

National Institute of Technology, Calicut
Department of Computer Science and Engineering
CS2094 – Data Structures Lab

Assignment 3 (Advanced)

Submission deadline (on or before): **03.03.2015 (Tuesday) 5pm**

Naming Conventions for submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG3A<Number>_<ROLLNO>_<FIRSTNAME>.zip (For example: ASSG3A_BxyyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

The source codes must be named as ASSG3A<Number>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG3A_BxyyyyyCS_LAXMAN_1.c).

Questions

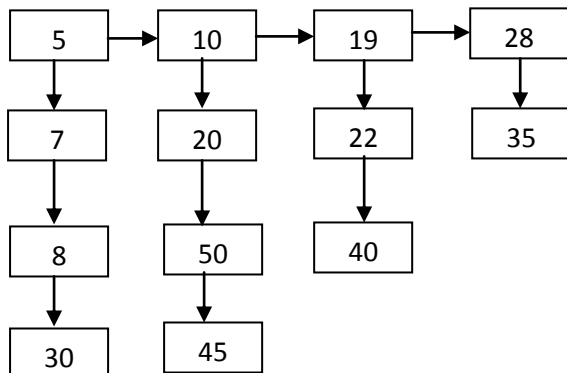
1. Flattening a Linked List

You are given a pointer p to node, having a key field, and two pointers to similar nodes. The node p is the first node in the “main” list. The two pointers associated with each node are,

- a. Next pointer to the next node in the linked list.
- b. Down pointer which points to another linked list.

The node p starts up a “main” list, in which each node is connected to the next one through the next pointers. Each node in this main list also starts up through its “down” pointers, a “down” list of nodes, each connected to the next through down pointers. For the nodes in the down list (other than the first node) the next pointers are necessarily *nil*. Each node belongs to only one down list.

All linked lists are sorted. See the following example.



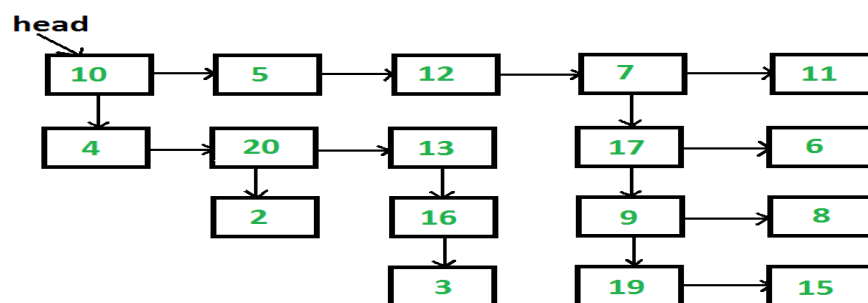
Write a function `flatten()` to flatten the lists into a singly linked list. The flattened linked list should also be sorted. Example: for the above input list, output list should be,

5→7→8→10→19→20→22→28→30→35→40→45→50

2. Flatten a multilevel linked list

Given a linked list where in addition to the next pointer, each node has a child pointer, which may or may not point to a separate list. These child lists may have one or more children of their own, and so on, to produce a multilevel data structure, as shown in below figure. You are given the head of the first level of the list. Flatten the list so that all the nodes appear in a single-level linked list. You need to flatten the list in way that all nodes at first level should come first, then nodes of second level, and so on. Each node is a C struct with the following definition.

```
struct list
{
    int data;
    struct list *next;
    struct list *child;
}
```



The above list should be converted to

10 → 5 → 12 → 7 → 11 → 4 → 20 → 13 → 17 → 6 → 2 → 16 → 9 → 8 → 3 → 19 → 15