Examen de
Programmation fonctionnelle
Durée : 2 h
Tous documents autorisés

Ce sujet comporte 2 pages.

1 Bases

1. Typer (sans détailler) les valeurs suivantes et évaluer (en détaillant) `x` :

```
#let p = fun a b p -> p a b;;

#let f = fun p -> p (fun x y -> x);;

#let s = fun p -> p (fun x y -> y);;

#let x = f (s (p 1 (p "deux" '3')));;
```

2. Donner un exemple de valeur pour le type suivant :

```
#type t = X of (int * string) list array * float;;
```

3. Donner l'itérateur pour le type suivant :

```
#type x = I of int | F of float | S of string * x;;
```

et utiliser le pour calculer l'espace mémoire utilisé par une valeur de type `x` sachant que :
- la représentation d'un constructeur coûte 1 (quelque soit le type associé) ;
- un entier coûte 4 et un flotant 8 ;
- une chaîne de caractère de longueur n coûte $n + 4$ (utiliser la fonction `Strig.length` qui retourne la longueur d'une chaîne).

2 Arithmétique en précision arbitraire

Pour s'affranchir de la limitation des entiers standards (type `int`), on décide de représenter les grands entiers par une liste de *chiffres* dans une base donnée. Pour faciliter les manipulations, on représente les nombres en commençant par les poids faibles. Par exemple, en base 10, le nombre 1729 est représenté par la liste `[9;2;7;1]`. Toutes les calculs seront faits pour une base donnée `base` définie globalement.

```
#let base = 10;;
val base : int = 10
```

1. Définir la fonction d'addition de deux grands entiers.

```
#somme [9;2;7;1] [2;4];;
 - : int list = [1; 7; 7; 1]
```

2. Définir les fonctions de conversion entre entier et grand entier. NB : la conversion en *petit* entier est éventuellement incorrecte.

```
#grand_entier 91;;
- : int list = [1; 9]

#petit_entier [0;0;0;2];;
- : int = 2000

#petit_entier [0;1;2;3;4;5;6;7;8;9];;
- : int = -860875030
```

3. Définir la fonction d'affichage sur la sortie standard d'un grand entier (en base 10).

```
#afficher (grand_entier 1984);;
1984- : unit = ()
```

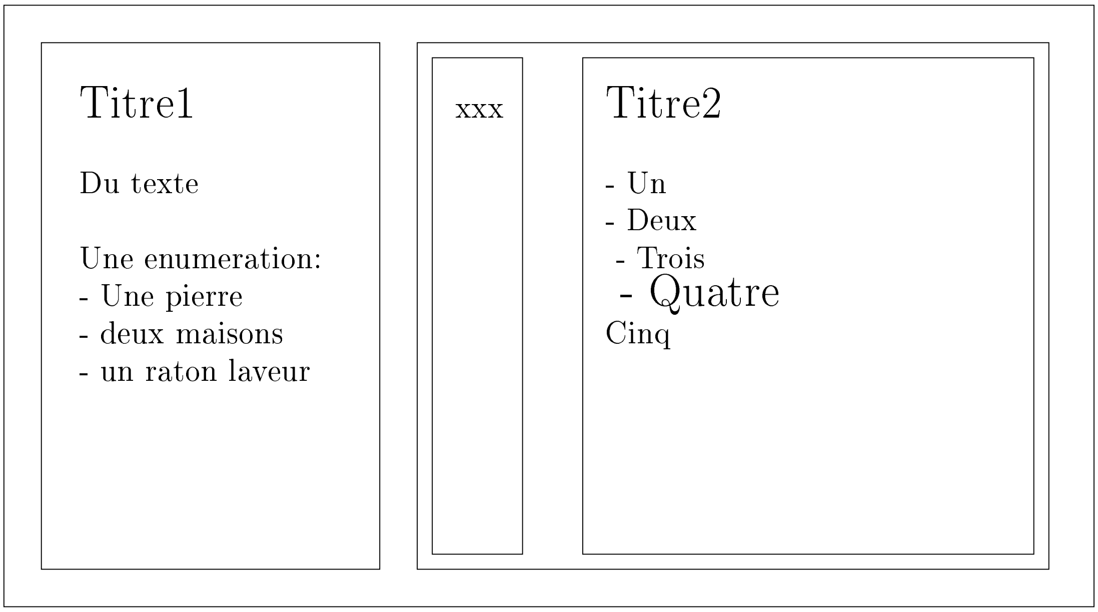

FIG. 1 – *Un exemple de document structuré*

4. Définir la fonction de multiplication de deux grands entiers.

```
#produit (grand_entier 123) (grand_entier 456);;
- : int list = [8; 8; 0; 6; 5]
```

5. Définir la fonction qui calcule en grand entier, la factorielle d'un petit entier.

```
#afficher (fact 100);;
93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518 28
625369792082722375825118521091686400000000000000000000000- : unit = ()
```

3 Document structuré

On considère un document structuré (typiquement une page HTML). Au niveau supérieur, un document est constitués de cadres. Un cadre est constitué de

- deux cadres ou
- d'une collection des éléments suivant :
  - du texte en vrac ;
  - un titre ;
  - une énumération d´éléments quelconques.

Dans l'exemple de la figure 1, la page contient un cadre contenant 2 pages. La première page est constituée d'un titre, de texte en vrac et d'une énumération (à la Prévert). La seconde page contient deux pages dont la seconde contient un titre et une énumération dont le troisième élément est une énumération dont le second élément est un titre (`Quatre`).

1. Définir le ou les types permettant de représenter un document structuré et écrire avec ce type la valeur représentant le document de la figure 1.
2. Définir la fonction qui *met à plat* un document structuré en retournant une liste de chaîne de caractères.