TD 1 : Programmation récursive
Miage L2 - Université de Nanterre

Chaque fois que vous implémentez une fonction, pensez à bien indenter votre code, à le commenter (sans pour autant paraphraser) et à le standardiser . . .

Exercice 1 :

A partir de l'algorithme suivant, écrivez une fonction récursive retournant le même résultat, puis déroulez les deux algorithmes pour n=256 et m=64.

```
int pgcd(int n, int m)
{
  while(m)
  {
    int r=n%m;
    n=m;
    m=r;
  }
  return m;
}
```

Exercice 2 :

Considérons la fonction suivante :

```
int mystere(int n)
{
  if(n<10) return 1;
  else return 1+mystere(n/10);
}
```

1. Donnez les résultats pour les valeurs suivantes de n : 5, 20, 98, 956.
2. Que calcule la fonction `mystere` pour des valeurs entières et positives de n?

Exercice 3 :

$x^n = x \times x^{n-1}$ lorsque $n > 0$ et $x^0 = 1$.

1. Ecrivez une fonction itérative qui calcule x^n, avec un réel x et un entier n.
2. Ecrivez une fonction récursive qui calcule x^n , avec un réel x et un entier n.

Exercice 4 :

L'algorithme dit d'exponentiation rapide permet de calculer la $n$ième puissance d'un nombre plus efficacement qu'en le multipliant n fois par lui-même. Il repose sur les deux faits suivants :

- $x^{2n} = (x^2)^n$
- $x^{2n+1} = x \times x^{2n}$

1. Ecrivez une fonction récursive prenant un réel x et un entier n en argument, et renvoyant en résultat x^n, en appliquant récursivement celle des deux égalités qui correspond.

2. Supposons que l'on calcule 2^{65} à l'aide de votre fonction. Donnez la suite des appels récursifs effectués, avec leur imbrication.

3. Combien y a t-il d'appel à la fonction provoqués lors du calcul de 2^{16} ? 2^{32} ? 2^{64} ? et par la méthode précédente (exercice précédent) ?

Exercice 5 :

Les coefficients binomiaux sont définis sur les entiers naturels $n \geq k \geq 0$ par

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Une relation de récurrence valable est la suivante :

$$\binom{n+1}{k+1} = \binom{k}{n} + \binom{n}{k+1}$$

avec $\forall n \geq 0 \ \binom{n}{0} = \binom{n}{n} = 1$.

Réalisez une fonction effectuant un calcul récursif des coefficients binomiaux qui n'utilise que l'addition des entiers.

Exercice 6 :

Implémentez l'algorithme du tri fusion vu en cours.

Exercice 7 :

La recherche d'un élément dans un tableau non trié est linéaire (il faut parcourir en moyenne un nombre d'éléments proportionnel au nombre d'éléments du tableau et au pire les n éléments du tableau). Si le tableau est trié (on suppose en ordre croissant), il est possible de faire une recherche beaucoup plus rapidement en utilisant une méthode dichotomique. On coupe le tableau en deux parties de longueurs à peu près égales. On compare l'élément cherché au dernier élément de la première partie. Puis selon le résultat du test, on cherche l'élément récursivement dans la première ou la deuxième partie. On peut ainsi écrire une fonction récursive qui donne le nombre d'éléments d'un tableau trié donné, inférieurs strictement à l'élément cherché. On peut écrire une deuxième fonction qui appelle la première et renvoie l'indice de la première occurence de l'élément recherché s'il est présent dans le tableau ou -1 s'il en est absent..