3. Linked Lists and Recursion (8 pts.)

Given the code in the image below answer the following questions regarding a call to **`llmystery(head, 2)`** assuming `head` points to the linked list of values:

1 2 3 4 5

Answer the following questions:

3.1. **True/False:** Will the original linked list be modified (values or pointers)? __**NO**_____

3.2. On the next page, show what will be printed by the code, and further show a call tree (box diagram) of the recursive calls (with relevant arguments) made during execution.

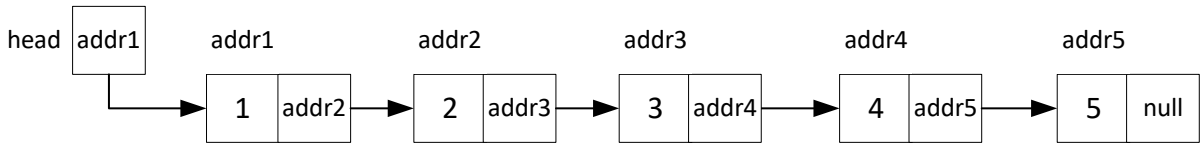
```
struct IntItem {
  int val;
  IntItem *next;
  IntItem(int v, IntItem* n) { val = v; next = n; }
};

void llmystery(IntItem* head, size_t loc);
int llmysteryHelper(IntItem* head, size_t loc, IntQueue& v);

void llmystery(IntItem* head, size_t loc)
{
  IntQueue q;
  int t = llmysteryHelper(head, loc, q);
  cout << t << endl; // *** COUT PRINT HERE ***
  while(!q.empty()) {
    cout << q.front() << " ";  // *** COUT PRINT HERE ***
    q.pop_front();
  }
  cout << endl;
}

int llmysteryHelper(IntItem* head, size_t loc, IntQueue& q)
{
  if(head == nullptr) {  return 0; }
  else if(loc > 0) {
    int t = llmysteryHelper(head->next, loc-1, q);
    q.push_back(head->val);
    return t;
  }
  else {
    q.push_back(head->val);
    return head->val + llmysteryHelper(head->next, loc, q) ;
  }
}
```

Show what will be printed by the call to `llmystery(head, 2)`

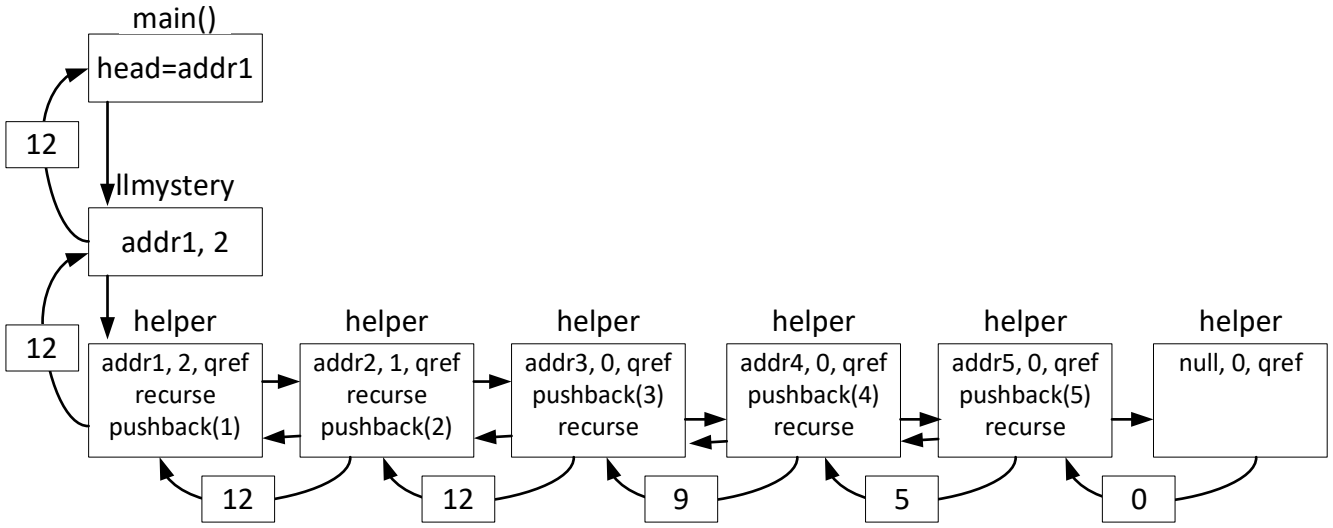
12
3 4 5 2 1

Upload a diagram of the call tree (box diagram) of the recursive calls made during execution.

Linked List



Call tree



Prints: 3 4 5 1 2