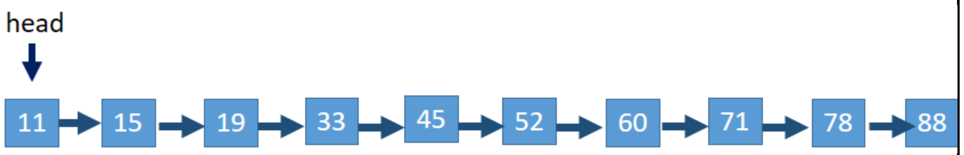
Online Assignment 1: List (27/11/2021 Saturday)

You have to re-implement the LL based implementation of the List ADT as per the specification of your offline assignment. This time the linked list will have to be implemented by using two arrays, Value and Link. Value will keep the elements and Link will keep the index mimicking the next pointers of a linked list node. Your implementation must only use the Arr based list implementations (of your offline assignment) to handle the two arrays, Value and Link. All the tasks in your offline assignment must be redone using this re-implementation.

An example is shown in Figure 1. Here the linked list is drawn conceptually in Figure 1(a) and the corresponding two arrays (Value and Link) are shown in Figure 1(b). NULL pointer is indicated by an index of -1 and the Head of the list is identified as ‘h’ in Figure 1(b).



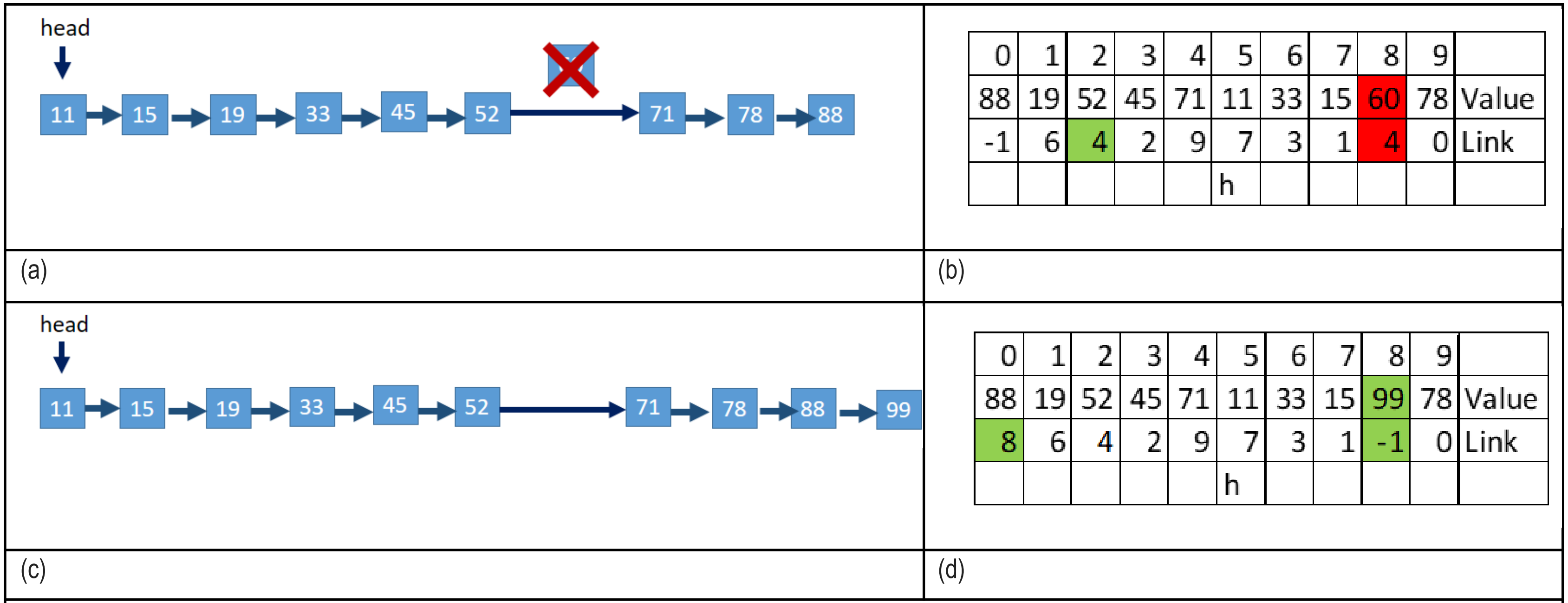
0	1	2	3	4	5	6	7	8	9	
88	19	52	45	71	11	33	15	60	78	Value
-1	6	8	2	9	7	3	1	4	0	Link
					h					

(a)

(b)

Figure 1: A conceptual Linked List and its implementation using 2 arrays, Value and Link. Value stores the elements (in this case integers) and Link keeps track of the next pointer by recording the index of the next element.

Figure 2 shows two consecutive operations on the list in Figure 1. First we Delete 60 (Figure 2) and then Append 99 (Figure 3).



(a)

0	1	2	3	4	5	6	7	8	9	
88	19	52	45	71	11	33	15	60	78	Value
-1	6	4	2	9	7	3	1	4	0	Link
					h					

(b)

(c)

0	1	2	3	4	5	6	7	8	9	
88	19	52	45	71	11	33	15	99	78	Value
8	6	4	2	9	7	3	1	-1	0	Link
					h					

(d)

Figure 2: Deleting 60 from the linked list is conceptually shown in (a). The red marked cells in (b) indicates that these are deleted are not being used anymore. So, the next insert/append may be done here. The green cell has been modified (from 8 to 4) as 52 now points to 71. Appending 99 in the resulting linked list (i.e., after the above-mentioned deletion) is conceptually shown in (c). The green cells in (d) are the modification needed in the two arrays.

Submission Guidelines:

1. **Create a directory with your 7-digit student id as its name**
2. **Put all the source files only into the directory created in step 1. Also create a readme.txt file briefly explaining the main purpose of the source files.**
3. **Zip the directory (compress in .zip format. Any other format like .rar, .7z etc. is not acceptable)**

4. **Upload the .zip file on Moodle in the designated assignment submission link. For example, if your student id is 1905xxx, create a directory named 1905xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 1905xxx. Compress the directory 1905xxx into 1905xxx.zip and upload the 1905xxx.zip on Moodle.**

Failure to follow the above-mentioned submission guideline may result in upto 10% penalty.