

# Devoir maison n°6 : Nombres Heureux

Jules Charlier, Thomas Diot, Pierre Gallois, Jim Garnier

1E1

## Problème A - Description de l'algorithme

1) L'image de ... est ...

---

2) a)

b)

---

3)

## Problème B - Trajectoire des nombres inférieurs à 100

1)

Réimplémentation de la fonction karma en python et en typst.

```
def karma(n):  
    """Calcule la somme 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 :

```
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
```

```
#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'exécuter 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 :

```
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)