Université Paris 6
Licence d'informatique

Module de Programmation
ANNÉE 2000-2001

Travaux Dirigés n° 2

Exercice 1 Évaluez les expressions ci-dessous. On supposera qu'à chaque question correspond une nouvelle session CAML interactive.

1

```
let l = 3::[];;
l=[3];;
1::l;;
[]@l@[4];;
l::[1];;
l@["34"; "2"];;
```

2

```
let xxx = (3,"45",true);;
let zzz = (3, ("45", true)) and ttt = ((3, "45"), true);;
xxx = zzz;;
xxx = ttt;;
let f (x,y,z) = (x, string_of_int x ^ y, not z);;
f xxx;;
```

3

```
type t = {x : int; y : int};;
let rec1 = {x = 3; y = 1};;
let rec2 = {y = 1; x = 3};;
rec1 = rec2;;
type t' = {xxx : int; yyy : t};;
let rec3 = {xxx = 1; yyy = {x = 4; y = 5}};;
rec3.yyy.x;;
```

4

```
type C = C1 | C2 | C3 of int * float * string list;;
let x = C3(1, 1.2, ["aaa"; "bbb"]);;
let f c = match c with
| C1 -> 1
| C3(x,y,z) -> x
| C2 -> 0;;
f x;;
```

Exercice 2 *(Listes)* Nous allons nous familiariser ici avec le type des listes. Définissez et donner le type des fonctions suivantes :

1 `head_liste`, `tail_liste` et `nth_liste` qui retournent respectivement le premier, le dernier et le `nth_liste`-ième élément d'une liste.

2 `rang_pair_liste` qui extrait la liste des éléments de rang pair d'une liste.

3 `somme_liste` qui calcule la somme des éléments d'une liste.

4 `for_all` qui détermine si un prédicat est vérifié par tous les éléments d'une liste.

5 `reverse_liste` qui renverse une liste.

6 Écrivez la fonction `merge` qui fusionne deux listes triées. En déduire une fonction de tri par dichotomie.

Exercice 3 *(Types sommes et listes)* Un étudiant s'il est présent lors d'une composition a une note qui est de type `float`. À la fin de l'année, il a passé 5 compositions et est censé avoir 5 notes d'importance égales. Compte tenu qu'une absence à une de ces compositions est éliminatoire, on se propose de dresser la liste des reçus parmi une liste d'étudiants. Pour cela on découpe le problème de la façon suivante :

1 *(Types)* Définissez le type `note` et le type `étudiant` qui rassemble les 5 notes d'un étudiant.

2 *(É)*crire une fonction `somme_notes` qui calcule la somme d'une liste de notes.

3 *(É)*crire une fonction `nombre_notes` qui calcule le nombre de notes effectives (étudiant présent) dans une liste de notes.

4 *(E)*n utilisant ces deux fonctions, écrire la fonction `reçu` qui pour un étudiant donné indique s'il est reçu ou non.

5 *(E)*n déduire une fonction `liste_des_reçus` qui dresse la liste des reçus parmi une liste d'étudiants inscrits.

6 *(É)*crire une fonction `somme_nombre_notes` qui calcule ces deux informations en un seul parcours de la liste. En déduire une nouvelle version de la fonction `reçu`.

Exercice 4 *(Représentation d'un jeu de cartes)*
Dans cet exercice, on se propose de modéliser en Caml le jeu de 32 cartes servant à la belote.

1 Une carte est définie par sa couleur (carreau, cœur, pique, trèfle) et sa hauteur (as, roi, dame, valet, dix, neuf, huit, sept). Définissez les types énumérés `couleur` et `hauteur` à cet effet.

2 Définissez le type `carte` comme un enregistrement contenant les informations de couleur et de hauteur.

3 Définissez les valeurs représentant l'as de trèfle, la dame de cœur et le 8 de carreau.

4 En utilisant le type `list` prédéfini en Caml, définissez le type `main` représentant un *ensemble* (une liste) de cartes.

5 En quoi le type `main` ne répond t-il pas entièrement à la question ? Donnez un contre-exemple.

Exercice 5 *(Le comptage des points)*
En fin de partie, les cartes sont évaluées de la façon suivante : quelle que soit la couleur de l'atout, l'as vaut 11 points, le roi vaut 4 points, la dame vaut 3 points, le dix vaut 10 points, le huit et le sept ne valent aucun point. Dans la couleur de l'atout, le valet vaut 20 points et le 9 vaut 14 points, mais dans les autres couleurs, le valet ne vaut que 2 points et le 9 ne vaut rien du tout.

1 Écrivez une fonction `valeur` donnant la valeur d'une carte. Commencez par réfléchir au type que devra posséder cette fonction.

2 Utilisez la fonction précédente pour définir une fonction `valeur_main` donnant la valeur d'une liste de cartes.

Exercice 6 *(Le rangement des cartes)*
Dans cet exercice, on souhaite ranger une `main` par couleur et par valeur (on rappelle que la valeur d'une carte dépend de sa couleur). L'ordre souhaité pour les couleurs est celui du bridge (Trèfle, Carreau, Cœur, Pique).

1 Écrivez un prédicat `inf` qui retourne le booléen `true` si son premier argument est une carte devant être placée avant la carte donnée en second argument.

2 Écrivez une fonction `insere` qui introduit une carte à l'endroit souhaité dans une main déjà convenablement rangée.

3 Écrivez une fonction `range` qui classe une main.