PARTIE ALGORITHMIQUE	TP1 L'algorithme de la dichotomie

1) Le jeu du nombre caché

Un joueur 1 (ou l'ordinateur) choisit un nombre compris entre 1 et 100. L'autre joueur doit trouver ce nombre en minimisant le nombre d'essais.

Appelons N le nombre secret du joueur 1 et R la réponse du joueur 2.

Le joueur 2 doit proposer des réponses et le joueur 1 répond selon les cas « trop petit », « trop grand » ou « gagné ».

Proposer un algorithme et écrire le programme en Python que vous appellerez Nombre_cache1.py

Aide : revoir le module random pour un tirage au sort aléatoire.

Modifier le programme précédent pour que le nombre maximal d'essais soit égal à 10.

2) Initiation à la dichotomie

Quelle technique de jeu peut employer un humain pour gagner le plus rapidement possible ? Proposer 2 méthodes.

Programmer en Python ces 2 méthodes que vous nommerez Nombre_cache2.py et Nombre_cache3.py

Vérifiez que les programmes fonctionnent correctement en affichant les différentes étapes.

3) Utilisation de la dichotomie pour résoudre une équation

f est une fonction définie sur l'intervalle $[a;b]$ et strictement monotone sur $[a;b]$. On cherche à résoudre numériquement l'équation $f(x) = 0$.

- On peut éliminer tout d'abord les cas où $f(a) = 0$ ou $f(b) = 0$.
- L'existence d'une racine α sur $]a;b[$ est une conséquence du fait que $f(a)$ et $f(b)$ sont de signes contraires, ce qui équivaut à $f(a) \times f(b) < 0$, et que f est continue sur $[a\ ;\ b]$.
- Si c est un réel de l'intervalle $]a, b[$, la position de α par rapport à c peut être testée par l'intruction suivante :

```
Si  f(a)×f(c)<= 0  alors
      Rechercher α sur ]a;c]
  Sinon
      Rechercher α sur ]c;b[
```

Dans la pratique on prendra pour c le milieu de l'intervalle $[a;b]$.

On itérera le processus jusqu'à obtenir par exemple $b - a < e$ pour *une précision e donnée*.

Algorithme

Si $f(a)\times f(b)\geq 0$ alors
    Afficher('Pas de racine sur]A;B[')
Sinon
    Tant que b-a>=e Faire

$$c \leftarrow \frac{a+b}{2}$$

Si $f(a)\times f(c)<=0$ alors
    $b \leftarrow c$
Sinon
    $a \leftarrow c$

Compléter le programme :

```
''' Recherche d'une solution par dichotomie'''

def f(x):
   return x**3+x+1

########Programme Principal ############
a=float(input('Entrez la borne a'))
b=float(input('Entrez la borne b'))
e=float(input('Entrez la précision e'))
```

Application à la recherche d'un élément dans une liste triée :

Soit une liste contenant N=256 nombres triés par ordre croissant.

1) Avec un algorithme séquentiel passant en revue les nombres un par un, quel est le nombre maximal de tests effectués ?
2) Même question avec un algorithme utilisant la dichotomie. Donner un exemple.