



QCM - Programmation fonctionnelle

27/02/2022

Durée: 15 min. Les documents sont interdits.

Ce contrôle est un questionnaire à choix multiples qui sera corrigé automatiquement. Il est donc important de remplir avec soin les réponses au stylo noir (le crayon de papier ou les stylos plus clairs peuvent poser des problèmes de lecture). Lorsque vous choisissez une réponse, il faut noircir complètement la case correspondante. Ne pas toucher aux cases situées tout en haut de la feuille, elles servent à repérer vos copies lors de la correction automatisée. Il y a 5 questions en tout, 2 points par question, -1 point par erreur et minimum 0 par question (pas de point négatif).

Attention, pensez à tourner la page

<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input checked="" type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0	<input type="checkbox"/>	0
<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1	<input type="checkbox"/>	1
<input type="checkbox"/>	2	<input checked="" type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2	<input type="checkbox"/>	2
<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3	<input type="checkbox"/>	3
<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input type="checkbox"/>	4	<input checked="" type="checkbox"/>	4
<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5	<input type="checkbox"/>	5
<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6	<input type="checkbox"/>	6
<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7	<input type="checkbox"/>	7
<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8	<input checked="" type="checkbox"/>	8	<input type="checkbox"/>	8	<input type="checkbox"/>	8
<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9	<input type="checkbox"/>	9

← codez votre numéro d'étudiant ci-contre, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :
BROILLET Virgile

Question 1 ♣ On considère la fonction suivante:

```
let rec f x l l' = if l = [] then x :: l' else (x mod 3) :: l
```

Indiquer, parmi les affirmations suivante celles qui sont justes.

- ☐ Dans tous les cas, *f* effectue un appel récursif en argument.
- ☒ La fonction *f* prend un argument de type *int*. ☐ La fonction *f* est récursive.
- ☐ La fonction *f* parcourt toute les listes passées ☐ Aucune de ces réponses n'est correcte.

Question 2 ♣ On considère la fonction suivante:

```
let rec f l =  
  match l with  
  | [] -> []  
  | x :: l' -> if x mod 3 = 0 then x :: f l' else f l'
```

Indiquer, parmi les affirmations suivante celles qui sont justes.

- ☐ Dans tous les cas, *f* effectue un appel récursif
- ☒ La fonction *f* parcourt toute la liste passée en argument. ☐ La fonction *f* prend un argument de type *string*.
- ☒ La fonction *f* est récursive. ☐ Aucune de ces réponses n'est correcte.



Question 3 ♣ On considère la fonction suivante:

```
let rec f l =  
  match l with  
  | [] -> []  
  | x :: l' -> if x = 3 then 9 :: f l' else (x * 3) :: f l'
```

Indiquer, parmi les affirmations suivante celles qui sont justes.

☒ Le code de `f` contient deux appels récursifs.

☒ La fonction `f` multiplie tous les éléments de son argument par 3.

☐ Si `l2 = f l` alors tous les éléments de `l2` sont dans `l`.

☒ La fonction `f` utilise le *pattern matching*.

☐ Aucune de ces réponses n'est correcte.

Question 4 ♣ On considère la fonction suivante:

```
let rec f e l = match l with  
  (* compléter ici *)
```

Indiquer, parmi les propositions suivantes, celles qui permettent de compléter le code de `f` de façon à ce que `f e l` soit une liste contenant chacun les éléments de `l` multiplié par `e`.

☒ `| x::l' -> (x * e) :: f e l'`

☐ `| [] -> if y = e then l else f e l`

☐ `| l'::x -> (x*e) :: l'`

☒ `| [] -> []`

☐ Aucune de ces réponses n'est correcte.

Question 5 ♣ On considère la fonction suivante:

```
let rec f e l = match l with  
  (* compléter ici *)
```

Indiquer, parmi les propositions suivantes, celles qui permettent de compléter le code de `f` de façon à ce que `f e l` renvoie `true` si `e` est dans `l`.

☒ `| x::l' -> x = e || f e l'`

☐ `| [] -> if y = e then false else f e l`

☐ `| l'::x -> x=e || l'`

☒ `| [] -> false`

☐ Aucune de ces réponses n'est correcte.