302-A02 Using linky to implement a stack ADT

Now that we have an integer linked list implementation (c.f. 302-A01), we can start re-using this abstract data type (ADT) to design and implement other abstract data types. What would it take to develop an integer stack data structure now that we have a linked list one? We could re-implement everything by hand, or try to leverage the linked list ADT we already have so that everytime it is updated with features or bug fixes, we automatically benefit from these efforts in our stack ADT.

Start by copying all the files developed in 302-A01 and create the following additional files;

- **stacky.h** will contain the headers (aka declarations) of our functions.
- **stacky.c** will contain the definitions of some of these functions
- **tests_stacky.c** will contain the main function which tests all of the above

Re-using integer linked list ADT for an integer stack ADT

Let us implement our integer stack ADT as a linked list of elements. Creating a new node of information, de-allocating it, displaying all the information nodes stored in the stack, are operations which are identical to those we already implemented in the linked list.

Therefore, we can use the same data structure (TCell) to store the elements of our stack. This also means that the following functions, listed in linky.h, can be re-used in our integer stack ADT.

Adding new functions

New operations must be available though; e.g. we must be able to **push** elements on the stack. However, pushing a new integer on the stack is simply adding a new TCell containing this integer at the front of the linked list. Similarly, in order to **pop** an element out of the list, one simply needs to access the last element and return its value after removing it from the linked list.

void TCell_push (TCell list, int value);**

- create a new TCell to store the integer *value*
- add it in front of the linked list **list*.

int TCell_pop (TCell list);**

- locate the first element of the linked list and store its int value in a temporary variable *value*
- Deallocate the TCell structure corresponding to the first element of the linked list and update the linking in the linked list accordingly
- Return the value *value*

Adding new tests

Use the **tests_stacky.c** main function to design and implement tests which are going to allow us to easily and visually validate the functions you implemented.