

## CONCOURS d'accès à l'ESI

**Epreuve : Algorithmique**

**Epreuve : Algorithmique et programmation**

**Code : ALGPRO**

**Date : Septembre 2012**

**Durée : 3 heures**

**Documents interdits**

### CONSIGNES :

- Chaque partie est à remettre sur une copie différente.
- N'oubliez pas de mettre en évidence sur votre copie le N° de la question traitée (**Question 1** : .....)
- Soignez votre travail car il en sera tenu compte de la présentation

### Partie 1 : Structures statiques et POO

Une entreprise de diffusion de films alimente des cinémas en films qu'ils souhaitent projeter. Elle dispose aussi de trois dépôts et TOUS les films sont stockés dans chacun d'eux. De plus cette entreprise dispose de la distance (en Km) qui sépare un cinéma de chacun des trois dépôts.

Le client (propriétaire d'une salle de cinéma) peut commander les films par téléphone ou Email et ensuite les films lui seront livrés à domicile. Les films sont livrés du dépôt vers le client un à un (un client à la fois) et les clients seront livrés une fois que toutes les commandes sont faites.

Nous disposons donc du tableau COM qui contient les commandes faites par les clients, dans lequel un 1 indique que le film est commandé et 0 dans le cas contraire. Vous constatez qu'un cinéma peut commander tous les films et un cinéma peut ne commander aucun film.

		cinémas					
		1	2	3	4	5	6
films	1	0	0	0	0	1	0
	2	0	0	1	0	1	0
	3	1	0	0	0	1	0
	4	1	0	1	0	1	1

Et du tableau DIST qui contient la distance (en Km.) entre un cinéma et les trois dépôts.

		cinémas					
		1	2	3	4	5	6
dépôts	1	12	1	6	4	7	2
	2	4	5	7	3	8	10
	3	5	2	1	2	5	4

De plus un découpage du problème a été effectué et il est le suivant :

Procédure LECT2D (var T : tab2; var L,C: entier) (\* lit un tableau à 2 dimensions d'entiers de L lignes et C colonnes\*)

Fonction ColVal (T : tab2; L , C ,Val : entier): booléen (\*donne Vrai si tous les éléments de la colonne C du tableau T sont tous égaux à Val. L étant le nombre de lignes\*)

Fonction PetitCol (T : tab2; L , Col: entier): entier (\* donne le plus petit élément de la colonne Col du tableau T.

L étant le nombre de lignes \*)

Type Tab2 = tableau [1..100, 1..100] d'entiers

**Question 1 :** Nous souhaitons connaître la plus petite distance à parcourir pour alimenter l'ensemble des cinémas qui ont passé une commande. Donnez l'analyse de l'algorithme principal en utilisant le découpage proposé, ci dessus. (par exemple , dans le jeu d'essai ci dessus cette distance est : 12 km. )

**Question 2 :** Donnez l'algorithme correspondant à cette analyse

**Question 3 :** Ecrivez le programme Pascal de cet algorithme.

Cette entreprise souhaite créer un site permettant d'afficher la liste des séances de films programmés dans les différents cinémas du pays. Chaque cinéma est caractérisé par son nom, son adresse, et la ville où il se trouve, ainsi que les salles disponibles. Chacune de ces salles dispose d'un numéro, sa capacité, et nous pouvons également afficher si elle est climatisée ou pas ainsi que la liste des séances programmées. Chaque séance est identifiée par un numéro, une heure de début et de fin, la salle de cinéma où elle est programmée, ainsi que le film prévu. Nous pouvons connaître le titre du film, l'année de sa production, le nom de son réalisateur et s'il s'agit d'une version originale ou traduite. Il doit être également possible de connaître la liste des acteurs qui ont joué dans ce film. Nous supposons qu'un réalisateur ne peut pas être acteur et inversement (un acteur ne peut pas être réalisateur).

Pour chaque artiste (acteur ou réalisateur), nous pouvons connaître le nom, le prénom, l'année de naissance, le nombre de films joués (si c'est un acteur) ou réalisés (si c'est un réalisateur). Si un acteur a eu un grand prix (ex. oscar) nous souhaitons connaître le nom du prix et l'année de son obtention (nous nous intéressons au dernier prix gagné). Pour les réalisateurs, nous souhaitons connaître, pour chacun d'eux, le film qui a eu le plus de succès.

