FIȘĂ DE LUCRU · CLASA A IX-A · FIȘA 6 · ACOPERĂ LECȚIILE IX.10–IX.11

# Divizibilitate, numere prime, cmmdc, factori primi

Durată: 100 min  Nivel: toate profilurile

**CE VEI ȘTI DUPĂ LECȚIE**

- Parcurgi divizorii până la √n și verifici primalitatea fără greșelile clasice (1, pătratele perfecte, depășirea).
- Folosești Euclid pentru cmmdc, cmmmc, fracții și probleme „cu grupe egale" sau „cu întâlniri".
- Descompui în factori primi și citești din descompunere: numărul divizorilor, zerourile de la final, cmmdc.

## Ce trebuie să știi (o pagină)

| Ce vrei | Cod / regulă | Exemplu |
|---|---|---|
| `d` divide pe `n` | `n % d == 0` | |
| toți divizorii, în perechi | `for (d = 1; d * d <= n; d++) if (n % d == 0) { d; if (d != n / d) n / d; }` | 12: (1, 12), (2, 6), (3, 4) |
| număr impar de divizori | doar pătratele perfecte | 36 are 9 divizori |
| prim | `n >= 2` și niciun `d` cu `2 ≤ d`, `d * d <= n` | 1 nu e prim, 2 e singurul prim par |
| cmmdc (Euclid) | `while (b != 0) { r = a % b; a = b; b = r; }` → `a` | (48, 18) → 6 |
| cmmdc prin scăderi | `while (a != b) if (a > b) a -= b; else b -= a;` | buclă infinită dacă un număr e 0 |
| cmmmc | `a / cmmdc * b` (întâi împarți) | (4, 6) → 12 |
| factori primi | `while (d * d <= n) { e = 0; while (n % d == 0) { n /= d; e++; } … d++; }` apoi `if (n > 1)` | 360 = $2^3 \cdot 3^2 \cdot 5$ |
| numărul divizorilor | $(e_1 + 1)(e_2 + 1)\ldots$ | 360 → 4 · 3 · 2 = 24 |
| cmmdc / cmmmc din factori | exponenții minimi / maximi | |
| zerouri la final în n! | `n/5 + n/25 + n/125 + …` | 100! → 24 |
| numere mari | `long long d` dacă `n` e aproape de $2 \cdot 10^9$ | `d * d` depășește `int` |

**ATENȚIE — GREȘELI FRECVENTE**

Trei capcane care pică teste: `1` declarat prim, divizorul `√n` numărat de două ori și `a * b / cmmdc` calculat în `int`. Verifică mereu pe `n = 1`, pe un pătrat perfect și pe numere mari.

## Nivelul 1 · Prezice

★ citește și prezice

**P1.** Completează tabelul.

| `n` | divizorii | câți | suma lor | prim? |
|---|---|---|---|---|
| 20 | | | | |
| 25 | | | | |
| 29 | | | | |
| 1 | | | | |
| 49 | | | | |

**P2.** Calculează `cmmdc(100, 8)` cu ambele variante ale lui Euclid. Câți pași (iterații ale buclei) face fiecare? Ce crezi că se întâmplă pentru `(1 000 000, 1)`?

**P3.** *Ca la BAC.*

```
citește n                    (număr natural, n ≥ 2)
d ← 2
k ← 0
┌cât timp n > 1 execută
│   ┌dacă n % d = 0 atunci
│   │   n ← [n / d]
│   │   k ← k + 1
│   │altfel
│   │   d ← d + 1
│   └■
└■
scrie k
```

a. Ce se afișează pentru `n = 360`? Spune în cuvinte ce numără algoritmul.

b. Scrie **toate** numerele de două cifre pentru care se afișează `6`.

c. Care este cel mai mic număr de trei cifre pentru care se afișează `2`?

**P4.** Ce afișează secvența pentru `n = 36`? Dar pentru `n = 20`? Ce problemă are pentru unele valori ale lui `n`?

```cpp
for (int d = 1; d * d <= n; d++)
    if (n % d == 0) cout << d << ' ' << n / d << ' ';
```

**P5.** Fă tabelul `(a, b, r)` pentru Euclid prin împărțiri, cu `a = 1071, b = 462`. Cât sunt cmmdc și cmmmc?

**P6.** Cu câte zerouri se termină `125!`? Dar `2026!`? (Doar împărțiri, fără calculator.)

**P7.** Folosind descompunerea, câți divizori au `720`, `1001` și `1024`? Care este cel mai mic număr cu exact **6** divizori?

