Unversité Paris Descartes
Ufr de Mathématiques et Informatique
Licence 1 MIA
Programmation fonctionnelle
Mai 2012

Durée de l'épreuve 40 mn

Modifiez le fichier cc3.ml qui vous est fourni. Sauvegardez régulièrement votre travail.

Vous disposez de la liste suivante qui correspond aux rendements des productions de blé et/ou de maïs dans quatre pays européens. Les rendements sont fournis en kg/ha pour les trois dernières années.

```
let production_globale =
[
{id=1;pays=Italie;                cereale=Ble;     rendement=(5444,5361,5316)};
{id=2;pays=Belgique;              cereale=Ble;     rendement=(9185,8650,9373)};
{id=3;pays=France;                cereale=Ble;     rendement=(7538,6981,6832)};
{id=4;pays=Italie;                cereale=Mais;    rendement=(7444,7361,7316)};
{id=5;pays=Belgique;              cereale=Mais;    rendement=(11185,10650,10373)};
{id=6;pays=Allemagne; cereale=Mais;    rendement=(9538,8981,9832)};
];;
```

1. Création d'un enregistrement **production_inconnue** dans lequel le champ id a pour valeur 0, le champ pays a pour valeur Pays_inconnu, le champ cereale a pour valeur Cereale_inconnue et le champ rendement a pour valeur 0 pour les trois années.

2. Définition de type
   1. Définir le type **t_pays** associé au champ pays.
   2. Définir le type **t_cereale** associé au champ cereale.
   3. Définir le type **t_production** permettant de définir une production de céréale pour un pays.

3. Création d'une sous liste
   1. Créer une fonction récursive terminale **extraire** qui crée à partir d'une **liste** donnée la liste des éléments qui respectent un **predicat** donné.
   2. Créer une fonction **rendement_moyen** qui calcule le rendement moyen pour une **production** donnée. Le résultat sera de type réel.
   3. Créer une fonction booléenne **production_en_augmentation** qui verifie si le rendement de la dernière année est supérieur au rendement moyen des trois dernières années.
   4. Créer la liste des productions dont le rendement de la dernière année est supérieur au rendement moyen des trois dernières années.

4. Calcul
   1. Créer une fonction **element** qui retourne le premier élément d'une **liste** qui correspond à un **predicat** donné, si aucun élément ne correspond au predicat la fonction retourne un élément **vide** fourni en argument.
   2. Créer une fonction booléenne **egale_production** qui vérifie qu'une **production** donnée correspond à la production d'une **céréale** donnée pour un **pays** donné.
   3. Créer une fonction **production_selectionnee** qui recherche la production pour une **céréale** et un **pays** donnés dans la liste **production_globale**. Si la production n'existe pas la fonction retourne une **production_inconnue**. Cette fonction fera appel à la fonction **element.**
   4. Calculer le rendement moyen de la production de blé en France.

Unversité Paris Descartes
Ufr de Mathématiques et Informatique
Licence 1 MIA
Programmation fonctionnelle
Mai 2012

Durée de l'épreuve 40 mn

Modifiez le fichier cc3.ml qui vous est fourni. Sauvegardez régulièrement votre travail.

Vous disposez de la liste suivante qui correspond à la liste de compétiteurs qui ont participé à une épreuve de lancer du poids ou de lancer du javelot.
La distance parcourue est exprimée en mètres et en centimètres.

```
let competition =
[
{dossard=1;nom="DUPONT";  prenom="Pierre";   discipline=Javelot;  distance=(80,10) };
{dossard=2;nom="MAX";     prenom="Luce";     discipline=Poids;    distance=(63,20) };
{dossard=3;nom="DUPONT";  prenom="Paul";     discipline=Javelot;  distance=(82,12) };
{dossard=4;nom="LEON";    prenom="Pedro";    discipline=Javelot;  distance=(79,62) };
{dossard=5;nom="ALEX";    prenom="Luc";      discipline=Javelot;  distance=(83,12) };
{dossard=6;nom="CAMEL";   prenom="Camille";  discipline=Poids;    distance=(63,90) };
{dossard=7;nom="DIXON";   prenom="John";     discipline=Poids;    distance=(63,10) };
{dossard=8;nom="SMITH";   prenom="John";     discipline=Javelot;  distance=(80,12) };
];;
```

1. Définition de type
   1. Définir le type t_discipline associé au champ discipline.
   2. Définir le type t_competiteur qui définit un enregistrement qui représente un compétiteur.
2. Création d'une sous-liste.
   1. Le distance parcourue par l'engin est exprimée en mètres et en centimètres, créer une fonction **distance_en_centimetres** qui convertit le **distance** en centimètres.
   2. Créer une fonction booléenne **qualification** qui pour une **discipline,** des **minima** et un **competiteur** donnés vérifie si le compétiteur est qualifié. Le compétiteur est qualifié s'il a réalisé dans la discipline concernée un jet au moins égale aux minima imposés. La distance minimale sera exprimée en mètres et en centimètres.
   3. Créer une fonction récursive classique **selection** qui crée, à partir d'une **liste** donnée, la liste des éléments qui respecte un **predicat** donné en respectant l'ordre initial des éléments dans la liste. On ne conservera donc que les éléments qui vérifient la condition définie par le prédicat.
   4. Créer une fonction récursive terminale **selection_terminale** qui produit la même liste dans le même ordre.
3. Trier une liste.
   1. Créer une fonction **ordre_alphabetique** qui retourne un entier négatif si un **competiteur1** est classé alphabétiquement avant un **competiteur2**. Vous tiendrez compte du fait qu'il peut y avoir des homonymes.
   2. Créer une fonction **liste_alphabetique** qui, pour une **liste** de compétiteurs donnée, crée une liste de compétiteurs triés par ordre alphabétique.
4. Créer la liste, dans l'ordre alphabétique, des compétiteurs qui ont réalisé les minima en javelot. Les minima sont de 80 m et 10 cm.