Langages fonctionnels (INFO406)
Examen – Session 1

L2 informatique, CMI Informatique, Maths Info, Peip 2 Info
Vendredi 28 février 2025 – 8 h 30 à 10 h 30 (+ 1/3 temps)

Aucun dispositif électronique autorisé
(calculette, ordinateur, téléphone portable, montre connectée, ...)

Remarques :

- *Le sujet comporte 5 pages.*
- *Les exercices sont indépendants : vous les traiterez dans l'ordre que vous souhaitez.*
- *Le barème est donné à titre indicatif.*

Consignes :

- *Précisez systématiquement le **profil** des fonctions que vous avez à écrire dans ce qui suit.*
- *Les fonctions récursives que vous écrivez doivent engendrer des **processus récursifs.***
- *Vous introduirez toute fonction jugée utile à la définition des fonctions qui vous sont demandées. Dans le cadre d'une **analyse fonctionnelle descendante**, vous donnerez les profils des fonctions introduites, vous indiquerez ce qu'elles font et les utiliserez comme si elles existaient. Vous les définirez ensuite.*
- *Des fonctions de base pour le traitement des chaînes de caractères vous sont fournies (cf. ci-dessous). **Utilisez-les.***
- *N'hésitez pas à passer une question si vous ne parvenez pas à écrire la fonction qui y est demandée. Vous pourrez malgré cela utiliser cette fonction dans la suite de votre devoir.*

Fonctions de base pour le traitement des chaînes de caractères.

```
open String;;

let est_vide = function ch -> (ch = "");;
(** string -> bool
teste si son paramètre est une chaîne vide *)

let prem_lettre = function ch -> sub ch 0 1;;
(** string -> string
renvoie la première lettre d'une chaîne de caractères *)

let reste = function ch -> sub ch 1 ((length ch) - 1);;
(** string -> string
renvoie la chaîne de caractères passée en paramètre privée de sa première lettre *)
```

Rappel : La fonction ***int_of_string*** traduit une chaîne de caractères représentant un entier en le nombre correspondant ; la fonction ***string_of_int*** fait le contraire.

Exercice 1 – Où l'on incite à la mobilité douce (16 pts)

Dans le cadre de la lutte contre le réchauffement climatique, l'État a mis en place pour ses agents le « forfait mobilité douce » (FMD). Ce dispositif se traduit par une prime versée aux agents qui font le choix d'un mode de transport alternatif et durable (vélo, covoiturage, …). Le montant de la prime dépend du nombre de jours où la mobilité douce est effectivement utilisée. Elle s'élève à :

- 100 euros pour 30 à 59 jours de mobilité douce,
- 200 euros pour 60 à 99 jours,
- 300 euros pour 100 jours et plus.

Nous allons dans ce qui suit écrire les fonctions permettant de calculer la prime pour un agent donné.

Partie 1 : Première version de la fonction *calculer_prime* (V0)

L'agent doit déclarer, pour une année civile donnée (du 1er janvier au 31 décembre), son *vecteur de déplacement*. Ce vecteur est une chaîne de 365 caractères (ou 366 si l'année est bissextile), correspondant chacun au mode de transport emprunté le jour *j* (vélo, covoiturage, véhicule à faible émission, véhicule thermique, bus, à pied, …). Certains modes comptent pour le calcul du nombre de jours de mobilité douce, d'autres pas, soit parce que cela ne correspond pas à de la mobilité douce (c'est le cas d'un véhicule thermique), soit parce d'autres dispositifs de prise en charge existent (cas des abonnements de bus) soit encore parce qu'il n'y a pas de frais liés à ce mode de transport (cas des déplacements à pied)

Le tableau ci-dessous dresse la liste des modes de transport avec, pour chacun, le code qui le représente et le nombre de jours de mobilité douce attribué.

Mode de transport	Pas de déplacement	Vélo	Covoit. avec véhicule thermique	Covoit. avec véhicule non thermique	Bus	À Pied	Véhicule thermique	Véhicule faible émission (élec./hydro / ...)
représenté par	'0'	'V'	'c'	'C'	'B'	'P'	't'	'E'
nb	0	1	1	1	0	0	0	1

Ainsi par exemple la chaîne « 00000CCEEV00...» (on suppose qu'il y a 365 ou 366 caractères) peut être considérée comme un vecteur de déplacement pour une année donnée :

- les 5 premiers jours de l'année (soit du 1er/01 au 5/01), la personne ne s'est pas déplacée ;
- les 6ème et 7ème jours (6/01 et 7/01), elle a fait du covoiturage avec un véhicule non thermique ;
- les 8ème et 9ème jours (8/01 et 9/01), elle est venue seule avec un véhicule à faible émission ;
- le 10ème, elle est venue en vélo ;
- les 2 suivants (11/01 et 12/01), elle ne s'est pas déplacée ;
- et ainsi de suite, jusqu'au 365 ou 366ème jour (le dernier jour de l'année : 31/12).

