CH8-prog.py

```python
# Algorithme naïf de recherche
def presence(e,L):
    for k in L:
        if k==e:
            return True
    return False

#Algorithme de recherche dichotomique
def presence_dichotomie(L,e):
    g=0
    d=len(L)-1
    while g<=d:
        m=(g+d)//2
        if e==L[m]:
            return True
        else: # on définit le nouveau tableau pour l'étape suivante
            if e>L[m]: # e se trouve dans la partie droite
                g=m+1
            else:      # e se trouve dans la partie gauche
                d=m-1
            # dans chaque cas, le terme d'indice m n'est plus pris en compte
            # dans le nouveau tableau puisque e n'est pas égal à L[m]
    # On sort de la boucle lorsque d devient plus grand que g
    return False

#Algorithme de résolution d'une équation du type f(x)=0
def solution(f,a,b,eps):
    if f(a)*f(b)>0:
        print("Cette fonction ne s'annule pas sur cet intervalle.")
    else :
        g=a
        d=b
        while d-g>eps:
            m=(g+d)/2
            if f(m)==0:
                return(m)
            else: # on définit les bornes du nouvel intervalle
                if f(g)*f(m)>0: # la solution se situe à droite de m
                    g=m
                else:            # la solution se situe à gauche de m
                    d=m
        return m

#Algorithme naïf d'exponentiation
def puissance(n,x):
    res=1
    for i in range(n):
        res=res*x
    return res

#Algorithme dichotomique d'exponentiation
def puissance2(n,x):
    p=1
    while n!=0:
        if n%2!=0:
            p=p*x
        x=x*x
        n=n//2
    return p

```