Université Grenoble Alpes
U.F.R. Sciences de l'Homme et de la Société
Master MIASHS - DCISS

T.D. Algorithmique n° 13

Séquences contiguës

Exercice 1 : Gestion de groupes

On utilise deux tableaux pour gérer ses groupes de TD d'une U.F.R. :
- un tableau d'étudiants contenant les noms des étudiants et leur n° de groupe de TD,
- un tableau des groupes de TD, contenant les noms des enseignants et les effectifs du groupe.

Le premier tableau est défini par les types suivants :

Etudiant : type agrégat **nom** : chaîne ; **groupe** : entier ≥ 0 fagregat

TabEtudiants : type tableau sur [0..NMAX] de Etudiant

On repère la fin de la liste des étudiants par le mot "FIN" qui joue le rôle de marque de fin de séquence : si le tableau comporte n éléments, le mot "FIN" est placé en position n (voir schéma).

TabEtudiants :

0	1	2	3	...	n	...
Mireille Ledisc	Luce Yfair	Paul Hochon	Ivan Skivolle	...	FIN	...
2	1	2	4	...		...

Le second tableau est défini par les types suivants :

Td : type agrégat **enseignant** : chaîne; **effectif** : entier ≥ 0 fagrégat
TabTd : type tableau sur [**1**..TDMAX] de Td

Les indices du second tableau correspondent **aux numéros des groupes de TD** du premier tableau. Ainsi, dans l'exemple suivant, Mireille Ledisc est dans le groupe de TD n°2, dont Mme Durand est l'enseignante. Il y a 22 étudiants dans ce TD.

TabTd :

1	2	3	4	5	...
M. Lepensec	Mme Durand	Mme Docteur	Mme Durand	FIN	...
26	22	25	12		...

Un enseignant peut avoir plusieurs groupes de TD, il peut donc apparaître plusieurs fois dans le second tableau. Comme pour le premier tableau, la chaîne de caractères "FIN" marque la fin de la liste des groupes de TD.

a) Donnez le contenu du lexique partagé, on y placera l'écran pour l'affichage des messages d'erreur.

b) Ecrire la fonction **vérification** qui vérifie si, pour un groupe donné, le nombre d'étudiants du premier tableau correspond bien à ce qui est mentionné dans le second tableau.

fonction **vérification**(**g** : entier ; **t1** : TabEtudiants ; **t2** : TabTd) ⟶ **booléen**
// vérification(g, t1, t2) renvoie vrai si le nombre d'étudiants du groupe g dans t1 est égal à la
// valeur correspondante dans t2

c) Ecrire l'action **supprimer** qui supprime un étudiant du premier tableau. On ne supprime pas physiquement un enregistrement mais on place la valeur 0 dans le champ groupe. Il conviendra de mettre à jour le second tableau. Si l'étudiant n'existe pas, on affichera un message d'erreur. Si un groupe de TD n'a plus d'étudiants, laisser simplement la valeur 0 dans l'effectif.

action **supprimer**(Consulté **e**: chaîne ; Modifié **t1** : TabEtudiants ; Modifié **t2** : Tabtd)
// Effet : supprime l'étudiant de nom e
// E.I. : t1 contient probablement un enregistrement de nom e
// E.F. : t1 contient -1 dans le champ groupe de l'enregistrement dont le nom est e. L'effectif
// correspondant dans t2 est aussi mis à jour ; si e non trouvé, t1 et t2 sont inchangés

d) Ecrire l'action **ajouter** qui, étant donné un nom d'étudiant et le numéro de son groupe de TD, met à jour les deux tableaux.
- Si le nom appartient déjà au tableau, cela signifie que l'étudiant a changé de groupe de TD, ou que l'étudiant a été supprimé auparavant (groupe = 0) ; il conviendra alors de ne pas rajouter un enregistrement, mais de mettre à jour les informations.
- S'il faut ajouter un enregistrement, utilisez si possible les cases "supprimées" par l'action précédente. Si le premier tableau est plein, il faut écrire un message d'erreur et rien ne doit alors avoir été modifié.
- On suppose que les données sont cohérentes.

action **ajouter**(Consulté **e**: chaîne ; Consulté **g** : entier > 0; Modifié **t1** : TabEtudiants ;
Modifié **t2** : Tabtd)
// Effet : ajoute l'étudiant e qui est dans le groupe de TD numéro g
// E.I. : indifférent
// E.F. : sauf en cas de tableau trop petit, le tableau t1 contient e et le numéro du groupe g.
// Les effectifs correspondants dans le second tableau sont aussi mis à jour.

Exercice 2 : File d'attente chez un médecin

On désire représenter dans un tableau (en contigu) une file d'attente chez un médecin. Chaque enregistrement contient les informations relatives à un patient. Deux variables, appelées *premier* et *dernier*, permettent de mémoriser l'indice du premier patient (le prochain appelé par le médecin) et celui du dernier patient arrivé dans la salle d'attente.
Chaque nouvel arrivant dans la salle d'attente est ajouté dans le tableau à la fin de la file (la variable *dernier* est modifiée). De même, chaque patient appelé par le médecin est supprimé de la file (la variable *premier* est modifiée). Des patients peuvent aussi abandonner leur tour en décidant de quitter ; les enregistrements sont alors décalés pour préserver leur contiguïté.
Lorsque la variable *dernier* arrive à la dernière case du tableau, les ajouts suivants se feront au début du tableau.

On utilisera le type Patient composé d'un nom et d'un numéro de sécurité sociale :

Patient type agrégat **nom** : chaîne, **numéroSS** : chaîne fagrégat

a) Comment représenter une file d'attente vide ? une file d'attente pleine ?

b) Combien de patients peut-on représenter dans un tableau de NMAX patients ?

c) Ecrire la classe FileDAttente, c'est-à-dire les variables de la partie privée qui permettent de définir une file d'attente, un constructeur qui crée une file vide et les méthodes suivantes :

action **arrivéePatient**(consulté **x** : Patient, élaboré **ok**: booléen)
// Effet : ajoute le patient x à la file. Le booléen est à faux en cas d'erreur ;

action **appelPatient**(élaboré ok : booléen)
// Effet : supprime le premier patient de la file d'attente et affiche son nom ok est à faux en cas d'erreur

action **abandonPatient**(consulté n : chaîne, élaboré ok : booléen)
// Effet : supprime le patient de nom n de la file ; ok est à faux en cas d'erreur

fonction **nombreDePatients**() → entier ≥0
// renvoie le nombre de patients présents dans la file d'attente

fonction **estPresent**(num : chaîne) → booléen
// renvoie vrai si le patient de numéro num est présent dans la file d'attente