Le vecteur de déplacement est utilisé pour déterminer le nombre de jours de mobilité douce. Pour cela, il est tout d'abord transformé en un vecteur « mobilité douce » : c'est une chaîne de caractères ne contenant que des 0 et des 1, chacun remplaçant un des caractères du vecteur de déplacement selon que le mode de transport est éligible au FMD, ou pas. Sur l'exemple précédent, cela donne, « 000001111100...». Il suffit alors de faire la somme des chiffres apparaissant dans ce vecteur pour avoir le nombre de jours de mobilité douce, et ainsi le montant de la prime.

1.1 – Ecrire la fonction ***somme*** qui fait la somme des chiffres apparaissant dans une chaîne de caractères correspondant à un vecteur « mobilité douce »

Exemple : # somme "0000011111000011000" ;;
- : int = 7

1.2 – Écrire la fonction ***est_eligible*** qui vérifie si un caractère correspond à un mode de transport éligible au FMD.

Exemple : # est_eligible "V" ;;
- : bool = True

1.3 – Écrire la fonction ***vecteur_mobilite_douce*** qui transforme un vecteur de déplacement en un vecteur « mobilité douce » contenant uniquement des '0' et des '1' (1 si mobilité douce OK).

Exemple :

vecteur_mobilite_douce "00000CCEEV00" ;;
- : string = "000001111100"

1.4 – Écrire la fonction ***calculer_primeV0*** qui prend en paramètre un vecteur de déplacement et renvoie le montant de la prime FMD accordée. Il s'agit là d'une première version de cette fonction, d'où le V0 dans son nom. Vous en construirez d'autres dans ce qui suit.

Partie 2

En préalable à tout calcul, la validité du vecteur « mobilité douce » est vérifiée en regard :

- des jours de fermeture officielle de l'établissement dans lequel l'agent est affecté :
- des congés et jours de télétravail déclarés par l'agent : c'est la même chose ; il doit y avoir un « 0 » dans le vecteur « mobilité douce » en regard d'un jour de télétravail ou de vacances.

Ces 2 calendriers sont représentés de la même manière que le vecteur « mobilité douce », à savoir une chaîne de 365 ou 366 caractères, chacun étant soit un « 0 » soit un « 1 ». Le « 1 » correspond à un jour de fermeture dans le premier calendrier, à un jour de télétravail ou de congé dans le second.

La validation consiste à vérifier qu'une mobilité n'a pas été déclarée un jour où l'établissement est fermé, un jour où l'agent est en télétravail ou encore un jour où il est en vacances. A ces jours doivent donc correspondre des « 0 » dans le vecteur « mobilité douce ». Si ce n'est pas le cas, le vecteur « mobilité douce » est corrigé : le « 1 » non OK devient un « 0 ».

Exemple (*) :

- soient vmd un vecteur de mobilité douce, vferm un vecteur de fermeture de l'établissement
- vmd = "001001111100..."
- vferm = "111110000000..." (établissement fermé du 1er au 5 janvier)
- le vecteur « mobilité douce » corrigé est alors "000001111100…".

Le "1" correspondant au 3 janvier dans le vmd initial (donc déclaré comme un jour où un déplacement mobilité douce a eu lieu) a été remplacé par un "0". vferm indique en effet que le 3 janvier était un jour de fermeture : il ne peut de fait pas y avoir de déplacement comptabilisé ce jour-là.

2.1 – Soit la fonction ***combine*** suivante :

```
let rec combine =
  function f ->
    function ch1 ->
      function ch2 ->
  if est_vide ch1 || est_vide ch2 then ""
  else
    let c = f (prem_lettre ch1) (prem_lettre ch2) in
    c ^ combine f (reste ch1) (reste ch2);;
```

2.1.1 – Quel est le profil de cette fonction ? Justifiez votre réponse. Vous pourrez utilisez pour cela les équations aux types.
2.1.2 – Donnez 2 exemples d'application de cette fonction.
2.1.3 – Expliquez ce qu'elle fait.

