

FINAL – LO42

**Les documents ne sont pas autorisés (La copie ou les idées du voisin non plus).
Toute réponse devra être claire et justifiée si nécessaire (toute ambiguïté sera mal interprétée).
Sauf indication contraire, dans le cas d’algorithmes les réponses doivent être rédigées en pseudo code.**

1) J’en suis tout retourné ! (5)

Soit G un graphe orienté. On définit le graphe inverse G' en inversant le sens de tous les arcs de G . La file des successeurs (respectivement le tableau des adresses des premiers successeurs) de G' constitue la file des prédécesseurs (respectivement le tableau des adresses des premiers prédécesseurs) de G et est notée FP (respectivement APP).

Le graphe G étant donné par FS et APS, construisez FP et APP moyennant deux lectures de FS seulement.

Méthode :

Par une première lecture de FS, on construit le tableau NP contenant le nombre de prédécesseurs de chaque sommet. On en déduit APP.

Par une deuxième lecture de FS, on construit finalement FP, en utilisant APP comme indicateur de pointage dans FP.

2) Le devoir de mémoire ! (2)

Voici une fonction récursive.

```

Fonction f (i : entier ; var a : réel ; var n : entier) : réel ;
Var x : réel ;
    b : booléen ;
Début
    lire (x, b) ;
    Si b Alors
        Si i > 0 Alors
            a ← a + x ;
            f ← f ( i+ 1, a, n) + carré (a – x) ;
        Sinon
            a ← x ;
            f ← racine_carré ( ( f ( i+ 1, a, n) + carré (a – x) ) /n)
        Fsi
    Sinon
        n ← i
        Si i > 0 Alors a ← a / i
        Fsi
        f ← 0 ;
    Fsi
Fin

```

On vous demande quelle est la valeur de r après l’instruction “ $r \leftarrow f(0, a, n)$;”

Vous devrez expliquer comment se font les différents calculs au cours des appels récursifs.

3) La chirurgie! (5)

Vous disposer d’un arbre binaire de recherche. La suppression d’un nœud peut relever d’un des quatre cas suivants :

- Le nœud est une feuille,
- Le nœud n’a qu’un fils gauche ou droit,
- Le fils droit n’a pas de fils gauche ou réciproquement.
- Le cas général où le nœud le plus à gauche du sous arbre droit peut remplacer le nœud supprimé (en l’ayant sorti au préalable du sous arbre droit).

Notez que ces cas ne sont pas disjoints !

Après avoir donné le type de données adapté à l’implantation de l’arbre binaire de recherche, vous écrirez une procédure permettant de supprimer un nœud dans l’arbre. La procédure recevra en paramètres l’arbre et la valeur à

supprimer dans cet arbre. On suppose que les valeurs stockées dans les nœuds sont de type compatible avec les opérateurs de comparaison classiques.

4) Entre la gauche et la droite le juste équilibre est délicat ! (5)

Vous connaissez tous les arbres H_équilibrés puisqu'ils ont été étudiés en cours.

- Rappelez la structure de données permettant d'implanter l'arbre H_équilibré.
- Expliquez ce qu'est une réorganisation par une rotation gauche-gauche et une réorganisation par une rotation gauche-droite.
- Donnez l'algorithme de réorganisation gauche-droite.

5) Palindrome (10)

Un palindrome est une phrase ou un mot entièrement symétrique. Par exemple, le mot LAVAL ou la phrase " Esope reste ici et se repose " sont des palindromes.

Le but de cet exercice est de définir un programme qui vérifie si une phrase saisie au clavier et **de longueur quelconque** est un palindrome. Par simplification, nous limiterons cette notion aux phrases symétriques au niveau des mots mais pas forcément des caractères (exemple : " Tous pour un et un pour tous "). Vous supposerez disponibles les sous programmes suivants :

*Fonction Lire_Mot() :chaîne de caractères
/*Opération qui permet la saisie d'un mot au clavier et le retourne sous forme de chaîne de caractères ;
retourne une chaîne vide (de longueur nulle) si aucun mot n'a été saisi*/*

*procédure Affiche_Phrase(phrase p)
/*Opération qui permet l'affichage sur écran de la phrase transmise en paramètre. */*

*Fonction Détruire_Phrase(phrase p)
/*Opération qui permet de détruire la phrase en libérant la place occupée. */*

Un type *chaîne* a été redéfini en C et le fichier *include string.h* fournit les prototypes de tous les outils, disponibles dans une bibliothèque, pour manipuler ces chaînes de caractères, notamment :

*Fonction chcmp(chaîne chn2,chaîne chn1) : int
/*retourne un entier <0 si chn2<chn1, = 0 si chn2=chn1, >0 si chn2>chn1*/*

*Fonction hcpy(chaîne chn2,chaîne chn1) : int
/*copie la chaîne chn1 dans la chaîne chn2*/*

*Fonction chlen (chaîne chn) : int
/*retourne la longueur de la chaîne de caractères*/*

- Donnez les déclarations des types des variables dynamiques nécessaires à l'implantation de ce programme
- Proposez une méthode de vérification de la caractéristique " palindrome " et donnez la spécification de toutes les opérations utiles à son implantation
- Ecrire en langage C les sous-programmes correspondants aux spécifications proposées en 2.