INFORMATICĂ · CLASA A X-A · RECAPITULARE · LECȚIA 13

# Recapitulare: materia clasei a X-a în formatul Bac

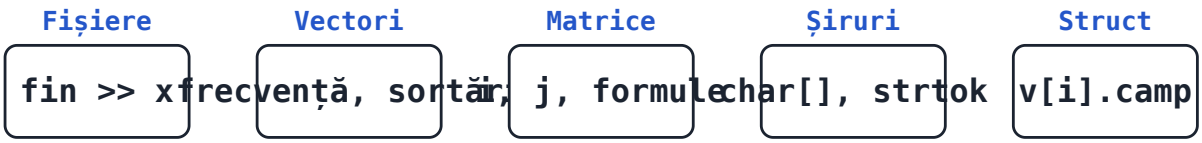
Durată: 60 min  Nivel: mate-info (Bac)  Înainte: lecțiile x-01, x-07, x-09, x-11, x-12

**CE VEI ȘTI DUPĂ LECȚIE**

- Rezolvi itemii II.2 și II.3 (șiruri de caractere, structuri, matrice) în timp de examen.
- Rezolvi o problemă III.3 cu fișier și algoritm eficient, inclusiv descrierea metodei.
- Îți faci o listă proprie de greșeli de evitat.

## 1. Ideea: ai deja uneltele, trebuie să le alegi

Clasa a X-a ți-a dat cinci „unelte". La Bac nu ți se spune pe care s-o folosești, de aceea o parte din lucru este să **recunoști tiparul** din enunț.




la Bac le combini: text din fișier, vector de structuri, matrice construită cu formulă

*Cele 5 unelte*

| Dacă în enunț vezi... | Folosești | Lecția |
|---|---|---|
| „fișierul `bac.txt` conține...", până la $10^6$ numere | `ifstream`, `while (fin >> x)`, **fără vector** | X.1 |
| „elementele tabloului", ștergere, inserare | vector, doi indici | X.3 |
| „valori mici", „de câte ori apare" | vector de frecvență | X.6 |
| „secvență", „consecutive" | o parcurgere cu `lung`, `lmax` | X.7 |
| „n linii și n coloane", „diagonală", „construiește tabloul" | matrice, formula `a[i][j]` | X.8, X.9 |
| „cuvânt", „text", „litere", „vocale" | `char[]`, `strtok`, ASCII | X.10, X.11 |
| „nume, an, preț", „variabila memorează datele despre..." | `struct`, vector de structuri | X.12 |

## 2. Planul de lucru pentru un subiect

| Subiect | Timp orientativ (din 180 min) | Sfat |
|---|---|---|
| I (grile) | 25–30 min | tabel de valori pe ciornă; elimină variantele imposibile |
| II.1 (pseudocod) | 25 min | a) tabel, b) verifică valorile, c) program complet, d) atenție la condiția negată |
| II.2 + II.3 | 25 min | declarări corecte, citește exact numele variabilelor date |
| III.1 (subprogram) | 25 min | doar definiția subprogramului, fără `main` dacă nu se cere |
| III.2 (program) | 25 min | citire, prelucrare, afișare exact în formatul cerut |
| III.3 (eficiență) | 30 min | a) descriere + justificare, b) program care citește din fișier |
| verificare | 15 min | relecturează `;`, acolade, tipuri, numele fișierului |

