Devoir surveillé n°1 Option info MPSI (Durée 2h)

L'usage de la calculatrice est interdit. Les raisonnements présentés devront être soigneusement justifiés et détaillés, quelques points seront dédiés à la présentation, l'orthographe et la propreté de votre copie. En particulier, il vous est demandé de souligner les résultats obtenus À LA RÈGLE ! Il n'est pas nécessaire de répondre à l'ensemble des questions pour avoir une bonne note. Si vous ne parvenez pas à résoudre une question, vous pouvez admettre le résultat dans la suite de l'énoncé.Lisez bien tout le sujet avant de commencer et identifiez les parties plus simples pour vous, et commencez par ces parties.

Exercice 1

Une preuve par induction

1) Rappeler le théorème du principe d'induction.

2) On considère la fonction suivante :

```
let rec fonction1 (a,b) = match (a,b) with
  | (n,m) when n=0 -> 1
  | (n,m) when n = m -> 1
  | (n,m)-> fonction1 (n,m-1) + fonction1 (n-1,m-1)
```

a) Quelle est le type de cette fonction ?

b) Cette fonction est-elle curryfiée ? Si non, donner une version curryfiée de cette fonction.

c) Calculer fonction1(2,4) et fonction1(1,3). Que calcule cette fonction ?

d) On tente de prouver par induction que cette fonction se termine bien. On pose la proposition suivante : P(a,b) : "La fonction *fonction1* se termine avec les arguments a et b.". Pour quel sous ensemble A de l'ensemble des couples d'arguments possibles a-t-on P(a,b) automatiquement vraie ?

e) Poser 2 fonctions φ_1 et φ_2 tel que pour tout $x \in E \backslash A$, on ait

$$P(\varphi_1(x)) \text{ est vrai et } P(\varphi_2(x)) \text{ est vrai } \Rightarrow P(x) \text{ est vrai}$$

Où E est l'ensemble des arguments possibles de la fonction.

f) En admettant que le théorème d'induction s'adapte avec 2 fonctions dans l'hypothèse d'hérédité, conclure par le principé d'induction (quel hypothèse reste-t-il à vérifier ?).

g) Quels problème cette fonction récursive présente-elle d'un point de vue mémoire ? Sans rentrer dans les détails (ni du code), comment aurait-on pu adapter le code de cette fonction ?

Exercice 2

Du codage récursif

1) Ecrire une fonction "bezout" de type int -> int -> int * int qui calcule des coefficients de Bezout de deux entiers.

2) Ecrivez une fonction récursive "somchiffres" de type int -> int qui calcule la somme des chiffres de l'entier passé en argument.

3) Ecrire une fonction "add" : int -> int -> int , récursive terminale, et effectuant la somme de ses deux arguments (supposés > 0). Les seules opérations autorisées sont l'addition ou la soustraction de 1.

4) Ecrire une fonction "mul" : int -> int -> int , utilisant une fonction auxiliaire récursive terminale, et effectuant le produit de ses deux arguments (supposés positifs). Les seules opérations autorisées sont l'addition, et la multiplication par 2 (ou la division par 2 d'un nombre pair).

5) Le jeu de fizzbuzz consiste à partant de n = 1 demander à chaque personne autour d'une table de citer le nombre entier suivant en respectant les règles suivantes données par ordre de priorité :

(a) Si n est un multiple de 35, la personne doit dire fizzbuzz

(b) Si n est un multiple de 5, la personne doit dire buzz

(c) Si n est un multiple de 7 ou contient un 7 dans son écriture décimale, la personne doit dire fizz

(d) Sinon, la personne doit simplement dire le nombre n.

L'objectif de cet qyestion est de construire un programme donnant la liste des mots à prononcer par les joueurs. Pour cela, nous allons décomposer le problème en sous-problèmes (on précisera le type de chacune des fonctions créées) :

a) Définir la fonctions suivante : la fonction "charlistofstring" qui convertit une chaîne de caractères en la liste de ses caractères (on pourra s'aider de l'aide CamL).

b) Définir la fonctions suivante : la fonction "contient c s" qui teste si le caractère c apparaît dans la chaîne de caractère s.

c) Définir la fonction "fizzbuzz" qui écrit à l´écran la valeur qui doit être prononcée en fonction de n.

d) En déduire la fonction "List-fizzbuzz" n qui affiche la séquence des fizzbuzz k pour k allant de 1 à n.

Exercice 3

Un peu de listes Dans cette partie, on représente un ensemble fini par la liste triée (dans l'ordre croissant) sans doublon de ses éléments. On identifiera alors l'ensemble et la liste le représentant. Par exemple l'ensemble $\{3, 5, 2, 4, 8\}$ sera identifié par la liste $[2, 3, 4, 5, 8]$.

1) Donné la liste associée à l'ensemble $\{1, 8, 5, 9, 7\}$

2) Ecrire une fonction calculant le cardinal d'un ensemble (son nombre d'éléments).

3) Ecrire une fonction renvoyant l'élément minimal d'un ensemble non vide. ´

4) Ecrire une fonction renvoyant l'élément maximal d'un ensemble non vide. ´

5) Ecrire une fonction testant l'appartenance d'un élément à un ensemble. ´

6) Ecrire une fonction "comprehension" prenant en argument un prédicat p et un ensemble l et renvoyant l'ensemble des éléments de l vérifiant p.

7) Ecrire une fonction insertion prenant en argument un élément x et un ensemble l et renvoyant l'ensemble contenant x et les éléments de l.

8) Ecrire une fonction renvoyant l'union de deux ensembles. Chaque liste ne devra être parcourue qu'au plus une fois.

9) Ecrire sur le même principe une fonction renvoyant l'intersection de deux ensembles, une fonction renvoyant la différence ensembliste de deux ensembles et une fonction testant l'inclusion d'un ensemble dans un autre.