

---

## Exercice 1

---

Une entreprise a doté ses employés d'un capteur sans fil qui émet un code numérique chaque fois qu'un employé traverse la porte d'entrée ou de sortie de l'entreprise. L'entreprise a moins de 1000 employés. À chaque passage, le capteur sans fil enregistre l'heure et la minute de passage ensemble avec le code de l'employé (Un code alphanumérique avec un maximum de 10 caractères).

On veut développer un programme en Haskell pour le calcul des heures de travail des employés de cette entreprise. Le programme reçoit de la ligne de commande, un premier paramètre qui représente le nom du fichier contenant les passages, et un second paramètre (optionnel), qui représente le code numérique d'un employé. Le fichier est relatif au passage d'une journée et est composé d'une série de lignes, dont chacune d'elles correspond à un passage et est composée de trois champs: heure minute code\_employé. Si le programme est appelé avec les deux paramètres sur la ligne de commande (le code employé est présent), alors doit afficher pour l'employé spécifié, le nombre total de minutes de travail. Pour déterminer le nombre total de minutes de travail, il est nécessaire de comparer l'heure du premier passage et l'heure du dernier passage de cet employé. Si par contre le programme est lancé avec un seul paramètre sur la ligne de commande (le code employé est absent), alors le programme doit afficher le nombre total d'employés différents qui ont travaillé ce jour (c'est-à-dire qui sont passés au moins une fois par de l'entrée)

exemple:

Étant donné le fichier passages.txt, son contenu est:

```
8 30 abc222
8 30 abc123
8 31 azx112
9 10 abc123
12 10 abc123
```

Le programme appelé horaire doit se comporter comme suit:

```
$ horaire passages.txt
```

Il y a 3 employés

\$ horaire passages.txt abc123

L'employé abc123 a travaillé pendant 220 minutes.

---

## Exercice 2

---

Réaliser un programme en langage Haskell pour l'enregistrement des cartes recharges entre groupe d'amis.

la liste des cartes est mémorisée dans un fichier texte, et est composée d'un nombre inconnu de lignes, dont chacune d'elle contient trois éléments:

le nom de l'expéditeur, le nom du destinataire, et le nom de la localité d'où la carte a été envoyée. Le nom des amis et de la localité sont supposés ne pas contenir des espaces et de longueur maximale 30 caractères chacune. Le programme reçoit comme premier paramètre sur la ligne de commande le nom du fichier texte, alors que le second paramètre peut être la chaîne new ou bien la chaîne find. La commande new requiert trois paramètres en plus sur la ligne de commande correspondant aux noms des amis et de localité, et doit ajouter une telle information à la fin du fichier. Le programme doit signaler avec un message d'erreur l'éventuelle tentative de ré-introduire une carte déjà existante.

La commande find est par contre suivie d'un seul paramètre sur la ligne de commande, correspondant au nom d'un ami. Dans ce cas le programme doit afficher la liste des amis qui ont expédié une carte à l'ami spécifié et la localité correspondante.

Exemple:

Supposons que le programme s'appelle carte et que le fichier carte.txt contienne les données suivantes.

Ndoubé Léa Bomono

Ndoubé Tchuenté Bomono

Léa Ebongue Melong

Tchuenta Lea Edea

Dans ce cas, en executant le programme des manières suivantes:

```
$carte carte.txt find Lea
```

Doit être généré l'output suivant:

Carte reçu par Lea:

1- Ndoube de Bomono

2- Tchuenta de Edea

par contre executant le programme avec la commnde suivante:

```
$ carte carte.txt new Lea tchuenta Nkongsamba
```

devra etre ajouter au fchier carte.txt et a la fin la ligne suivante:

Lea Tchente Nkongsamba

---

### Exercice 3

---

Nous voulons développer un programme en langage Haskell pour gérer de façon numérique les enregistrements a l'examen. Les enregistrements sont stockés dans un fichier texte portant le nom log.txt et contient les données de N étudiants, où N est le nombre entier écrit a la première ligne du fichier. Après cette première ligne, chaque ligne suivante contient les données relatives à un seul étudiant, et indiquant le numéro d'immatriculation de l'étudiant (nombre entier compris entre 1 et 999999) et la note obtenue (nombre entier entre 12 et 20, ou zéro pour indiquer que l'examen n'a pas encore été passé). Le programme peut être exécuté de deux manières différentes.