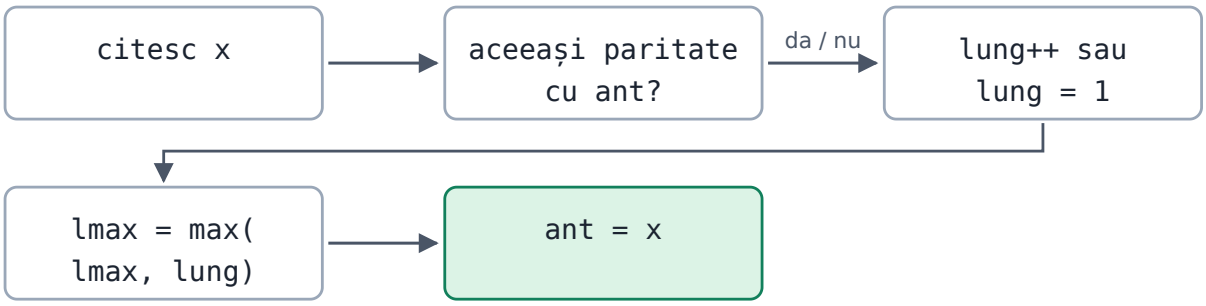
**DE CE AȘA?**

**De ce Subiectul I primește cel mai puțin timp pe item?** Fiecare grilă se rezolvă cu un tabel de valori pe ciornă, în 2–3 minute. Dacă rămâi blocat la una, treci mai departe și revii: punctajul unei grile este același cu al oricăreia dintre celelalte.

## 3. Problemă tip III.3 rezolvată

**Enunț.** Fișierul `bac.txt` conține un șir de cel mult $10^6$ numere naturale din intervalul $[1, 10^9]$. Se cere lungimea maximă a unei secvențe de numere consecutive din șir care au aceeași paritate. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei utilizate.

**a) Descrierea metodei.** Se citesc numerele din fișier unul câte unul, păstrând doar numărul anterior. Pentru fiecare număr citit se verifică dacă are aceeași paritate cu anteriorul: dacă da, lungimea secvenței curente crește cu 1, altfel lungimea redevine 1. După fiecare actualizare se compară lungimea curentă cu maximul. Algoritmul este eficient ca timp, deoarece datele sunt parcurse o singură dată (complexitate liniară), și ca memorie, deoarece se folosesc doar câteva variabile simple, indiferent de numărul de valori din fișier.




repetă pentru fiecare număr din fișier; la final afișezi lmax

*Metoda*

**b) Programul.**

```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ifstream fin("bac.txt");
    int ant, x, lung = 1, lmax = 1;
    fin >> ant;
    while (fin >> x)
    {
        if (x % 2 == ant % 2) lung++;
        else lung = 1;
        if (lung > lmax) lmax = lung;
        ant = x;
    }
    cout << lmax;
    return 0;
}
```

**DE CE AȘA?**

**De ce nu citim totul într-un vector?** Pot fi $10^6$ numere; un vector cu $10^6$ `int` ocupă ~4 MB, ceea ce încape, dar enunțul cere explicit *memorie eficientă*. Algoritmul are nevoie doar de numărul anterior, deci păstrăm doar atât. **De ce `lung = 1` (nu 0) la ruperea secvenței?** Numărul curent începe o secvență nouă, care are deja un element.

**Urmărire** pentru `3 5 2 4 6 7`:

| x | ant | aceeași paritate? | lung | lmax |
|---|---|---|---|---|
| (3 citit în `ant`) | | | 1 | 1 |
| 5 | 3 | da (impare) | 2 | 2 |
| 2 | 5 | nu | 1 | 2 |
| 4 | 2 | da (pare) | 2 | 2 |
| 6 | 4 | da | 3 | 3 |

| x | ant | aceeași paritate? | lung | lmax |
|---|---|---|---|---|
| 7 | 6 | nu | 1 | 3 |

Se afișează 3 (secvența `2 4 6`).

**Intrare (conținutul fișierului) / ieșire**

| Fișier | Ieșire |
|---|---|
| `3 5 2 4 6 7` | `3` |
| `8` | `1` |

LA BAC

Un algoritm care memorează toate numerele într-un vector și apoi le parcurge primește punctaj parțial la eficiență (memorie neconstantă). Unul cu două bucle imbricate ($O(n^2)$) pierde și punctajul de timp. Descrierea metodei **în cuvinte** (varianta a) se punctează separat: scrie-o chiar dacă programul nu iese.

