Introduction à la programmation

Exercices sur les listes

Supprimer un élément d'une liste

Soit un élément x et soit une liste l ne contenant aucun doublon mais contenant un élément égal à x. Écrire une fonction $SupprimeListe$ qui, à partir de x et de l renvoie une autre liste contenant tous les éléments de l dans le même ordre sauf x. Si l ne contient pas x, alors la fonction devra renvoyer l telle quelle.

Une liste est-elle la permutation d'une autre ?

Soient deux listes l_1 et l_2 ne contenant aucun doublon. Proposer un algorithme capable de déterminer si une des listes est la permutation de l'autre liste. Écrire la fonction correspondante en *Caml*.

Le nombre de multiples de 5 dans une liste

Soit une liste d'entiers comportant ou non des doublons. Écrire une fonction qui à partir d'une telle liste renvoie le nombre d'éléments dans cette liste qui sont des multiples de 5.

Éliminer d'une liste tous les multiples de 5

Soit une liste l d'entiers comportant ou non des doublons. Écrire une fonction *suppr_listeM5* qui, à partir d'une telle liste renvoie une autre liste contenant les mêmes éléments que l dans le même ordre, à l'exception des éléments qui sont multiples de 5.

Éliminer d'une liste tous les éléments qui vérifient P

Nous cherchons à généraliser le propos. Nous cherchons à pouvoir éliminer d'une liste les éléments qui sont multiples de 3, ou des diviseurs de 100, ou encore des nombres qui représentent une année bissextile et ainsi de suite. De façon plus générale, nous cherchons à pouvoir éliminer d'une liste l'ensemble de tous les nombres qui vérifient une propriété P, sans avoir à écrire une nouvelle fonction pour chaque propriété. Soit une fonction $proprieteP$ qui étant donné un élément x renvoie *true* si x vérifie la propriété P et *false* autrement. Écrire une fonction ***suppr_listeP*** qui à partir d'une liste l et d'une fonction $proprieteP$ renvoie une autre liste contenant les mêmes éléments que l et dans le même ordre, à l'exception de ceux qui vérifient la propriété P.

Que fait la fonction suivante ?

```
let rec merp = function
 [] -> true
|l -> let ll=suppr_listeP (function a -> a=list_length l) l in
    (list_length ll=list_length l-1)&(merp ll);;
```

où `list_length` est une fonction prédèfinie en *Caml* et qui renvoie le nombre d'éléments que possède une liste.