

TD 3: Algorithmique fonctionnelle

Récurtivité

Nga Nguyen - Stefan Bornhofen - Peio Loubière

1 Division entière

1.1 Quotient

Écrire une fonction récursive qui calcule le quotient de la division entière de deux nombres strictement positifs (sans utiliser l'opérateur div).

```
1 Fonction quotient(a, b : Entier) : Entier  
2 Variables  
3   res : Entier  
4 Début  
5   Si (b > a) Alors  
6     res ← 0  
7   Sinon  
8     res ← 1 + quotient(a-b,b)  
9   FinSi  
10  retourner res  
11 Fin
```

1.2 Reste

Écrire une fonction récursive qui calcule le reste de la division entière de deux nombres strictement positifs (sans utiliser l'opérateur mod).

```

1 Fonction reste(a, b : Entier) : Entier
2 Variables
3   res : Entier
4 Début
5   Si (b > a) Alors
6     res ← a
7   Sinon
8     res ← reste(a-b, b)
9   FinSi
10  retourner res
11 Fin

```

2 Binôme $C(n,p)$

On sait que les coefficients du binôme $C(n,p)$, utilisés dans le calcul de $(a+b)^n$, sont définis par : $C(n,p) = \frac{n!}{p!(n-p)!}$ et qu'ils vérifient la propriété : $C(n,p) = C(n-1,p-1) + C(n-1,p)$.
Exemple : $(a+b)^3 = C(3,0) * a^3 + C(3,1) * a^2b + C(3,2) * ab^2 + C(3,3)b^3$

Écrire une fonction récursive qui calcule $C(n,p)$.

```

1 Fonction Binome(n, p : Entier) : Entier
2 Début
3   Si  $0 \leq p$  et  $p \leq n$  Alors
4     Si  $n = 0$  Alors
5       retourner 1
6     Sinon
7       Si  $p = 0$  Alors
8         retourner 1
9       Sinon
10        retourner  $C(n-1,p-1) + C(n-1,p)$ 
11      FinSi
12    FinSi
13  Sinon
14    retourner 0
15  FinSi
16 Fin

```

3 Chaîne de caractères

3.1 Palindrome

Un palindrome est une chaîne de caractères qui se lit de gauche à droite ou de droite à gauche, comme " radar ", " elle ", " été ", " esope reste ici et se repose ", si l'on fait abstraction des blancs. Écrire un algorithme récursif qui permet de vérifier si une chaîne de caractères est un palindrome ou non.

```
1 Fonction Palindrome(s : Chaîne) : Booléen
2 Variables l : Entier
3 Début
4   l ← longueur(s)
5   Si (l ≤ 1) Alors
6     retourner vrai
7   Sinon
8     retourner (extrait(s,1,1) = extrait(s,l,1)) et
9       Palindrome(extrait(s,2,l-2))
10  FinSi
11  Fin
```

3.2 Miroir

La fonction miroir donne l'inverse d'une chaîne de caractères, c'est-à-dire la chaîne formée des mêmes caractères mais dans l'ordre inverse, par exemple miroir("séminaire") = "erianimés". Écrire un algorithme récursif qui permet de réaliser cette fonction.

```
1 Fonction Miroir(s : Chaîne) : Chaîne
2 Variables l : Entier
3 Début
4   l ← longueur(s)
5   Si (l ≤ 1) Alors
6     retourner s
7   Sinon
8     retourner extrait(s,l,1) & Miroir(extrait(s,1,l-1))
9   FinSi
10  Fin
```

4 Puissance

Soit une fonction récursive **puissance** qui prend en paramètre un réel n et un entier k et renvoie le calcul de n^k .

- Question 1 :** Écrire la fonction récursive **puissance** qui calcule n^k de façon naïve en considérant que $n^k = n * n^{k-1}$ et que $n^0 = 1$.
- Question 2 :** Transformer la version précédente en récursivité terminale.
- Question 3 :** Il existe une autre méthode de calcul de n^k plus efficace : la méthode par dichotomie. Elle repose sur le fait que si k est pair alors $n^k = (n * n)^{k/2}$ et $n^k = n * n^{k-1}$ sinon.
Écrire la fonction récursive **puissance-dic** qui calcule n^k par dichotomie.
- Question 4 :** Transformer la version précédente en récursivité terminale.

```

1  Fonction puissance(n, k : Entier) : Entier
2  Début
3      Si k = 0 Alors
4          retourner 1
5      Sinon
6          retourner n*puissance(n,k-1)
7      FinSi
8  Fin
9
10 

---



---


11
12 Fonction puissanceRT(n, k, acc : Entier) : Entier
13 Début
14     Si k = 0 Alors
15         retourner acc
16     Sinon
17         retourner puissanceRT(n,k-1,acc*n)
18     FinSi
19 Fin
20
21 Fonction puissance(n, k : Entier) : Entier
22 Début
23     retourner puissanceRT(n,k,1)
24 Fin
25
26 

---



---


27
28 Fonction puissance-dic(n, k : Entier) : Entier
29 Début
30     Si k = 0 Alors
31         retourner 1
32     SinonSi (k mod 2 = 0)
33         retourner puissance-dic(n*n,k/2)
34     Sinon
35         retourner n*puissance-dic(n,k-1)
36     FinSi
37 Fin
38
39 

---



---


40
41 Fonction puissance-dicRT(n, k, acc : Entier) : Entier
42 Début
43     Si k = 0 Alors
44         retourner acc
45     SinonSi (k mod 2 = 0)
46         retourner puissance-dicRT(n*n,k/2,acc)
47     Sinon
48         retourner puissance-dicRT(n,k-1,acc*n)
49     FinSi
50 Fin
51
52 Fonction puissance(n, k : Entier) : Entier
53 Début
54     retourner puissance-dicRT(n,k,1)
55 Fin

```