

CC 4 (2011-2012)

Dans tout les cas, une solution récursive devra être sous forme terminale.

Exercice 1

On définit les types

```
t_info3 = (string * int *float) list
```

```
t_exo = {info1 : int ; info2 : int ; info3 : t_info3} ;;
```

1°/ Indiquer parmi les expressions suivantes celles qui sont valides pour une variable x de type t_exo . On justifiera les réponses, soit par un commentaire et une version syntaxiquement exacte, soit par une indication d'une interprétation dans le cas où la syntaxe est exacte.

- $(x.info1, x.info2)$
- $x.list$
- $(x.info3 ; "university")$
- `let f a y z = if a y then match (z.info3,y.info3) with (c,i,r),(cc,ii,rr) -> r +. rr ;;`

2°/ Ecrire une fonction qui permet de concaténer toutes les chaînes contenues dans un argument de type t_info3 .

3°/ Même question en considérant que l'argument est de type t_exo .

Exercice 2

Une culture est caractérisée par son nom, la date de plantation, la superficie cultivée, la date de récolte, les dépenses de culture, le prix de vente. On suppose qu'a été défini le type t_date .

```
type t_date = {jour : int ; mois : int ; annee : int}
```

1°/ Définir un type enregistrement **$t_culture$** à 6 champs qui traduit cette information.

2°/ Ecrire une fonction **antérieur** booléenne qui vérifie que deux dates sont dans l'ordre chronologique.

Ecrire une fonction **coherent** booléenne qui vérifie, pour une culture donnée de type $t_culture$, que la date de récolte est postérieure à la date de plantation.

3°/ Ecrire une fonction **rendement** qui indique, pour une culture donnée de type $t_culture$, le bénéfice relativement à la superficie cultivée.

Une ferme est caractérisée par une liste d'enregistrements de type $t_culture$.

4°/ Ecrire une fonction **culture** qui établit, pour une ferme donnée, la liste des noms des cultures de bénéfice positif.

5°/ Ecrire une fonction **benefice** qui établit le bénéfice cumulé d'une ferme.

6°/ Ecrire une fonction **present** qui vérifie qu'une culture indiquée par son nom est présente dans une ferme. Un appel pourrait être : `present "carotte" x`

7°/ Ecrire une fonction **fusion** qui permet d'établir la liste des cultures présentes dans deux fermes. La liste résultat, liste de chaînes, ne devra pas comporter de doublon.

Généraliser la fonction à la fusion d'une liste de fermes. On écrira la fonction **union**.