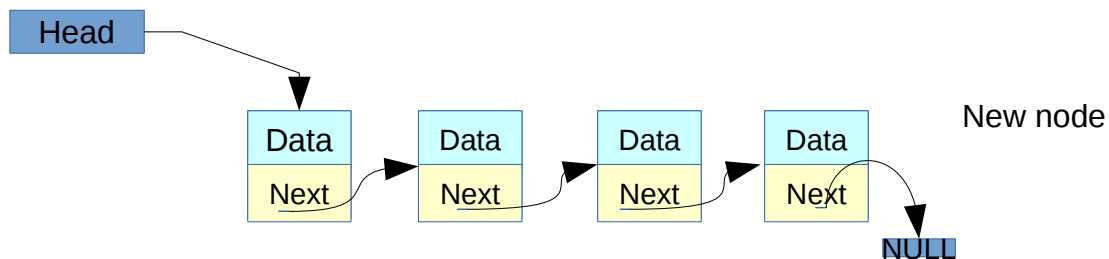


Aim: The aim of this laboratory is to work with pointers. It extends the previous laboratory class.

Description: In the previous implementation, the new node was added to the head and elements were removed from the head. As such, the last element to be inserted will be the first element to be removed. Such a data structure is called LIFO (Last-In First-Out) or a stack.

Today, you are supposed to insert the new items to the tail of the existing list and remove items from the head. So, the first item to be removed will be the first time to be inserted into the linked list. This behavior is called FIFO (First-in First-Out) or a queue. The diagram below explains.



We will be using a similar code base to the previous laboratory class. You are supposed to implement the `int tail_insert(void *data, list_t *list)` function. Test it with the main function. If the implementation is correct the names of the cities will be printed in the order in which they were entered. You are required to implement this part within the laboratory class itself. Show your code to an instructor before leaving the laboratory class.

Next, you are supposed to implement `int insert_sorted(void *data, list_t *list)` where the given item will be added to the correct position in a already sorted list. So if you use this function the names of cities will be printed in their alphabetical order. You may use the `strcmp` function useful for this implementation.

The final implementation is `void delete_list(list_t *list)` which should delete all the elements in the linked list and the pointers within the node.

Your implementation of `insert_sorted` and `delete_list` will be marked. However submit a single tarball containing all the files which are in the given skeleton code.

Questions: Answer the following questions. They do not carry any marks but will be useful in future (not just for this laboratory class);

1. Suppose you have lots of nodes in the list, what do you think about the time to insert a new element?
2. Can you have the same FIFO behavior by adding new elements to the head? (hint: rethink removal)
3. Is it a good idea to have a function like `inset_sorted` when the data type is `void *`? Explain.

Submission: Submit your answer via Moodle. You should submit a tarball with the `linked.c`, `main.c` Makefile and `linked.h` Deadline is 23rd October 2019.