

Contrôle continu n°1

Durée 2h – Aucun document autorisé

- **Vous devez respecter le type des fonctions tel qu'il est donné par l'énoncé ou les exemples.**
- **Le compte rendu devra contenir les fonctions, leur type, des jeux de tests représentatifs et des commentaires. Chacun de ces points sera évalué.**

Le but de ce problème est d'écrire un ensemble de fonctions permettant de réaliser la gestion d'un magasin de location de skis.

Une date est donnée par un couple d'entiers (jour, mois). Pour plus de simplicité, on limitera la saison aux mois de janvier à mars, donc du premier janvier noté (1,1) (jour 1 de l'année) au 31 mars noté (31,3), jour 90 de l'année. On suppose que février ne compte que 28 jours.

- On caractérise une réservation par le nom du skieur et les dates de début et de fin de la réservation. On définit à cet effet le type :

```
# type reservation={debut:int*int;fin:int*int;skieur:string};;
```

- A toute paire de ski on associe sa taille et sa liste ordonnée dans le temps de réservations. On définit donc le type :

```
# type ski={taille:int;location:reservation list};;
```

- Le stock du magasin de location de ski est alors géré à l'aide d'une liste de skis:

```
#let stock =  
  [{taille=46;location=[{debut=4,1;fin=12,1;skieur="dupond"};  
{debut=14,1;fin=29,1;skieur="durand"}]};  
  {taille=42;location=[{debut=6,1;fin=21,1;skieur="martin"};  
{debut=22,2;fin=10,3;skieur="dupuis"};{debut=11,3;fin=18,3;skieur="brice"}]};  
  {taille=42;location=  
    [{debut=4,1;fin=12,1;skieur="dusse"};{debut=14,2;fin=28,2;skieur="napoleon"}]} ];;  
  
#stock : ski list = ...
```

I. Gestion du stock

1. Écrire une fonction **long** calculant la longueur d'une liste de type quelconque et l'utiliser pour calculer le nombre de skis dans l'objet **stock**.
2. Écrire une fonction **date** définie par filtrage permettant de connaître le numéro dans l'année associé à une date donnée.

```
Ex :      #date(4,1);;                #date(12,3);;
      - : int = 4                      - : int = 71
```

3. Écrire une fonctionnelle **nbjours**, curryfiée, donnant le nombre de jours entre deux dates données, ces dates étant comptées.

Ex : #nbjours (10,1)(14,1);
- : int = 5 (* 5 dates : 10,11,12,13 et 14 janvier *)

4. Écrire une fonction **nbjoursloc**, donnant le nombre total de jours de location d'une paire de skis donné.

```
Ex : let ski1={taille = 42 ; location =  [{debut = 6, 1; fin = 21, 1; skieur = "martin"};
                                           {debut = 22, 1; fin = 30, 1; skieur = "dupuis"};
                                           {debut = 1, 2; fin = 14, 2; skieur = "brice"}]};;
```

```
#ski1 : ski =...
      #nbjoursloc(ski1);
      -: int = 39
```

5. Écrire une fonctionnelle **inclus**, curryfiée, à résultat booléen qui étant données trois dates indique si la troisième est incluse entre les deux premières.

```
Ex :      #inclus (11,1) (2,2) (31,1);;
          - : bool = true
```

6. Écrire une fonctionnelle **dispo**, curryfiée, qui étant donnés une date et un objet de type `ski`, indique si cette paire de ski est disponible à la date donnée.

```
Ex :
    #dispo (15,2) skil;;
    - : bool = true
    #dispo (15,1) skil;;
    - : bool = false
```

7. Écrire une fonctionnelle **disponible**, curryfiée, qui étant donnés une date, une taille et le stock de skis, indique si une paire de ski de cette taille est disponible à la date donnée.

Ex :		
#disponible (15,2) 42 stock;;		#disponible (15,1) 42 stock;;
- : bool = true		- : bool = false

```
#disponible (15,2) 43 stock;;
- : bool = false
```

II. Gestion des skieurs et des factures

Le magasin ne reculant devant aucun sacrifice, accorde une réduction de 10 % aux groupes de plus de trois personnes. Ceci l'oblige à distinguer les loueurs de skis « individuels » des loueurs « en groupe ».

On définit donc le type somme :

```
# type loueur = Indiv of string | Groupe of string list ;;
```

Le fichier client est alors constitué d'une liste de loueurs :

```
#let clients=[Indiv "dupond";Indiv "durand";Indiv "martin";  
              Groupe ["brice";"napoleon";"dusse"]];;
```

```
#clients : loueur list = [Indiv "dupond"; Indiv "durand"; Indiv "martin";  
                        Groupe ["brice"; "napoleon"; "dusse"]]
```

1. Écrire une fonction **nbskieurs** calculant le nombre total de skieurs dans le fichier **clients**.

```
Ex :      #nbskieur(clients);;  
        - : int = 6
```

2. Le tarif des locations de ski est de 5€ par jour pour les réservations de moins d'une semaine et de 4€ par jour à partir du 8ème jour inclus. Écrire une fonction **tarif** qui à une réservation donnée associe le tarif de cette réservation.

```
Ex: #tarif {debut = 6, 1; fin = 21, 1; skieur = "martin"};;  
    - : int = 71
```

3. Écrire une fonctionnelle **prix**, curryfiée, qui à un nom de personne associe en fonction du stock du magasin le prix total à payer par cette personne (hors réduction accordée aux groupes). Attention : une même personne peut avoir réservé plusieurs paires de skis différents et à des dates différentes.

```
Ex : #prix "martin" stock;;          #prix "brice" stock;;  
    - : int = 71                    - : int = 63
```

4. Écrire une fonctionnelle **facture**, curryfiée, qui en fonction du stock du magasin associe à un loueur (individuel ou groupe) le prix total de la facture à payer. Pour un groupe, on facturera la somme des prix de tout le groupe et on accordera une réduction de 10 %.

```
Ex : #facture stock (Indiv "martin") ;;  
    - : int = 71  
    #facture stock (Groupe ["brice"; "napoleon"; "dusse"]) ;;  
    - : int = 157
```