

```

'''
TP3 : Les files
Groupe : Ipein/SP3/GB
Date : 30-09-2023
'''

# file.py
def creer_file(): return []
def file_vide(f): return len(f)==0
def sommet(f): return f[0]
def taille(f): return len(f)
def enfiler(f,x): f.append(x)
def defiler(f): return f.pop(0)

#ex 1
def afficher(f):
    for i in range(taille(f)):
        print(sommet(f))
        s = defiler(f)
        enfiler(f,s)

def defiler_jusqua(f,x):
    # à la fin :
    # [] if x n'existe pas
    # [x,*,*,*] if x existe
    # sol 1
    t = taille(f)
    for i in range(t):
        if sommet(f)==x:
            break
        else:
            defiler(f)
    # sol 2
    while not file_vide(f) and sommet(f)!=x:
        defiler(f)
def appartient(f,x):
    test = False
    t = taille(f)
    for i in range(t):
        s = defiler(x)
        enfiler(f,s)
        if s == x: test = True
    return test

from pile import *
def inverser(f):
    p = creer_pile()
    # inverser l'ordre via une pile
    # vider f dans p
    while not file_vide(f):
        empiler(p, defiler(f))
    # vider p dans f
    while not pile_vide(p):
        enfiler(f, depiler(p))

# Ex 2
def nb_hamming(n):
    f2 = creer_file(); enfiler(f2,1)
    f3 = creer_file(); enfiler(f3,1)
    f5 = creer_file(); enfiler(f5,1)
    for i in range(n):
        k = min(sommet(f2),sommet(f3),sommet(f5))
        print(k)
        for f in (f2,f3,f5):
            if sommet(f)==k:
                defiler(f)
            enfiler(f2,2*k)
            enfiler(f3,3*k)
            enfiler(f5,5*k)

# Ex 3 : reste à faire

```