1 Dans l'algorithme de recherche dichotomique dans une liste triée, il est nécessaire que notre liste de départ soit triée dans l'ordre croissant. Écrire une fonction qui prend en paramètre une liste d'entiers et renvoie *True* si elle est triée dans l'ordre croissant et *False* sinon.

2 Dans cet exercice, on utilise la fonction de recherche dichotomique dans une liste triée donnée en cours que l'on rappelle ici, sans les commentaires :

```
def dicho(L, a):
    """recherche si l'élément a est présent dans la liste triée dans l'ordre croissant L et renvoie un
     indice si c'est le cas et False sinon"""
    g = 0
    d = len(L) - 1
    while g <= d:
        m = (g + d) // 2
        if L[m] == a:
            return(m)
        elif L[m] < a:
            g = m + 1
        else:
            d = m - 1
    return(False)
```

1. On utilise la liste triée $L = [12, 15, 16, 18, 21, 24, 25, 27, 31, 33, 35, 36, 38]$. Suivre l'algorithme pas à pas, en donnant les valeurs de g, d et en explicitant la sous-liste dans laquelle on cherche l'élément dans le cas où :
   (a) $a = 25$
   (b) $a = 15$
   (c) $a = 34$
2. Que se passe t-il si l'on cherche un élément a dans une liste vide ?
3. Cet algorithme fonctionne t-il toujours si l'on remplace *while* $g <= d$ par *while* $g < d$. Si vous pensez que ce n'est pas le cas, trouver un contre-exemple.

3 Dans cet exercice, on utilise la fonction d'exponentiation rapide donnée en cours que l'on rappelle ici, sans les commentaires :

```
def exporapide(a, n):
    r = 1
    while n > 0:
        if n % 2 == 1: # cas où n est impair
            r = r * a
        a = a * a
        n = n // 2
    return(r)
```

On choisit $a = 2$ et $n = 31$.

1. Donner les valeurs de a, n et r tout au long de l'exécution de la fonction.
2. Combien de multiplications effectue l'algorithme ?

3. La méthode de calcul se base sur les multiplications suivantes :

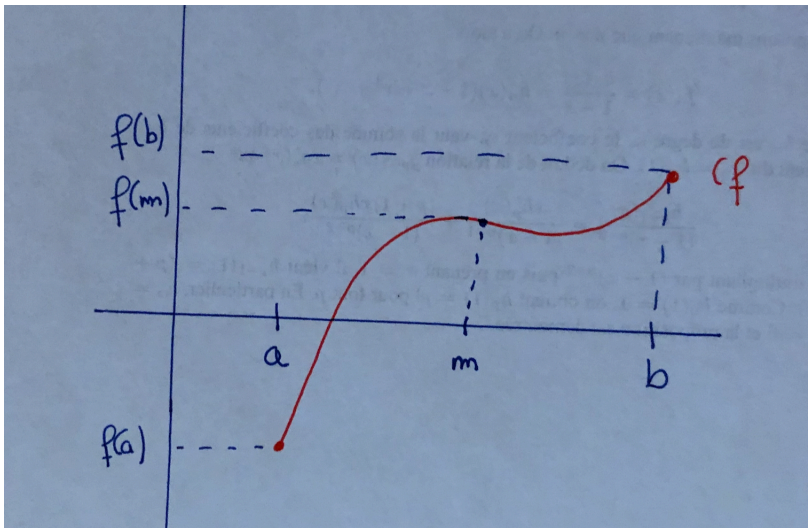
$$2^{31} = 2 \times 2^{30}$$
$$2^{30} = 2^{15} \times 2^{15}$$
$$2^{15} = 2 \times 2^{14}$$
$$2^{14} = 2^7 \times 2^7$$
$$2^7 = 2 \times 2^6$$
$$2^6 = 2^3 \times 2^3$$
$$2^3 = 2 \times 2^2$$
$$2^2 = 2 \times 2$$

   Ici nous avons 8 multiplications, comment expliquer la différence avec la réponse à la question précédente ?

4. Trouver une façon de calculer 2^{31} avec seulement 7 multiplications.

4 Le but de l'exercice est d'écrire un algorithme qui permet de chercher une valeur approchée d'un zéro d'une fonction f continue sur l'intervalle $[a, b]$. On suppose pour cela que f s'annule une unique fois, comme sur le graphique ci-dessus. Le principe est le suivant :

- on considère le milieu du segment $[a, b]$, que l'on note m.
- si le zéro recherché est dans $[a, m]$, on ne change pas a et on pose $b = m$
- si le zéro recherché est dans $[m, b]$, on ne change pas b et on pose $a = m$
- on réitère le procédé avec le nouvel intervalle $[a, b]$ obtenu.



C'est bien un algorithme dichotomique car la longueur de l'intervalle d'étude est divisée par 2 à chaque étape.

1. Montrer que tester la condition $f(a) \times f(m) < 0$ permet de savoir dans quel intervalle se situe le zéro recherché.
2. On souhaite obtenir un encadrement du zéro recherché avec une précision notée *eps* et on va utiliser une boucle *while*. Quelle est la condition que l'on doit mettre dans cette boucle ?
3. Écrire complètement la fonction *zerodicho*(f, a, b, *eps*) renvoyant une valeur approchée à *eps* près de l'unique zéro de f sur l'intervalle $[a, b]$.
4. Comment trouver une valeur approchée de $\sqrt{2}$? de π ?