TD de programmation fonctionnelle et logique
TD 2 : récursivité

Suite et fin de la factorielle récursive et impérative

```
# let rec fact_imp n =
   let a =ref(0)
   in match n with
        0 -> a := 1
      | _ -> let b = ref(fact_imp (n-1))
             in a := n;
   print_int !a;;
             val fact_imp : int -> unit = <fun>
```

Expressions à évaluer

```
# let x = 1;;

# let y = 4;;

# let x = 3 and y = x + 1 in
   let x = x + y and y = x - y
     in (x,y);;

# let x = 3
   in let  y = x + 1
     in let x = x + y
       in let y = x - y
         in (x,y);;
```

Fonctions récursives simples sur les tableaux

1. Écrivez une fonction qui teste si deux tableaux sont égaux.
2. Écrivez une fonction qui calcule l'élément maximum d'un tableau d'entiers.

Palindromes

Un palindrome est un mot qui se lit bien à l'endroit qu'à l'envers (cf. figure 1).

Rappels : l'expression `String.length s` renvoie la longueur de la chaîne de caractères `s` ; l'expression `String.sub s a b` construit une chaîne de `b` caractères égale à la sous-chaîne de `s` débutant par son a^ecaractère.

– Écrire une fonction à valeurs booléennes qui est vrai si le mot qui lui est passé en argument est un palindrome.

– Même question mais en ne construisant pas de sous-chaînes de caractères.

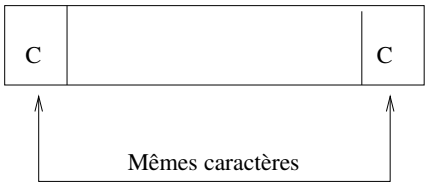

FIG. 1: Structure d'un palindrome

Calcul de (cos(nx), sin(nx))

Écrire une fonction qui prend en entrée un entier `n` et une paire de valeurs réelles qui sont en fait les valeurs du cosinus et du sinus d'un certain angle `x`, et qui renvoie la paire `(cos(nx), sin(nx))`. Autrement dit, le deuxième argument de la fonction est une paire `(a,b)` telle que `a = cos x` et `b = sin x` . Le schéma de calcul doit bien évidemment être récursif.

On pourra se servir des formules de trigonométrie suivantes :

```
cos(nx) = cos((n-1)x) cos(x) - sin((n-1)x) sin(x)
sin(nx) = sin((n-1)x) cos(x) - cos((n-1)x) sin(x)
```