

Examen LO21 A2024
Durée : 1h30
Documents non autorisés

Exercice 1 (3+5) +-1

On souhaite gérer les candidatures pour la poursuite des études au sein d'un établissement d'enseignement supérieur. Pour cela, on dispose de deux listes :

-la liste des candidats : chaque candidat de cette liste est caractérisé par son nom, sa moyenne générale et le code de la spécialité (un entier) qu'il a choisie. La liste des candidats est triée par ordre croissant sur le code des spécialités choisies par les candidats. On suppose que les candidats à la même spécialité sont triés par ordre décroissant sur leurs moyennes.

-la liste des spécialités offertes par l'établissement : chaque spécialité est caractérisée par son code (un entier), le nombre de places disponibles et la moyenne minimale que doit posséder tout candidat pouvant être accepté dans cette spécialité. La liste des spécialités est triée par ordre croissant sur le code des spécialités proposées par l'établissement.

1) Etant donné un entier et une liste de candidats, écrire de manière récursive l'algorithme du sous-programme permettant de construire la liste des candidats à la spécialité ayant comme code cet entier. Chaque élément de cette nouvelle liste est caractérisé par le nom et la moyenne générale du candidat.

2) Etant données une liste de candidats et une liste de spécialités, écrire l'algorithme itératif qui construit la liste des noms des candidats acceptés par l'établissement. Le nombre de candidats acceptés dépend du nombre de places et seuls les meilleurs candidats sont acceptés en priorité (ayant les meilleures moyennes).

Exercice 2 (3+5) +-1

On considère des arbres binaires de recherche d'entiers. Dans ces arbres, On suppose que deux nœuds différents possèdent nécessairement deux valeurs différentes.

1) Etant donné un arbre binaire d'entiers et un entier, écrire l'algorithme récursif optimal du sous-programme **Appartient** qui teste si cet entier est la valeur d'un des nœuds de l'arbre.

2) Etant donné un arbre binaire de recherche d'entiers et un entier, écrire l'algorithme récursif du sous-programme **Chemin** qui construit la liste des valeurs des nœuds constituant le chemin depuis la racine jusqu'au nœud contenant cette valeur. Par exemple, pour l'arbre suivant :

a = 5 Chemin(a,7) produit la liste (5, 8, 7)
 / \ Chemin(a,8) produit la liste (5,8)
 3 8
 / \
 6 7

Exercice 3 (2+2) +-1

Soit le langage régulier L construit sur le vocabulaire terminal $V=\{a,b,c\}$ et contenant tous les mots de la forme suivante : a^*b^*c (mots composés de zéro ou plusieurs **a**, suivi de zéro ou plusieurs **b**, suivi d'un **c**).

1) Donner l'automate permettant de reconnaître les mots de L (automate associé à L).

2) Donner la grammaire G associée au langage L.

Note : 1) Pour chaque sous-programme, vous devez préciser clairement son entête (son nom, ses paramètres avec leurs types et éventuellement le type du résultat retourné dans le cas où le sous-programme est une fonction) et le nommer en utilisant le nom donné dans l'énoncé des exercices. 2) Vous pouvez utiliser les opérations élémentaires (données en cours) sur les listes et les arbres sans donner leurs algorithmes, mais en rappelant leurs entêtes.