FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA I2 (INTENSIV) · LECȚIA PY9-I2

Clasa list, în profunzime, și tablourile din C++

Durată: 200 min Nivel: intensiv · Python + C++

CE VEI ȘTI DUPĂ LECȚIE

- Prezici efectele metodelor extend(), clear(), remove(), sort(), reverse(), copy() când lista are mai multe nume sau e trimisă unei funcții.
- Găsești greșelile tipice din programele C++ cu tablouri: indici în afara limitelor, convenții amestecate, = în loc de ==.
- Scrii programe C++ cu tablouri unidimensionale și le testezi pe cazuri normale și limită.
- Compari aceeași soluție în Python și în C++.

Ce trebuie să știi (o pagină)

Python	Efect	Întoarce
`a.extend(c)` / `a += c`	adaugă pe loc elementele lui `c` (listă, `range`, text)	`None`
`a = a + b`	listă **nouă**; `b` trebuie să fie listă	lista nouă
`a.clear()` / `a = []`	golește **obiectul** (se vede prin toate numele) / doar numele `a` indică altă listă	`None` / —
`a.remove(x)`	șterge **prima** apariție (`ValueError` dacă lipsește)	`None`
`a.reverse()`	inversează ordinea, pe loc (nu sortează)	`None`
`a.sort(key=f, reverse=True)` / `sorted(a, …)`	sortează **stabil**, după `f(element)`: pe loc / într-o listă nouă	`None` / lista nouă
`a.copy()`	copie superficială (lista e nouă, elementele sunt aceleași)	lista nouă

C++	Ce înseamnă
`int stoc[100];`	100 de `int`, indici `0 … 99`, dimensiune fixă
`int t[5] = {4, 7};`	`4 7 0 0 0`; `= {0}` sau tablou global: zerouri; local neinițializat: valori întâmplătoare
`t[i]`	elementul de pe poziția `i`: citire, modificare
`for (int i = 0; i < n; i++)`	parcurgere `0 … n − 1`; la convenția `1 … n`, declari `n_max + 1`

C++	Ce înseamnă
indice în afara tabloului	**nicio eroare**: comportament nedefinit
`b = a;` între tablouri	eroare de compilare; copiezi cu `for`
`&&`, `||`, `==`	`and`, `or`, comparație (`=` este atribuire!)

ATENȚIE — GREȘELI FRECVENTE

Compilează mereu cu `g++ -std=c++17 -Wall`. Avertismentele (`suggest parentheses around assignment`, `invokes undefined behavior`) îți arată exact bug-urile de la nivelul 2.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează? Atenție la ultimul rând înainte de `print`.

```
a = [1]
b = a
a += [2]
a.append([3, 4])
a.extend(range(5, 7))
a = a + [7]
print(a, b, len(b))
```

P2. Ce conțin `x` și `y` după apeluri?

```
def reseteaza(l, mod):
    if mod == 1: l.clear()
    else: l = [0]
x = [4, 5]; y = [4, 5]
reseteaza(x, 1); reseteaza(y, 2)
print(x, y)
```

P3. Ce afișează?

```
s = [2, 7, 2, 9, 2]
s.remove(2)
s.remove(s[-1])
print(s, s.count(2))
```

P4. Coduri de produs. Ce afișează? (`"B"` = 66, `"a"` = 97, `"1"` < `"9"`.)

```
c = ["b2", "B10", "a", "B9"]
print(sorted(c), sorted(c, key=str.lower))
```

P5. Scrie cele două liste afișate.

```
cuv = ["mar", "kiwi", "pere", "fag", "ananas"]
print(sorted(cuv, key=len, reverse=True))
t = sorted(cuv, key=len); t.reverse(); print(t)
```

P6. Ce afișează programul C++?

```cpp
int v[6] = {3, 8};
v[v[0]] = 5;
v[5] = v[1] + v[3];
for (int i = 0; i < 6; i++)
    cout << v[i] << " ";
```

P7. Ce afișează? Ce poziții vizitează bucla?

```cpp
int a[8] = {4, 1, 7, 8, 9, 2, 6, 5};
int s = 0, c = 0;
for (int i = 1; i < 8; i += 2) {
    s += a[i];
    if (a[i] > a[i - 1])
        c++;
}
cout << s << " " << c;
```

P8. Temperaturile unei săptămâni sunt `-3 2 5 5 -1 5 0`. Ce zi afișează programul C++ și ce afișează varianta Python de sub el? De ce diferă?

```cpp
int t[7] = {-3, 2, 5, 5, -1, 5, 0};
int p = 0;
for (int i = 1; i < 7; i++)
    if (t[i] >= t[p])
        p = i;
cout << p + 1;
```

Varianta Python: `t = [-3, 2, 5, 5, -1, 5, 0]` și `print(t.index(max(t)) + 1)`.

