



+82/1/18+

QCM - Programmation fonctionnelle

06/03/2023

Durée: 15 min. Les documents sont interdits.

Ce contrôle est un questionnaire à choix multiples qui sera corrigé automatiquement. Remplir avec soin les réponses au stylo noir en noircissant complètement les cases choisies. Ne pas toucher aux cases situées tout en haut de la feuille. 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 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"/> | 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"/> | 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"/> | 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"/> | 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"/> | 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"/> | 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"/> | 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"/> | 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 | <input type="checkbox"/> | 9 |

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

Nom et prénom :

BROILLET...Vingile...

Question 1 ♣ Indiquer, parmi les propositions suivantes, celles qui correspondent à un décorateur pouvant être utilisé avec une fonction de type `int option -> int option`. Indication: dans chaque proposition, il y a un seul paramètre qui peut être une fonction.

- ☐ `fun a -> fun b -> 1 + (fun c -> a b)`
- ☒ `fun a -> fun b -> fun c -> if b c = 0 then b a else b c`
- ☒ `fun a -> fun b -> a (a b)`
- ☐ `fun a -> fun b -> fun c -> b a + b c`
- ☐ Aucune de ces réponses n'est correcte.

Question 2 ♣ Soit le code à compléter suivant:

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

Indiquer, parmi les propositions suivantes, celles qui *peuvent* être utilisées pour compléter ce code de façon à ce que `f` renvoie la liste résultant de la concaténation des listes obtenues en appliquant `g` aux éléments de `l`.

- ☒ `| x::l' -> g x @ f g l'`
- ☐ `| x::l' -> f (g x) l'`
- ☒ `| [] -> []`
- ☐ `| x::l' -> f g x :: f g l'`
- ☐ Aucune de ces réponses n'est correcte.



Question 3 ♣ Soit le code à compléter suivant:

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

Indiquer, parmi les propositions suivantes, celles qui *peuvent* être utilisées pour compléter ce code de façon à ce que f calcule la liste dont les éléments sont obtenus en appliquant g à tous les éléments de l.

- ☒ | [] -> []
☐ | x::l' -> f x :: f g l'
☐ | x::l' -> g x :: l'
☐ | [] -> g l
☐ Aucune de ces réponses n'est correcte.

Question 4 ♣ Soit le code à compléter suivant:

```
type 'a arbre = Vide | Noeud of 'a * 'a arbre * 'a arbre;;  
let f g a = match a with (* compléter ici *)
```

Indiquer, parmi les propositions suivantes, celles qui *peuvent* être utilisées pour compléter ce code de façon à ce que f renvoie true si pour au moins un des éléments de a, g renvoie true

- ☐ | Noeud (x,y,z) -> g x && f g y && f g z
☐ | Vide -> true
☐ | Noeud (x,y,z) -> Noeud(g x, f g y, f g z)
☐ | Vide -> Vide
☒ Aucune de ces réponses n'est correcte.

Question 5 ♣ Soit le code à compléter suivant:

```
type 'a arbre = Vide | Noeud of 'a * 'a arbre * 'a arbre;;  
let f g a = match a with (* compléter ici *)
```

Indiquer, parmi les propositions suivantes, celles qui *peuvent* être utilisées pour compléter ce code de façon à ce que f renvoie l'arbre obtenu en appliquant g à chacun des éléments de a.

- ☒ | Vide -> Vide
☒ | Noeud (x,y,z) -> Noeud (g x, f g y, f g z)
☐ | Noeud (x,y,z) -> Noeud (x, g y, g z)
☒ | Vide -> false
☐ Aucune de ces réponses n'est correcte.