FIȘĂ DE LUCRU · CLASA A IX-A · FIȘA 3 · ACOPERĂ LECȚIILE IX.5–IX.6

Bucle: while, do-while, for

Durată: 100 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Numeri exact de câte ori se execută o buclă și ce valoare are contorul la ieșire.
- Alegi între while, do-while și for și traduci corect repetă … până când.
- Folosești break, continue și bucle imbricate (desene, tabele, numărări pe perechi).

Ce trebuie să știi (o pagină)

Regula	Cod / exemplu	Reține
`while` : testul **înainte**	`while (c) { ... }`	corpul poate rula de 0 ori
`do-while` : testul **după**	`do { ... } while (c);`	corpul rulează cel puțin o dată; `;` la final
`for` = `while` scris compact	`for (i = a; i <= b; i++)` = `i = a; while (i <= b) { corp; i++; }`	la ieșire `i = b + 1`
câte execuții are `for (i = a; i <= b; i += k)`	⌊(b − a) / k⌋ + 1 (dacă a ≤ b)	`i = 1, 4, 7, 10` → 4
`repetă … până când C`	`do { ... } while (!C);`	condiția se **neagă**
`cât timp C execută …` cu `repetă`	`dacă C atunci repetă … până când not C`	`dacă` -ul acoperă cazul cu 0 execuții
`break`	iese din bucla **cea mai apropiată**	la bucle imbricate, doar din cea interioară
`continue`	sare la următoarea iterație	în `for` se execută pasul; în `while` **nu**
bucle imbricate	interioara rulează **complet** pentru fiecare pas al exterioarei	desen: exterioara = rânduri
câte execuții are corpul interior	`j = 1..n` → n^2; `j = 1..i` → $n(n+1)/2$	n = 4: 16, respectiv 10

ATENȚIE — GREȘELI FRECVENTE

O buclă se termină doar dacă variabila din condiție **se apropie** de oprire la fiecare pas. Verifică întotdeauna: se modifică? poate „sări” peste valoarea de oprire?

Nivelul 1 · Prezice

★ citește și prezice

P1. Completează tabelul: de câte ori se execută corpul și ce valoare are `i` după buclă?

secvența	execuții	`i` la final
`for (i = 0; i < 20; i += 4) ...`		
`for (i = 20; i > 1; i /= 2) ...`		
`for (i = 1; i <= 100; i *= 3) ...`		
`for (i = 10; i < 10; i++) ...`		
`i = 10; do { i++; } while (i < 10);`		
`for (i = -3; i <= 3; i += 2) ...`		

P2. Două contoare care se apropie. Ce afișează?

```
for (int i = 1, j = 10; i < j; i += 2, j -= 3)
    cout << i << ',' << j << ' ';
```

P3. Același corp, două bucle. Ce afișează? Ce afișează dacă `x` și `y` pornesc de la 30? Pentru ce valori de start dau cele două bucle rezultate **diferite**?

```
int x = 50, k1 = 0, k2 = 0;
while (x < 50) { x += 7; k1++; }
int y = 50;
do { y += 7; k2++; } while (y < 50);
cout << k1 << ' ' << x << ' ' << k2 << ' ' << y;
```

P4. `break` și `continue` în același `while`. Ce afișează? Fă un tabel cu `i` și `s`.

```
int i = 0, s = 0;
while (i < 10)
{
    i++;
    if (i % 3 == 0) continue;
    if (s > 12) break;
    s += i;
}
cout << i << ' ' << s;
```

P5. Desenează ce afișează pentru `n = 4`. Ce formă ar avea pentru `n = 5`?

```
for (int i = 1; i <= n; i++)
{
    for (int j = 1; j <= n; j++)
        if (i == j || i + j == n + 1) cout << 'X';
        else cout << '.';
    cout << '\n';
}
```

P6. *Ca la BAC.* Ce se afișează pentru `n = 5`? Dar pentru `n = 8`?

```
citește n                    (număr natural)
k ← 0
i ← 1
┌cât timp i ≤ n execută
│   j ← i
│   ┌cât timp j ≤ n execută
│   │   k ← k + 1
│   │   j ← j * 2
│   └■
│   i ← i + 1
└■
scrie k
```

P7. Ce afișează? Completează un tabel cu `a`, `b` după fiecare iterație.

```cpp
int a = 1, b = 100, pasi = 0;
do
{
    a = a * 2;
    b = b - a;
    pasi++;
} while (a < b);
cout << a << ' ' << b << ' ' << pasi;
```

P8. *Grilă.* Care secvențe afișează exact același lucru ca `for (i = 1; i <= 5; i++) cout << i;`?

