

Contrôle continu - IPF

lundi 8 décembre 2014 - 30 minutes

Nom - Prénom :

Compléter la session suivante. Il n'y a pas d'erreur syntaxique mais des erreurs de typage peuvent exister. Dans ce cas, indiquez la réponse *erreur de typage* et expliquez en quelques mots pourquoi.

Répondez directement sur la feuille.**Echauffement : session OCaml**

let h x = x*.x ;;

val h : float -> float = <fun>

let f (x,t) = x-.t ;;

let w (x, y) = y && (x>0);;

[true ; 0.0];;

let j x y = x y y;;

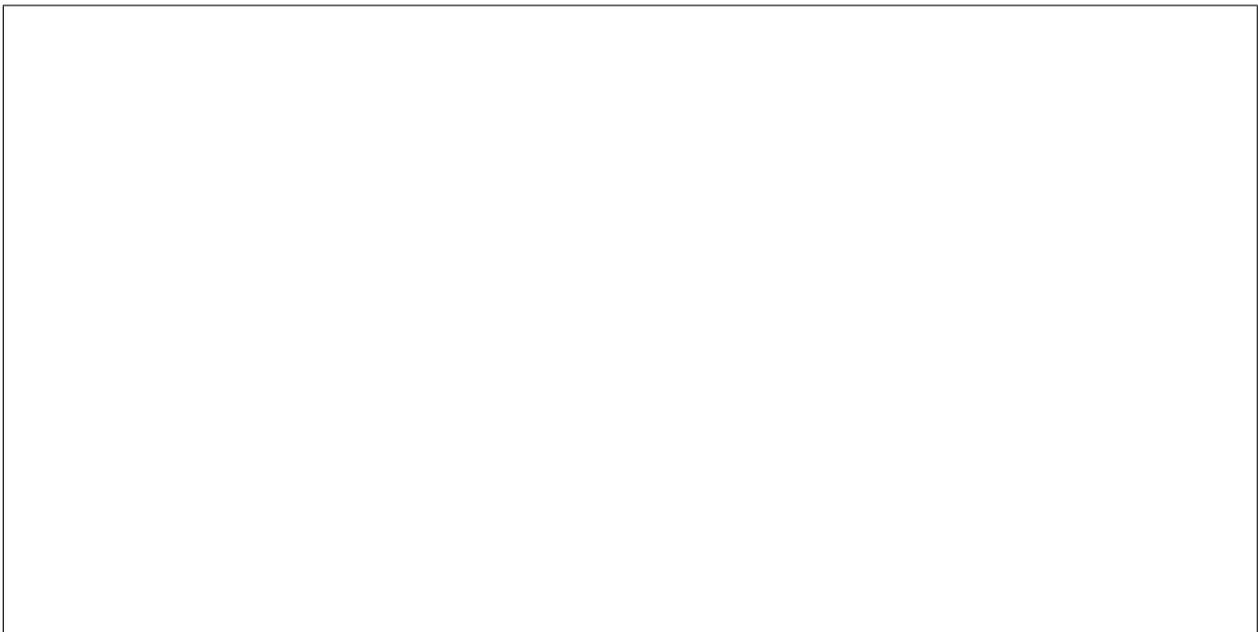
Fonction 1 : Programmez la fonction dont l'interface est la suivante

```
(* interface longueur
type : 'a list -> int
arguments : l
précondition : aucune
postcondition : retourne la longueur (nb d'éléments) de la liste l
test(s) : longueur [1.5; 2.4; 3.5; 1.2] = 4 *)
```



Fonction 2 : Programmez la fonction dont l'interface est la suivante

```
(* interface separe_positifs_negatifs
arguments : une liste l
précondition : l est une liste d'entiers
postcondition : 2 listes (la première contenant les éléments positifs de l
et la seconde les négatifs de l)
test(s) : separe_positifs_negatifs [-1;2;3;-4;5;-6;7] = ([2;3;5;7], [-1;-4;-6]) *)
```



Fonction 3 : Programmez la fonction dont l'interface est la suivante

```
(* interface concat
arguments : 2 listes l et m
précondition : aucune
postcondition : le résultat est une liste contenant les éléments de la
                 liste l suivies des éléments de la liste m
tests : concat [1;3;5] [2;4] = [1;3;5;2;4] *)
```

Il est interdit d'utiliser la fonction @.

Fonction 4 : Re-programmez la fonction concat de manière récursive terminale

Il est toujours interdit d'utiliser la fonction @.