1I001 — Eléments de programmation 1

Interrogation 0 – Groupe 11.1
Jeudi 18 Septembre 2014

Numéro d'étudiant :

Durée : 10 minutes — Aucun document autorisé hormis la carte de référence. Les réponses doivent **tenir dans l'encadré** prévu à cet effet et les feuilles doivent **rester agrafées**.

Exercice 1. Questions de cours

1. Qu'est ce qu'une *expression* en langage Python ?

   Une expression en Python est une écriture formelle textuelle que l'on peut saisir au clavier en respectant les contraintes induites la syntaxe du langage. La propriété fondamentale d'une expression est d'être évaluable, c'est-à-dire que chaque expression possède une valeur que l'on peut obtenir par calcul.

2. Que signifie *int* en langage Python ?

   En langage Python, *int* (integer) est un type représentant les entiers compris entre -2 147 483 648 et 2 147 483 647.
   Remarque : pour les entier compris entre $-\infty$ et -2 147 483 647 ou entre 2 147 483 648 et $+\infty$, le type correspond est appelé "long" (long integer). La conversion du type int vers le type long est automatique et transparente pour le programmeur.

3. Quel est le type de l'expression suivante : 19.0 ?

   L'expression atomique 19.0 est de type *float*.
   Remarque : pour les férus de formalisme mathématique, ce résultat peut-être surprenant. Dans la majorité des langages informatique, l'ensemble des nombres de type entier n'est pas compris dans l'ensemble des nombres de type flottant. En effet, en mathématiques, 19.0 est à la fois une entier et un nombre réel : il est évident que $19.0 = 19$. Alors qu'en Python ces deux expressions sont de types différents. Par conséquent, l'expression composée suivante :

   ```
   >>> type(19) == type(19.0)
   ```

   renverra False. Attention, le test égalité 19.0 == 19 est toute de même évalué à *True*.

4. En python, quelles sont les deux valeurs de type *bool* possibles ?

   Les deux valeurs de type *bool* possible sont : *True* et *False*.

Exercice 2. Exercice

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

Numéro d'étudiant :

1. Donner les trois relations qui doivent être vérifier par a, b et c pour que le triangle soit correctement défini.

Les longueurs des côtés du triangle doivent satisfaire les inégalités triangulaires :

$$a \leq b + c$$
$$b \leq a + c$$
$$c \leq a + b$$

2. Ecrire la fonction `perimetre_triangle` qui prend en argument la longueur de chacun des côtés d'un triangle et retourne son périmètre.

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

    Retourne le perimetre d'un triangle defini par la
    longueur de ses trois cotes.
    """
    return a + b + c

# Jeu de tests
assert perimetre_triangle(3,2,3) == 8
assert perimetre_triangle(1,1,1) == 3
assert perimetre_triangle(0,0,0) == 0
```