S1) `i = 1; while (i <= 5) { cout << i; i++; }`
S2) `i = 0; while (i < 5) { i++; cout << i; }`
S3) `i = 1; while (i <= 5) { i++; cout << i; }`
S4) `i = 1; do cout << i++; while (i < 5);`
S5) `i = 6; while (--i) cout << 6 - i;`

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Pentru `n = 4` trebuie afișat `1,2,3,4` (fără virgulă la final), dar programul afișează `1,2,3,4,`. Repară-l în **două** feluri: o dată cu un `if` în buclă, o dată fără niciun `if`.

```cpp
for (int i = 1; i <= n; i++) cout << i << ',';
```

P10. Programul vrea suma numerelor de la 1 la 10, sărind multiplii lui 4, dar nu se mai oprește. De ce? Repară-l și spune ce sumă trebuie să afișeze.

```cpp
int i = 1, s = 0;
while (i <= 10)
{
    if (i % 4 == 0) continue;
    s += i;
    i++;
}
cout << s;
```

P11. Trebuie să deseneze un dreptunghi cu `n` rânduri și `m` coloane de `*`. Ce afișează pentru `n = 2, m = 3`? Repară.

```
for (int i = 1; i <= n; i++)
    for (int j = 1; j <= m; j++)
        cout << '*';
    cout << '\n';
```

P12. *Puzzle cu rânduri amestecate.* Aranjează rândurile ca programul să deseneze triunghiul aliniat la dreapta (exemplu pentru `n = 4`). Un rând este în plus.

```
A)      for (int j = 1; j <= i; j++) cout << '*';
B)  }
C)  int n; cin >> n;
D)      for (int j = 1; j <= i - 1; j++) cout << ' ';
E)  for (int i = 1; i <= n; i++) {
F)      cout << '\n';
G)      for (int j = 1; j <= n - i; j++) cout << ' ';
```

```
   *
  **
 ***
****
```

P13. *Transcriere ca la BAC.* Un elev a tradus pseudocodul în `do { ... } while (p > n);`. Ce afișează programul lui pentru `n = 10`? Ce ar trebui să afișeze? Repară condiția, apoi scrie un algoritm echivalent cu `cât timp`. Este nevoie de un `dacă` în față?

```
citește n                    (n număr natural, n ≥ 1)
p ← 1
k ← 0
┌repetă
│   p ← p * 3
│   k ← k + 1
└până când p > n
scrie k, " ", p
```

P14. *Completează golurile.*

a. Pentru `n = 10` să afișeze `10 8 6 4 2`, iar pentru `n = 9` să afișeze `9 7 5 3 1`: `for (int i = ____; _________; _________) cout << i << ' ';`

b. Pentru `n = 100` să afișeze `1 2 4 8 16 32 64`: `int i = 1; while (_________) { cout << i << ' '; i = _________; }`

Nivelul 3 · Scrie cod

★★ scrie cod

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

P15. Pușculița. Ana pune 5 lei în prima săptămână și în fiecare săptămână cu 2 lei mai mult decât în cea dinainte (5, 7, 9, ...). După câte săptămâni are cel puțin `S` lei?

`S`	5	12	100	1
săptămâni	`1`	`2`	`9`	`1`

P16. PIN-ul. Utilizatorul are 3 încercări să introducă un PIN valid (un număr cu 4 cifre, între 1000 și 9999). Folosește `do-while`. Afișează `Acceptat la incercarea k` sau, după 3 încercări greșite, `Card`

`blocat`.

intrare	ieșire
`12 99999 4321`	`Acceptat la incercarea 3`
`1000`	`Acceptat la incercarea 1`
`5 6 7`	`Card blocat`

P17. Tabla de șah. Desenează o tablă `n × n` cu `#` pe pătratele în care `i + j` e par și `.` în rest (`i`, `j` pornesc de la 1). Teste: `n = 1` → `#`; `n = 3` → `#.#` / `.#.` / `#.#` (pe trei rânduri).

