

```

'''
TP3 : Les files
Groupe : Ipein/SP1/GB
Date : 24-09-2023
'''

# 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):
    # ajouter x en fin de f
    f.append(x)
def defiler(f):
    # le sommet est à la position 0
    return f.pop(0)

# Ex1
def afficher(f):
    for i in range(taille(f)):
        x = defiler(f)
        print(x)
        enfiler(f,x)

def defiler_jusqua(f,x):
    while taille(f)>0 and sommet(f)!=x:
        defiler(f)
    # sortie de while :
    # if file vide or sommet(f)==x

def appartient(f,x):
    from copy import copy
    f1 = copy(f)
    defiler_jusqua(f1,x)
    if file_vide(f1): return False
    return True

def creer_pile():
    return []

p = creer_pile()

def pile_vide(p):
    #return True if len(p)==0 else False
    return len(p)==0

vide = pile_vide(p)

def sommet(p):
    if not pile_vide(p):
        return p[-1] #p[len(p)-1]
    else:
        #print("pile vide")
        raise Exception("pile vide")

s = sommet(p)

def taille(p):
    return len(p)

def empiler(p,x):
    p.append(x)

def depiler(p):
    return p.pop()#par défaut index=-1

def inverser(f):
    #from pile import *
    p = creer_pile()
    #vider f dans p
    while taille(f)>0:
        x = defiler(f)
        empiler(p,x)

    #vider p dans f
    while not pile_vide(p):
        x = depiler(p)
        enfiler(f,x)

# Ex 2
def nb_hamming(n):
    f2,f3 = creer_file(),creer_file()
    f5 = creer_file()

```

```

enfiler(f2,1)
enfiler(f3,1)
enfiler(f5,1)
for i in range(n):
    k= min(sommet(f2),sommet(f3),sommet(f5))
    print(k,end=", ")
    if k == sommet(f2):
        enfiler(f2,defiler(f2)*2)
    elif k == sommet(f3):
        enfiler(f3,defiler(f3)*3)
    else:
        enfiler(f5,defiler(f5)*5)

# Ex 4
def creer_file(n): return [0,0,[0]*n]
def taille(f): return f[0]-f[1]
def file_vide(f): return taille(f)==0
def file_pleine(f): return taille(f)== len(f[2])
def enfiler(f,x):
    if not file_pleine(f):
        Q = f[0]
        f[2][Q] = x
        f[0] += 1

def defiler(f):
    if not file_vide(f):
        T = f[1]
        x = f[2][T]
        f[1] += 1
        return x

```