

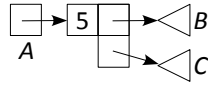
**TD d'algorithmique INF220 – TD5 – Arbres**

**Exercice 1 – Codage d'un arbre**

Que renvoie `écritureParenthesee(T)`, si  $T$  est l'arbre représenté à droite ?

**Exercice 2 – Comptages dans les arbres**

• Q1. On va compter les feuilles d'un arbre d'entiers. Pour cela on définit l'algorithme récursif `nombreFeuilles` qui prend en entrée un arbre d'entiers  $A$ , et renvoie le nombre de feuilles de l'arbre. Si l'arbre  $A$  a deux fils qui sont les arbres  $B$  et  $C$ , que `nombreFeuilles(B) = 4` et que



**Algorithme écritureParenthesee**

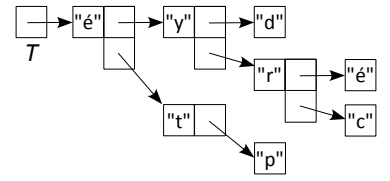
Entrée : arbre de chaînes de caractères  $A$

Type de sortie : chaîne de caractères

Variables : chaîne de caractères `resultat`, entier  $i$

Début

1. Si `estFeuille(A)` alors :
  2. `resultat`  $\leftarrow$  `racine(A)`
  4. Sinon :
  5. `resultat`  $\leftarrow$  "("
  6. Pour  $i$  de 1 à `longueur(enfants(A))` faire :
  7. `resultat`  $\leftarrow$  `Concatene(resultat,Concatene(écritureParenthesee(Case(enfants(A),i)),";"))`
  8. Fin pour
  9. `resultat`  $\leftarrow$  `Concatene(Concatene(resultat,")"),racine(A))`
  10. Fin si
  11. renvoyer `resultat`
- Fin



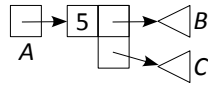
**TD d'algorithmique INF220 – TD5 – Arbres**

**Exercice 1 – Codage d'un arbre**

Que renvoie `écritureParenthesee(T)`, si  $T$  est l'arbre représenté à droite ?

**Exercice 2 – Comptages dans les arbres**

• Q1. On va compter les feuilles d'un arbre d'entiers. Pour cela on définit l'algorithme récursif `nombreFeuilles` qui prend en entrée un arbre d'entiers  $A$ , et renvoie le nombre de feuilles de l'arbre. Si l'arbre  $A$  a deux fils qui sont les arbres  $B$  et  $C$ , que `nombreFeuilles(B) = 4` et que



**Algorithme écritureParenthesee**

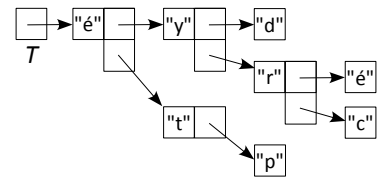
Entrée : arbre de chaînes de caractères  $A$

Type de sortie : chaîne de caractères

Variables : chaîne de caractères `resultat`, entier  $i$

Début

1. Si `estFeuille(A)` alors :
  2. `resultat`  $\leftarrow$  `racine(A)`
  4. Sinon :
  5. `resultat`  $\leftarrow$  "("
  6. Pour  $i$  de 1 à `longueur(enfants(A))` faire :
  7. `resultat`  $\leftarrow$  `Concatene(resultat,Concatene(écritureParenthesee(Case(enfants(A),i)),";"))`
  8. Fin pour
  9. `resultat`  $\leftarrow$  `Concatene(Concatene(resultat,")"),racine(A))`
  10. Fin si
  11. renvoyer `resultat`
- Fin



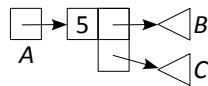
**TD d'algorithmique INF220 – TD5 – Arbres**

**Exercice 1 – Codage d'un arbre**

Que renvoie `écritureParenthesee(T)`, si  $T$  est l'arbre représenté à droite ?

