

Compléments de Programmation

Licence 1 UPEC 2023/2024

TM 4 : Récursivité

Exercice 1: Affichage de nombres Dans une classe `Exo1`

1. Écrivez une fonction récursive `public static void afficheNombresDecroissant (int n)` qui prend en paramètre un entier `n` et qui affiche les nombres de 1 à `n` par ordre décroissant.
2. Écrivez une fonction récursive `public static void afficheNombresCroissant (int n)` qui prend en paramètre un entier `n` et qui affiche les nombres de 1 à `n` par ordre croissant.

Exercice 2: Affichage d'étoiles Dans une classe `Exo2`

1. Écrivez une fonction récursive `public static void ligne (int n)` qui prend en paramètre un entier `n` et qui affiche une ligne de `n` étoiles. Par exemple, l'appel à `ligne(5)` doit afficher
2. Écrivez une fonction récursive `public static void triangle (int n)` qui prend en paramètre un entier `n` et qui affiche un triangle de `n` lignes avec sur la première ligne, une étoile, sur la deuxième, 2 étoiles, et ainsi de suite jusqu'à la dernière ligne où il y a `n` étoiles. Par exemple, l'appel à `triangle(10)` doit afficher

```
* * * * *
```

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Exercice 3: Tableaux Dans une classe `Exo3`

1. Écrivez une fonction récursive, qui prend en paramètre un tableau d'entiers `tab` un entier `x` et un entier `j` et qui renvoie `true` si `x` se trouve dans `tab` entre la case 0 et la case `j-1`. La fonction renvoie `false` sinon.
2. Écrivez une fonction récursive, qui prend en paramètre un tableau d'entiers `tab` un entier `x` et un entier `j` et qui multiplie par `x` toutes les valeurs de `tab` entre la case 0 et la case `j-1`.
3. Écrivez une fonction récursive, qui prend en paramètre un tableau d'entiers `tab` et un entier `j` et qui renvoie `true` si le tableau `tab` contient deux cases adjacentes avec la même valeur entre la case 0 et la case `j-1`. La fonction renvoie `false` sinon.
4. Écrivez une fonction récursive, qui prend en paramètre un tableau d'entiers `tab`, deux entiers `i, j` et qui renvoie `true` si le tableau est palindrome entre la case `i` et la case `j-1`. Cela veut dire que `tab[i]==tab[j-1]` et `tab[i+1]==tab[j-2]`, etc... La fonction renvoie `false` sinon. Attention à la (les ?) condition d'arrêt !
5. Écrivez une fonction récursive, qui prend en paramètre un tableau d'entiers `tab`, deux entiers `i, j` et qui renverse les valeurs du tableau entre la case `i` et la case `j-1`. Cela veut dire que `tab[i]` s'échange avec `tab[j-1]`, `tab[i+1]` s'échange avec `tab[j-2]`, etc... Attention à la (les ?) condition d'arrêt !

Exercice 4: Avec des mots ! Dans une classe `Exo4`

1. Écrivez une fonction récursive `palindrome` qui prend en paramètre une chaîne de caractères et qui renvoie `true` si le mot passé en paramètre est un palindrome, `false` sinon. Par exemple, `palindrome("radar")` et `palindrome("ressasser")` doivent renvoyer `true`, alors que `palindrome("enseignant")` doit renvoyer `false`.
Pour cela, on utilisera seulement les méthode `charAt`, `length` et `substring` de la classe `String`
2. Écrivez une fonction récursive `verlan` qui prend en paramètre une chaîne de caractères et qui renvoie la chaîne de caractères dont toutes les lettres écrites sont dans l'ordre inverse. Par exemple, `verlan("hello")` doit renvoyer `"olleh"`.
On précise que l'on peut concaténer deux chaînes de caractères en utilisant le `+` : par exemple, `"hello" + "toto"` renvoie `"hellototo"`.

Exercice 5: Algorithme d'Euclide

On rappelle que si a et b sont deux entiers positifs tels que $a \geq b \geq 0$ et $(a, b) \neq (0, 0)$, alors le pgcd de a et b vérifie :

$$\text{pgcd}(a, b) = \begin{cases} b & \text{si } a \bmod b = 0, \\ \text{pgcd}(b, a \bmod b) & \text{sinon,} \end{cases}$$

où r est le reste dans la division euclidienne de a par b , c'est à dire que $r = a \bmod b$.

Écrivez une fonction récursive `pgcd` qui prend en paramètre deux entiers positifs ou nuls a et b et qui renvoie le pgcd de a et de b en exploitant la propriété précédente.

Par exemple, `pgcd(2, 3)` renvoie 1 et `pgcd(42, 54)` renvoie 6.

Exercice 6: Exercices de compréhension, à faire sur feuille.

Pour chacun des fonctions ci-dessous, cherchez à comprendre

1. est-ce que la fonction termine toujours ?
2. Si elle termine, qu'est-ce qu'elle calcule ?

Fonction 1 :

```
public static int boucle1(int n){
    return boucle1(n-1);
}
```

Fonction 2 :

```
public static int boucle2(int n){
    if (n == 0){
        return n-1;
    }
    else{
        return boucle2(n-1);
    }
}
```

Fonction 3 :

```
public static int boucle3(int n){
    if (n <= 0){
        return 0;
    }
    else{
        return n + boucle3(n-1);
    }
}
```