**Question 4 :** Tracer un schéma indiquant les différentes classes en précisant les relations entre elles. Préciser si les classes sont abstraites.

**Question 5 :** Quelle structure de données proposez-vous pour conserver les acteurs qui ont joué dans chaque film ? Précisez la déclaration en java.

**Question 6 :** Nous souhaitons permettre à l'administrateur du site de rajouter des acteurs à la structure de donnée proposée dans la question 5. Cependant, la méthode de rajout que nous appellerons ajouterActeur doit retourner une erreur si la date de naissance de l'acteur est supérieure à la date de réalisation du film. Quelle solution proposer vous ? Donnez le code correspondant en java.

**Question 7 :** Nous voulons à présent associer à chaque acteur dans un film un entier représentant le degré d'importance de son rôle dans ce film (on associera l'entier 1 au premier rôle, l'entier 2 au second rôle...etc). Pour cela, nous créons une nouvelle classe que l'on nommera Role et qui contiendra deux attributs : un objet de type Acteur et un entier représentant l'importance du rôle que joue l'acteur dans le film. Expliquer ce qu'il faut faire pour pouvoir afficher les noms et prénoms des acteurs d'un film dans l'ordre décroissant de l'importance de leurs rôles. Ecrire les fragments de codes java correspondants.

## Partie 2 : Structures de données et de fichiers dynamiques

On désire ranger  $n$  données en vue de réaliser l'opération "Suivant" qui consiste à fournir à la demande le prochain élément dans l'ordre d'un élément donné. Grâce à cette opération, on va pouvoir réaliser l'opération de requête à intervalle qui est très utilisée dans les systèmes de gestion de base de données.

Une requête à intervalle est tout simplement l'opération qui consiste à récupérer (ou afficher) toutes les données comprises entre deux valeurs  $a$  et  $b$  données.

Comme application, on peut considérer une épreuve d'algorithmique ayant pour but de sélectionner les étudiants ayant une note supérieure ou égale à 12. Les correcteurs corrigent et insèrent (à distance) les notes dans une structure de données commune. Pour afficher les résultats, il suffit de trouver dans la structure de données, le premier étudiant avec une note supérieure ou égale à 12 puis utiliser la fonction "Suivant" de manière répétitive. Grâce à cette fonction "Suivant" la structure de donnée se comporte comme une liste linéaire chaînée ordonnée ordinaire.

- Une première solution consiste à allouer un tableau de  $m$  éléments ( $m \ll n$ ) et commencer à ranger les données. Si le tableau ne suffit pas on alloue un autre tableau de  $m$  éléments et on continue à ranger les autres données. Si besoin est, un troisième tableau est demandé. Etc. On a ainsi une liste linéaire chaînée de tableaux dynamiques. Afin de répondre avec efficacité à l'opération projetée, la liste est maintenue ordonnée.
- Une autre solution consiste à ranger les éléments dans un arbre de recherche binaire (maintenu équilibré par les techniques existantes) et donc on n'a plus à gérer l'espace occupé par les données.
- Enfin, une troisième possibilité est utilisée si les données sont très volumineuses pour être contenues en mémoire centrale. Il s'agit tout simplement de ranger les données dans un B-arbre d'ordre  $m$ .

### Questions

*Question 1 : Décrire les structures de données et écrire la fonction "Suivant" dans chaque cas.*

*Question 2 : Etudier les espaces mémoire (RAM) consommés par les différentes structures de données et les coûts de l'opération "Suivant" dans chaque cas.*

Les arbres de recherche binaire et les B-arbres sont implémentés avec le champ "Père".

N.B

- Pour la liste de tableaux, l'algorithme démarre d'un élément se trouvant dans le maillon  $P$  à la position  $D$ .
- Pour l'arbre de recherche binaire, l'algorithme démarre d'un élément se trouvant dans le nœud  $P$
- Pour le B-arbre, l'algorithme démarre d'un élément se trouvant dans le bloc  $P$  à la position  $D$ . On suppose que le bloc est déjà chargé en mémoire dans le buffer  $Buf$ .

### **Barème de notation** (sur 40 pts):

Partie 1 (sur 20)

question 1 : 3 pts, question 2 : 5 pts, question 3 : 2 pts,  
question 4 : 4 , question 5 : 1, question 6 : 2 , question 7 : 3

Partie 2 (sur 20)

question 1 : 15 pts, question 2 : 5 pts