



## Vous devez faire seulement un exercice parmi les exercices 3 et 4

### 3. Boucles et dessins. ( / 10 pts.)

Rédiger un script qui reproduise (en respectant les couleurs) le dessin de la figure 1 en utilisant l'instruction `for` et le module `turtle`.

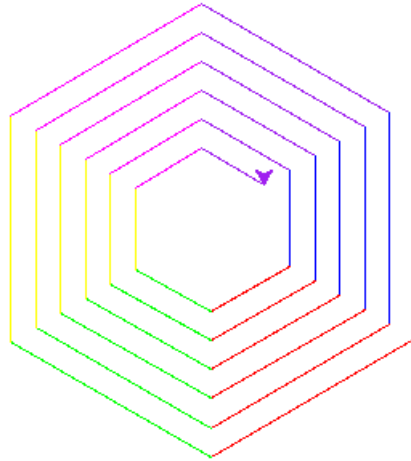


FIGURE 1. Exercice 3

### 4. Le nombre $\pi$ . ( / 10 pts.)

Le nombre  $\frac{\pi^2}{12}$  peut-être évalué par la somme infinie :

$$\frac{\pi^2}{12} = 1 - \frac{1}{4} + \frac{1}{9} - \frac{1}{16} + \dots + (-1)^k \frac{1}{(k+1)^2} + \dots$$

Si on effectue la somme des 1000 premiers termes, que vaut l'approximation de  $\frac{\pi^2}{12}$  ?

## Vous devez faire seulement un exercice parmi les exercices 5 et 6

### 5. Monnaie optimale. ( / 15 pts.)

Rédiger un script *Python* qui demande à l'utilisateur de taper une somme en CHF (sans les centimes) et qui décompose cette somme en billets (de 1000, 200, 100, 50, 20 ou 10 CHF) et en pièces (5, 2 ou 1 CHF) de manière optimale, c'est-à-dire qui détermine le nombre minimal de billets et de pièces nécessaires pour décomposer la somme donnée par l'utilisateur. Utiliser l'algorithme glouton vu en classe. Voici un exemple de fonctionnement du programme:

```
CHF: 2578
```

```
1000 1000 200 200 100 50 20 5 2 1
```

### 6. Suites. ( / 15 pts.)

On considère la fonction  $f : [0; 1] \rightarrow [0; 1]$  définie par

$$f(x) = \begin{cases} 2x & \text{si } 0 \leq x \leq \frac{1}{2} \\ 2(1-x) & \text{si } \frac{1}{2} \leq x \leq 1 \end{cases}$$

Pour un nombre  $u_0 \in [0; 1]$ , on considère la suite de nombres donnée par

$$u_0, \quad u_1 = f(u_0), \quad u_2 = f(u_1), \quad u_3 = f(u_2), \quad \dots$$

- (1) Calculer les 50 premiers éléments de cette suite pour  $u_0 = \frac{2}{3}$ .
- (2) Rédiger un script Python qui affiche les 10 premiers éléments de cette suite pour un nombre  $u_0$  donné par l'utilisateur en utilisant l'instruction `def` et une boucle.
- (3) Exécuter le script pour  $u_0 = \frac{2}{15}$ . Que se passe-t-il avec les éléments de la suite ?