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Declarația potrivită. Un depozit are **250** de produse; pentru fiecare se reține cantitatea în kilograme, **cu zecimale**. Ce este greșit la fiecare declarație? Care e cea bună?

```cpp
int cant[250];            // (a)
double cant[249];         // (b)
double a[250];            // (c)
double cantitate_kg[250]; // (d)
double cant[250.0];       // (e)
```

P10. Programul trebuie să afișeze suma celor `n` note (`1 ≤ n ≤ 30`). Pentru `3` / `7 8 9` afișează `15` în loc de `24`. Explică de unde vine 15. Ce se mai întâmplă pentru `n = 30`? Repară-l.

```cpp
int note[30];
int main() {
    int n, s = 0;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> note[i];
    for (int i = 0; i < n; i++)
        s += note[i];
    cout << s;
    return 0;
}
```

P11. Coșul golit prea devreme. Fiecare client își pune produsele în `cos`, iar la final comanda se arhivează și coșul se golește. Ce afișează? Repară programul în **două** feluri.

```python
comenzi = []
cos = []
for client in [["pâine", "lapte"], ["mere"]]:
    for p in client:
        cos.append(p)
    comenzi.append(cos)
    cos.clear()
print(comenzi)
```

P12. Rânduri amestecate (C++). Ordonează rândurile ca programul să afișeze în câte zile temperatura a fost **mai mare decât în ziua precedentă** (`n ≤ 31`). Un rând este în plus: care și ce s-ar întâmpla dacă l-ai folosi? Ce afișează pentru `8` / `3 5 5 7 2 4 6 1`?

```
A)      if (t[i] > t[i - 1])
B)  for (int i = 0; i < n; i++)  cin >> t[i];
C)  cout << c;
D)  for (int i = 1; i < n; i++)
E)          c++;
F)  int t[31], n, c = 0;  cin >> n;
G)  for (int i = 0; i < n; i++)
```

P13. Traducere greșită. Programul Python citește scorurile din 10 runde (pot fi și negative) și afișează scorul maxim și de câte ori a fost obținut. Traducerea în C++ are **trei** greșeli. Găsește-le; pentru care dintre ele te avertizează `g++ -Wall`?

```python
scor = [0] * 10
for r in range(10):
    scor[r] = int(input())
print(max(scor), scor.count(max(scor)))
```

```cpp
int scor[10];
for (int r = 1; r <= 10; r++)
    cin >> scor[r];
int mx = 0, ap = 0;
for (int r = 0; r < 10; r++)
    if (scor[r] > mx)
        mx = scor[r];
for (int r = 0; r < 10; r++)
    if (scor[r] = mx)
        ap++;
cout << mx << " " << ap;
```

Test: `-2 -5 -1 -3 -1 -4 -7 -1 -6 -2` → `-1 3`.

P14. Funcția trebuie să șteargă **ultima** apariție a lui `x`. Pentru `[4, 7, 4, 1]` și `x = 4` dă `[4, 7, 4]`, iar pentru `[4, 7, 5, 4]` se oprește cu eroare. Repară-o.

```python
def sterge_ultima(a, x):
    i = len(a) - a[::-1].index(x)
    a.pop(i)
```

Nivelul 3 · Scrie cod

★★ scrie cod

Toate în **C++**. Verifică **toate** testele (pe rânduri separate sau cu spații: `cin` le citește la fel).

P15. Stocul magazinului. Se citesc `n ≤ 100`, stocurile produselor `1 … n`, apoi `q` vânzări de forma `p k` (produsul `p`, `k` bucăți). O vânzare cu produs inexistent sau cu stoc insuficient afișează `refuz` și nu modifică nimic. La final afișează produsele epuizate (stoc 0), în ordine, sau `toate in stoc`.

intrare	ieșire
`4` / `10 3 0 5` / `3` / `2 3` / `4 6` / `1 10`	`refuz` / `1 2 3`
`2` / `5 5` / `2` / `1 2` / `3 1`	`refuz` / `toate in stoc`
`1` / `4` / `1` / `1 4`	`1`

P16. Zilele caniculare. Se citesc 7 temperaturi. Afișează maximul urmat de `:` și de **toate** zilele (1…7) în care a fost atins, apoi, pe rândul următor, amplitudinea (maxim − minim).

intrare	ieșire
`3 5 -1 5 2 0 5`	`5: 2 4 7` / `6`
`-4 -4 -4 -4 -4 -4 -4`	`-4: 1 2 3 4 5 6 7` / `0`
`9 1 2 3 4 5 6`	`9: 1` / `8`

P17. Recorduri. Se citesc scorurile din 10 runde. O rundă este *record* dacă scorul ei este strict mai mare decât toate scorurile anterioare (prima rundă nu se numără). Afișează rundele-record (1…10), apoi numărul lor.

intrare	ieșire
`5 3 8 8 2 9 1 10 10 4`	`3 6 8` / `3`
`10 9 8 7 6 5 4 3 2 1`	(rând gol) / `0`
`-5 -4 -3 -2 -1 0 1 2 3 4`	`2 3 4 5 6 7 8 9 10` / `9`

