FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 4 · LECȚIA PY9-04

Cifrele unui număr

Durată: 100 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești algoritmi care desfac un număr în cifre și construiesc altul nou.
- Repari greșelile tipice: copia uitată, inițializarea minimului, p actualizat greșit, cazul n = 0.
- Scrii programe cu cifre în varianta aritmetică și le verifici, unde se poate, cu str().

Ce trebuie să știi (o pagină)

Operația	Python	Exemplu (`n = 4712`)
ultima cifră	`n % 10`	`2`
taie ultima cifră	`n //= 10`	`471`
ultimele `k` cifre	`n % 10**k`	`n % 100 = 12`
cifra de pe poziția `k` (de la dreapta, de la 0)	`n // 10**k % 10`	`k = 2` → `7`
prima cifră	`while n > 9: n //= 10`	`4`
adaugă `c` la dreapta	`n = n * 10 + c`	`c = 5` → `47125`
adaugă `c` la stânga	`n = c * p + n`, `p = 10**(nr. cifre)`	`c = 5` → `54712`
oglindit	`r = r * 10 + n % 10`	`2174`
număr nou, ordine păstrată	`r = c * p + r`, apoi `p *= 10`	
`n` poate fi 0	`while True:` … `if n == 0: break`	0 are o cifră
verificare cu text	`len(str(n))`, `int(str(n)[::-1])`	`4`, `2174`

SFAT

Înainte de buclă, întreabă-te: **mai am nevoie de `n` după buclă?** Dacă da, lucrează pe o copie (`m = n`).

Nivelul 1 · Prezice

★ citește și prezice

P1. Pentru `n = 70315`, ce dau `n % 1000`, `n // 1000`, `n // 10**3 % 10`, `n // 10 % 100` și `divmod(n, 100)`?

P2. Ce afișează pentru `n = 4096`?

```
s = 0
p = 1
while n > 0:
    s += n % 10
    p *= n % 10
    n //= 10
print(s, p)
```

P3. Ce afișează pentru `n = 52814`? Dar pentru `n = 918082`? Calculează și `n % 11` pentru fiecare. Ce observi?

```
s = 0
semn = 1
while n > 0:
    s += semn * (n % 10)
    semn = -semn
    n //= 10
print(s)
```

P4. Ce afișează `print(x, y)`?

```
x = 0
x = x * 10 + 3
x = x * 10 + 0
x = x * 10 + 7
y = 0
p = 1
y = 3 * p + y; p *= 10
y = 0 * p + y; p *= 10
y = 7 * p + y; p *= 10
```

P5. Ce valori au `a` și `b` la final, pentru `n = 90817`? Care dintre bucle calculează prima cifră?

```
a = n
while a >= 10:
    a //= 10
b = n
while b > 0:
    b //= 10
```

P6. De câte ori se execută bucla `while n > 0: n //= 10` pentru `n = 10**17`? Câte cifre are `2**64`?

P7. Ce afișează?

```
print(str(4050)[::-1], int(str(4050)[::-1]), len(str(10**5)))
print(str(12) + str(34), int(str(12) + str(34)) + 1)
```

P8. *Ca la Bac.* Ce se afișează pentru `n = 2769`? Dar pentru `906`? Pentru ce numere `n` se afișează chiar `n`?

```
citește n
m ← 0; p ← 1
cât timp n ≠ 0 execută
    c ← n % 10
    dacă c > 5 atunci c ← 9 − c
    m ← c · p + m; p ← p · 10
    n ← [n / 10]
scrie m
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul ar trebui să afișeze cifra minimă. Pentru `573` afișează `0`. Repară-l, astfel încât să meargă și pentru `n = 0`.

```python
n = int(input())
cmin = 0
while n > 0:
    if n % 10 < cmin:
        cmin = n % 10
    n //= 10
print(cmin)
```

P10. Verificarea de palindrom spune `NU` pentru `121`. De ce? Pentru ce număr afișează totuși `DA`? Repară-l.

```python
n = int(input())
r = 0
while n > 0:
    r = r * 10 + n % 10
    n //= 10
if r == n:
    print("DA")
else:
    print("NU")
```

P11. Programul șterge toate cifrele egale cu `k`, păstrând ordinea celorlalte. Pentru `n = 12321`, `k = 2` afișează `10301` în loc de `131`. Găsește rândul greșit.

```python
r = 0
p = 1
while n > 0:
    c = n % 10
    if c != k:
        r = c * p + r
    p *= 10
    n //= 10
print(r)
```

P12. *Rânduri amestecate.* După `n = int(input())`, ordonează rândurile ca programul să **dubleze fiecare cifră** (`352 → 335522`). Un rând este în plus. Ce afișează pentru `70`?

```
A)  r = (c * 10 + c) * p + r
B)  p *= 100
C)  c = n % 10
D)  n //= 10
E)  r, p = 0, 1
F)  while n > 0:
G)  p *= 10
H)  print(r)
```

P13. *Traducere din C++.* Codul C++ `while (n) { s += n % 10; n /= 10; }` (cu `n` de tip `int`) a fost tradus în Python astfel. Pentru `n = 123`, ce afișează? Câte pași face bucla? Ce s-a pierdut la traducere?

```python
s = 0
while n:
    s += n % 10
    n /= 10
print(s)
```

P14. Ambele rânduri dau eroare. Ce tip de eroare și cum le repari? (Pentru `583` trebuie să obții `16` și `9`.)