P18. Ceasul digital. Câte momente `hh:mm` dintr-o zi (00:00 … 23:59) au `hh + mm = s`? Folosește două bucle imbricate.

s	0	82	30	70
momente	1	1	24	13

P19. Scara de numere. Afișează primii `n` termeni ai șirului `1 2 2 3 3 3 4 4 4 4 …` (fiecare `v` apare de `v` ori). Atenție: te poți opri în mijlocul unui grup.

n	ieșire
`7`	`1 2 2 3 3 3 4`
`1`	`1`
`10`	`1 2 2 3 3 3 4 4 4 4`

P20. Playlist pe repeat. Un playlist are `k` melodii numerotate 1…k și se reia de la capăt după ultima. Pornești de la melodia `s`. Afișează numerele primelor `n` melodii ascultate. Intrare: `k s n`.

intrare	`4 3 6`	`5 1 3`	`3 3 7`
ieșire	`3 4 1 2 3 4`	`1 2 3`	`3 1 2 3 1 2 3`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Problema inversă. Pentru ce valori naturale ale lui `x` secvența afișează `3`? Dar `4`? Ce regulă observi?

```
int k = 0;
while (x > 1) { x = (x + 1) / 2; k++; }
cout << k;
```

P22. O singură buclă. Afișează tabla înmulțirii `n × n` (ca în lecție) folosind **un singur** `for`, cu contorul `t` de la 0 la `n·n − 1`. Indiciu: din `t` poți calcula rândul și coloana.

P23. Rădăcina pătrată fără `sqrt`. Ce afișează pentru `n = 30`? Ce calculează, în general? **De ce funcționează?** (Indiciu: cât fac 1 + 3, 1 + 3 + 5, 1 + 3 + 5 + 7?)

```cpp
int k = 0, impar = 1;
while (n >= impar) { n -= impar; impar += 2; k++; }
cout << k;
```

P24. Două bucle imbricate = n^2? Nu mereu. Rulează algoritmul de la P6 pentru `n = 1000` și `n = 100000`. Compară `k` cu `n²` și cu `n`. Explică rezultatul: câte valori ale lui `i` fac un singur pas interior? Câte fac doi pași?

P25. Mini-proiect: ghicește numărul. Programul citește întâi numărul secret (ca să-l poți testa), apoi încercările jucătorului. După fiecare încercare greșită afișează `mai mare` sau `mai mic`, iar la final `Ai ghicit din k incercari`. Test: `37 50 25 37` → `mai mic`, `mai mare`, `Ai ghicit din 3 incercari`. Apoi gândește: dacă secretul e între 1 și 100, care e cea mai bună strategie și câte încercări îți ajung **sigur**?

Probleme de lucru în clasă (pbinfo)

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

- [ ] #459 **suma1** — ușoară
  https://www.pbinfo.ro/probleme/459/suma1
- [ ] #348 **afisareputeri** — ușoară
  https://www.pbinfo.ro/probleme/348/afisareputeri
- [ ] #351 **piramida** — ușoară
  https://www.pbinfo.ro/probleme/351/piramida
- [ ] #339 **numere** — ușoară
  https://www.pbinfo.ro/probleme/339/numere

Temă pentru acasă (pbinfo)

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

- [ ] #331 **afisarenumereimpare** — ușoară
  https://www.pbinfo.ro/probleme/331/afisarenumereimpare
- [ ] #326 **ultimulpar** — ușoară
  https://www.pbinfo.ro/probleme/326/ultimulpar
- [ ] #350 **produscartezian** — ușoară
  https://www.pbinfo.ro/probleme/350/produscartezian
- [ ] #457 **patrat** — ușoară
  https://www.pbinfo.ro/probleme/457/patrat
- [ ] #2530 **desenclepsidra** — ușoară
  https://www.pbinfo.ro/probleme/2530/desenclepsidra