FIȘĂ DE LUCRU · CLASA A IX-A · FIȘA 5 · ACOPERĂ LECȚIA IX.9

Cifrele unui număr

Durată: 90 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești pe hârtie orice buclă care taie cifre și spui în cuvinte ce face.
- Construiești numere noi din cifre: oglindit, cifre păstrate în ordine (metoda cu p), cifre transformate.
- Prinzi capcanele: n = 0, copia uitată, cifra folosită după ce a fost tăiată, depășirea.

Ce trebuie să știi (o pagină)

Ce vrei	Cod	Exemplu (n = 52384)
ultima cifră / taie ultima cifră	`n % 10` / `n / 10`	`4` / `5238`
ultimele k cifre / taie k cifre	`n % 10^k` / `n / 10^k`	`n % 100 = 84`, `n / 100 = 523`
cifra de pe o poziție anume	`n / 10^k % 10`	`n / 100 % 10 = 3` (sutele)
parcurgere (cifrele ies de la dreapta)	`while (n > 0) { c = n % 10; …; n = n / 10; }`	4, 8, 3, 2, 5
și pentru n = 0	`do { … } while (n > 0);`	0 are o cifră
prima cifră	`while (n > 9) n = n / 10;`	`5`
oglindit	`og = og * 10 + c;`	`48325`
număr nou, ordinea inversă	`rez = rez * 10 + c;` (doar pentru cifrele păstrate)	pare: `482`
număr nou, aceeași ordine	`rez = rez + c * p; p = p * 10;` (doar pentru cifrele păstrate)	pare: `284`
start corect	suma 0, produsul 1, maximul 0, minimul 9	

ATENȚIE — GREȘELI FRECVENTE

Bucla îl „consumă” pe `n`: la final e 0. Dacă ai nevoie de el (palindrom, a doua parcurgere), salvează o **copie** înainte. Iar dacă rezultatul poate avea mai multe cifre decât încap în `int` (oglinditul lui 1 999 999 999), folosește `long long`.

Nivelul 1 · Prezice

★ citește și prezice

P1. Pentru `n = 70409`, completează tabelul.

`n % 100`	`n / 100 % 10`	`n / 1000`	`n % 1000 / 10`	`n / 10 % 10`	`n % 10000 / 1000`	`n / 10000 + n % 10`

P2. Ce afișează secvența pentru `n = 2468`? Dar pentru `n = 99999`? De câte ori se execută bucla în fiecare caz?

```
while (n > 9)
    n = n / 10 + n % 10;
cout << n;
```

P3. Pseudocod ca la BAC.

```
citește n (număr natural)
k ← 0
┌cât timp n > 9 execută
│   ┌dacă n % 10 > [n / 10] % 10 atunci
│   │   k ← k + 1
│   └■
│   n ← [n / 10]
└■
scrie k
```

a. Ce se afișează pentru `n = 13529`? b) Scrie cel mai mic și cel mai mare număr de **patru** cifre pentru care se afișează `3`.

P4. Aplici secvența de oglindire de **două ori** la rând (oglindești, apoi oglindești rezultatul). Ce obții pornind de la `1230`, `5005`, `9000`? Pentru ce numere nu te întorci de unde ai plecat?

P5. Completează tabelul pentru `n = 3572`, apoi spune ce face secvența.

```
int rez = 0, p = 1;
while (n > 0)
{
    int c = n % 10;
    if (c % 2 == 1) c = c - 1;
    rez = rez + c * p;
    p = p * 10;
    n = n / 10;
}
```

pas	`c` citit	`c` pus	`rez`	`p`	`n`
1					
2					
3					
4					

P6. Ce valoare are `x` la final pentru `n = 2580`? Dar pentru `n = 909`? Descrie **în cuvinte** ce face secvența.

```
int x = 0, p = 1;
while (n > 0) { x = x + (9 - n % 10) * p; p = p * 10; n = n / 10; }
```

P7. De câte ori se execută corpul fiecărei bucle?

`n`	`while (n > 0) n = n / 10;`	`do n = n / 10; while (n > 0);`
7		
10		
999		
1000		
0		

P8. Ce afișează secvența pentru `n = 123456`? Dar pentru `n = 7005`?

```
int s = 0;
while (n > 0) { s = s + n % 100; n = n / 100; }
cout << s;
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul trebuia să spună dacă `n` este palindrom. Pentru ce valori ale lui `n` afișează `DA`? Repară-l.

```
int n, og = 0;
cin >> n;
while (n > 0) { og = og * 10 + n % 10; n = n / 10; }
if (og == n) cout << "DA"; else cout << "NU";
```

P10. Programul numără cifrele pare. Pentru `246` afișează corect `3`, dar pentru `135` afișează `1`. Ce greșeală a făcut elevul? Mai dă un număr de o cifră pentru care greșește.

```
while (n > 0)
{
    n = n / 10;
    if (n % 10 % 2 == 0) k++;
}
```

P11. Elevul trebuia să **șteargă zerourile** din `n`, păstrând ordinea cifrelor (`50203 → 523`). Ce afișează programul lui pentru `50203`? Repară-l schimbând un singur rând și adăugând o variabilă.

```
int rez = 0;
while (n > 0)
{
    int c = n % 10;
    if (c != 0) rez = rez * 10 + c;
    n = n / 10;
}
cout << rez;
```

P12. *Rânduri amestecate.* Ordonează rândurile ca programul să construiască numărul format din cifrele lui `n` **mai mari decât 4**, în ordinea din `n`. Un rând este în plus. Ce obții pentru `70596`?

