



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 checked="" 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 ♣ On souhaite définir un type arbre binaire générique dont les données se trouvent dans les nœuds. On souhaite compléter la définition suivante:

type 'a arbre = (* compléter ici *)

Indiquer, parmi les propositions suivantes, celles qui peuvent être utilisées pour compléter la définition.

- ☐ | Conse of 'a * 'a arbre list
- ☐ | Consh of arbre list
- ☐ | Consq of int
- ☐ | Consf of 'a * arbre * arbre
- ☒ | Aucune de ces réponses n'est correcte.

Question 2 ♣ Soit le type suivant:

type expr = Cst of int | Plus of expr * expr | Mult of expr * expr

On souhaite définir une fonction d'évaluation pour ces expressions en complétant la définition suivante:

let rec eval e = match e with (* compléter ici *)

Indiquer, parmi les propositions suivantes, celles qui peuvent être utilisées pour compléter la définition.

- ☐ | Cst x -> eval x
- ☒ | Plus (x1,x2) -> eval x1 + eval x2
- ☐ | Mult(x1,x2) -> eval (x1 * eval x2)
- ☒ | Mult (x1,x2) -> eval (x1 * x2)
- ☐ | Aucune de ces réponses n'est correcte.



Question 3 ♣ Soit le type suivant:

`type arbre = Vide | Noeud of int * arbre * arbre`

On souhaite définir une fonction qui calcule la longueur (nombre de nœuds de la racine à la feuille) minimale d'une branche:

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

Indiquer, parmi les propositions suivantes, celles qui peuvent être utilisées pour compléter la définition.

- ☐ | Noeud (x1,x2,x3) -> min x1 (min x2 x3)
- ☒ | Noeud (x1,x2,x3) -> min (f x2) (f x3)
- ☒ | Vide -> 0
- ☒ | Vide -> f (min x)
- ☐ Aucune de ces réponses n'est correcte.

Question 4 ♣ Soit le type suivant:

`type arbre = Vide | Noeud of int * arbre * arbre`

On souhaite définir une fonction qui renvoie true si un élément appartient à un arbre de recherche:

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

Indiquer, parmi les propositions suivantes, celles qui peuvent être utilisées pour compléter la définition.

- ☐ | Noeud (x1,x2,x3) -> if x1 = e then x else if x1 > e then Noeud(x1,f e x2, x3) else Noeud(x1, x2, f e x3)
- ☐ | Vide -> true
- ☒ | Noeud (x1,x2,x3) -> if x1 = e then f e x2 else f e x3
- ☒ | Vide -> false
- ☐ Aucune de ces réponses n'est correcte.

Question 5 ♣ Soit le type suivant:

`type arbre = Vide | Noeud of int * arbre * arbre`

On souhaite définir une fonction qui renvoie la somme de a et des éléments d'un arbre:

`let rec f a x = match x with (* compléter ici *)`

Indiquer, parmi les propositions suivantes, celles qui peuvent être utilisées pour compléter la définition.

- ☐ | Vide -> 0
- ☐ | Noeud (x1,x2,x3) -> f (a+x1) x2 x3
- ☒ | Noeud (x1,x2,x3) -> if x1 = 0 then a else f a x2 + f a x3
- ☒ | Noeud (x1,x2,x3) -> x1 + f (f a x2) x3
- ☐ Aucune de ces réponses n'est correcte.