

PROGRAMMATION FONCTIONNELLE

EXAMEN 2ÈME ANNÉE

Cette épreuve est à rendre sous forme électronique *et* papier. Sur papier, commentez vos œuvres. Sur ordinateur, ne mettez pas votre nom dans les fichiers. Vous placerez vos réponses dans un répertoire dont le nom est votre numéro de copie.

Une réponse sans commentaires/documentation ou sans code vaut 0. Une fonction qui ne marche pas *ou qui ne donne pas le même résultat que l'exemple*, même si elle compile, vaut 0 si vous n'expliquez pas dans quelles circonstances elle ne marche pas ou pourquoi. De même, précisez vos choix de conception. Si vos fonctions sont récursives terminales, mentionnez-le.

Les seuls documents autorisés sont vos cours et TDs et le *manuel* de OCaml. Vous avez 2h.

Exercice 1. (environ 2 points) Écrire une fonction qui, à partir d'une fonction f et d'un entier n , calcule la somme $\sum_{i=0}^{i=n} f(i)$.

Exemple

```
# let f n = n + 1;;
val f : int -> int = <fun>

# somme f 5;;
- : int = 21
```

Exercice 2. (environ 5 points) Si $(u_n)_{n \in \mathbf{N}}$ et $(v_n)_{n \in \mathbf{N}}$ sont deux suites, le produit de convolution de u et v , noté $u \otimes v$, est la suite définie par $\forall n \in \mathbf{N}, (u \otimes v)_n = \sum_{i=0}^{i=n} u(i) \cdot v(n-i)$.

Écrire une fonction qui, à partir de deux fonctions u et v et d'un entier n , calcule $(u \otimes v)_n$.

Exemple

```
# let u n = n + 1;;
val u : int -> int = <fun>

# let v n = 2 * n;;
val v : int -> int = <fun>

# convolution u v 5;;
- : int = 70
```

Qu'est-ce que `convolution u v` ? Quelle est la complexité de cette fonction, en nombre de multiplications $u(i) \cdot v(n-i)$ et en fonction de n ?

Exercice 3. (environ 3 points) Écrire une fonction qui enlève d'une liste un élément sur 2

Exemple :

```
# enlever ['a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h'];;
- : char list = ['a'; 'c'; 'e'; 'g']
```

Exercice 4. (environ 9 points) Écrire une fonction qui mélange *aléatoirement* une liste. Vous pourrez par exemple utiliser la fonction `Random.bool` qui sert à tirer à pile ou face ou `Random.int` qui sert à choisir aléatoirement un entier.

Exemple :

```
# melanger ['a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h'];;
- : char list = ['b'; 'c'; 'f'; 'a'; 'g'; 'h']
```

Exercice 5. (environ 6 points) Écrire une fonction qui transforme un tableau en son symétrique, sans modifier le tableau original.

Exemple :

```
# let t = [| 'a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h' |];;
val t : char array = [| 'a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h' |]

# retourner t
- : char array = [| 'h'; 'g'; 'f'; 'e'; 'd'; 'c'; 'b'; 'a' |]

# t;;
- : char array = [| 'a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h' |]
```

Exercice 6. (environ 5 points) Écrire une fonction qui détermine si une liste est un palindrome, c'est-à-dire si elle est égale à son symétrique.

Exemple :

```
# palindrome ['a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h'];;
- : bool = false

# palindrome ['a'; 'b'; 'c'; 'd'; 'c'; 'b'; 'a'];;
- : bool = true
```

Exercice 7. (environ 10 points) Écrire une fonction qui transforme un titre entièrement en majuscules en un titre à l'anglo-saxonne, c'est-à-dire où la première lettre de chaque mot est en majuscules.

Exemple :

```
# a_l_anglo_saxonne "HARRY POTTER 7: HARRY POTTER AND THE DEATHLY HALLOWS";;
- : string = "Harry Potter 7: Harry Potter And The Deathly Hallows"
```

—

Bon courage.