

Programmation fonctionnelle (PF1)
Examen session 1 – année 2023-2024 – Durée : 2 heures

Les exercices sont indépendants. Vous pouvez écrire des fonctions auxiliaires si nécessaire.

N'oubliez pas d'écrire à chaque fois la signature de chaque fonction.

Les surveillants ne pourront pas répondre à vos questions.

Exercice 1 (5 points)

On souhaite manipuler des figures géométriques qui sont soit des cercles, soit des carrés, via le type suivant :

```
data Figure = Cercle Double | Carré Double
```

où le paramètre de type Double est soit le rayon du cercle, soit le côté du carré. Ainsi, (Cercle 2) est un cercle de rayon 2, et (Carré 3) est un carré de côté 3.

Question 1 : écrivez la fonction `surface` qui permet de connaître la surface d'une Figure donnée. Par exemple :

```
surface (Carré 2)  
→ 4.0  
surface (Cercle 3)  
→ 28.27
```

(NB : on dispose d'une variable prédéfinie `pi` qui vaut 3.14159265).

Question 2 : en réutilisant si nécessaire cette fonction `surface`, écrivez une fonction `surfaceTotale` qui prend en paramètres une liste de figures et renvoie la somme des surfaces de toutes les figures. Par exemple :

```
surfaceTotale [Carré 2, Cercle 3]  
→ 32.27
```

Question 3 : en réutilisant si nécessaire les fonctions définies dans les questions précédentes, écrivez une fonction `lePetit` qui renvoie la figure la plus petite (celle ayant la plus petite surface) d'une liste. On partira du principe que la liste donnée en paramètre n'est pas vide. Par exemple, on aura :

```
lePetit [Carré 4, Cercle 3]  
→ Carré 4
```

Exercice 2 (5 points)

Écrivez une fonction `compresser` prenant en paramètres une chaîne de caractères et renvoyant la liste de paires contenant chaque caractère ainsi que le nombre de fois consécutives où ce caractère apparaît. Exemple :

```
compresser "aaaaaabbbaabcc"  
→ [('a',6), ('b',3), ('a',2), ('b',1), ('c',2)]
```

Exercice 3 (5 points)

On dispose d'une liste infinie de nombres pseudo-aléatoires. On veut transformer cette liste de manière à ce qu'elle ne contienne que des nombres entre 1 et 6, pour simuler des lancers de dés.

Question 1 : Écrivez une fonction `queDesDés` prenant en paramètre une liste infinie d'entiers positifs ou nuls, et renvoyant une liste infinie d'entiers compris entre 1 et 6.

Question 2 : On dispose désormais d'une liste infinie de valeurs comprises en 1 et 6. Écrivez une fonction `nbLancers` qui renvoie le nombre de fois où il a fallu lancer le dé avant d'obtenir la valeur 6. Par exemple, si la liste `dés` vaut `[1, 2, 3, 1, 6, 5, 6, 6, 6, 5, 4, 1, 3, 1, ...]` alors on aura :

```
nbLancers dés  
→ 5
```

Question 3 : Écrivez une fonction `nbLancers2` qui renvoie le nombre de fois où il a fallu lancer le dé avant d'obtenir deux fois de suite la valeur 6. Par exemple, si la liste `dés` vaut `[1, 2, 3, 1, 6, 5, 6, 6, 6, 5, 4, 1, 3, 1, ...]` alors on aura :

```
nbLancers2 dés  
→ 8
```

Exercice 4 (5 points)

Je dispose d'une bibliothèque contenant des livres que je n'ai pas tous lus. Je décide de la modéliser sous la forme d'une liste de triplets de type `(String, Int, Bool)`, chaque triplet représentant le titre du livre, son nombre de pages, et un booléen indiquant si je l'ai lu. Par exemple, le livre « Blaze » de 340 pages que je n'ai pas encore lu sera représenté sous la forme : `("Blaze", 340, False)`.

Question 1 : Écrivez une fonction `pagesLues` qui prend en paramètre une liste de livres et renvoie le nombre total de pages dans les livres que j'ai lus. Vous utiliserez la syntaxe des listes en compréhension. Exemple :

```
pagesLues [("Blaze", 340, False), ("Plasmas", 170, True), ("Spectres", 526, True)]  
→ 696
```

Question 2 : Écrivez une fonction `enfinLu` qui prend en paramètre un titre et une liste de livres et qui note que le livre dont le titre a été passé en paramètres a été lu. Si le livre n'existe pas dans la liste, ou s'il était déjà noté comme lu, on ne change rien. L'ordre de la liste en sortie n'a pas d'importance (vous pouvez changer l'ordre). Exemples :

```
enfinLu "Blaze" [("Blaze", 340, False), ("Plasmas", 170, True)]  
→ [("Blaze", 340, True), ("Plasmas", 170, True)]
```

```
enfinLu "Plasmas" [("Blaze", 340, False), ("Plasmas", 170, True)]  
→ [("Blaze", 340, False), ("Plasmas", 170, True)]
```

```
enfinLu "Spectres" [("Blaze", 340, False), ("Plasmas", 170, True)]  
→ [("Blaze", 340, False), ("Plasmas", 170, True)]
```

Question 3 : Écrivez une fonction `choisisPourMoi` qui prend en paramètres une liste de livres, et qui renvoie le titre d'un livre non lu de la liste. Attention, il est possible que la liste ne contienne aucun livre non lu. Il faut gérer ce cas.