

# ULCO - L2 Mathématiques

## Examen Algorithmique - 2h

5 janvier 2017

### Exercice 1 : Listes chaînées

On considèrera une liste chaînée de **joueur**, où **joueur** est une structure qui contient :

- un tableau de 50 caractères "nom", composé uniquement de lettres minuscules entre 'a' et 'z' où 'a' < 'b' < ... < 'z' ;
- un entier "nb\_buts" ;
- un entier "nb\_matches" ;
- un réel "salaire" ;
- un booléen "blesse" qui sera Vrai si le joueur est blessé et Faux sinon ;
- un pointeur nommé "**suis**vant" sur le **joueur** suivant s'il existe, sinon "**suis**vant" vaudra "Null".

Dans cet exercice,

1. vous écrirez la structure "joueur" décrite ci-dessus ;
2. vous écrirez **une fonction "infoBlesses" retournant** la somme des salaires des joueurs blessés, ainsi que le salaire moyen de ces mêmes joueurs ;
3. vous écrirez une fonction "insérerDernier" qui prend en paramètres un pointeur "Tete" pointant sur le début d'une liste, et un **joueur** nommé "nouveau" (qu'on considèrera initialisé), et qui insère **le joueur** "nouveau" en dernière position ;
4. vous écrirez une fonction "tableauxTries" qui prend en paramètres 2 tableaux T1 et T2 de 50 caractères, et qui renvoie Vrai si T1 contient un "nom" qui est avant celui contenu dans T2 **pour** l'ordre alphabétique, et Faux sinon ;
5. on supposera maintenant que la liste est triée dans l'ordre alphabétique des noms des joueurs (champ "nom" de la structure "joueur"). Vous écrirez une fonction "insérer" qui prend en paramètres un pointeur "Tete" pointant sur le début d'une liste, et un **joueur** nommé "nouveau" (qu'on considèrera initialisé), et qui insère **le joueur** "nouveau" à sa place dans la liste (donc en fonction de son "nom"). Vous pourrez utiliser la fonction "tableauxTries" même si vous n'avez pas réussi à l'écrire.

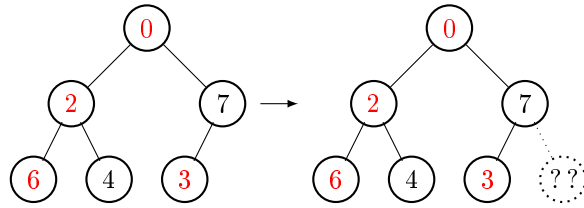
### Exercice 2 : Arbres binaires

Nous allons considérer un arbre binaire dont les noeuds contiennent des entiers. La racine de l'arbre contient la valeur 0, son sous-arbre gauche est composé des noeuds contenant des entiers pairs, et son sous-arbre droit des noeuds contenant des entiers impairs.

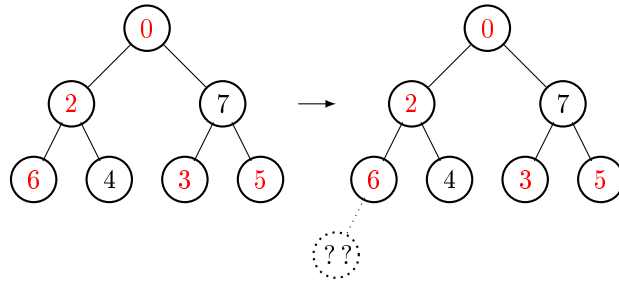
Dans cet exercice,

1. vous écrirez la structure d'un noeud comme décrit ci-dessus ;
2. vous écrirez une fonction "afficher" qui prend en paramètres un pointeur "Racine" pointant sur la racine d'un arbre binaire, et qui permet d'afficher dans un ordre que vous définirez toutes les valeurs de cet arbre binaire ;
3. vous écrirez une fonction "insérerSousArbre" qui prend en paramètres un pointeur "Racine" pointant sur la racine d'un arbre binaire, et un élément nommé "nouveau", et qui complète l'arbre binaire avec l'entier "nouveau" en tant que feuille d'un noeud qui n'a pas 2 fils (en complétant en priorité un noeud qui a un fils par rapport à un noeud qui n'en a pas). Voir les exemples ci-après.

Exemple 1 :



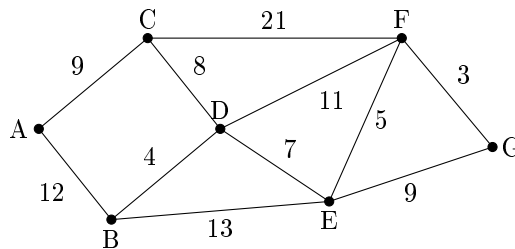
Exemple 2 :



- vous écririez une fonction "insérer" qui prend en paramètres un pointeur "Racine" pointant sur la racine d'un arbre binaire, et un élément nommé "nouveau" (qu'on considèrera initialisé, donc avec une valeur), et qui insère l'élément "nouveau" dans l'arbre binaire au bout d'une feuille du sous-arbre auquel il appartient (gauche si pair, droit si impair).

### Exercice 3 : Graphes

On considère le graphe  $G$  suivant :



- Le graphe  $G$  possède-t-il une chaîne eulérienne ? Si oui en donnez une, si non justifier.
- A l'aide de l'algorithme de Dijkstra, déterminez le plus court chemin allant de  $A$  à  $G$ . Vous donnerez les différentes étapes de l'algorithme.