1I001 – MIPI 14.1

Durée : 20 minutes

le 30 septembre, 2014

Le seul document permis est la carte de référence.

Exercice 1 :

Question 1 : L'aire d'un octogone régulier (octogone dont tous les côtés ont même longueur), où les côtés sont de longueurs a est calculée avec la formule $2(1+\sqrt{2})a^2$. Écrire une fonction avec un paramètre qui calcule l'aire d'un octogone régulier.

Correction :

```
import math

def aire_oct_reg(a):
    """Number -> float
    Hypothese: a >= 0
    Renvoie l'aire d'un octogone regulier de cote de longeur a."""
    return 2*(1+math.sqrt(2))*a*a

#jeu de test
assert aire_oct_reg(1) == 2*(1+math.sqrt(2))
assert aire_oct_reg(0) == 0
assert aire_oct_reg(2) == 8*(1+math.sqrt(2))
```

Question 2 : Écrire une fonction avec deux paramètres qui calcule le volume d'un prisme octogonal (8 faces sont des rectangles et 2 faces sont des octogones réguliers)comme présenté dans la figure suivante). On rappelle que le volume de ce polyèdre est égal à l'aire d'un octogone multipliée par la hauteur du polyèdre.

[1]Source: http://etc.usf.edu/clipart/43100/43135/prism-oct1_43135.htm

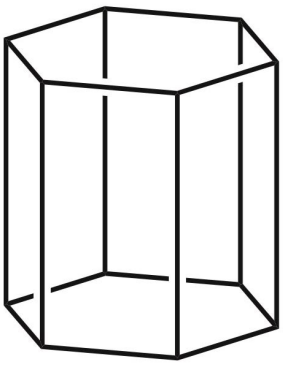

Figure 1: Prisme octogonal.[1]

Correction :

```
def vol_oct_prisme(a,h):
    """ Number * Number -> float
    Hypothese: a >= 0 and h >= 0
    Renvoie le volume d'un prisme octogonal dont les octogones ont des cotes
    de longueurs a et dont la hauteur est h."""
    return aire_oct_reg(a)*h

#jeu de test
assert vol_oct_prisme(1,1) == 2*(1+math.sqrt(2))
assert vol_oct_prisme(0,1) == 0
assert vol_oct_prisme(1,0) == 0
assert vol_oct_prisme(2,2) == 16*(1+math.sqrt(2))
```

Exercice 2 : Un fournisseur de tapis vend des tapis de largeur identique mais de longueur 80, 40, 15 et 1 qu'il produit à partir des grands rouleaux, d'une largeur fixée.

Écrire une fonction qui prend une valeur entière en entrée (cette valeur correspond à la longueur totale d'un rouleau) et calcule, de manière gloutonne, (c'est-à-dire du plus grand au plus petit), le nombre total de tapis qu'il peut produire avec ce rouleau.

Plus précisément, la fonction détermine d'abord le nombre maximal de tapis de longueur 80 que l'on peut produire, puis, sur ce qui reste, détermine le nombre maximal de tapis de longueur 40, et ainsi de suite. La fonction retourne donc le nombre total de tapis produits avec ce rouleau.

Par exemple :

```
>>> nombre_tapis(160)
2

>>> nombre_tapis(40)
1

>>> nombre_tapis(16)
2

>>> nombre_tapis(342)
12
```

Correction :

```
def nombre_tapis(val):
    """ int -> int
    Hypothese: val >= 0
    Renvoie le nombre de tapis produits a partir d'un rouleau de longueur 'val'
    pour des tapis de longueurs 80, 40, 15 et 1.
    Le calcul est fait d'une maniere gloutonne du plus long tapis au plus petit.
    """

    #taille15: int
    taille15 = 0
    #taille40 : int
    taille40 = 0
    #taille80 : int
    taille80 = val // 80
    #reste : int
    reste = val - 80 * taille80

    taille40 = reste // 40
```

```
    reste = reste - 40 * taille40

    taille15 = reste // 15
    reste = reste - 15 * taille15

    return taille80 + taille40 + taille15 + reste

#jeu de test
assert nombre_tapis(160) == 2
assert nombre_tapis(40) == 1
assert nombre_tapis(16) == 2
assert nombre_tapis(342) == 12
```