



SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CMPT 125 - Introduction to Computing Science and Programming II

Linked-Lists

Linked List - Recap

- Chain of separate elements
- Head points to the first element
- Tail points to the last element
- Each element has a data part which contains value and a pointer which points to the next node in the list

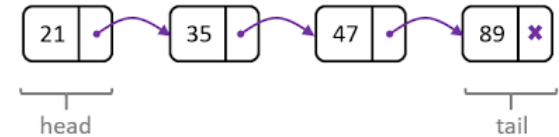


Fig1: Linked List
Source: [101computing](https://www.101computing.com/linked-list/)

Linked List - Operations

- **LL_add_to_head(LL_t *list, int value):** Add element to the head of the list
- **LL_add_to_tail(LL_t *list, int value):** Add element to the tail of the list
- **LL_remove_from_head(LL_t *list):** Remove element from the head of the list
- **LL_size(const LL_t *list):** Return the size of the list
- **LL_print(const LL_t *list):** Prints all elements of the list from head to tail
- **LLnode_free(node_t *node):** Frees memory used by the node
- **LL_free(LL_t *list):** Frees memory used by the list

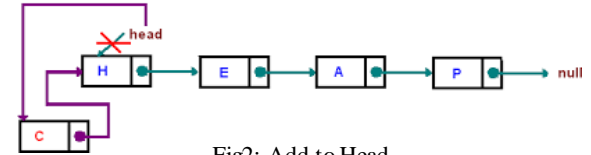


Fig2: Add to Head

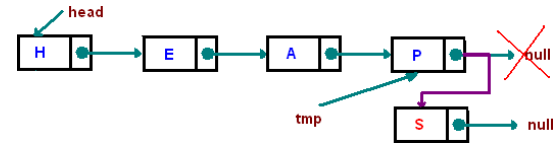


Fig3: Add to Tail

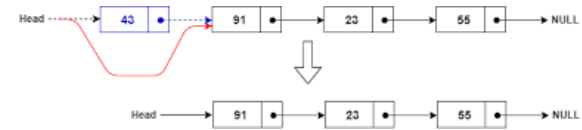


Fig4: Remove from Head

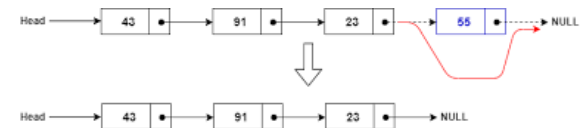


Fig5: Remove from Tail

Steps to compile code

- Unzip and open the directory in VSCode

In the terminal, run:

- `> cd LL`
- `> make`
- `> ./driver_LL`

Exercise

- Read and understand the functions defined in LL.c
- Implement the functions:
 - **LL_remove_from_tail()**: removes element from tail of the list
 - **LL_print_reverse()**: prints list elements in reverse order. Try doing it in $O(N)$ time.
 - **to_array()**: gets a linked list and creates an array with same values
 - **array_to_list()**: gets an array and creates a linked list
 - **are_equal()**: check if two linked lists are equal (equal length and same values in order)
- Add more test cases to test the functions you implement

Exercise

- Read and understand the functions defined in queue.c.
This implementation uses linked lists and not arrays
- Implement the following functions:
 - **size(queue* q):** returns the size of the queue
 - **are_equal(queue* q1, queue* q2):** checks if two queues are equal
 - **to_array():** gets a queue of ints and creates an array with same values
- Write your own test cases to test the functions you implement