

1I001 — Éléments de programmation 1

Interrogation 1 – Groupe 11.1 Jeudi 25 Septembre 2014

Numéro d'étudiant :

Durée : 15 minutes — Aucun document autorisé hormis la carte de référence. Les réponses doivent tenir dans l'encadré prévu à cet effet.

Exercice 1. Questions de cours

1. Précisez les différences entre une *instruction* et une *expression*

Les principes d'évaluation des expressions (arithmétiques) vues lors du cours précédent expliquent comment passer d'une expression à sa valeur. En comparaison, une instruction ne possède pas de valeur et ne peut donc pas être évaluée. On parle donc plutôt de principe d'interprétation des instructions.

2. Définissez le mot *prédicat*

En langage Python, les fonctions qui retournent un booléen en fonction de leur paramètres sont appelées prédicats.

3. En Python, quel est le type de l'expression suivante : $(42 + 12)/3$? Justifiez votre réponse

L'expression composée $(42 + 12)/3$ est de type *float*.
En effet, la division réelle désignée par «/» retourne forcément un type *float*.

4. En Python, quel est le type de l'expression suivante : $33//5$? Justifiez votre réponse

L'expression composée $33//5$ est de type *int*.
En effet, la division euclidienne désignée par «//» retourne forcément un type *int*.

Exercice 2. Exercice

On considère un triangle défini par ses trois cotés de longueurs entières respectives a , b et c .

1. Écrire la fonction `aire_triangle` qui prend en argument la longueur de chacun des côtés d'un triangle et retourne son aire.

Rappel : la formule du calcul de l'aire d'un triangle est : $Aire = \sqrt{(p-a)(p-b)(p-c)}$ avec $p = \frac{a+b+c}{2}$.

```
import math # necessaire pour pouvoir utiliser la racine carree

def aire_triangle(a,b,c):
    """ int^3 -> int
    Hypothese: (a <= b + c) and (b <= a + c)
    and (c <= a + b)

    Retourne l'aire d'un triangle defini par la
    longueur de ses trois cotes.
    """
    # p : float
        p = (a + b + c) / 2
        return math.sqrt(p * (p - a) * (p - b) * (p - c))

# Jeu de tests
assert aire_triangle(3, 4, 5) == 6.0
assert aire_triangle(13, 14, 15) == 84.0
assert aire_triangle(1, 1, 1) == math.sqrt(3 / 16)
assert aire_triangle(2, 3, 5) == 0.0 # c'est un triangle plat...
```