

---

**Generadors de Collatz****P61669\_ca**

---

Sigui  $n$  qualsevol natural estrictament positiu. Considereu el procés següent: Si  $n$  és parell, dividiu-lo per dos. Altrament, multipliqueu-lo per 3 i sumeu-li 1. Quan arribeu a 1, pareu. Per exemple, començant en 3, s'obté la seqüència de Collatz  $S(3)$ :

3, 10, 5, 16, 8, 4, 2, 1.

Des de l'any 1937 es conjectura que aquest procés acaba per a qualsevol  $n$  inicial, encara que no ho ha sabut demostrar ningú. En aquest problema suposarem que la conjectura és certa.

1. Escriviu una funció

```
def collatz (n: int) → Iterator[int]
```

que, donat un natural  $n \geq 1$ , generi la seqüència de Collatz  $S(n)$ .

2. Escriviu una funció

```
def collatz_length (n: int) → int
```

que, donat un natural  $n \geq 1$ , retorni la llargada d' $S(n)$ .

3. Escriviu una funció

```
def collatz_highest (n: int) → int
```

que, donat un natural  $n \geq 1$ , retorni l'element més alt d' $S(n)$ .

4. Escriviu una funció

```
def collatz_highest_position (n: int) → tuple[int, int]
```

que, donat un natural  $n \geq 1$ , retorni una tupla amb la posició de l'element més alt d' $S(n)$  i el seu valor. Les posicions es comencen a comptar des de zero.

5. Escriviu una funció

```
def collatz_11(int) → bool
```

que, donat un natural  $n \geq 1$ , indiqui si  $S(n)$  conté algun múltiple de 11 (excepte en la primera posició).

6. Escriviu una funció

```
def collatz_first_common (n1: int, n2: int) → int
```

que, donats dos naturals  $n_1, n_2 \geq 1$ , retorni el primer element comú d' $S(n_1)$  i  $S(n_2)$ .

7. Escriviu una funció

```
def collatz_common_elements (n1: int, n2: int) → int
```

que, donats dos naturals  $n_1, n_2 \geq 1$ , retorni el nombre d'elements comuns en  $S(n_1)$  i  $S(n_2)$ .

8. Escriviu una funció

```
def collatz_union (n1: int, n2: int) → list[int]
```

que, donats dos naturals  $n_1, n_2 \geq 1$ , retorni tots els elements d' $S(n_1)$  i  $S(n_2)$  en ordre estrictament creixent.

9. Escriviu una funció

```
def collatz_top_numbers() → Iterator[int]
```

que generi la llista infinita de valors inicials pels quals el seu element més alt és més gran que qualsevol dels valors inicials menors. Altrament dit, es vol la seqüència descrita per  $[i \mid \max S(i) > \max S(j) \ \forall j < i]$ .

10. Escriviu una funció

```
def collatz_first_missing(n: int) → int
```

que, donat un natural  $n \geq 1$ , retorni el natural no nul més petit que no pertanyi a  $S(n)$ .

## Observacions

- Implementeu totes les funcions de forma senzilla, clara i concisa treient profit de Python.
- Descarregueu-vos el fitxer `code.py`. El programa principal i l'esquelet de les funcions ja se us dona implementat.
- El Jutge dóna puntuacions parcials per a cada apartat, però l'avaluació va a càrrec del professor.

## Exemple d'entrada

```
collatz 3
collatz 14
collatz 15
collatz_length 3
collatz_highest 3
collatz_highest_position 3
collatz_11 14
collatz_11 11
collatz_first_common 14 15
collatz_common_elements 14 15
collatz_union 7 6
collatz_top_numbers 12
collatz_first_missing 3
```

## Exemple de sortida

```
3 10 5 16 8 4 2 1
14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
8
16
(3, 16)
True
False
40
9
1 2 3 4 5 6 7 8 10 11 13 16 17 20 22 26 34 40 52
1 2 3 7 15 27 255 447 639 703 1819 4255
6
```

## **Informació del problema**

Autor : Jordi Petit

Generació : 2024-05-02 20:39:14

© *Jutge.org*, 2006–2024.

<https://jutge.org>