TP 1 (Rappel et prérequis)

Environnement Python

Pour utiliser Python :

1. Préconisation : Installer un IDE comme
    a. **Jupyter Notebook (via Anaconda)**
    b. **PyCharm**
2. Alternative : Utiliser une plateforme proposant un compilateur Python en ligne comme : https://repl.it/languages/python3

TP 1 – Exercice 1

Enoncé :

Donner une définition, avec signature et hypothèse(s) éventuelle(s), de la fonction **isocele** qui, étant donné trois nombres positifs a, b et c définissant les trois côtés d'un triangle, renvoie le booléen vrai si ce triangle est isocèle, le booléen faux sinon. (L'expression "définissent les trois côtés d'un triangle" vise à exclure des valeurs telles que a = 1, b = 1, c = 20 qui ne permettent pas de tracer un triangle, car c est "trop grand" par rapport aux deux autres valeurs.)

Exemple :

>>> isocele(4, 2, 3) => False

>>> isocele(4, 3, 3) => True

>>> isocele(4, 4, 4) => True

TP 1 - Exercice 2

Enoncé :

Une façon de calculer l'aire d'un triangle consiste à ordonner les valeurs des côtés, en définissant trois valeurs *u*1, *u*2 et *u*3 telles que *u*1 soit égale à la plus petite des trois valeurs *a*, *b* et *c*, que *u*2 soit la valeur intermédiaire et *u*3 la plus grande, de telle sorte que *u*1 *u*2 *u*3 puis à calculer **:**

$$A = \frac{1}{2}\sqrt{u_1^2 u_3^2 - \left(\frac{u_1^2 - u_2^2 + u_3^2}{2}\right)^2}$$

Donner une définition, avec signature et hypothèse(s) éventuelle(s), de la fonction **aire_ordonne** qui, étant donné trois nombres positifs a, b et c définissant les trois côtés d'un triangle, renvoie l'aire du triangle, calculée en appliquant la formule ci-dessus. La fonction doit utiliser les fonctions min et max pour définir des variables correspondant aux variables a, b et c triées.

TP 1 - Exercice 2

Exemple :

>>> aire_ordonne(4, 2, 3) => 2.9047375096555625

>>> aire_ordonne(4, 3, 3) => 4.47213595499958

>>> aire_ordonne(4, 4, 4) => 6.928203230275509

>>> aire_ordonne(3,4,5) => 6.0

>>> aire_ordonne(13,14,15) => 84.0

>>> aire_ordonne(1,1,1) => 0.4330127018922193

TP 1 - Exercice 3

Enoncé :

Donner une définition, avec signature et hypothèse(s) éventuelle(s), de la fonction **definit_triangle** qui, étant donné trois nombres a, b et c, renvoie le booléen vrai si les trois valeurs définissent un triangle, le booléen faux sinon.

On rappelle que a, b et c définissent un triangle s'ils sont tous les trois strictement positifs et strictement inférieurs à (a + b + c)/2

Exemple :

>>> definit_triangle(1, 1, 20) => False

>>> definit_triangle(4, 2, 3) => True

>>> definit_triangle(4, 4, 4) => True

TP 1 - Exercice 4

Enoncé :

Donner une définition, avec signature et hypothèse(s) éventuelle(s), de la fonction **nb_triangles_speciaux** qui, étant donné deux entiers strictement positifs n et p, renvoie le nombre de triangles dont les côtés sont des valeurs entières comprises entre n et p (inclus) et dont l'aire est égale au périmètre.

On considère que les triangles dont les côtés sont (a, b, c) et (b, a, c) par exemple sont identiques et comptent pour un seul triangle : on ne compte que les triangles tels que a b c. N.B. cet exercice ne nécessite pas d'utiliser de n-uplets ou de listes.

Exemple :

>>> nb_triangles_speciaux(1, 20) => 4

En effet, seuls les triangles dont les côtés sont (5, 12, 13), (6, 8, 10), (7, 15, 20) et (9, 10, 17) ont un périmètre égal à leur aire.