**UNIVERSITÉ D'ORLÉANS** **DEUG MIAS**
**IF 3.01 - Informatique** **Année 2001-2002**

*Feuille d'exercices n° 5*

---

**Exercice 1** Gestion d'une compagnie d'avions

```
# type avion = {reference : string;
                mes       : int;
                tpsvol    : int;
                capacite  : int;
                loue      : bool};;
type avion = {
  reference : string;
  mes : int;
  tpsvol : int;
  capacite : int;
  loue : bool;
}
```

Étant donnée la référence d'un avion, ainsi qu'une liste d'avions, on peut afficher ses caractéristiques :

```
# let rec caracteristiques flotte ref = match flotte with
          []   -> failwith "Avion non trouve !"
        | a::m -> if (a.reference = ref) then a
                                          else caracteristiques m ref;;
val caracteristiques : string -> avion list -> avion = <fun>
```

Ceci suffit, puisque dans ce cas, CAML va afficher toutes les donnees sur l'avion en question.

```
# let rec cherche flotte n = match flotte with
          []   -> []
        | a::m -> if ((not a.loue) && (a.capacite >= n))
                           then a.reference::(cherche m n)
                           else cherche m n;;
val cherche : avion list -> int -> string list = <fun>
```

Pour la fonction de mise à jour :

```
# let rec maj flotte heures loue ref = match flotte with
                  []   -> []
                | a::r -> if (a.reference = ref)
                          then   {reference = a.reference;
                                  mes       = a.mes;
                                  tpsvol    = (a.tpsvol + heures);
                                  capacite  = a.capacite;
                                  loue      = loue}::r
                          else a::(maj r heures loue ref);;
val maj : avion list -> int -> bool -> string -> avion list = <fun>
```

Cherchons les avions qui ont plus de $h$ heures de vol :

```
# let rec test_tpsvol flotte h = match flotte with
          []   -> []
        | a::m -> if (a.tpsvol >= h) then (a.reference)::(test_tpsvol m h)
                                     else test_tpsvol m h;;
val test_tpsvol : avion list -> int -> string list = <fun>
```

Recherchons maintenant l'avions le plus vieux :

```
# let rec vieux flotte = match flotte with
          []   -> failwith "erreur"
        | [a]  -> (a.reference, a.mes)
        | a::m -> let (ref,annee) = vieux m in
                if (a.mes <= annee) then (a.reference, a.mes)
                                     else (ref, annee);;
val vieux : avion list -> string * int = <fun>
```

Nous allons maintenant chercher une liste de $m$ avions possédant exactement $p$ places et non loués. La fonction renverra une erreur si une telle liste n'existe pas. Par contre, s'il existe plus que $m$ avions correspondants à la demande, on n'en renverra que $m$ :

```
# let rec sous_flotte flotte p m = match m with
          0 -> []
        | n -> match flotte with
                  []   -> failwith "erreur"
                | a::r -> if ((not a.loue) && (a.capacite = p))
                          then a::(sous_flotte r p (n-1))
                          else  sous_flotte r p n;;
val sous_flotte : avion list -> int -> int -> avion list = <fun>
```

Maintenant, si on veut louer $m$ avions ayant $p$ places, il faut les supprimer de la flotte :

```
# let rec est_loue a l = match l with
          []   -> failwith "erreur"
        | b::r -> if (a=b) then {b with loue=true}::r
                           else b::(est_loue a r);;
val est_loue : avion -> avion list -> avion list = <fun>
# let rec sont_loues m l = match m with
          []   -> l
        | a::r -> sont_loues r (est_loue a l);;
val sont_loues : avion list -> avion list -> avion list = <fun>
# let location flotte p m = sont_loues (sous_flotte flotte p m) flotte;;
val location : avion list -> int -> int -> avion list = <fun>
```

```
# let annee_act = 2001;
val annee_act : int = 2001
# let rec supprime a l = match l with
          []   -> failwith "erreur"
        | b::r -> if (a=b) then r else b::(supprime a r);;
val supprime : 'a -> 'a list -> 'a list = <fun>
# let rec suppr_liste m l = match m with
          []   -> l
        | a::r -> suppr_liste r (supprime a l);;
val suppr_liste : 'a list -> 'a list -> 'a list = <fun>
# let rec suppr_vieux flotte age = let rec trop_vieux f a = match f with
                  []   -> []
                | b::m -> if (b.mes >= (annee_act - a))
                      then trop_vieux m a
                      else b::(trop_vieux m a) in
        suppr_liste (trop_vieux flotte age) flotte;;
val suppr_vieux : avion list -> int -> avion list = <fun>
```

