

```

# ===== #
# ENSAM (PSI) 2015 - Planche 270 - Exercice II #
# (2eme exercice du recueil d'exercices pour les révisions) #
# ===== #

# Fonction construisant la liste des diviseurs propres d'un entier naturel n
# passé en argument. On suppose que cet entier est supérieur ou égal à 2.
def LDP(n):
    L = []
    dpmax = n//2
    for i in range(1,dpmax+1):
        if n%i == 0:
            L += [i]
    return(L)

# Fonction déterminant la somme des diviseurs propres d'un entier naturel n
# passé en argument.
# 1ère version (naïve car utilisant LDP et construisant une liste... inutile)
def SDP(n):
    # On utilise LDP pour obtenir la liste des diviseurs propres de n
    L = LDP(n)
    # On renvoie la somme des termes de L
    return(sum(L))

# Fonction déterminant la somme des diviseurs propres d'un entier naturel n
# passé en argument.
# 2ème version (issue de LDP mais ne construisant pas une liste inutile)
def SDP2(n):
    S = 0
    dpmax = n//2
    for i in range(1,dpmax+1):
        if n%i == 0:
            S += i
    # On renvoie S
    return(S)

# Fonction donnant les nombres parfaits inférieurs à K passé en argument.
def Parfaits(K):
    for i in range(1,K+1):
        if i == SDP(i):
            print(i,"est parfait.")

# Fonction donnant la liste des couples (p,q) de nombres amicaux tels que
# p < q <= K
def Amicaux(K):
    A = []
    # Liste des SDP des entiers compris entre 1 et K
    L = []
    for i in range(1,K+1):
        L += [SDP(i)]
    # Recherche des nombres amicaux
    for p in range(1,K):
        for q in range(p+1,K+1):
            if L[p-1] == q and L[q-1] == p:
                print(p,"et",q,"sont amicaux.")
                A += [[p,q]]
                # Si on préfère manipuler des tuples, on peut remplacer la ligne
                # précédente par :
                # A += [(p,q)]
    return(A)

```