```python
print(sum(str(n)))          # suma cifrelor
print(max(str(n)) + 1)      # cifra maximă plus 1
```

Nivelul 3 · Scrie cod

★★ scrie cod

Scrie varianta aritmetică. Unde se poate, verifică rezultatul și cu `str()`.

P15. Cifra de control. Aduni cifrele lui `n`, apoi cifrele sumei și tot așa, până rămâne o singură cifră.

intrare	98765	0	9	10	999999999999
ieșire	8	0	9	1	9

P16. Contorul. Fiecare cifră crește cu 1, iar 9 devine 0 (ca la un contor mecanic, dar **fără** transport).

intrare	1929	9	0	345	99
ieșire	2030	0	1	456	0

P17. Capetele. Afișează numărul format din prima și ultima cifră a lui `n`.

intrare	52814	7	90	100
ieșire	54	77	90	10

P18. Cel mai mare număr din aceleași cifre. Fără liste: pentru fiecare cifră `c` de la 9 la 0, parcurgi `n` și adaugi la dreapta rezultatului fiecare apariție a lui `c`.

intrare	3920	1001	5	0	4747
ieșire	9320	1100	5	0	7744

P19. Palindrom. Afișează `DA` dacă `n` se citește la fel de la ambele capete, altfel `NU`. Pe rândul al doilea afișează verificarea cu text, `str(n) == str(n)[::-1]`.

intrare	12321	1221	120	7	0
ieșire	`DA` / `True`	`DA` / `True`	`NU` / `False`	`DA` / `True`	`DA` / `True`

P20. Șterge cifra a k-a. Citește `n` și `k` (`k` numărat **de la stânga**, de la 1). Afișează numărul fără acea cifră. *Indiciu:* cu `p = 10**(nr - k)`, unde `nr` este numărul de cifre, rezultatul este `n // (p * 10) * p + n % p`.

intrare (`n`, `k`)	`52814`, `2`	`52814`, `1`	`52814`, `5`	`7`, `1`	`1000`, `1`
ieșire	`5814`	`2814`	`5281`	`0`	`0`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Numărare inteligentă. Câte numere de trei cifre sunt egale cu oglinditul lor? Pentru câte numere de trei cifre oglinditul oglinditului **nu** este numărul inițial? Răspunde întâi prin raționament, apoi verifică cu un program.

P22. De ce funcționează? Arată că, pentru `n ≥ 1`, cifra de control de la P15 este `n % 9`, cu excepția multiplilor lui 9, unde este `9`. *Indiciu:* `10 = 9 + 1`, deci `10 % 9 = 1`, `100 % 9 = 1`, … Ce legătură are asta cu regula de divizibilitate cu 9?

P23. Când str() nu mai merge. Rulează `len(str(2**20000))`. Ce se întâmplă? Numără cifrele lui `2**20000` cu varianta aritmetică. Ce ordin de complexitate are bucla, în funcție de numărul de cifre?

P24. Constrângere. Din `n` elimini **exact o cifră**. Care este cel mai mare număr care se poate obține? Folosește ideea de la P20 pentru fiecare poziție. Teste: `52814 → 5814`, `1000 → 100`, `2919 → 919`, `10 → 1`, `9 → 0`.

P25. Mini-proiect: numărul cardului. Numerele de card se verifică cu **algoritmul lui Luhn**: pornind de la ultima cifră (poziția 1), cifrele de pe pozițiile **pare** se dublează, iar dacă rezultatul depășește 9, se scade 9. Se adună toate cifrele obținute; numărul este valid dacă suma se divide cu 10. Scrie programul (`VALID` / `INVALID`). Teste: `79927398713 → VALID`, `79927398710 → INVALID`, `4539578763621486 → VALID`, `1234567812345678 → INVALID`. Ce greșeli de tastare prinde verificarea (o cifră greșită? două cifre vecine inversate?)? **P26. Numere cu nume.** Un număr cu `k` cifre este *Armstrong* dacă suma cifrelor ridicate la puterea `k` este chiar el (`153 = 1³ + 5³ + 3³`). Un număr este *Harshad* dacă se divide cu suma cifrelor lui (`18`, pentru că `18 % 9 = 0`). Citește `n ≥ 1` și afișează pe două rânduri `Armstrong: DA` / `NU` și `Harshad: DA` / `NU`. Apoi, prin program, află numerele Armstrong de trei cifre și câte numere Harshad sunt între 1 și 100.

intrare	153	154	9474	5	10	1729
Armstrong	`DA`	`NU`	`DA`	`DA`	`NU`	`NU`
Harshad	`DA`	`NU`	`NU`	`DA`	`DA`	`DA`

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] #66 **numaruldecifre** — ușoară
  https://www.pbinfo.ro/probleme/66/numaruldecifre
- [ ] #69 **oglindit** — ușoară
  https://www.pbinfo.ro/probleme/69/oglindit
- [ ] #360 **cifra** — ușoară
  https://www.pbinfo.ro/probleme/360/cifra

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #68 **ciframaxima** — ușoară
  https://www.pbinfo.ro/probleme/68/ciframaxima
- [ ] #3932 **stergezerouri** — ușoară
  https://www.pbinfo.ro/probleme/3932/stergezerouri
- [ ] #368 **nrpal** — ușoară
  https://www.pbinfo.ro/probleme/368/nrpal
- [ ] #1059 **schimbcifre** — medie
  https://www.pbinfo.ro/probleme/1059/schimbcifre