```
A)  rez = rez + c * p;
B)  n = n / 10;
C)  int rez = 0, p = 1;
D)  }
E)  if (c > 4) {
F)  rez = rez * 10 + c;
G)  while (n > 0) {
H)  p = p * 10;
I)  int c = n % 10;
J)  }
```

P13. Completează golurile astfel încât secvența să afișeze cifra **minimă** a lui `n`, corect **și pentru** `n = 0`.

```
int mn = ____;
do
{
    int c = ____;
    if (____) mn = c;
    n = ____;
} while (____);
cout << mn;
```

P14. Programul trebuia să afișeze cifra de control (9875 → 29 → 11 → 2), dar nu se mai oprește. Scrie primele 4 valori pe care le ia `n` după fiecare pas al buclei mari și explică de ce bucla nu se termină.

```
int s = 0;
while (n > 9)
{
    while (n > 0) { s = s + n % 10; n = n / 10; }
    n = s;
}
```

Nivelul 3 · Scrie cod

★★ scrie cod

Verifică **toate** testele înainte să spui că e gata.

P15. Biletul norocos. Un bilet de autobuz are 6 cifre (poate începe cu zerouri, deci `090009` se citește `90009`). Biletul e norocos dacă suma primelor trei cifre e egală cu suma ultimelor trei. Afișează `DA` / `NU`. Teste: `123402 → DA`, `111222 → NU`, `90009 → DA`, `0 → DA`.

P16. Codul seifului. Codul este **cel mai mare număr** care se poate forma cu exact aceleași cifre ca `n`. Indiciu: pentru fiecare cifră de la 9 la 0, numără de câte ori apare. Teste: `30725 → 75320`, `9090 → 9900`, `1000 → 1000`, `5 → 5`.

P17. Kilometrajul. Mașina are la bord `n` kilometri. Câți kilometri mai are de mers până când kilometrajul arată din nou un palindrom (strict mai mare decât `n`)? Teste: `12345 → 76`, `999 → 2`, `1221 → 110`, `0 → 1`.

P18. Numărul cardului (algoritmul Luhn). Pornind de la ultima cifră, dublezi fiecare a doua cifră (a 2-a, a 4-a… de la dreapta); dacă dublul trece de 9, scazi 9. Aduni toate cifrele obținute. Numărul e `VALID` dacă suma se termină în 0. Folosește `long long`. Teste: `79927398713 → VALID`, `79927398710 → INVALID`, `4539148803436467 → VALID`, `18 → VALID`.

P19. Jocul 7-BUM. Copiii numără de la 1 la `n`, dar în loc de orice număr divizibil cu 7 **sau** care conține cifra 7 strigă „BUM". Câte BUM-uri se strigă? Teste: `15 → 2`, `100 → 30`, `6 → 0`.

P20. Extrasul de cont. Afișează `n` (până la 10^{18}) cu punct între grupele de câte trei cifre. Atenție la grupele care încep cu zero.

intrare	`1234567`	`1000000`	`2026`	`999`	`0`
ieșire	`1.234.567`	`1.000.000`	`2.026`	`999`	`0`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. De ce funcționează? Secvența din P2 dă mereu același rezultat ca cifra de control (suma cifrelor, repetat), deși adună altceva. Explică de ce: scrie `n = 10·q + r` și compară `n` cu `q + r`. De ce se oprește bucla sigur?

P22. Oglinda care înmulțește. Există un număr de patru cifre al cărui oglindit este exact de 4 ori mai mare decât el. Scrie un program care îl găsește. Apoi încearcă toți factorii de la 2 la 9: pentru care mai există un astfel de număr?

P23. Palindrom fără oglindit. Verifică dacă un `int` este palindrom **fără să construiești oglinditul** (care poate să nu încapă în `int`). Indiciu: cu `p` = cea mai mare putere a lui 10 cel mult egală cu `n`, compari prima cifră (`n / p`) cu ultima, apoi le tai pe amândouă. Ce se întâmplă cu zerourile din interior, de exemplu la `1001` și `100101`?

P24. Ce se întâmplă dacă… citești `n = -123` în programul pentru suma cifrelor cu `while (n > 0)`? Dar dacă schimbi condiția în `while (n != 0)`? Repară programul ca să dea `6`.

P25. Cel mai mic număr cu suma cifrelor s. Pentru `s` dat (1 ≤ s ≤ 45), afișează cel mai mic număr natural care are suma cifrelor `s`, **fără să încerci numerele pe rând**. Teste: `20 → 299`, `10 → 19`, `9 → 9`, `45 → 99999`. Explică de ce construcția ta dă **cel mai mic** număr.

Probleme de lucru în clasă (pbinfo)

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

- [ ] `#69` **oglindit** — ușoară
  https://www.pbinfo.ro/probleme/69/oglindit
- [ ] `#368` **nrpal** — ușoară
  https://www.pbinfo.ro/probleme/368/nrpal
- [ ] `#360` **cifra** — ușoară
  https://www.pbinfo.ro/probleme/360/cifra

- [ ] #340 **control** — ușoară
  https://www.pbinfo.ro/probleme/340/control

Temă pentru acasă (pbinfo)

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

- [ ] #3932 **stergezerouri** — ușoară
  https://www.pbinfo.ro/probleme/3932/stergezerouri
- [ ] #4932 **numarnou** — ușoară
  https://www.pbinfo.ro/probleme/4932/numarnou
- [ ] #365 **cifre-comune** — ușoară
  https://www.pbinfo.ro/probleme/365/cifre-comune
- [ ] #4934 **semioglindit9** — medie
  https://www.pbinfo.ro/probleme/4934/semioglindit9
- [ ] #1059 **schimbcifre** — medie
  https://www.pbinfo.ro/probleme/1059/schimbcifre