**P8.** *Ca la BAC.*

```
citește n                    (număr natural nenul)
k ← 0
┌pentru i ← 1, n execută
│   a ← n
│   b ← i
│   ┌cât timp b ≠ 0 execută
│   │   r ← a % b
│   │   a ← b
│   │   b ← r
│   └■
│   ┌dacă a = 1 atunci
│   │   k ← k + 1
│   └■
└■
scrie k
```

a. Ce se afișează pentru `n = 12`? Dar pentru `n = 10`? Ce numără algoritmul?

b. Pentru ce numere de două cifre se afișează `n - 1`?

c. Scrie toate valorile lui `n` pentru care se afișează `4`.

## Nivelul 2 · Găsește și repară

★★ găsește și repară

**P9.** Verificarea de primalitate de mai jos greșește pentru cinci valori din `[1, 50]`. Care sunt? Repară cele **două** greșeli.

```cpp
bool prim = true;
for (int d = 2; d * d < n; d++)
    if (n % d == 0) prim = false;
```

**P10.** Elevul a mutat un rând în Euclid. Ce afișează pentru `(12, 18)`? Dar pentru `(18, 12)`? Ce calculează, de fapt? Repară.

```cpp
while (b != 0) { a = b; int r = a % b; b = r; }
cout << a;
```

**P11.** Programul trebuia să afișeze factorii primi ai lui `n`, fiecare de câte ori apare. Ce afișează pentru `360`? Dar pentru `12` și `8`? De ce poate apărea chiar un număr care **nu e prim**? Repară schimbând un singur cuvânt.

```cpp
for (int d = 2; d <= n; d++)
    if (n % d == 0) { cout << d << ' '; n /= d; }
```

**P12.** Pentru `a = 60000, b = 90000` (variabile `int`), `cout << a * b / cmmdc(a, b);` afișează un rezultat greșit, deși `cmmdc` este corect. De ce? Rescrie expresia fără `long long`. Ce trebuie să afișeze?

**P13.** *Rânduri amestecate.* Programul simplifică fracția `a/b` (pentru `45 60` afișează `3/4`). Ordonează rândurile; unul este în plus.

```
A)  int r = x % y; x = y; y = r;
B)  cout << a / y << '/' << b / y;
C)  int a, b; cin >> a >> b;
D)  }
```

```
E)  while (y != 0) {
F)  cout << a / x << '/' << b / x;
G)  int x = a, y = b;
```

**P14.** *Completează golurile.* Secvența afișează **cel mai mare divizor al lui `n` mai mic decât `n`** (`n ≥ 2`), în cel mult √n pași. Teste: `91 → 13`, `97 → 1`, `100 → 50`.

```
int d = 2;
while (_________ && _________) d++;
if (d * d > n) cout << ____; else cout << ____;
```

## Nivelul 3 · Scrie cod

★★ scrie cod

Pentru fiecare program, verifică **toate** testele date înainte să spui că e gata.

**P15. Echipele.** La o tabără sunt `a` băieți și `b` fete. Vrem cât mai multe echipe, toate cu **aceeași** componență, fără să rămână nimeni pe dinafară. Afișează numărul de echipe, câți băieți și câte fete are fiecare.

| **intrare** | **`12 18`** | **`7 5`** | **`100 75`** | **`24 24`** |
|---|---|---|---|---|
| ieșire | `6 2 3` | `1 7 5` | `25 4 3` | `24 1 1` |

**P16. Trei semafoare.** Trei semafoare se fac verzi împreună la momentul 0 și apoi o dată la `a`, `b`, respectiv `c` secunde. După câte secunde se fac din nou verzi toate trei deodată? (`a, b, c ≤ 10⁶`)

| **intrare** | **`4 6 10`** | **`7 11 13`** | **`5 5 5`** | **`1000000 999999 999998`** |
|---|---|---|---|---|
| ieșire | `60` | `1001` | `5` | `499998500001000000` |

**P17. Prime gemene.** Afișează prima pereche de numere prime gemene `p, p + 2` cu `p ≥ n`. Teste: `1 → 3 5`, `6 → 11 13`, `12 → 17 19`, `100 → 101 103`, `3 → 3 5`.