**Exercice 2 – Comptages dans les arbres**

• Q1. On va compter les feuilles d'un arbre d'entiers. Pour cela on définit l'algorithme récursif `nombreFeuilles` qui prend en entrée un arbre d'entiers  $A$ , et renvoie le nombre de feuilles de l'arbre. Si l'arbre  $A$  a deux fils qui sont les arbres  $B$  et  $C$ , que `nombreFeuilles(B) = 4` et que



**Algorithme écritureParenthesee**

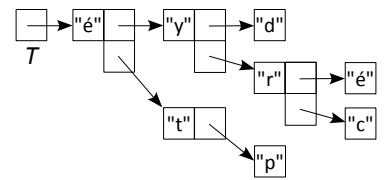
Entrée : arbre de chaînes de caractères  $A$

Type de sortie : chaîne de caractères

Variables : chaîne de caractères `resultat`, entier  $i$

Début

1. Si `estFeuille(A)` alors :
  2. `resultat`  $\leftarrow$  `racine(A)`
  4. Sinon :
  5. `resultat`  $\leftarrow$  "("
  6. Pour  $i$  de 1 à `longueur(enfants(A))` faire :
  7. `resultat`  $\leftarrow$  `Concatene(resultat,Concatene(écritureParenthesee(Case(enfants(A),i)),";"))`
  8. Fin pour
  9. `resultat`  $\leftarrow$  `Concatene(Concatene(resultat,")"),racine(A))`
  10. Fin si
  11. renvoyer `resultat`
- Fin



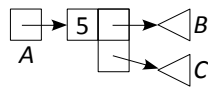
**TD d'algorithmique INF220 – TD5 – Arbres**

**Exercice 1 – Codage d'un arbre**

Que renvoie `écritureParenthesee(T)`, si  $T$  est l'arbre représenté à droite ?

**Exercice 2 – Comptages dans les arbres**

• Q1. On va compter les feuilles d'un arbre d'entiers. Pour cela on définit l'algorithme récursif `nombreFeuilles` qui prend en entrée un arbre d'entiers  $A$ , et renvoie le nombre de feuilles de l'arbre. Si l'arbre  $A$  a deux fils qui sont les arbres  $B$  et  $C$ , que `nombreFeuilles(B) = 4` et que



**Algorithme écritureParenthesee**

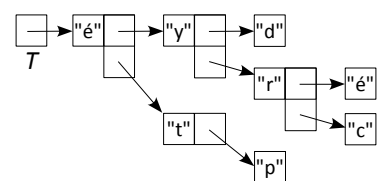
Entrée : arbre de chaînes de caractères  $A$

Type de sortie : chaîne de caractères

Variables : chaîne de caractères `resultat`, entier  $i$

Début

1. Si `estFeuille(A)` alors :
  2. `resultat`  $\leftarrow$  `racine(A)`
  4. Sinon :
  5. `resultat`  $\leftarrow$  "("
  6. Pour  $i$  de 1 à `longueur(enfants(A))` faire :
  7. `resultat`  $\leftarrow$  `Concatene(resultat,Concatene(écritureParenthesee(Case(enfants(A),i)),";"))`
  8. Fin pour
  9. `resultat`  $\leftarrow$  `Concatene(Concatene(resultat,")"),racine(A))`
  10. Fin si
  11. renvoyer `resultat`
- Fin



**nombreFeuilles(C) = 6**, quelle valeur renvoie **nombreFeuilles(A)** ? Quelle est l'initialisation de l'algorithme **nombreFeuilles** ?

- Q2. Déduisez-en l'écriture de l'algorithme récursif **nombreFeuilles**.
- Q3. En faisant de légères modifications de l'algorithme **nombreFeuilles**, écrivez un algorithme récursif **nombreNoeuds** qui calcule le nombre de nœuds de l'arbre d'entiers fourni en entrée (nœuds = toutes les cases de l'arbre qui contiennent un entier).
- Q4. En appelant les algorithmes **nombreFeuilles** et **nombreNoeuds**, écrivez un algorithme **nombreNoeudsInternes** qui renvoie le nombre de nœuds qui ne sont pas des feuilles dans l'arbre en entrée.

### Exercice 3 – Recherche dans un arbre

- Q1. Écrivez un algorithme récursif **recherche** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie VRAI si  $i$  se trouve dans une des cases de  $A$ , et FAUX sinon.
- Q2. Écrivez un algorithme récursif **profondeur** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie la profondeur de l'entier  $i$  (0 s'il est à la racine, 1 s'il est dans un fils de la racine, 2 s'il est dans un de leurs fils, etc.) s'il se trouve dans l'arbre, et -1 sinon.

