```python
def creer_file():
   return list()
def file_vide(f):
   return len(f)==0
def taille(f): return len(f)
def enfiler(f,x):
   f.append(x)
def defiler(f):
   return f.pop(0)
def sommet(f):
   return f[0]

# ex1
def afficher(f):
   t = taille(f)
   for i in range(t):
       s = defiler(f)
       print(s)
       enfiler(f,s)

def defilerJusqua(f,x):
   # à la fin : [] ou [x,*,*,*]
   while taille(f)>0 and sommet(f)!=x:
       defiler(f)

from copy import copy

def appartient(f,x):
   # x in f: à éviter avec les files/piles
   # f = [s,*,*,*,*]
   f1 = copy(f)
   defilerJusqua(f1,x)
   # à la fin : f1 =[] ou f1=[x,*,*,*]
   return not file_vide(f1)
   #return taille(f1)>0
   #return taille(f1)
"""
from pile import *
def inverser_file(f):
   #f = [a,b,c,d]
   #p = []
   # empiler(p,defiler(f))
   #p = [a] => f= [b,c,d]
   #p = [a,b] => f= [c,d]
   #p = [a,b,c] => f= [d]
   #p = [a,b,c,d] => f= []
   #p : dernier entré, premier sorti
   # enfiler(f,depiler(p))
   # p = [a,b,c] => f[d]
   # p = [a,b] => f[d,c]
   # p = [a] => f[d,c,b]
   # p = [] => f[d,c,b,a]
   # etape 1 : vider f dans p
   while not file_vide(f):
       empiler(p,defiler(f))
   # etape 2 : vider p dans f
   while not pile_vide(p):
       enfiler(f,depiler(p))
   #méthode 2
   t = taille(f)
   for i in range(t):#i < taille(f)
       empiler(p,defiler(f))
"""
# Ex 3
# nb hamming = 2^i * 3^j * 5^k
# f2 = [6,8]
# f3 = [6,9,12]
# f5 = [5,10,15,20]
# k = min(f2,f3,f5) => k : nb hamm
# print(k), k=4
# nbres : 1,2,3,5
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,end=", ")
       if k == sommet(f2): defiler(f2)
       if k == sommet(f3): defiler(f3)
       if k == sommet(f5): defiler(f5)
       enfiler(f2,2*k)
       enfiler(f3,3*k)
```

```
        enfiler(f5,5*k)

nb_hamming(10)
# Ex 3
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 sommet(f):
    t = f[1]
    return f[2][t]
def enfiler(f,x):
    n = len(f[2])
    if taille(f)< n:
        q = f[0]
        f[2][q] = x
        f[0]+=1
def defiler(f):
    if not file_vide(f):
        x= sommet(f)
        f[1]+=1
        return x
```