

```
#####
#####TP-TD 12: variables locals/globals-Fonctions récursives#####
#####
'''
Exercice 1
Soit U la suite récurrente suivante:
U(1)=10 et U(n)=3U(n-1)+2 si n>1
Ecrire une fonction de paramètre n qui retour le n eme terme de U.
'''

# -----Méthode irérative (utilisant des boucles)-----
def suite(n):
    u=10
    for i in range(1,n):
        u=3*u+2
    x=10
    y='bjr'
    z=10+x
    print('locals=',locals())
    return u

# -----Méthode récursive (qui appelle à elle même avec un autre paramètre)-----
def suite_recur(n):
    if n==1:return 10
    return (3*suite_recur(n-1)+2)

'''
Exercice 2
Ecrire une fonction récursive puiss_recur qu accepte x et n (x supposé un réel et n est u_
retourne x**n.
'''
def puiss_recur(x,n):
    if n==1:return x
    return x*puiss_recur(x,n-1)
'''
Exercice 3
Ecrire une fonction récursive multi_recur qu accepte x et n (x supposé un réel et n est u_
retourne nx.
'''
def multi_recur(x,n):
    if n==0: return 0
    # if n==1: return x
    # if n==2: return 2*x
    return x+multi_recur(x,n-1)
'''
Exercice 4
Ecrire une fonction récursive multi_recur qu accepte x et n (x supposé un réel et n est u_
retourne nx.
'''
def fact(n):
    if n==1: return 1
    return n*fact(n-1)
'''
Exercice 5
Ecrire une fonction without_car qu accepte une chaine s et un caractere c et retourne une
sans c. (par plusieurs méthodes)
'''
```

```

# 1ere méthode
def without_car1(s='',car=''):
    new_s=''
    for c in s:
        #if c==car:continue
        if c!=car:
            new_s+=c
        #else:s+=car
    return new_s

# 2eme méthode
def without_car2(s='',car=''):
    return s.replace(car,'')

# 3eme méthode
def without_car3(s='',car=''):
    return ''.join(s.split(car))

# 4eme méthode
def without_car4(s='',car=''):
    s0=s
    while True:
        i=s0.find(car)
        if i==-1:break
        s0=s0[:i]+s0[i+1:]
    return s0

## i=s0.find(car)
## while i!=-1:
##     s0=s[:i]+s[i+1:]
##     i=s0.find(car)
##
## else:
##     return s

#5eme méthode
def without_car5(s='',car=''):

    i=s.find(car)
    if i==-1: return s
    else:
        return without_car5(s[:i]+s[i+1:],car)

```