

<i>Paradigmes des langages de programmation</i>			
<i>Section</i>	1 ^{ère} année du Master Recherche en Informatique	<i>Année universitaire</i>	2015/2016
<i>Enseignant responsable</i>	Mme. Sajeh ZAIRI CHIH	<i>Établissement</i>	Faculté des Sciences de Tunis
<i>Date : 04/05/2016 - Durée : 1H</i>		<i>Documents non autorisés</i>	

Devoir surveillé

Exercice 1 (16pt)

Soit le programme déclarative suivant :

```

declare Pascal, Droite, Gauche, Add in

  fun {Droite L}
    case L
      of H|R then H|{Droite R}
      else [0]
    end
  end

  fun {Gauche L}
    0|L
  end

  fun {Add L1 L2}
    case L1#L2
      of H1|R1 # H2|R2 then L=H1+H2|{Add L1 L2}
      else nil
    end
  end

  fun {Pascalle N}
    if N==1 then [1]
    else
      {Add {Gauche {Pascal N-1}} {Droit {Pascal N-1}}}
    end
  end
end

```

1. Que permet de faire la fonction **Droite**. Donner le résultat de l'appel de cette fonction pour un input que vous proposez.
2. Que permet de faire la fonction **Gauche**. Donner le résultat de l'appel de cette fonction pour un input que vous proposez.
3. Que permet de faire la fonction **Add**. Donner le résultat de l'appel de cette fonction pour un input que vous proposez.
4. Que permet de faire la fonction **Pascal**. Donner le résultat de l'appel de cette fonction pour N=5.
5. Quels sont les environnements contextuels des fonctions **Pascal**, **Droite**, **Gauche** et **Add**.
6. Donner la traduction de la fonction **Pascal** en langage noyau.
7. Est-ce que les fonctions **Pascal**, **Droite** et **Add** sont récursives terminales.
8. Rappeler les différentes techniques qu'on peut exploiter pour faire de la récursivité terminale en utilisant des listes.
9. Utiliser une des techniques précédentes pour rendre récursives terminales les fonctions de la question 7 qui ne le sont pas.
10. Si dans la question 7, vous avez trouvé qu'il y a des fonctions récursives terminales expliquez quelle technique a été utilisée pour chacune d'elles.

Exercice 2 (4pt)

Ecrire une fonction LengthL qui permet de calculer le nombre d'éléments élémentaires d'une liste.

Exemple : { LengthL [1 [a b c] s [[8 9] 5 h [1 5 9]]] } = 12

Pour écrire cette fonction vous pouvez utiliser la fonction suivante qui permet de tester si l'élément X est une constante ou une liste :

```

fun {IsCons X}
  case X
    of _|_ then false
    else true
  end
end

```

Les copies claires sont bien appréciées