S'il est exécuté en passant le mot stat comme premier paramètre sur la ligne de commande, le programme doit ensuite fournir les statistiques suivantes: nombre d'étudiants promus (et sur le total, exprimé avec un chiffre après la virgule) la moyenne des élèves promus (indiqué avec un seul chiffre après la virgule).

Le programme peut également être exécuté en passant le mot note comme premier paramètre comme deuxième paramètre le numéro matricule de l'étudiant et en dernier la note obtenu par l'étudiant. Dans ce cas, le programme doit insérer la note de l'étudiant dans le fichier, mais signaler une erreur dans le cas où l'étudiant n'est pas inscrit a l'examen (c'est-à-dire son numéro matricule n'apparaît pas dans le fichier) ou a déjà passé l'examen (c'est-à-dire possède déjà une note non nulle dans le registre). Par exemple, si le fichier log.txt contenait les données suivantes:

3

33467 30

24356 0

224678 18

et le programme - supposé être appelé examen - a été exécuté avec la ligne de commande suivante:

examen stat

alors le programme devrait produire la sortie suivante:

promu = 2 (66,7%)

note moyenne = 24,0

Si par contre le programme est activé avec:

examen note 24356 24

après l'exécution du programme, le fichier log.txt doit contenir les éléments suivants données:

3

33467 30

24356 24

224678 18

---

#### Exercice 4

---

Un professeur veut mettre en place un programme qui lui permet de faire des statistiques sur la fréquentation des cours universitaires tenues par lui. Chaque cours universitaire est caractérisé par un code (par exemple 06AZNDI). Chaque fois que l'enseignant fait une leçon, il doit utiliser le programme pour entrer les informations liées à cette leçon, et en particulier: la date et le nombre d'étudiants présents à la leçon. Les informations sont stockées dans un fichier de travail appelé `lessons.txt`. Ce fichier est composé d'un nombre variable, non connu a priori, de lignes, dont chacune conserve les informations sur une seule leçon. Le fichier peut contenir des informations

concernant de nombreux cours différents, librement entremêlées. Le format de chaque ligne du fichier est comme suit:

`code date nombreEtudiant`

où:

- le code est le code du cours (10 caractères maximum, sans espaces);
- date est la date de la leçon, représentée par un nombre entier compris entre 1 et 365;
- nombreEtudiant est le nombre d'étudiants présents, représentés sous la forme d'un nombre entier positif.

Le programme est appelé avec deux arguments sur la ligne de commande: le premier argument indique le code du cours concerné, tandis que le second indique l'opération effectuer. L'opération peut être I pour "insertion" ou S pour "statistiques".

En particulier:

- dans le cas de l'insertion d'une nouvelle leçon (liée au cours indiqué sur la ligne de commande), le programme demandera à l'utilisateur les informations nécessaires (date et nombre d'étudiants)

et mettra à jour le fichier de travail en y ajoutant une ligne. Après traitement de cette instruction, le programme se termine.

- affichage des statistiques d'un cours. Dans ce cas, le programme calcule et affiche, pour le cours indiqué sur la ligne de commande, les informations suivantes: date du cours avec le plus grand nombre d'étudiants, date de la leçon avec le moins d'étudiants, nombre moyen d'étudiants ayant assisté aux cours. Le programme se termine après.

Par exemple, en supposant que le programme s'appelle `registre`, et que le fichier `lessons.txt` est initialement vide,

une interaction possible avec le programme est la

suivante (notez que `c:>` est l'invite de commande du système d'exploitation):

```
c:> registre 06AZNDI I
```

Date: 101

Étudiants: 40

c:> registre 04KKZWE I

Date: 104

Étudiants: 99

c:> registre 06AZNDI I

Date: 98

Étudiants: 45

c:> registre 06AZNDI S

Le nombre minimum d'étudiants a été atteint le 101

Le nombre maximum d'étudiants a été atteint le 98

La moyenne des étudiants est de 42,5