Exercice 1 : On considère la fonction définie sur $\mathbb{R}$ par :

$$f(x) = 0,5x^2 - x - 12$$

1. Calculer l'image de -1 par f.
2. À l'aide d'un développement, vérifier que $f(x) = (0,5x + 2)(x - 6)$.
3. À l'aide d'un développement, vérifier que $f(x) = 0,5(x - 1)^2 - 12,5$.
4. Déterminer les antécédents de 0 en résolvant une équation.
5. En déduire les solutions de l'inéquation $f(x) \geqslant 0$.
6. Déterminer les antécédents de -12 en résolvant une équation (en choisissant la forme adaptée).
7. Construire le tableau de variation de f (vous pouvez vous aider d'un logiciel comme GeoGebra pour tracer la courbe et déterminer le minimum).

Exercice 2 : **Formule de Héron pour les triangles**

1. ABC est un triangle tel que $AB = 8$ cm, $AC = 15$ cm, $BC = 17$ cm.
   a. Montrer que ABC est rectangle en A.
   b. En déduire l'aire de ABC.
2. On attribue à Héron d'Alexandrie (1^{er} siècle après J.C) la démonstration d'une formule permettant de calculer l'aire $\mathscr{A}$ d'un triangle dont on connaît les trois côtés a, b et c, sans avoir besoin de calculer la hauteur :

$$\mathscr{A} = \sqrt{p(p-a)(p-b)(p-c)}$$

   où p est la moitié du périmètre du triangle.
   a. En appliquant la formule de Héron, retrouver l'aire de ABC.
   b. Montrer qu'un triangle équilatéral de côté c a une aire de $\dfrac{\sqrt{3}}{4}c^2$.
   c. Appliquer cette formule pour calculer la surface d'un panneau de signalisation d'un danger ayant la forme d'un triangle équilatéral de côté 125 cm.

Exercice 3 : **Méthode de Héron pour le calcul d'une racine carrée**

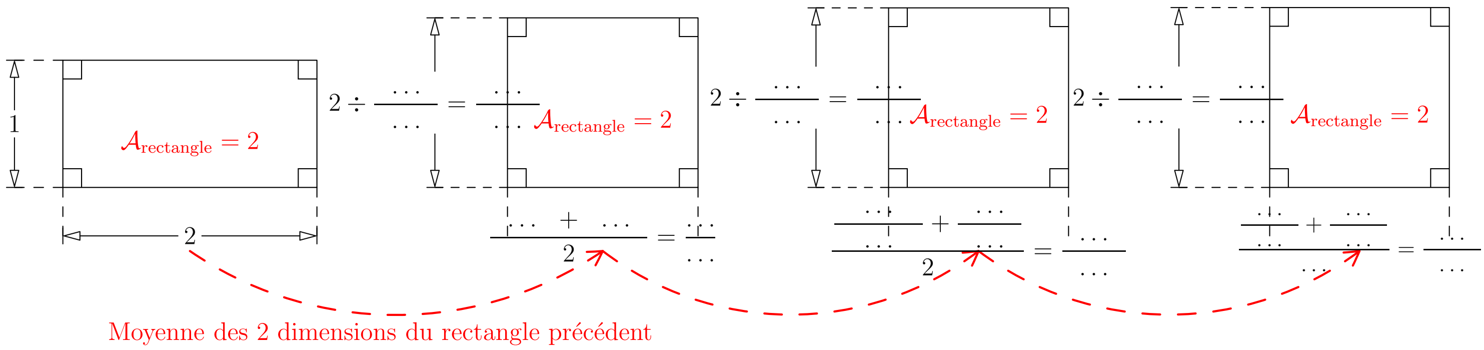
La méthode décrite ci-dessous a pour but d'encadrer $\sqrt{a}$ (a réel strictement supérieur à 1) avec une précision arbitraire. Cette méthode déjà connue par les Babyloniens, est également attribuée au grec Héron d'Alexandrie (1^{er} siècle). Il l'expose dans le premier tome de son ouvrage *Metrica*, ouvrage qui a été découvert en 1896. Chez les mathématiciens grecs, extraire la racine carrée de $\sqrt{a}$ revient à trouver un carré dont l'aire soit égale à a. En prenant un rectangle de côté arbitraire $L_0 = x$ et de même aire, il est nécessaire que l'autre côté ait pour longueur $\ell_0 = \dfrac{a}{x}$.

Pour le rendre « moins rectangle », il suffit de considérer un nouveau rectangle dont les dimensions vérifient :

- la longueur est la **moyenne arithmétique** des dimensions du rectangle précédent ;
- pour que l'aire reste égale à a, on **divise** a par cette nouvelle longueur pour trouver l'autre dimension du rectangle.

En réitérant infiniment le processus, le rectangle se transforme petit à petit en un carré de même aire. Cette constatation est à la base de la méthode de Héron.

1. Compléter les valeurs obtenues avec cet algorithme, pour l'approximation de $\sqrt{2}$:



Remarque : On admet que $\sqrt{2}$ est toujours compris entre ces deux dimensions. Ainsi, dans tous les cas, $\ell_0 < \sqrt{2} < L_0$, $\ell_1 < \sqrt{2} < L_1$, … À chaque itération de l'algorithme de Héron, on obtient un encadrement de plus en plus fin de $\sqrt{2}$.

On souhaite automatiser la recherche d'un encadrement de $\sqrt{2}$ d'amplitude inférieure à un niveau de précision donné par l'utilisateur. On considère la fonction `racine_heron` ci-contre écrite en langage naturel et qui prend en paramètres le nombre $a > 1$ dont on veut la racine carrée ainsi que la précision `nb_decimales` exprimée en nombre de chiffres après la virgule.

2. En vous inspirant des calculs effectués dans la partie précédente, complétez cette fonction et sa traduction en Python.
3. Testez cette fonction dans un environnement Python pour obtenir un encadrement de $\sqrt{2}$ d'amplitude inférieure à 10^{-5}.
4. Modifiez la fonction pour qu'elle affiche aussi le nombre nécessaire d'étapes pour obtenir l'encadrement demandé (rajouter un compteur). Quelle est le nombre nécessaire d'étapes pour obtenir un encadrement à 10^{-12} ?
5. Déposez votre fichier heron_mon_prenom.py dans le cours Moodle de la classe.

racine_heron

Fonction racine_heron(a, nb_decimales) **:**
  longueur ← . . . . .
  largeur ← . . . . .
  Tant que $|longueur - largeur| > 10^{-nb\_decimales}$ **faire :**
    longueur ← . . . . .
    largeur ← . . . . .
  renvoie arrondi(largeur, nb_decimales), arrondi(longueur, nb_decimales)

Code Python :

```python
def heron_racine(a, nb_decimales) :
    """cette fonction calcule un encadrement de la racine carrée de a avec le nombre
    de décimales spécifié"""
    longueur = ...
    largeur = ...
    while abs(longueur - largeur) > 10**(-nb_decimales) :
        longueur = ...
        largeur = ...
    return round(largeur,nb_decimales), round(longueur,nb_decimales)
```