Examen final

Exercice 1 **(7 points = 2+2+(1,5+1,5)) (Fonctions récursives)**

1) Utiliser la règle de récursion pour montrer que la fonction $mult3$ est PR :
$mult3 = \lambda xyz.x * y * z$
2) Utiliser la règle de composition pour montrer que la fonction SommeCarre est PR :
SommeCarre = $\lambda xy.x^2+y^2$
3)
a) Montrer par récurrence sur l'entier k≥1, que la fonction $S1_k$ est PR, pour tout k :

$$S1_k = \boldsymbol{\lambda x_1 \ldots x_k}.\sum_{i=1}^{k} x_i = \boldsymbol{\lambda x_1 \ldots x_k}.x_1 + x_2 + \cdots + x_k$$

b) En déduire, par récurrence sur l'entier k≥1, que la fonction $S2_k$ est PR, pour tout k :

$$S2_k = \boldsymbol{\lambda x_1 \ldots x_k}.\sum_{i=1}^{k} S1_i(x_1,\ldots,x_i) = \boldsymbol{\lambda x_1 \ldots x_k}.x_1 + (x_1 + x_2) + \cdots + (x_1 + \cdots + x_k)$$

Remarque : Les fonctions suivantes, vues en cours, sont considérées PR :
$\oplus$, $\otimes$, Moins, Fact, Abs, Exp, Sg, $\overline{Sg}$, Pred, mod, div, les fonctions constantes d'arité 0, les fonctions constantes d'arité 1.

Exercice 2 **(6 points = 2+1+1+2) (Machines de Turing)**

On considère la machine de Turing $MT = (S, E, I)$, avec :
$S = \{0, 1, *\}$, $E = \{q_0, q_1, q_2, q_3, q_4, q_f\}$,
$I = \{$

$(1)\ \boldsymbol{q_0 1\ Dq_0}$,	$(5)\ \boldsymbol{q_1 0\ Dq_f}$,	$(9)\ \boldsymbol{q_3 *\ 0q_2}$,
$(2)\ \boldsymbol{q_0 *\ Dq_0}$,	$(6)\ \boldsymbol{q_1 *\ 0q_2}$,	$(10)\ \boldsymbol{q_3 0\ Dq_4}$,
$(3)\ \boldsymbol{q_0 0\ Gq_1}$,	$(7)\ \boldsymbol{q_2 0\ Gq_3}$,	$(11)\ \boldsymbol{q_4 0\ Dq_4}$,
$(4)\ \boldsymbol{q_1 1\ Gq_1}$,	$(8)\ \boldsymbol{q_3 1\ 0q_2}$,	$(12)\ \boldsymbol{q_4 1\ 1q_f}\}$

1- Dérouler cette machine pour : **a)** x=2, **b)** $(x_1,x_2) = (1,2)$, **c)** $(x_1,x_2,x_3) = (1,2,3)$.
2- Déduire toutes les fonctions calculables par cette machine.
3- Ecrire la machine de Turing MT_Sg qui calcule la fonction $Sg = \boldsymbol{\lambda x}.\begin{cases}0 & si\ x = 0\\ 1 & sinon\end{cases}$
4- En déduire la machine de Turing MT_f qui calcule la fonction $f = \boldsymbol{\lambda xy}.Sg(x) * y$

Exercice 3 **(7 points = 1+2+(1,5+1,5+1)) (Langage CAML)**

1- Ecrire une fonction **nbElem** calculant le nombre d'éléments d'une liste.
2- En utilisant la fonction nbElem, écrire une fonction récursive, notée **recupElem x n**, qui calcule la sous-liste de x composée des n premiers éléments de x, avec 0 ≤ n ≤ nbElm x.
Exemples : a) recupElem [1; 2; 6; 4] 3 = [1; 2; 6], **b)** recupElem [1; 2; 6; 4] 0 = []
3- On considère la fonction suivante, notée **verif**, utilisant les 2 fonctions précédentes :

```
let rec verif x y = if nbElem x < nbElem y then false
                    else if recupElem x (nbElem y)=y then true else verif (tl x) y ;;
```

a) Dérouler cette fonction pour : **i)** x = [1; 4;1;5;6] et y = [1;5;6], **ii)** x = [1; 4;1;5;6] et y = [1;6]
b) Que fait la fonction **verif** ?

Bon courage