Problems for the Final Review

1. Write a function, which restricts linked lists to their first \( n \) nodes, i.e., it places NULL in the address field of the \( n \)-th node and deletes all further nodes.

2. Write a function, which concatenates two given linked lists.

3. Write a function, which out of a given circular list makes a single linked list with NULL in its last node.

4. What is the largest possible depth of the Huffman tree of a list consisting of 16 characters? What is the smallest possible depth in this case?

5. Write a function, which for a given binary tree computes the average depth of leaves.

6. Provide an example of an input, so that the tree, which is the result of running the insert function provided below, is the full binary tree of depth 3. Assume that keys are integers. In your solution specify input keys only.

```c
TNode* insert(TNode* T_ptr, TNode* new_ptr)
{
    if(T_ptr==NULL)
        return new_ptr;
    if(new_ptr->key < T_ptr->key)
        T_ptr->left_link=insert(T_ptr->left_link,new_ptr);
    else
        T_ptr->right_link=insert(T_ptr->right_link,new_ptr);
    return T_ptr;
}
```

7. Problems related to Merge Sort

   (i) Draw the tree of recursive calls for Merge Sort algorithm acting on an array consisting of 7 elements.

   (ii) How many times function merge is called during execution?

   (iii) Trace modifications done to the array \([7,1,6,2,5,3,4]\) after merging on every level of the tree of recursive calls (this is not the exact execution).

   (iv) Modify the code of function mergesort, so that the number of times function merge is called is computed automatically.
8. Problems related to Heap Sort

(i) Trace the execution of Heap Sort algorithm applied to the array [7,1,6,2,5,3,4].

(ii) What is the largest number of swap operations, which may occur during execution of make_heap(a,7)?