

Kaprekarjev par

Vhodne podatke preberete v zanki s standardnega vhoda in rezultate izpišite na standardni izhod.

```
if __name__ == '__main__':  
    import sys  
    for line in sys.stdin:  
        print(kaprekar(int(line))[-1])
```

Postopek, ki za dani člen Kaprekarjevega zaporedja poišče naslednji člen:

1. Najprej določimo nov člen zaporedja, tako da števke v zapisu danega člena uredimo v padajočem vrstnem redu.
2. Nato poiščemo število, ki ima števke v zapisu danega člena urejene v naraščajočem zaporedju.
3. Naslednji člen zaporedja je razlika med njima.

Vhod

S standardnega vhoda preberite število k_0 . Tvorite zaporedje k_0, k_1, \dots, k_n , dokler se število k_n prvič ponovi v zaporedju.

Izhod

Na standardni izhod izpišite to število.

Primer

Vhod	Izhod	Pojasnilo
23132	74943	23132 → 20988 → 95931 → 85932 → 74943 → 62964 → 71973 → 83952 → 74943

Collatzevo zaporedje

Vhodne podatke preberete v zanki s standardnega vhoda.

```
if __name__ == '__main__':
    import sys
    for line in sys.stdin:
        print(max_collatz(int(line)))
```

Collatzevo zaporedje

Definicija Collatzevega zaporedja:

$$\text{collatz}(n) = \begin{cases} \frac{n}{2}, & \text{če je } n \text{ sodo število} \\ 3n + 1, & \text{sicer} \end{cases}$$

Collatzevo zaporedje $n_{i+1} = \text{collatz}(n_i)$, $n_0 = n$, $i \in \mathbb{N}$,

se je do sedaj za vsako izbrano začetno vrednost vedno končalo v ciklu $4 \rightarrow 2 \rightarrow 1$.

Definicija problema

Za dano začetno vrednost n poiščite največji člen Collatzevega zaporedja.

Vhodni podatki

V zanki preberite s standardnega vhoda začetne vrednosti n .

Na izhod izpišite največjo vrednost Collatzevega zaporedja za dano začetno vrednost.

Primeri

Vhod	Izhod
27	9232
123	628
245	736
3212	3616
45	136

Vsota cikla permutacije

Vhodne sezname preberete v zanki s standardnega vhoda in izpišete vsoto elementov cikla, ki vsebuje element 0 na standardni izhod.

```
if __name__ == '__main__':  
    import sys  
    for line in sys.stdin:  
        print(cikel0(eval(line)))
```

Permutacija

Permutacija je razporedba elementov seznama. Vseh možnih permutacij oziroma različnih vrstnih redov elementov danega seznama, ki ima različne elemente, je $n!$. Tu je n število elementov in $n! = 1 \times 2 \times \dots \times n$.

```
0 1 2 3 4 5 6 7 8 9  
[8, 2, 7, 4, 1, 0, 9, 3, 6, 5]
```

Cikel

```
8 -> 6 -> 9 -> 5 -> 0
```

Primer

Vhod

```
[8, 2, 7, 4, 1, 0, 9, 3, 6, 5]
```

Izhod

```
28
```

Koda TAKO

Vhodne podatke preberete v zanki s standardnega vhoda in izpišete na standardni izhod `True` oziroma `False`, odvisno od tega, ali je prišlo do napake ali ne.

```
if __name__ == '__main__':
    import sys
    for nim in sys.stdin:
        print(check_isbn(nim.strip()))
```

Kodo `TAKO` sestavlja 10 števk. Običajno se zapiše v obliki

`x-xxx-xxxxx-x`, kjer smo z `x` označili številke 0–9.

Štiri skupine števk so ločene z znakom `'-'`. Prva skupina vsebuje eno samo števko, sledi skupina treh in petih števk, na koncu je dodana še ena števka. Prvih 9 števk je poljubnih, zadnja pa je z njimi določena in jo imenujemo *preizkusna vsota*.

Če se pri prepisovanju kode zgodi napaka, lahko v večini primerov to ugotovimo.

Zadnja števka, preizkusna vsota, je izbrana tako, da je vsota produktov števil, ki pripadajo števkom po vrsti od leve proti desni s števili [10, 9, 8, 7, 6, 5, 4, 3, 2, 1], deljiva z 11. Za kodo `0-306-40615-2` izračunamo vsoto

$$10*0 + 9*3 + 8*0 + 7*6 + 6*4 + 5*0 + 4*6 + 3*1 + 2*5 + 2 = 132$$

Število 132 je deljivo z 11. $132 = 11 * 12$. Sklepamo, da do napake ni prišlo.

Definicija problema

Napiši funkcijo, ki preveri pravilnost vpisane kode `TAKO`.

Primeri

1. primer

Vhod

```
0-306-40615-2
```

Izhod

```
True
```

2. primer

Vhod

```
0-306-40615-3
```

Izhod

```
False
```