

### Exercice 6.3

Écrire une fonction qui recherche une clé dans une liste d'associations et renvoie la valeur associée à cette clé.

Écrire une fonction qui met à jour une liste d'associations, c'est-à-dire qui change la valeur associée à une clé si cette clé existe déjà dans la liste et ajoute un nouveau couple sinon.

## 6.2 La concaténation : modifier ou copier

### 6.2.1 Modifier

Imaginons maintenant que deux bibliothèques décident de fusionner. Pour construire le catalogue de la nouvelle bibliothèque, il faut fabriquer une nouvelle liste qui contient les éléments des anciens catalogues des deux bibliothèques. Cette opération, qui à partir de deux listes  $x_1, \dots, x_n$  et  $y_1, \dots, y_p$ , construit la liste  $x_1, \dots, x_n, y_1, \dots, y_p$  s'appelle la *concaténation*. Pour simplifier les dessins nous allons considérer des listes caractères et non des listes de chaînes de caractères. La concaténation peut se programmer ainsi

```
static List append (List x, List y) {
    if (x == null) return y;
    List p = x;
    while (p.tl != null) p = p.tl;
    p.tl = y;
    return x;}

```

Si on définit deux listes

```
List b = new List('a', new List('p', new List('p', null)));
List c = new List('e', new List('n', new List('d', null)));

```

et que l'on affiche leur concaténation

```
printList(append(b,c));

```

on obtient la liste

```
a p p e n d

```

En effet, dans le cas où la liste  $x$  n'est pas vide, l'instruction  $p = x$ ; commence par mettre la première cellule de la liste  $x$  dans la variable  $p$