**nombreFeuilles(C) = 6**, quelle valeur renvoie **nombreFeuilles(A)** ? Quelle est l'initialisation de l'algorithme **nombreFeuilles** ?

- Q2. Déduisez-en l'écriture de l'algorithme récursif **nombreFeuilles**.
- Q3. En faisant de légères modifications de l'algorithme **nombreFeuilles**, écrivez un algorithme récursif **nombreNoeuds** qui calcule le nombre de nœuds de l'arbre d'entiers fourni en entrée (nœuds = toutes les cases de l'arbre qui contiennent un entier).
- Q4. En appelant les algorithmes **nombreFeuilles** et **nombreNoeuds**, écrivez un algorithme **nombreNoeudsInternes** qui renvoie le nombre de nœuds qui ne sont pas des feuilles dans l'arbre en entrée.

### Exercice 3 – Recherche dans un arbre

- Q1. Écrivez un algorithme récursif **recherche** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie VRAI si  $i$  se trouve dans une des cases de  $A$ , et FAUX sinon.
- Q2. Écrivez un algorithme récursif **profondeur** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie la profondeur de l'entier  $i$  (0 s'il est à la racine, 1 s'il est dans un fils de la racine, 2 s'il est dans un de leurs fils, etc.) s'il se trouve dans l'arbre, et -1 sinon.

**nombreFeuilles(C) = 6**, quelle valeur renvoie **nombreFeuilles(A)** ? Quelle est l'initialisation de l'algorithme **nombreFeuilles** ?

- Q2. Déduisez-en l'écriture de l'algorithme récursif **nombreFeuilles**.
- Q3. En faisant de légères modifications de l'algorithme **nombreFeuilles**, écrivez un algorithme récursif **nombreNoeuds** qui calcule le nombre de nœuds de l'arbre d'entiers fourni en entrée (nœuds = toutes les cases de l'arbre qui contiennent un entier).
- Q4. En appelant les algorithmes **nombreFeuilles** et **nombreNoeuds**, écrivez un algorithme **nombreNoeudsInternes** qui renvoie le nombre de nœuds qui ne sont pas des feuilles dans l'arbre en entrée.

### Exercice 3 – Recherche dans un arbre

- Q1. Écrivez un algorithme récursif **recherche** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie VRAI si  $i$  se trouve dans une des cases de  $A$ , et FAUX sinon.
- Q2. Écrivez un algorithme récursif **profondeur** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie la profondeur de l'entier  $i$  (0 s'il est à la racine, 1 s'il est dans un fils de la racine, 2 s'il est dans un de leurs fils, etc.) s'il se trouve dans l'arbre, et -1 sinon.

**nombreFeuilles(C) = 6**, quelle valeur renvoie **nombreFeuilles(A)** ? Quelle est l'initialisation de l'algorithme **nombreFeuilles** ?

- Q2. Déduisez-en l'écriture de l'algorithme récursif **nombreFeuilles**.
- Q3. En faisant de légères modifications de l'algorithme **nombreFeuilles**, écrivez un algorithme récursif **nombreNoeuds** qui calcule le nombre de nœuds de l'arbre d'entiers fourni en entrée (nœuds = toutes les cases de l'arbre qui contiennent un entier).
- Q4. En appelant les algorithmes **nombreFeuilles** et **nombreNoeuds**, écrivez un algorithme **nombreNoeudsInternes** qui renvoie le nombre de nœuds qui ne sont pas des feuilles dans l'arbre en entrée.

### Exercice 3 – Recherche dans un arbre

- Q1. Écrivez un algorithme récursif **recherche** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie VRAI si  $i$  se trouve dans une des cases de  $A$ , et FAUX sinon.
- Q2. Écrivez un algorithme récursif **profondeur** qui prend en entrée un entier  $i$  et un arbre d'entiers  $A$ , et renvoie la profondeur de l'entier  $i$  (0 s'il est à la racine, 1 s'il est dans un fils de la racine, 2 s'il est dans un de leurs fils, etc.) s'il se trouve dans l'arbre, et -1 sinon.