

Kaprekarjevo zaporedje

```
def kaprekar(x):
    n = len(x)
    mini = ''.join(sorted(x))          # uredi številke po velikosti v premem
    maxi = ''.join(sorted(x, reverse=True)) # in obratnem vrstnem redu
    x = str(int(maxi) - int(mini))     # x je razlika
    x = x.zfill(n)                    # doda vodilne ničle, da dolžina ostane n
    return x

def kaprekar_(n):
    visited = set(n)
    while True:
        n = kaprekar(n)
        if n in visited:
            break
        visited.add(n)
    return n

def kaprekar_cycle(n):
    nn = [kaprekar_(n)]
    while True:
        m = kaprekar(nn[-1])
        if m == nn[0]:
            break
        nn.append(m)
    return ' '.join(nn)

if __name__ == '__main__':
    import sys
    for n in sys.stdin:
        n = int(n)
        print(kaprekar_cycle(str(n)))
```

Koda TAKO

```
#!/usr/bin/env python3

def check_isbn(n):
    n = [int(x) for x in list(n) if x != '-']
    t = 0
    s = 0
    for i in range(10):
        t += n[i];
        s += t;
    return s % 11 == 0

if __name__ == '__main__':
    import sys
    line = sys.stdin.readline()
    print(check_isbn(line.strip()))
```

Filter praštevil

```
#!/usr/bin/env python3
from random import randint

def prime_filter(l):
    m = max(l)
    p = list(range(2,m+1))
    primes = []
    while p:
        primes.append(p[0])
        p = [x for x in p if x % primes[-1] != 0]
    return [x for x in l if x in primes]

if __name__ == '__main__':
    import sys
    line = sys.stdin.readline()
    print(sum(prime_filter(eval(line))))
```

Izštevanka 7

```
#!/usr/bin/env python3
import sys

def izstevanka7(n):
    nn = list(range(n))
    r = 0
    k = 1
    while len(nn) > 1:
        if k % 7 == 0 or k % 10 == 7:
            del nn[r]
        else:
            r = (r + 1)
            r = r % len(nn)
        k += 1
        if k > 99:
            k = 1
    return nn[0]

if __name__ == "__main__":
    import sys
    line = sys.stdin.readline()
    n = int(line)
    print(izstevanka7(n))
```

Verižni ulomki

...

!/usr/bin/env python

```
def verizni(v): a = 1 b = v[-1] v = v[::-1] for x in v[1:]: b, a = x * b + a, b return b, a
```

```
def inverse(a, b): res = [] if a < b: a, b = b, a res.append(0) while b > 0: res.append(a // b) a, b = b, a % b return res
```

```
if name == 'main': import sys line = sys.stdin.readline() line = eval(line) print(*verizni(line)) ``
```