

## Série n°3

**Exercice 1 :**

- Ecrire l'analyse et l'algorithme d'une fonction itérative qui permet de calculer le pgcd de deux entiers par la méthode d'Euclide :

$$\text{Pgcd}(a,0)=a$$

$$\text{Pgcd}(a,b)=\text{Pgcd}(b,a \bmod b)$$

- Donner une solution récursive

**Exercice 2 :**

Ecrire l'analyse et l'algorithme d'une fonction récursive qui permet de calculer le pgcd de deux entiers par la méthode de différence :

$$\text{Pgcd}(a,b)=a \text{ ou } b \text{ si } a=b$$

$$\text{Pgcd}(a,b)=\text{Pgcd}(a,b-a) \text{ si } b>a$$

$$\text{Pgcd}(a,b)=\text{Pgcd}(a-b,b) \text{ si } a>b$$

**Exercice 3:**

La suite de Fibonnaci est définie par:

$$\begin{cases} F(1)=1, F(2)=1 \\ F(n)=F(n-1)+F(n-2) \quad \forall n \geq 2 \end{cases}$$

Ecrire l'analyse et l'algorithme d'une fonction récursive qui permet de calculer  $F(n)$ .

**Exercice 4 :**

La combinaison de  $p$  par  $n$  est défini par

$$C(n,p) = n! / ((n-p)! * p!)$$

- 1) Calculer  $C(n-1,p-1) + C(n-1, p)$ .
- 2) Dédire une fonction récursive qui permet de calculer  $C(n,p)$  (donner analyse et algorithme)

**Exercice 5:**

Donner l'analyse et l'algorithme d'une procédure récursive qui permet d'inverser une chaîne.

**Exercice 6 :**

0) Def fn inconnue (ch : chaîne) : booléen

1) Tr ← vrai

2) P ← pos (" ", ch) // un seul espace

3) Tant que (P <> 0) et (Tr = vrai) faire

Si (ch [1] <> ch [P+1])

alors

Tr ← faux

Sinon

Ch ← sous\_chaine (ch, P+1, long (ch)-P)

P ← pos (" ", ch) // un seul espace

Finsi

Fin tant que

4) Inconnue ← Tr

5) Fin inconnue.

**Questions :**

1- Exécuter à la main la fonction inconnue pour chacune des chaînes suivantes :

Ch	Résultat de la fonction
“programme Pascal”	
“algorithme”	
“analyse avec algorithme”	

2- En déduire le rôle de cette fonction ?

3- Donner une solution récursive pour cette fonction.

**Exercice 7:**

On se propose de trier un tableau T de réels par ordre décroissant de la manière suivante :

- Pour chaque élément de T[i] on détermine le nombre d'éléments qui lui sont inférieurs (nb).
- On place T[i] dans le tableau R à la position n-nb.
- On répète ce traitement pour tous les éléments du tableau.

Exemple : T = 

2.5	1.5	0.2	3.5	4.00
-----	-----	-----	-----	------

T[1]= 2.5, nb= 2, t[1] sera placé à la position 5-2=3 dans R

R= 

		2.5		
--	--	-----	--	--

Ecrire un programme qui permet de :

- Remplir le tableau T par n réels aléatoires ( $2 \leq n \leq 30$ ).
- Trier T selon la méthode décrite ci-dessus.
- Afficher le contenu du tableau R.

**Question**

- 1- Analyser le problème en module en le décomposant en modules et déduire l'algorithme du programme principal.
- 2- Analyser chacun des modules envisagés et en déduire les algorithmes correspondants.

**Exercice 8:**

- Ecrire l'algorithme de la fonction recherche dichotomique qui permet de vérifier si un entier existe ou non dans un tableau t d'entiers trié.
- Ecrire une version récursive pour la solution.