**P18. Perfect, abundent sau deficient?** Compară `n` cu suma divizorilor lui **mai mici decât el**: egală → `perfect`, mai mare → `abundent`, mai mică → `deficient`. Lucrează în O(√n). Teste: `6 → perfect`, `12 → abundent`, `7 → deficient`, `1 → deficient`, `945 → abundent`, `33550336 → perfect`.

**P19. Descompunerea frumoasă.** Afișează descompunerea în formatul de mai jos (exponentul 1 nu se scrie). Trebuie să meargă instant pentru `n ≤ 2·10⁹`.

| **intrare** | **ieșire** |
|---|---|
| `360` | `360 = 2^3 * 3^2 * 5` |
| `1024` | `1024 = 2^10` |
| `97` | `97 = 97` |
| `2147483646` | `2147483646 = 2 * 3^2 * 7 * 11 * 31 * 151 * 331` |

**P20. Zerourile produsului.** Se citesc `n` și `n` numere naturale nenule (până la $10^9$). Cu câte zerouri se termină produsul lor? **Nu** calcula produsul.

| intrare | `3` / `10 25 4` | `2` / `125 8` | `3` / `2 3 7` | `2` / `5 5` | `4` / `1000000000` (de 4 ori) |
|---|---|---|---|---|---|
| ieșire | `3` | `3` | `0` | `0` | `36` |

## Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

**P21. Problema inversă.** Care este cel mai mic `n` pentru care `n!` se termină cu cel puțin 100 de zerouri? Există vreun `n` pentru care `n!` se termină cu **exact 5** zerouri? Explică ce se întâmplă la trecerea de la `24!` la `25!`.

**P22. De ce funcționează?** Arată că `cmmdc(a, b) · cmmmc(a, b) = a · b`, comparând exponenții fiecărui factor prim (indiciu: `min(x, y) + max(x, y) = ?`). Verifică pe 360 și 324.

**P23. Contraexemplul.** Un elev spune: „un număr este prim dacă nu se divide cu 2, 3, 5 și 7". Care este cel mai mic număr **compus** pentru care regula greșește? Pentru ce alte numere până la 10 mai greșește? Până la ce limită ar fi corectă regula dacă ar trata separat numerele 1, 2, 3, 5, 7?

**P24. Suma divizorilor fără divizori.** Dacă `n = p^e · …`, suma divizorilor este produsul, pe factori, al sumelor `1 + p + p² + … + p^e`. Scrie programul care o calculează **în timpul descompunerii** (O(√n)) și verifică pe `360` (rezultat `1170`) și pe `10⁶`.

**P25. Mini-proiect: numere prietene.** Două numere sunt prietene dacă fiecare este egal cu suma divizorilor proprii ai celuilalt (ca 220 și 284). Găsește **toate** perechile mai mici decât 10 000. De ce e important să calculezi suma divizorilor în O(√n) și nu în O(n)?

## Probleme de lucru în clasă (pbinfo)

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

- [ ] **#45 verifprim** — ușoară
  https://www.pbinfo.ro/probleme/45/verifprim
- [ ] **#376 suma-divizori** — ușoară
  https://www.pbinfo.ro/probleme/376/suma-divizori
- [ ] **#58 cmmdc** — ușoară
  https://www.pbinfo.ro/probleme/58/cmmdc
- [ ] **#62 factorizare** — ușoară
  https://www.pbinfo.ro/probleme/62/factorizare

## Temă pentru acasă (pbinfo)

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

- [ ] #391 **fractii** — ușoară
  https://www.pbinfo.ro/probleme/391/fractii
- [ ] #64 **numarperfect** — ușoară
  https://www.pbinfo.ro/probleme/64/numarperfect
- [ ] #439 **factzero** — ușoară
  https://www.pbinfo.ro/probleme/439/factzero
- [ ] #63 **factorizare1** — ușoară
  https://www.pbinfo.ro/probleme/63/factorizare1

---