Pour ramener tous les avions de $p$ places en réparation, c'est à dire les supprimer (provisoirement), il suffit d'adapter la fonction précédente.
Voyons maintenant comment trier. L'a encore, c'est assez simple, si on reprend la fonction de tri (*quicksort* par exemple) déjà utilisée dans un TD précédent :

```
# let rec separe inf a l = match l with
          []   -> ([],[])
        | b::m -> let (x,y) = separe inf a m in
                if (inf b a) then (b::x,y) else (x,b::y);;
val separe : ('a -> 'b -> bool) -> 'b -> 'a list -> 'a list * 'a list = <fun>
# let rec quicksort inf l = match l with
          []   -> []
        | b::m -> let (x,y) = separe inf b m in
                (quicksort inf x)@[b]@(quicksort inf y);;
val quicksort : ('a -> 'a -> bool) -> 'a list -> 'a list = <fun>
```

Il suffit maintenant de bien définir l'ordre sur les avions :

```
# let moins_vole a b = (a.tpsvol <= b.tpsvol);;
val moins_vole : avion -> avion -> bool = <fun>
# let tri = quicksort moins_vole;;
val tri : avion list -> avion list = <fun>
```

Bon... Reste à tester tout ça...
Nous nous intéressons à une petite compagnie ayant une dizaine d'avions. Lorsque la réponse de CAML est trop longue, je ne mettrai pas tout...

```
# let flotte = [
{reference = "AB103"; mes = 1997; tpsvol    = 15346 ;
capacite  = 159; loue     = false};
{reference = "RT275"; mes = 1993; tpsvol    = 45673 ;
capacite  = 187; loue     = true};
{reference = "XD775"; mes = 1997; tpsvol    = 17637 ;
capacite  = 159; loue     = false};
{reference = "WD654"; mes = 1991; tpsvol    = 74763 ;
capacite  = 87; loue      = false};
{reference = "VG275"; mes = 1989; tpsvol    = 85437 ;
capacite  = 159; loue     = false};
{reference = "SG375"; mes = 2000; tpsvol    = 153 ;
capacite  = 159; loue     = false};
{reference = "TD753"; mes = 1987; tpsvol    = 84673 ;
capacite  = 187; loue     = false};
{reference = "VH258"; mes = 1994; tpsvol    = 47890 ;
capacite  = 187; loue     = true};
{reference = "RD256"; mes = 1992; tpsvol    = 63478 ;
capacite  = 87; loue      = false};
{reference = "UF454"; mes = 1999; tpsvol    = 1534 ;
capacite  = 184; loue     = false}];;
val flotte : avion list =
  [{reference="AB103"; mes=1997; tpsvol=15346; capacite=159; loue=false};
   {reference="RT275"; mes=1993; tpsvol=45673; capacite=187; loue=true};
   {reference="XD775"; mes=1997; tpsvol=17637; capacite=159; loue=false};
   {reference="WD654"; mes=1991; tpsvol=74763; capacite=87; loue=false};
   {reference="VG275"; mes=1989; tpsvol=85437; capacite=159; loue=false};
   {reference="SG375"; mes=2000; tpsvol=153; capacite=159; loue=false};
   {reference="TD753"; mes=1987; tpsvol=84673; capacite=187; loue=false};
   {reference="VH258"; mes=1994; tpsvol=47890; capacite=187; loue=true};
   {reference="RD256"; mes=1992; tpsvol=63478; capacite=87; loue=false};
   {reference="UF454"; mes=1999; tpsvol=1534; capacite=184; loue=false}]
# caracteristiques flotte "VH258";;
- : avion =
{reference="VH258"; mes=1994; tpsvol=47890; capacite=187; loue=true}

# cherche flotte 100;;
- : string list = ["AB103"; "XD775"; "VG275"; "SG375"; "TD753"; "UF454"]

# maj flotte 12 false "VH258";;
- : avion list =
[ ...
 {reference="VH258"; mes=1994; tpsvol=47902; capacite=187; loue=false};
  ... ]
# test_tpsvol flotte 47900;;
- : string list = ["WD654"; "VG275"; "TD753"; "RD256"]
```