P18. Contorul de energie. Se citesc 13 indexuri ale contorului unui apartament (în kWh): la 1 ianuarie și la sfârșitul fiecărei luni. Construiește tabloul `consum` cu consumul fiecărei luni (1...12), afișează-l, apoi luna cu cel mai mare consum (prima, la egalitate).

intrare	ieșire
`1000 1210 1400 1560 1700 1800 1890 1990 2100 2220 2380 2600 2850`	`210 190 160 140 100 90 100 110 120 160 220 250` / `12`
`500` de 13 ori	`0` de 12 ori / `1`

P19. reverse() de mână. Se citesc `n ≤ 100` și `n` numere. Inversează tabloul **pe loc** (fără al doilea tablou), prin interschimbări, apoi afișează-l. Câte interschimbări faci?

intrare	ieșire
`5` / `1 2 3 4 5`	`5 4 3 2 1`
`4` / `7 -1 0 9`	`9 0 -1 7`
`1` / `42`	`42`

P20. remove() pentru toate aparițiile. Se citesc `n ≤ 100`, `n` numere și `x`. Elimină din tablou **toate** aparițiile lui `x`, într-o **singură** parcurgere, păstrând ordinea celorlalte. Afișează noul `n` și elementele.

intrare	ieșire
`7` / `3 0 5 0 0 7 0` / `0`	`3` / `3 5 7`
`3` / `2 2 2` / `2`	`0` / (rând gol)
`4` / `1 2 3 4` / `9`	`4` / `1 2 3 4`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Același program, două limbaje. Numără perechile de valori **pare** aflate pe poziții **consecutive**. Scrie soluția în Python (funcție pe listă) și în C++ (tablou). Propune un set de teste care verifică: o pereche la **începutul** și una la **sfârșitul** listei, trei pare la rând, un singur element, listă vidă. De ce bucla merge până la `n - 2` (C++: `i < n - 1`)? Ce s-ar întâmpla în fiecare limbaj dacă ai merge până la `n - 1` și ai compara `a[i]` cu `a[i + 1]`?

P22. Tip și dimensiune. (a) Câți octeți ocupă `int consum[40]`, `double consum[40]` și `long long consum[40]`? Verifică cu `sizeof`. (b) Pentru consumul lunar (kWh, cu zecimale) al celor 40 de apartamente dintr-un bloc, ce declarație alegi? (c) Furnizorul adună consumul anual **în Wh** al celor 20 000 de apartamente dintr-un cartier, fiecare cu circa 300 000 Wh pe an. Încape suma într-un

`int`? (Cel mare `int` este `INT_MAX`, din `<climits>`.) Ce tip alegi? (d) De ce Python nu are problema de la (c)?

P23. Problemă inversă. (a) Există o listă de **numere** `a` pentru care `sorted(a, reverse=True)` diferă de `sorted(a)` urmat de `reverse()`? Argumentează. (b) Construiește cea mai scurtă listă de perechi `[nume, preț]` pentru care cele două variante (cu `key` = prețul) dau rezultate **diferite**.

P24. Ordinea sortărilor contează. În lecție, clasamentul s-a obținut cu `elevi.sort()`, apoi `elevi.sort(key=nota, reverse=True)`. Ce obții dacă **inversezi** ordinea celor două sortări? Ce obții dacă **sari** peste prima? Rulează pe lista din lecție și explică fiecare rezultat folosind stabilitatea.

P25. Mini-proiect: jurnalul meteo. Scrie în C++ **și** în Python un program care citește 7 temperaturi, apoi comenzi până la `0`: `1 z t` (ziua `z` primește temperatura `t`), `2 z` (afișează temperatura zilei `z`), `3` (afișează prima zi cu temperatura maximă și valoarea). O zi în afara intervalului 1…7 afișează `zi invalida`. Test (rânduri): `12 15 9 18 11 14 10` / `2 3` / `1 3 21` / `3` / `1 8 30` / `2 0` / `0`. Ieșire așteptată: `9`, `3 21`, `zi invalida`, `zi invalida`. Apoi scoate validarea din ambele programe și explică ce s-ar întâmpla la `2 0` în Python și în C++. Care limbaj te-ar fi „prins”?

Probleme de lucru în clasă (pbinfo)

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

- [ ] **#486** **minmax0** — ușoară
  https://www.pbinfo.ro/probleme/486/minmax0
- [ ] **#491** **suma2** — ușoară
  https://www.pbinfo.ro/probleme/491/suma2
- [ ] **#493** **constr** — ușoară
  https://www.pbinfo.ro/probleme/493/constr

Temă pentru acasă (pbinfo)

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

- [ ] **#487** **numarare2** — ușoară
  https://www.pbinfo.ro/probleme/487/numarare2
- [ ] **#553** **pozminmax** — ușoară
  https://www.pbinfo.ro/probleme/553/pozminmax
- [ ] **#161** **inlocuire** — ușoară
  https://www.pbinfo.ro/probleme/161/inlocuire