

Hôtel

Consignes Chaque fonction doit être accompagnée de sa spécification et d'un jeu de test. Les questions impliquent l'écriture de fonctions supplémentaires dont le comportement n'est pas spécifié dans l'énoncé. Elles pourront être des fonctions locales ou globales à votre guise, mais dans les deux cas, elles doivent être aussi accompagnées d'une spécification. Les fonctions locales ne peuvent pas être testées directement, employez les avec prudence.

On peut, bien sûr, utiliser les fonctions définies dans les questions précédentes pour répondre à une question.

Rappel La forme spéciale `verifier` n'est pas décrite dans votre carte de référence. En voici la syntaxe qui complète la section *Grammaire du langage* de la carte de référence.

```
<forme-spéciale> ::= ...  
                   ou  (verifier <symbole> <expression-test>*)
```

Où la grammaire d'<expression-test> est décrite dans la carte de référence, et un <symbole> est n'importe quel nom qui peut être utilisé comme nom de fonction. De telle manière que pour tester la fonction `toto:int→int`, on pourra écrire, par exemple :

```
(verifier toto  
  (toto 18)  -> 42  
  (toto -7.3) -> ERREUR)
```

Devoir

Dans ce devoir nous considérerons un hôtel, dont les clients sont représentés par leur nom (une chaîne de caractères), les chambres par la liste des clients qui y résident actuellement, les étages par la liste des chambres qui s'y trouvent, et l'hôtel lui-même par la liste de ses étages. L'hôtel est donc représenté par un objet de type `LISTE[LISTE[LISTE[string]]]`. Voici un exemple d'hôtel, qui a trois étages :

```
((("Jean-Robert") () ("Loic"))  
  (( ) ("Caroline" "Franck" "Jacques" "Jean-Christophe"))  
  (("Joelle" "Lionnel") () ("Alexandre" "Anne")))
```

Question 1 Écrire une fonction `nb-chambres-vides` qui, étant donné un hôtel, compte le nombre de chambres vides dans l'hôtel. Dans l'exemple, il y en a 4.

Question 2 Écrire une fonction `etage-vide?` qui, étant donné un hôtel, vérifie si il y a dans l'hôtel, un étage où toutes les chambres sont vides. Ce n'est pas le cas dans l'exemple.

Question 3 Écrire une fonction `clients` qui, étant donné un hôtel, retourne la liste des clients. On pourra supposer qu'il n'y a pas deux clients du même nom dans l'hôtel. Voici la liste des clients de l'exemple:

```
("Jean-Robert" "Loic" "Caroline" "Franck" "Jacques"  
 "Jean-Christophe" "Joelle" "Lionnel" "Alexandre" "Anne")
```

Question 4 Dans cette question on suppose qu'un unique client quitte l'hôtel, sans que les autres occupants de sa chambre n'aient à partir. Par exemple, si Anne quitte l'hôtel de l'exemple, l'hôtel sera ensuite dans l'état suivant:

```
((("Jean-Robert") () ("Loic"))  
 (()) ("Caroline" "Franck" "Jacques" "Jean-Christophe"))  
 (("Joelle" "Lionnel") () ("Alexandre")))
```

Écrire une fonction `depart` qui, étant donné un hôtel et un client, retourne le nouvel état de l'hôtel après le départ du client. Si le client n'est pas dans l'hôtel, l'hôtel initial est retourné.