On remarque ici que la mise à jour précédente n'a pas été réellement effectuée. C'est simplement qu'il faut mémoriser le résultat :

```
# let flotte = maj flotte 12 false "VH258";;
...
# test_tpsvol flotte 47900;;
- : string list = ["WD654"; "VG275"; "TD753"; "VH258"; "RD256"]
# vieux flotte;;
- : string * int = "TD753", 1987
# let flotte = location flotte 159 4;;
val flotte : avion list =
[{reference="AB103"; mes=1997; tpsvol=15346; capacite=159; loue=true};
 {reference="RT275"; mes=1993; tpsvol=45673; capacite=187; loue=true};
 {reference="XD775"; mes=1997; tpsvol=17637; capacite=159; loue=true};
 {reference="WD654"; mes=1991; tpsvol=74763; capacite=87; loue=false};
 {reference="VG275"; mes=1989; tpsvol=85437; capacite=159; loue=true};
 {reference="SG375"; mes=2000; tpsvol=153; capacite=159; loue=true};
 {reference="TD753"; mes=1987; tpsvol=84673; capacite=187; loue=false};
 {reference="VH258"; mes=1994; tpsvol=47914; capacite=187; loue=false};
 {reference="RD256"; mes=1992; tpsvol=63478; capacite=87; loue=false};
 {reference="UF454"; mes=1999; tpsvol=1534; capacite=184; loue=false}]
# let flotte = suppr_vieux flotte 10;;
val flotte : avion list =
[{reference="AB103"; mes=1997; tpsvol=15346; capacite=159; loue=true};
 {reference="RT275"; mes=1993; tpsvol=45673; capacite=187; loue=true};
 {reference="XD775"; mes=1997; tpsvol=17637; capacite=159; loue=true};
 {reference="WD654"; mes=1991; tpsvol=74763; capacite=87; loue=false};
 {reference="SG375"; mes=2000; tpsvol=153; capacite=159; loue=true};
 {reference="VH258"; mes=1994; tpsvol=47914; capacite=187; loue=false};
 {reference="RD256"; mes=1992; tpsvol=63478; capacite=87; loue=false};
 {reference="UF454"; mes=1999; tpsvol=1534; capacite=184; loue=false}]
# tri flotte;;
- : avion list =
[{reference="SG375"; mes=2000; tpsvol=153; capacite=159; loue=true};
 {reference="UF454"; mes=1999; tpsvol=1534; capacite=184; loue=false};
 {reference="AB103"; mes=1997; tpsvol=15346; capacite=159; loue=true};
 {reference="XD775"; mes=1997; tpsvol=17637; capacite=159; loue=true};
 {reference="RT275"; mes=1993; tpsvol=45673; capacite=187; loue=true};
 {reference="VH258"; mes=1994; tpsvol=47914; capacite=187; loue=false};
 {reference="RD256"; mes=1992; tpsvol=63478; capacite=87; loue=false};
 {reference="WD654"; mes=1991; tpsvol=74763; capacite=87; loue=false}]
```