## 4. Greșelile care costă cele mai multe puncte

ATENȚIE — GREȘELI FRECVENTE

> - Numele fișierului scris greșit sau citit cu `cin` în loc de `fin`. - Afișare în alt format decât cel cerut (spații, rânduri noi, mesaje în plus). - Variabile neinițializate (sume, contoare, maxime). - Depășire de tip: produse și sume mari în `int`. - `strtok` care distruge textul de care mai ai nevoie. - La matrice: inversarea lui `n` (linii) cu `m` (coloane). - Uitarea cazului „nu există" (afișează mesajul cerut!). - Terminatorul `'\0'` uitat după ștergeri în șiruri.

## 5. Mini-test de Bac (60 de minute)

Rezolvă pe hârtie, apoi verifică la final. Fiecare grilă are o singură variantă corectă.

### Subiectul I (8 grile)

**1.** Ce afișează?

```
int v[] = {4, 7, 1, 9}, i = 0;
while (v[i] % 2 == 0) i++;
cout << i;
```

a) 0 b) 1 c) 2 d) 3

**2.** Fișierul `a.in` conține `3 8 5 6 10`. Ce afișează?

```
ifstream fin("a.in"); int x, c = 0;
while (fin >> x) if (x > 5) c++;
cout << c;
```

a) 2 b) 3 c) 4 d) 5

**3.** Matricea 4×4 are `a[i][j] = i - j` (indici de la 1). Care este suma elementelor de pe diagonala secundară? a) −6 b) 0 c) 6 d) 4

**4.** Ce afișează?

```
char s[] = "informatica";
cout << strlen(s) << ' ' << s[strlen(s) - 1];
```

a) `10 a` b) `11 a` c) `11 \0` d) `12 a`

**5.** Ce afișează?

```
char s[20] = "bac";
strcat(s, "2026");
cout << strlen(s);
```

a) 4 b) 6 c) 7 d) 20

**6.** Se dă `struct P { int x, y; } a = {2, 5}, b = {4, 1};`. Care expresie are valoarea 1? a) `a == b` b) `a.x > b.x` c) `a.x < b.x && a.y > b.y` d) `P.x == 2`

**7.** În tabloul `(2, 4, 6, 8, 10, 12, 14)` se caută `x = 12` cu căutare binară (`st = 1`, `dr = 7`, `mij = (st + dr) / 2`). De câte ori se compară `x` cu un element al tabloului până la găsire? a) 1 b) 2 c) 3 d) 4

**8.** Pentru `a = (3, 1, 4, 1, 5)` și `s[i] = s[i-1] + a[i]`, care este suma `a[2] + a[3] + a[4]`, calculată ca `s[4] - s[1]`? a) 5 b) 6 c) 8 d) 9

### Subiectul II (3 exerciții)

**II.1 (șiruri).** Un text are cel mult 100 de caractere; cuvintele sale, formate din litere mici, sunt separate prin câte un spațiu. Scrie un program care afișează, pe câte o linie, cuvintele care sunt palindroame. Dacă nu există, afișează `nu exista`.

**II.2 (struct).** Se citesc `n` (≤ 100) produse, fiecare cu denumire (un cuvânt) și preț (număr întreg). Afișează denumirile produselor cu prețul maxim, în ordinea citirii.

**II.3 (matrice).** Se citește `n` (2 ≤ n ≤ 20). Construiește în memorie un tablou cu `n` linii și `n` coloane în care elementele de pe cele două diagonale sunt egale cu 1, iar celelalte cu 0. Afișează tabloul.

### Subiectul III (eficiență)

Fișierul `numere.txt` conține cel mult $10^6$ numere naturale. Se cere lungimea celei mai lungi secvențe **strict crescătoare** de elemente consecutive. a) Descrie metoda; b) scrie programul.

## 6. Greșeli frecvente (fișa de reținut)

| Greșeala | Efect | Corect |
|---|---|---|
| vector de $10^6$ elemente „pentru siguranță” | memorie / punctaj eficiență | doar variabile simple |
| `cin` în loc de `fin` | nu citește fișierul | `fin >> x` |
| linii vs. coloane inversate | tablou transpus | `i` pentru linii, `j` pentru coloane |

| Greșeala | Efect | Corect |
|---|---|---|
| `==` pe șiruri / structuri | compară adrese / nu compilează | `strcmp`, câmp cu câmp |
| uitat „nu exista” | pierzi punctul de la caz particular | tratează explicit |

## 7. La BAC

> **LA BAC**
>
> Barem: pentru un program complet se acordă puncte pe etape (declarări, citire, prelucrare, afișare). Chiar un program cu o mică greșeală de logică primește majoritatea punctelor dacă etapele sunt vizibile. Nu lăsa foaia goală.

