

TD : Fonctions fondamentales en R

Exercice 1 : PGCD

Objectif : Écrire une fonction en R qui calcule le **Plus Grand Commun Diviseur (PGCD)** de deux entiers naturels a et b , en utilisant l'algorithme d'Euclide.

Instructions :

- Implémentez la fonction `gcd(a, b)` selon l'algorithme suivant :
 - Tant que $b \neq 0$, faire :
 - stocker temporairement b
 - affecter à b le reste de la division de a par b
 - affecter à a la valeur temporaire
 - Retourner a
- Tester votre fonction sur les valeurs $a = 36$, $b = 60$.

Exercice 2 : Facteurs premiers uniques

Objectif : Écrire une fonction en R qui retourne les **facteurs premiers sans répétition** d'un entier naturel n .

Instructions :

- Implémentez la fonction `prime_factors(n)` qui :
 - Itère à partir de $i = 2$ jusqu'à $\sqrt{n}$
 - Divise n autant de fois que possible par i , si $i \mid n$
 - Stocke i dans une liste de facteurs
 - Si $n > 1$ à la fin, l'ajouter aussi à la liste
 - Retourne les facteurs **sans répétition**
- Tester la fonction sur l'exemple $n = 84$

Solution attendue Ex1 :

```
gcd <- function(a, b) {  
  while (b != 0) {  
    temp <- b  
    b <- a %% b  
    a <- temp  
  }  
  return(a)  
}  
  
# Exemple d'utilisation  
gcd(36, 60) # Résultat : 12
```

Solution attendue Ex 2 :

```
prime_factors <- function(n) {  
  factors <- c()  
  i <- 2  
  while (i * i <= n) {  
    while (n %% i == 0) {  
      factors <- c(factors, i)  
      n <- n / i  
    }  
    i <- i + 1  
  }  
  if (n > 1) {  
    factors <- c(factors, n)  
  }  
  return(unique(factors))  
}  
  
# Exemple d'utilisation  
prime_factors(84) # Résultat : 2 3 7
```