

PROGRAMMATION FONCTIONNELLE**Exercice 1 :**

- 1°) Ecrire une fonction `fcarre` ayant comme argument une fonction `f` de \mathbb{R} dans \mathbb{R} et telle que l'évaluation de l'expression `(fcarre f)` retourne une fonction de \mathbb{R} dans \mathbb{R} qui à toute valeur x de \mathbb{R} associe $[f(x)]^2$.
Il vous est demandé d'écrire la fonction `fcarre` de telle sorte que dans le calcul de $[f(x)]^2$, le calcul de $f(x)$ ne soit effectué qu'une seule fois, c'est-à-dire que la fonction `f` ne soit appelée qu'une seule fois.
- 2°) En utilisant les fonctions Scheme `sin` et `cos` permettant de calculer respectivement un sinus et un cosinus, écrire à l'aide de `fcarre` l'appel permettant de calculer la valeur de $g(4)$ ou g est la fonction telle que $g(x) = (\sin x + \cos x)^2$.

Exercice 2 :

L'objectif de cet exercice est d'écrire une fonction qui teste l'égalité de deux ensembles A et B représentés respectivement par deux listes A et B .

1°) Ecrire une fonction `CS` qui a comme arguments un symbole x , représentant un élément x , et une liste E , représentant un ensemble E , et telle que l'évaluation de l'expression `(CS x E)` retourne une liste $(B F)$ telle que si $x \in E$, alors $B = \#t$ et F est une liste représentant $E - \{x\}$, et si $x \notin E$, alors $B = \#f$ et F est une liste représentant E .

Remarque : l'objectif de cette fonction est de faire deux choses (tester si un élément appartient à une liste et, si cela est le cas, retourner la liste dans laquelle cet élément a été supprimé) en ne faisant qu'un seul parcours de la liste.

2°) Ecrire une fonction `identique?` qui a comme arguments deux listes A et B , représentant deux ensembles A et B , et telle que l'évaluation de l'expression `(identique? A B)` retourne la valeur logique *vrai* si les ensembles A et B contiennent les mêmes éléments, et la valeur logique *faux* sinon.

Exercice 3 :

1°) Le but de cette question est d'écrire une fonction `permut` qui a comme argument une liste L et qui retourne la liste de toutes les permutations possibles des éléments de L .

Par exemple, l'évaluation de `(permut ())` doit retourner `(())`,

celle de `(permut '(3))` doit retourner `((3))`,

celle de `(permut '(2 3))` doit retourner `((2 3) (3 2))`,

celle de `(permut '(1 2 3))` doit retourner `((1 2 3) (2 1 3) (2 3 1) (1 3 2) (3 1 2) (3 2 1))`.

- a) Soit une fonction `distribuer` qui a comme argument une variable `e` et une liste `L` ayant n éléments, et qui retourne une liste de $n+1$ sous-listes, la $k^{\text{ème}}$ sous-liste étant obtenue en insérant `e` en $k^{\text{ème}}$ position dans `L`.

Par exemple, l'évaluation de `(distribuer 1 '(2 3))` doit retourner
`((1 2 3) (2 1 3) (2 3 1))`.

Proposer le principe de la fonction `distribuer` puis l'écrire en Scheme.

- b) En utilisant la fonction `distribuer`, écrire la fonction `permut`.

2°) Le but de cette question est d'écrire une fonction `permut2` qui a comme argument une liste `L` de taille n contenant les n premiers entiers non nuls, et qui retourne la liste des permutations σ des éléments de `L` telles que $\forall i \in \{1, \dots, n\}, \sigma(i) \neq i$.

Par exemple, l'évaluation de `(permut2 '(1))` doit retourner `()`,

celle de `(permut2 '(1 2))` doit retourner `((2 1))`,

celle de `(permut2 '(1 2 3))` doit retourner `((2 3 1) (3 1 2))`.

- a) Que fait la fonction `F` suivante :

```
(define F (lambda (L P)
```

```
  (append-map (lambda (e) (if (P e) (list e) ())) L) ))?
```

- b) Ecrire la fonction `permut2` en utilisant les fonctions `permut` et `F`.