

esprit Ecole Supérieure Privée d'Ingénierie et de Technologies	Objectif (8): Les arbres binaires de recherche	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire: 2018-2019

Une école d'ingénieur nouvellement créée souhaite informatiser son système de scolarité. Dans cette école, il existe cinq niveaux de scolarité (1ère année → 5ème année) et dans chaque niveau il existe une seule classe. Les informations des étudiants sont initialement enregistrées dans un fichier texte. Chaque étudiant est caractérisé par :

- Id (entier)
- Nom (chaîne de caractère)
- Prénom (chaîne de caractère)
- Niveau (entier de 1 à 5)
- Moyenne (réel)

Pour faciliter la recherche d'un étudiant, on se propose de mettre les données dans un arbre binaire de recherche.

1. Déclarez les structures nécessaires.
2. Développez la fonction **abr initialiser(etudiant e)** qui permet de créer et retourner l'arbre dont la racine contient l'information étudiant
3. Développez la fonction **abr inserer (abr a,etudiant e)** qui permet d'insérer un étudiant en feuille de l'arbre a selon les moyennes des étudiants ;
4. Développez la fonction **abr construireArbre (abr a,char nomFich[])** qui permet de créer un arbre binaire de recherche à partir du fichier texte fourni. Cette fonction prend comme paramètres l'arbre à construire, le nom du fichier texte et retourne l'arbre créé.
5. Ecrire une fonction qui permet d'afficher les étudiants selon l'ordre décroissant de leur moyenne.

void afficher(abr a)

6. Ecrire une fonction qui calcule le nombre total des étudiants

int calculerNombre(abr a)

Nous souhaitons organiser les étudiants selon leurs moyennes dans des listes doublement chaînées comme suit :

- Les étudiants ayant une moyenne supérieure ou égale à 16 seront insérés dans la liste **TresBien**,
- Les étudiants ayant une moyenne supérieure ou égale à 14 et inférieur à 16 seront insérés dans la liste **Bien**,
- Les étudiants ayant une moyenne supérieure ou égale à 12 et inférieur à 14 seront insérés dans la liste **AssezBien**,
- Les étudiants ayant une moyenne supérieure ou égale à 10 et inférieur à 12 seront insérés dans la liste **Passable**,

7. Développez la fonction **réursive construireListe** permettant de créer **une liste doublement chaînée** à partir de l'arbre binaire selon la mention. Cette fonction prend

esprit Ecole Supérieure Privée d'Ingénierie et de Technologies	Objectif (8): Les arbres binaires de recherche	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire: 2018-2019

comme paramètres l'arbre binaire de recherche contenant tous les étudiants, l'adresse de la liste à retourner et l'intervalle [BI ..BS[reflétant la mention.

void construireListe(abr a,LDC *l, float BI, float BS);

NB: BI c'est la borne inférieure et BS c'est la borne supérieure de l'intervalle.

<i>Mention</i>	<i>[BI ..BS[</i>
TresBien	[16 ..20[
Bien	[14 ..16[
AssezBien	[12 ..14[
Passable	[10 ..12[