2.2 – Utilisez la fonction *combine* pour écrire la fonction ***corriger_vecteur*** qui prend en paramètres :

- un vecteur mobilité douce
- un vecteur « correctif » (fermeture de l'établissement, jours de télétravail, jours de congés, …)

et renvoie le vecteur mobilité douce corrigé (cf. exemple (*) fourni ci-dessus). Vous aurez recours à une application partielle.

2.3 – Utilisez les fonctions précédentes (y compris *calculer_primeV0,* pour limiter ce que vous aurez à réécrire) pour écrire la fonction ***calculer_primeV1*** qui calcule le montant de la prime FMD accordée en tenant compte cette fois-ci des correctifs à appliquer. Cette fonction prend en paramètres :

- un vecteur de déplacement (ex : "00000CCEEV00BBttB00"),
- un vecteur « fermeture »,
- un vecteur « congés et télétravail »,

et fait les corrections nécessaires avant de calculer le montant de la prime.

Partie 3 – Décomposition fonctionnelle et abstraction

On cherche maintenant à savoir quelle est la part de mobilités « actives » (déplacement en vélo ou à pied) dans l'ensemble des modes de transport auquel un agent a recours.

On procède ici de la même manière que pour les mobilités douces :

- on considère le vecteur de déplacement que l'on va, cette fois-ci, transformer en un vecteur « mobilité active » en sélectionnant les mobilités actives (« 1 » si mobilité active, « 0 » sinon)
- on applique les correctifs pour les périodes de fermeture, les congés et le télétravail ;
- on peut alors compter le nombre de jours de mobilité active.

En comparant ce nombre au nombre total de jours de mobilités déclarés, il est alors possible de calculer des pourcentages.

3.1 – Généraliser le comportement de la fonction *vecteur_mobilite_douce* pour obtenir une fonction ***vecteur_mobilite*** qui permettra d'obtenir le vecteur « 0-1 » de n'importe quel type de mobilité choisie (ex : le vecteur «0-1 » des mobilités actives, le vecteur « 0-1 » des déplacements thermiques, …). Vous indiquerez le profil de cette fonction en justifiant votre réponse.

3.2 – Ecrire la fonction ***part_mobilite_active*** qui prend en paramètres :

- un vecteur de déplacement (ex : "00000CCEEV00BBttB00"),

- un vecteur « fermeture »,
- un vecteur « congés et télétravail »,

et calcule le pourcentage de mobilité_active par rapport à l'ensemble des modes de déplacement utilisés. Vous réutiliserez bien entendu les fonctions écrites précédemment et en introduirez d'autres si besoin, dans le cadre d'une démarche de décomposition fonctionnelle.

3.3 – Comment peut-on généraliser la fonction *part_mobilite_active* pour calculer plus seulement le pourcentage de mobilité active par rapport à l'ensemble des modes de déplacement utilisés mais un pourcentage de n'importe quelle mobilité par rapport à n'importe quelle autre ? (ex : part des mobilités actives par rapport à l'ensemble des mobilités douces, part du co-voiturage, …). Expliquez.

Exercice 2 – Où l'on joue avec les mots (7 pts dont 3 de bonus)

On cherche à déterminer des sous-séquences de caractères communes à des chaînes de caractères données. Soit **C** une chaîne de caractères notée C_1, C_2, C_3, …, C_N où les Ci représentent chacun un caractère. Une sous-séquence de **C** est une chaîne de caractères Ci1, Ci2, …, Cik, les Cij étant des caractères de **C** pris dans leur ordre d'apparition dans **C**.

Par exemple, si **C** = "bateau" alors "ba", "bat", "bt" ou encore "aau" sont des sous-séquences de **C** (il y en a d'autres, bien sûr).

Si l'on considère deux chaînes de caractères ch1 et ch2, les sous-séquences communes à ch1 et ch2 sont les sous-séquences de ch1 qui sont aussi sous-séquences de ch2.

Par exemple :

ch1	ch2	Des exemples de sous-séquences communes (liste non exhaustive)
"bateau"	"banane"	"ba", "bae", "baa", "aa", "ae"
"bateau"	"abattue"	"batu", "bate", "bau", "atu", "au", ...
"ananas"	"banane"	"anan", "aa", "aan", ...
"ananas"	"maracas"	"aaas" , "aaa" , "as" , ...

1 – Ecrire la fonction ***verif_sseq_commune*** qui vérifie qu'une chaîne de caractères donnée est une sous-séquence commune à deux chaînes données.

2 – Bonus - Ecrire la fonction ***plus_longue_sseq_commune*** qui calcule la taille de la plus longue sous-séquence commune à deux chaînes de caractères données.
A noter : ce n'est pas (du tout !!) une bonne idée que de chercher à construire les sous-séquences.