
struct ListNode {
   int val;
   struct ListNode *next;
};

int length(struct ListNode *head) {
   int len = 0;
   while (head != NULL) {
      len++;
      head = head->next;
   }
   return len;
}

int findMergeNode(struct ListNode *headA, struct ListNode *headB) {
   int lenA = length(headA);
   int lenB = length(headB);

   // Move the pointer of the longer list by the difference in lengths
   while (lenA > lenB) {
      headA = headA->next;
      lenA--;
   }
   while (lenB > lenA) {
      headB = headB->next;
      lenB--;
   }
```

```
    // Traverse both lists until we find a common node
    while (headA != headB) {
        headA = headA->next;
        headB = headB->next;
    }

    // Return the value of the common node
    return headA->val;
}

int main() {
    // Create the first linked list: 1 -> 2 -> 3 -> 4 -> 5
    struct ListNode *headA = (struct ListNode *)malloc(sizeof(struct ListNode));
    headA->val = 1;
    headA->next = (struct ListNode *)malloc(sizeof(struct ListNode));
    headA->next->val = 2;
    headA->next->next = (struct ListNode *)malloc(sizeof(struct ListNode));
    headA->next->next->val = 3;
    headA->next->next->next = (struct ListNode *)malloc(sizeof(struct ListNode));
    headA->next->next->next->val = 4;
    headA->next->next->next->next = (struct ListNode *)malloc(sizeof(struct ListNode));
    headA->next->next->next->next->val = 5;
    headA->next->next->next->next->next = NULL;

    // Create the second linked list: 6 -> 7 -> 4 -> 5
    struct ListNode *headB = (struct ListNode *)malloc(sizeof(struct ListNode));
    headB->val = 6;
    headB->next = (struct ListNode *)malloc(sizeof(struct ListNode));
    headB->next->val = 7;
    headB->next->next = headA->next->next->next; // Merge point
    printf("Merge node value: %d\n", findMergeNode(headA, headB));

    // Free memory
    free(headA->next->next->next->next);
    free(headA->next->next->next);
    free(headA->next->next);
    free(headA->next);
    free(headA);

    free(headB->next);
    free(headB);

    return 0;
}
```



```
Merge node value: 4


...Program finished with exit code 0
Press ENTER to exit console.
```