

# Probleme cu tupluri

## Problema 1

Se citește de la tastatură un număr natural  $n$  nenul. Să se creeze și afișeze un tuplu format din numere naturale divizibile cu 2 și cu cifra unităților mai mică decât restul împărțirii la 9 a numărului respectiv mai mici sau egale decât  $n$ .

Exemplu: pentru  $n = 27$ , se va afișa  $(10, 12, 14, 16, 20, 22, 24, 26)$

Indicație: folosiți comprehensiunea de tupluri

```
In [ ]: n = int(input())
t = tuple(x for x in range(0, n + 1, 2) if x % 2 == 0 and x % 10 < x % 9)
print(t)
```

## Problema 2

În scopul acestei probleme, considerați reprezentarea de  $(a, b)$  pentru un număr complex. Se citesc de la tastatură numerele întregi  $a, b, c, d$ . Să se afișeze numărul complex  $(x, y)$  reprezentat de produsul numerelor complexe reprezentate de  $(a, b)$  și  $(c, d)$  și modulul acestuia calculat cu funcția `sqrt()`.

Exemplu: pentru  $(1,2)$  și  $(3,4)$ , produsul va fi  $(-5, 10)$  și modulul acestuia va fi aproximativ  $11,18$

```
In [ ]: import math

a = int(input())
b = int(input())
c = int(input())
d = int(input())
p = (a * c - b * d, a * d + c * b)
print(p, math.sqrt(sum(x ** 2 for x in p)))
```

## Problema 3

Se citesc de la tastatură 4 numere întregi,  $a, b, c$  și  $d$ , cu  $a < b$  și  $c < d$ . Să se afișeze intersecția și reuniunea intervalelor  $(a, b)$  și  $(c, d)$  sub formă de interval sau reuniune de intervale sau mulțime.

```
In [ ]: a = int(input())
b = int(input())
c = int(input())
d = int(input())
if a >= d or b <= c:
    inter = {} # vom considera aceasta ca fiind multimea vida, desi este un abuz de atribuire
    reunii = "reuniune disjuncta"
elif c < a:
    if b < d:
        inter = (a, b)
        reunii = (c, d)
    else:
        inter = (a, d)
        reunii = (c, b)
else:
    if b < d:
        inter = (c, b)
        reunii = (a, d)
    else:
        inter = (c, d)
        reunii = (a, b)
print(inter, reunii)
```

## Problema 4

Se citesc de la tastatură, pe o singură linie, un număr întreg  $a$ , un sir de numere întregi de lungime cel puțin 3 și un număr întreg  $b$ ,  $b > a$ . Să se afișeze câte elemente din interiorul sirului sunt cuprinse în intervalul  $(a, b)$ . Scoateți elementele cu valoare maximă din sir și afișați-l sub formă de tuplu.

Exemplu: pentru sirul  $12345678884234$

se va afișa

4

[2, 3, 4, 5, 6, 7, 4, 2, 3]

```
In [ ]: a, *t, b = (int(x) for x in input().split())
c = 0
mx = t[0]
for i in t:
    if a < i < b:
        c += 1
    if i > mx:
        mx = i
print(c)
ct = t.copy()
for i in ct: #folosim o copie pentru a nu modifica lista pe masura ce o parcurgem
    if i == mx:
        t.remove(i)
print(t)
```

## Problema 5

Se citesc de la tastatură două numere întregi  $a, b$  cu  $a < b$ , și un număr natural nenul  $r$ . Să se afișeze sub formă de tuplu termenii seriei aritmetice cu primul termen  $a$  și rația  $r$  care se găsesc în intervalul  $[a, b]$ . Să se afișeze suma lor.

*Indicație: nu folosiți formule matematice pentru sumă*

```
In [ ]: a = int(input())
b = int(input())
r = int(input())
t = tuple(a + r * i for i in range((b - a) // r + 1))
print(t, sum(t))
```

## Problema 6

Apreciați rezultatul următoarei secvențe de cod, cu precizările că  $a < b$ ,  $c < d$ ,  $b < d$  și  $a$  este diferit de  $c$ .

Un indicu poate fi găsit în comentariile acestei celule.

```
In [ ]: a = int(input())
b = int(input())
c = int(input())
d = int(input())
if a < c:
    if c < b < d:
        e, f = c, b
    elif d < b:
        e, f = c, d
    else:
        e, f = 0, -1
else:
    e, f = a, b
print(b - a + d - c - f + e + 1)
```