

```

def gagnant_rec(i,j):
    '''determine si la position (i,j) est gagnante ou perdante'''
    if (i,j)==(0,0):
        return False
    for k in range(1,j+1):
        if gagnant_rec(i-k,min(i-k,2*k))==False:
            return True
    return False

def gagnant(n): #position de départ avec n allumettes
    return gagnant_rec(n,n-1)

print('-----sans memoisation:-----')
#for i in range(45,50):
#    print(i,gagnant(i))

##Avec Memoisation
d={(0,0):False}
def gagnant_rec_mem(i,j):
    '''determine si la position (i,j) est gagnante ou perdante'''
    if (i,j) in d:
        return d[(i,j)]

    for k in range(1,j+1):
        if gagnant_rec_mem(i-k,min(i-k,2*k))==False:
            #s'il existe une position accessible perdante, alors (i,j) est gagnante
            d[(i,j)]=True
            return True
    #si toutes les positions accessibles sont gagnantes, alors (i,j) est perdante
    d[(i,j)]=False
    return False

def gagnant_mem(n):
    return gagnant_rec_mem(n,n-1)

print('-----avec memoisation:-----')
for i in range(100):
    if gagnant_mem(i)==False:
        print(i)

##on constate que les postions perdantes sont exactement les nombres de Fibonacci
##(au moins jusqu'à 100, mais on pourrait démontrer que c'est toujours vrai)

```