Contrôle continu - IPF

mardi 18 décembre 2012 - 30 minutes

Nom - Prénom : .....

Compléter la session suivante. Il n'y 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 n = n*n;;
val h : int -> int = <fun>

# let f (x,t) = x*.t ;;
```

```
# f (2,6);;
```

```
type lettre = Voyelle of char | Consomme of char;;

[ Voyelle 'e' ; Consomme 'f'];;
```

```
let l = [] ;;
```

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

```
(* interface base2
type : int -> int list
argument : x
precondition : x >=0
postcondition : base2 est une liste contenant la représentation
                en base 2 du nombre x (des bits de poids faible
                à gauche vers les bits de poids fort à droite)
tests : base2 4 = [0;0;1] base2 7 = [1;1;1] *)
```

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

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

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

```
(* interface return_last
arguments : l
précondition : **prévoir le cas de liste vide**
postcondition : retourne le dernier élément de la liste l
test(s) : return_last [1.0;0.0; 2.0] = 2.0
*)
```

Fonction 4 : Programmez la fonction dont l'interface est la suivante

```
(* interface produit [n..m]
type : int*int -> int
argument : (n,m)
precondition : n<=m
postcondition : retourne le produit de tous les éléments
compris entre n et m
tests : produit 2 4 = 2*3*4 = 24 *)
```