Université Moulay Ismail **Année universitaire : 2023/2024**
Faculté des sciences de Meknès **Filière : MIP**
Département d'Informatique **Module : INFO2**

TD N°2 : Les fonctions récursives

Exercice 1 :

Ecrire une **fonction récursive** qui calcule la somme de N premiers nombres entiers naturels:

S=1+2+3+……+(N-1)+N

Solution :

Pour écrire la forme **récursive** de la **fonction** somme, il faut chercher tout d'abord la récurrence mathématique.

S(0) = 0
S(N)=N+S (N-1)

```
Fonction somme (N : entier) () : entier
        Debut
                Si N=0 alors
                        Retour (N)
                Sinon
                        Retour( N+somme(N-1))
                Finsi
        Fin
```

Programme en Python :

```
def somme (N: int):
   if N==0 :
      return N
   else :
      return (N+somme(N-1))

N=int(input("Entrer une valeur entière"))
print("La somme de N premiers nombres entiers naturels : ", somme(N))
```

Exercice 2 :

Ecrire une **fonction récursive** qui calcule le N^{ieme} terme da la suite numérique définie par :

$U_0 = 2$
$U_1 = 2$
$U_2 = 2$
$U_n = 6*U_{n-1} + 4*U_{n-2} - 5*U_{n-3}$ pour tout $n > 2$

Solution :

```
Fonction suite (N : entier) () : entier
        Debut
                Si N<= 3 alors
                        Retour (2)
                Sinon
                        Retour(6*suite(N-1) + 4*suite(N-2) - 5*suite(N-3))
                Finsi
        Fin
```

Programme en Python :

```
def suite(N: int):
   if(N<=2) :
      return 2
   else :
      return (6*suite(N-1)+4*suite(N-2)-5*suite(N-3))

N=int(input("Entrer une valeur entière"))
print("Le Nieme terme da la suite numérique est: ", suite(N))
```

Exercice 3 :

On appelle palindrome une chaîne de caractère qui donne la même chaîne selon que l'on la lire de gauche à droite ou inversement. Autrement dit, le premier caractère est égal au dernier caractère, le deuxième caractère est égal à l'avant dernier caractère, etc.
Une définition récursive d'un palindrome est :

- La chaîne vide est un palindrome.
- La chaîne constituée d'un seul caractère est un palindrome.
- aXb est un palindrome si a = b et si X est un palindrome.

Écrire une fonction récursive qui teste si une chaîne de caractères et qui renvoie vrai si elle palindrome et faux si elle ne l'est pas.

Solution :

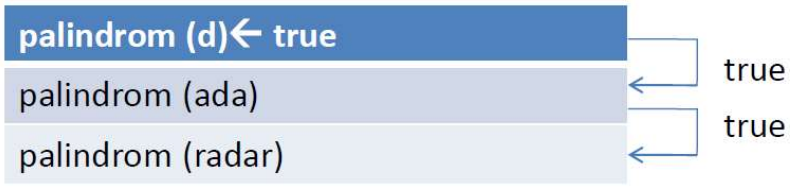
Exemple : radar

1. Décrire la condition d'arrêt : quand peut-on trouver facilement la solution ?
   La chaîne constituée d'un seul caractère est un palindrome ou de 0 caractère est palindrome
2. Réduire le problème à un problème d'ordre inférieur pour que la condition d'arrêt soit atteinte un moment donné
   a X b est un palindrome si a = b et si X est un palindrome.

Cas de base : La chaîne constituée d'un seul caractère est un palindrome
Cas général : aXb est un palindrome si a = b et si X est un palindrome.

```
Si a=b alors
        palindrome (r ada r) ← palindrom(ada)
Sinon
```

```
        retourner (false)
Si a=b alors
        palindrom(ada ) ← palindrom( d )
Sinon
        retourner (false)
palindrom(d) ← true
```



Récursivité non terminale

```
Fonction palindrome (tableau ch(): caractère , entier: i , entier j) : booléen
        Début
                Si ( i ≥ j)  alors
                        retourner (VRAI)
                sinon
                        Si ch[i] = ch[j]
                                retourner ( palindrome(ch, i + 1, j –1))
                        Sinon
                                retourner (false)
                        FinSi
                FinSi
        Fin
```

Programme en python

```
def palindrome(ch, i, j):
    if(i>=j) :
        return True
    elif (ch[i]==ch[j]):
        return (palindrome(ch, i+1, j-1))
    else:
        return False


mot=input("Entrer une valeur entière")
print("Le mot", mot," est Palindrome : ", palindrome(mot,0,len(mot)-1))
```

Exercice 4 :

1. Écrire de manière récursive une fonction qui donne le plus grand commun diviseur (pgcd) de deux entiers.
2. Trouver les relations de récurrence en utilisant une méthode de détermination du pgcd basée sur la soustraction

Solution :

- **En informatique la récursivité consiste à remplacer une boucle par un appel à la fonction elle-même.**

- On va créer une fonction qui pour fournir son résultat, elle va s'appeler elle-même un certain nombre de fois. Cet appel de la fonction par elle-même est la récursivité. Toutefois, il est impératif que ces auto-appels de la fonction PGCD s'arrêtent.

1. Décrire la condition d'arrêt : quand peut-on trouver facilement la solution ?
   Cas de base (a=b) :
   Si a=b le PGCD c'est a ou b donc on arrête donc retourner (a) ou retourner (b)
2. Réduire le problème à un problème d'ordre inférieur pour que la condition d'arrêt soit atteinte un moment donné
   Cas général (a>b) :

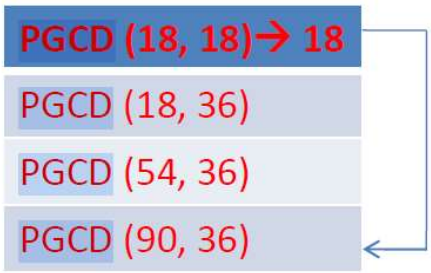
```
Si (a > b) Alors
        Resultat← PGCD(a -b, b)
Sinon
        Resultat← PGCD(a, b -a)
FinSi
        Retourner (Resultat)
```

A=90 B=36 → PGCD (90,36)→PGCD (90-36=54, 36)

A=54 B=36 → PGCD (54-36=18, 36)

A= 18 B=36 →PGCD (18, 36-18=18)

A=18 B=18 → A=18



Récursivité terminale

```
Fonction PGCD (a, b : entier) : entier
        Variable
                résultat : entier
        Debut
                Si (a = b) Alors
                        résultat = a
                Sinon Si (a > b) Alors
                                résultat = PGCD(a -b, b)
```

```
                Sinon
                        résultat = PGCD(a, b -a)
                FinSi
        FinSi
        Retourner (résultat)
Fin
```

Programme en Python :

```
def PGCD(a,b):
    if(a==b) :
        resultat=a
    elif (a>b):
        resultat= PGCD(a-b,b)
    else:
        resultat= PGCD(a, b-a)
    return resultat

a=int(input("Entrer une valeur entière a="))
b=int(input("Entrer une valeur entière b="))
print("Le PGCD est : ", PGCD(a,b))
```