La récursivité est une méthode de résolution des problèmes algorithmiques qui consiste à appeler un sous-programme dans son propre corps.

Un sous-programme récursif est un module qui **s'appelle lui-même** avec d'autres paramètres jusqu'à ce qu'une condition d'arrêt soit atteinte.

Comment concevoir un sous-programme récursif ?

Dans l'écriture des programmes récursifs on retrouve généralement les étapes suivantes :

1. *Trouver une décomposition récursive du problème*
   (a) Trouver l'élément de récursivité qui permet de définir les cas plus simples (*ex.* une valeur numérique qui décroît, une taille de données qui diminue).
   (b) Exprimer la solution dans le cas général en fonction de la solution pour le cas plus simple.
2. *Trouver la condition d'arrêt de récursivité et la solution dans ce cas*
   – Vérifier que la condition d'arrêt est atteinte après un nombre fini d'appels récursifs dans tous les cas
3. *Réunir les deux étapes précédentes dans un seul programme*

Forme générale :

Entête du module avec présence obligatoire de(s) paramètre(s)
Début
« Instructions »
Si CONDITION_D'ARRET **Alors**
« instruction du point d'arrêt »
Sinon {exécution}
« Appel du module récursif avec **changement obligatoire de(s) paramètre(s)**»
FinSi
Fin

Point d'arrêt

Appel récursif du module.
Au moins un paramètre doit être changé à chaque appel sinon on risque d'entrer dans une boucle infinie.

Exercice 1 :

Soit l'algorithme de la fonction **Inconnu** suivant :

```
Fonction Inconnu  (n : entier) : entier
Début
    Si (n=0) Alors
        Retourner 0
    Sinon
        Retourner  n + Inconnu (n-1)
    Fin Si
Fin
```

Questions :

1) Quel est le résultat retourné par la fonction **Inconnu** pour **n=5**.
2) Déduire le rôle de cette fonction ?

Exercice 2 :

Soit l'algorithme de la fonction **Inconnu** suivant :

```
Fonction Inconnu ( ……) : ……..
Début
    Si Long(ch)=0 Alors
        Retourner 0
    Sinon
        Si ch[Long(ch)-1] ∈ ["0".."9"] Alors
            d←Valeur (ch[Long(ch)-1])
            Retourner d + Inconnu (Sous-chaine (ch, 0, Long (ch)-1)
        Sinon
            Retourner  Inconnu (Sous-chaine (ch, 0, Long (ch)-1)
        FinSi
    Fin Si
Fin
```

Travail demandé :

1- Compléter l'entête de la fonction **Inconnu** en complétant la déclaration des paramètres et le type de retour.
2- Dresser le tableau de déclaration des objets locaux de la fonction **Inconnu**.
3- Quel est le résultat retourné par la fonction **Inconnu** pour **ch**="**Bac22G3**".
4- Déduire le rôle de cette fonction ?

Exercice 3 :

Soit l'algorithme de la fonction **Inconnu** suivant :

```
Fonction Inconnu ( a,b : Réel) : ……..
Début
    Si a - b ≥ 0 Alors
        Retourner  a
    Sinon
        Retourner  Inconnu (b,a)
    FinSi
Fin
```

Questions :

En se référant à l'algorithme **Inconnu** et pour chacune des propositions ci-après, remplir la case par la lettre correcte :

Proposition	**1**	**2**	**3**	**4**
Réponse				

1- Le type de la fonction **Inconnu** peut être :
a) Octet **b)** Réel **c)** Entier long

2- La condition d'arrêt du traitement récursif est :
a) a – b ≥ 0 **b)** Retourner a **c)** Retourner Inconnu (b,a)

3- Pour a = 9 et b = 9, le résultat retourné par la fonction **Inconnu** est égal à :
a) 9 **b)** 12 **c)** 3

4- Le rôle de la fonction **Inconnu** est de :
a) Calculer le PPCM de a et b **b)** Calculer le PGCD de a et b **c)** Rechercher le maximum de a et b

Exercice 4 :

Soit l'algorithme de la fonction **F** suivant:

```
Fonction F (n : entier) : chaîne
Début
    i ← 2
    ch ←""
    Répéter
        Si  n mod i=0  Alors
            Ch1← Convch(i)
            Ch ←ch+ch1+ "*"
            n ←n div i
        Sinon
            i←i +1
        Finsi
    Jusqu'à (n =1)
    ch← effacer ( ch , long ( ch )-1 , long ( ch ) )
    Retourner ch
Fin
```

Questions:

1- Quel est le résultat retourné par la fonction **F** pour **n=30** et pour **n=17**
2- Déduire le rôle de cette fonction ?
3- Proposer un algorithme récursif de la fonction **F**.

Itératif vs récursif :

Ecrire l'algorithme des modules récursifs nommés :

1) **Factorielle** permettant de calculer le factoriel d'un entier $n \geq 0$, avec n !=n*(n-1)*(n-2)*……*3*2*1
2) **Palindrome** permettant de vérifier si une chaine donnée non vide est palindrome ou non.
   Exemple : radar, été, aziza…
3) **Premier** permettant de vérifier si un entier *n* positif est premier ou non.
4) **PGCD** permettant de déterminer le pgcd de deux entiers naturels *a* et *b* par la méthode d'Euclide et la différence.
5) **Occurrence** permettant de déterminer le nombre d'occurrences d'un caractère *Car* dans une chaîne *ch*.
6) **Remplissage** permettant de remplir un tableau *T* par *N* entiers positifs.
7) **Affichage** permettant d'afficher un tableau *T* par *N* entiers positifs.
8) **Dichotomique** qui vérifie l'existence d'un entier *x* dans un tableau *T* trié dans l'ordre croissant contenant *N* entiers, en utilisant la technique de la recherche dichotomique.