Devoir maison n°6 : Nombres Heureux

Jules Charlier, Thomas Diot, Pierre Gallois, Jim Garnier
1E1

Problème A - Description de l'algorithme

1) En éxécutant l'algorithme pour $n = 12\,345$, on trouve que son image est de 55.

En l'éxécutant encore, on confirme bien que $157 \to 75 \to 74 \to 65$.

---

2) A chaque itération, l'algorithme calcule le dernier chiffre r en base 10 du nombre d, ajoute à la somme p le carré de r et pose d comme le d précédant, sans son dernier chiffre en base 10.

L'algorithme calcule donc la somme des carrés des chiffres en base 10 du nombre n.

---

3) a) Les nombres 157, 175, 517, 571, 715 et 751 ont les mêmes chiffres en base 10 : leurs images par la fonction karma sont donc identiques.

b) Soit $p \in \mathbb{N}^*$. Alors le nombre $n_0 = \overline{1\ldots1}$ avec p chiffres 1 est un antécédant de p par la fonction karma. De plus, si $k \in \mathbb{N}^*$, alors le nombre $10^k n_0$ a p chiffres 1 et k chiffres 0 : c'est donc aussi un antécédant de p par karma.

Donc p a une infinité d'antécédants par la fonction karma.

c) Montrons par l'absurde que 157 n'a pas d'antécédant à 3 chiffres par la fonction karma, en supposant que $\overline{abc}$ soit un antécédant de 157.

Donc 157 n'a pas d'antécédant à 3 chiffres. Cependant, 8852 est un antécédant à 4 chiffres de 157.

Problème B - Trajectoire des nombres inférieurs à 100

1) Réimplémentation de la fonction `karma` en python et en typst.

```python
def karma(n):
    """Calcule la somme des carrés
des chiffres de n"""
    d = n
    p = 0
    while d != 0:
        r = d%10
        d //= 10
        p += r**2
    return p
```

```
#let karma(n) = {
  let d = n
  let p = 0
  while d != 0 {
   let r = calc.rem-euclid(d, 10)
    d = calc.div-euclid(d, 10)
    p += calc.pow(r, 2)
  }
  p
}
```

Création de la fonction `heureux` permettant de déterminer si un nombre `n` est heureux ou non :

```python
def heureux(n):
    """
    Renvoie True si le nombre converge
dans le puit 1,
    sinon False
    """
    image = karma(n)
    while image != 1:
        # Teste si le nombre est arrivé
dans la boucle
        if image == 89:
            return False
        image = karma(image)
    return True
```

```javascript
#let heureux(n) = {
  let image = karma(n)
  while image != 1 {
    if image == 89 {
      return false
    }
    image = karma(image)
  }
  true
}
```

Seul un nombre de la boucle (89) est testé pour simplifier le programme, la perte de vitesse engendrée est négligeable.

Pour obtenir la liste des nombreux heureux strictement inférieurs à 100, il suffit d'éxecuter le code python `[n for n in range(1, 101) if heureux(n)]`.

On récapitule le résultat dans le tableau suivant :

	0	1	2	3	4	5	6	7	8	9
0	X	**H**	m	m	m	m	m	**H**	m	m
1	**H**	m	m	**H**	m	m	m	m	m	**H**
2	m	m	m	**H**	m	m	m	m	**H**	m
3	m	**H**	**H**	m	m	m	m	m	m	m
4	m	m	m	m	**H**	m	m	m	m	**H**
5	m	m	m	m	m	m	m	m	m	m
6	m	m	m	m	m	m	m	m	**H**	m
7	**H**	m	m	m	m	m	m	m	m	**H**
8	m	m	**H**	m	m	m	**H**	m	m	m
9	m	**H**	m	m	**H**	m	m	**H**	m	m

Il n'existe pas de nombre strictement inférieur à 100 qui soit ni heureux ni malheureux.

2) On utilise l'algorithme suivant pour calculer la probabilité qu'un nombre choisit au hasard entre 1 et 100 soit heureux :

```python
def probabilite(Nmax):
    """
    Calcule la probabilite qu'un nombre inferieur à Nmax soit heureux
    """
    return (
        [
            heureux(n) for n in range(1, Nmax+1)
        ].count(True)
        / Nmax
    )
```

On obtient une probabilité de 20%.

Problème C - Trajectoires des nombres de chiffres

1)

---

2)

---

3)