## 8. Mai departe (pentru cine vrea mai mult)

**Cronometrează-te.** la un subiect real de la `subiecte.edu.ro` (varianta C++), setează 3 ore și lucrează pe hârtie, ca la examen. Corectează cu baremul și notează unde pierzi puncte: de obicei se repetă aceleași 3–4 greșeli.

**Subiectul III.3 cu $10^6$ numere.** Antrenează-te pe probleme cu fișiere mari: la `pbinfo`, caută „bac” în titlu. Obișnuiește-te să te întrebi: „am nevoie de tot vectorul?”

**Ce urmează la clasa a XI-a.** Subprograme, recursivitate, backtracking, grafuri și arbori. Tot ce ai învățat acum (vectori, matrice, șiruri, struct) devine „materia primă” a acestor algoritmi.

## Rezumat

- Recunoaște tiparul din enunț: fișier + parcurgere, frecvențe, secvențe, matrice cu formulă, șiruri, struct.
- III.3: descrie metoda, apoi programează cu o singură parcurgere și memorie constantă.
- Scrie toate etapele: declarări, citire, prelucrare, afișare, cazul „nu există”.
- Fă greșelile *acum*, pe hârtie, nu la examen.

## Probleme de lucru (pbinfo)

Începe cu acestea, în ordine: sunt alese să urmeze pașii din lecție.

- [ ] `#2809` **crescator1** — medie
  Format III.3: fișier mare, o singură parcurgere.
  https://www.pbinfo.ro/probleme/2809/crescator1
- [ ] `#2827` **sir12** — medie
  https://www.pbinfo.ro/probleme/2827/sir12
- [ ] `#2825` **chenar2** — ușoară
  Format II.3: construire matrice.
  https://www.pbinfo.ro/probleme/2825/chenar2

## Temă (pbinfo)

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

- [ ] #2834 **sir13** — medie
  https://www.pbinfo.ro/probleme/2834/sir13
- [ ] #2824 **numere25** — medie
  https://www.pbinfo.ro/probleme/2824/numere25
- [ ] #2839 **parimpar1** — medie
  https://www.pbinfo.ro/probleme/2839/parimpar1
- [ ] #2807 **matrice10** — ușoară
  https://www.pbinfo.ro/probleme/2807/matrice10
- [ ] #2797 **cuvinte9** — ușoară
  https://www.pbinfo.ro/probleme/2797/cuvinte9
- [ ] #2304 **cuvinte7** — ușoară
  https://www.pbinfo.ro/probleme/2304/cuvinte7

## Grile din subiectele de BAC (pbinfo)

Exerciții de tip grilă din variantele de Bacalaureat, rezolvate online pe pbinfo, cu verificare automată a răspunsului.

- [ ] #1034 **Exercițiul #1034**
  program cu vector dat: ce valoare se afișează
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1034
- [ ] #415 **Exercițiul #415**
  condiția în care nu există cea mai mare valoare strict negativă dintr-un tablou
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=415
- [ ] #1050 **Exercițiul #1050**
  interclasarea a două tablouri date: succesiunea obținută
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1050
- [ ] #1051 **Exercițiul #1051**
  căutare binară: pașii pentru x dat
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1051
- [ ] #333 **Exercițiul #333**
  șiruri: secvența care verifică dacă un șir este palindrom
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=333
- [ ] #514 **Exercițiul #514**
  strlen("bine")
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=514

- [ ] #107 **Exercițiul #107**
  declararea corectă a unei variabile de tip structură
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=107

- [ ] #212 **Exercițiul #212**
  câte elemente sunt strict deasupra diagonalei secundare
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=212

