

Tutorial on Lists - Master 1 Bio-informatics

Exercise 1

Write a python program that creates and uses a linked list.
You will implement the following functions :

```
def push_front(l, e):
    # Adds an element at the beginning of the list
def first( l ):
    # returns the first cell of the list
def end( l ):
    # returns the last cell of the list
def next( l, cell ):
    # returns the cell that follow the cell 'cell'.
def value( cell ):
    # returns the value stored inside a cell.
```

Exercise 2

With that implementation, write the following functions :

```
def push_back(l, e):
    # Adds an element at the end of the list
def insert( l, cell, e ):
    # We suppose that 'cell' is a cell of the list 'l'. This function
    # create a new cell containing the value 'e' and then inserts this new cell
    # just after the cell 'cell'.
def copy( l ):
    # returns a copy of the linked list l
def middle(l):
    # returns the element in the middle of the linked list 'l'
def size(l):
    # returns the number of elements in the liked list 'l'.
def move( l1, l2 ):
    # Move all the cells of l2 inside l1 (at the end). The list l2 should be
    # empty .
def search(l,e):
    # returns the first cell that contains 'e' as value.
```

Exercise 3

Repeat the previous two exercises using doubly linked lists.

Extra work (hard) : you can implement the "dancing links" algorithm of Knuth :
<https://arxiv.org/abs/cs/0011047>.