## Citește și exersează în plus

- Subiecte de la examene (Bac) https://www.pbinfo.ro/articole/16/subiecte-de-la-examene
- Subiecte de Bac (site-ul Ministerului Educației) — subiectele și baremele oficiale pe ani; caută „Informatică” (de verificat linkul înainte de publicare) https://subiecte.edu.ro/
- Kilonova — probleme și concursuri românești — probleme de algoritmică în limba română, cu corector automat https://kilonova.ro/
- InfoArena — arhiva educațională — clasic pentru pregătirea olimpiadei; probleme cu fișiere .in/.out ca la Bac https://www.infoarena.ro/

# Răspunsuri

## Mini-test de Bac

**Subiectul I:** 1-b, 2-b, 3-b, 4-b, 5-c, 6-c, 7-b, 8-b.

1. `v[0] = 4` e par, `i` devine 1; `v[1] = 7` e impar, bucla se oprește: **1**.
2. Numerele mai mari ca 5: 8, 6, 10 → **3**.
3. Pentru `i + j = 5`: `a[1][4] = −3`, `a[2][3] = −1`, `a[3][2] = 1`, `a[4][1] = 3`; suma este **0**.
4. `informatica` are 11 litere, iar ultimul caracter este `a`: **`11 a`**.
5. `"bac2026"` are 3 + 4 = **7** caractere (20 este dimensiunea tabloului, nu lungimea șirului).
6. `2 < 4` și `5 > 1`: expresia **c** este adevărată. (a nu compilează; b este falsă; d folosește numele tipului în loc de variabilă.)
7. `mij = 4` → `a[4] = 8 < 12`, `st = 5`; `mij = 6` → `a[6] = 12`, găsit. **2** comparații.
8. `s = (3, 4, 8, 9, 14)`; `s[4] − s[1] = 9 − 3 =` **6** (1 + 4 + 1).

**II.1**

```cpp
#include <iostream>
#include <cstring>
using namespace std;

char s[101];

int main()
{
    cin.getline(s, 101);
    bool gasit = false;
    for (char *p = strtok(s, " "); p; p = strtok(NULL, " "))
    {
        int n = strlen(p);
        bool pal = true;
        for (int i = 0; i < n / 2; i++)
            if (p[i] != p[n - 1 - i]) pal = false;
        if (pal)
        {
            cout << p << '\n';
            gasit = true;
        }
    }
    if (!gasit) cout << "nu exista";
    return 0;
}
```

Observă: un cuvânt cu o singură literă ( `a` ) este palindrom (n/2 = 0 pași, `pal` rămâne adevărat).

**II.2**

```cpp
#include <iostream>
using namespace std;

struct produs { char den[31]; int pret; };
produs v[101];

int main()
{
    int n, mx = 0;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        cin >> v[i].den >> v[i].pret;
        if (v[i].pret > mx) mx = v[i].pret;
    }
    for (int i = 1; i <= n; i++)
        if (v[i].pret == mx) cout << v[i].den << '\n';
    return 0;
}
```

**II.3**

```cpp
#include <iostream>
using namespace std;

int a[21][21], n;

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++)
            a[i][j] = (i == j || i + j == n + 1) ? 1 : 0;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++) cout << a[i][j] << ' ';
        cout << '\n';
    }
    return 0;
}
```

**Subiectul III**

a) **Metoda.** Citesc numerele pe rând și păstrez doar numărul anterior. Dacă numărul curent este mai mare decât cel anterior, lungimea secvenței curente crește cu 1; altfel secvența se rupe și lungimea devine 1. Reține maximul lungimilor. Fișierul se parcurge o singură dată (timp liniar), iar memoria este constantă (câteva variabile), indiferent de câte numere conține fișierul.

```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ifstream fin("numere.txt");
    int ant, x, lung = 1, lmax = 1;
    fin >> ant;
    while (fin >> x)
    {
        if (x > ant) lung++;
        else lung = 1;
        if (lung > lmax) lmax = lung;
        ant = x;
    }
    cout << lmax;
    return 0;
}
```

Pentru `1 2 2 3 4 5 1`: `1 2` (lung 2), `2` rupe (lung 1), `2 3 4 5` (lung 4), `1` rupe: răspuns **4**.

---