FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 14 · LECȚIA PY9-14

Modelul liniar: listă, stivă, coadă, listă de frecvențe

Durată: 150 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești pas cu pas operațiile pe o stivă și pe o coadă și prezici ce rămâne în ele.
- Construiești și citești o listă de frecvențe.
- Alegi modelul potrivit (stivă, coadă, acces direct, frecvențe) și parcurgerea potrivită (cu sau fără memorare).

Ce trebuie să știi (o pagină)

Noțiune	În Python	Observație
listă cu `n` elemente	`a = [7, 2, 9]`, `len(a)`	indici `0 … n − 1`; `a[-1]` = ultimul
acces direct	`a[k]`	un singur pas, oricare ar fi `k`
parcurgere secvențială	`for x in a:` sau `for i in range(len(a)):`	treci pe rând prin toate
perechi consecutive	`for i in range(len(a) - 1):` … `a[i]`, `a[i + 1]`	`i` se oprește la penultimul
stivă (LIFO)	`s.append(x)`, `s.pop()`, vârful `s[-1]`	scoți ultimul adăugat
coadă (FIFO)	`q.append(x)`, `q.pop(0)`, primul `q[0]`	scoți primul adăugat
listă vidă?	`if s:` / `if not s:`	`pop()` pe listă vidă: `IndexError`
listă de frecvențe pentru valori `0 … M`	`fr = [0] * (M + 1)`, apoi `fr[x] += 1`	poziția = valoarea, elementul = de câte ori apare
fără memorare	prelucrezi fiecare valoare imediat	suma, numărul, maximul, comparația cu precedenta
cu memorare	`a.append(x)`, apoi încă o parcurgere	când ai nevoie de un rezultat aflat abia la final (media, minimul)

SFAT

La stive și cozi, desenează lista după **fiecare** operație, pe orizontală, cu vârful (sau începutul) marcat. Greșelile apar aproape mereu la a treia-a patra operație, când „ții minte” în loc să scrii.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează?

```
s = []
for x in [4, 7, 1]:
    s.append(x)
s.pop()
s.append(9)
print(s, s[-1])
```

P2. Ce afișează cele două `print`-uri?

```
q = ["Ana", "Bob"]
q.append("Cris")
print(q.pop(0))
q.append("Dan")
print(q.pop(0), q)
```

P3. Aceeași listă de operații (`+X` = adaugă `X`, `-` = scoate și afișează), o dată pe o stivă, o dată pe o coadă. Ce afișează? (Stiva `s` și coada `q` primesc aceleași operații.)

```
ops = ["+A", "+B", "-", "+C", "-", "-"]
s, q = [], []
for op in ops:
    if op == "-":
        print(s.pop(), q.pop(0))
    else:
        s.append(op[1])
        q.append(op[1])
```

P4. Ce listă de frecvențe se afișează?

```
n = 2026100
fr = [0] * 10
for c in str(n):
    fr[int(c)] += 1
print(fr)
```

P5. Lista de frecvențe `fr = [0, 2, 0, 3, 1]` a fost construită dintr-o listă de numere. Ce afișează programul? Câte numere avea lista inițială și care a fost cel mai frecvent?

```
fr = [0, 2, 0, 3, 1]
for v in range(len(fr)):
    for k in range(fr[v]):
        print(v, end=" ")
```

P6. Un element este „vârf" dacă este mai mare decât **ambii** vecini. Ce afișează? De ce `range` începe de la 1 și se oprește înainte de `len(a) - 1`?

```python
a = [1, 5, 2, 2, 7, 3, 8]
c = 0
for i in range(1, len(a) - 1):
    if a[i] > a[i - 1] and a[i] > a[i + 1]:
        c += 1
print(c)
```

P7. Ce afișează `a = [10, 20, 30, 40, 50]` și `print(a[len(a) // 2], a[-2], a[-len(a)])`?

P8. Programul nu folosește nicio listă. Ce afișează pentru `n = 7` și valorile `5 3 4 6 6 2 7` (câte una pe rând)? Ce numără, în cuvinte?

```python
n = int(input())
prec = int(input())
c = 0
for _ in range(n - 1):
    x = int(input())
    if x > prec:
        c += 1
    prec = x
print(c)
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul trebuie să servească clienții de la ghișeu **în ordinea sosirii**. Afișează `Carla`, `Bogdan`, `Ana`. Ce model a folosit, de fapt, autorul? Repară-l.

```python
coada = ["Ana", "Bogdan", "Carla"]
while coada:
    print("Servit:", coada.pop())
```

P10. Programul numără perechile de elemente consecutive egale. Pentru `[3, 3, 8, 1, 1]` dă `IndexError`. De ce? Ce trebuie să afișeze după reparare?

```python
a = [3, 3, 8, 1, 1]
c = 0
for i in range(len(a)):
    if a[i] == a[i + 1]:
        c += 1
print(c)
```

P11. Programul afișează câte note de fiecare fel (1–10) are clasa. Pentru notele `9 7 9 4` afișează nouă rânduri corecte și apoi se oprește cu eroare; pentru `9 10 7` se oprește imediat. Găsește greșeala și repar-o.

```python
note = list(map(int, input().split()))
fr = [0] * 10
for x in note:
    fr[x] += 1
for v in range(1, 11):
    print(v, fr[v])
```

P12. Funcția trebuie să spună dacă parantezele dintr-un text sunt corect închise. Pentru `"(()"` răspunde `True`, iar pentru `"())("` se oprește cu `IndexError`. Găsește **două** greșeli.

```
def echilibrat(t):
    s = []
    for c in t:
        if c == "(":
            s.append(c)
        else:
            s.pop()
    return True
```

P13. *Rânduri amestecate.* Programul citește numărul de candidați `k` (numerotați 1…k), apoi pe un rând voturile, și afișează candidatul cu cele mai multe voturi și numărul lui de voturi. Un rând este în plus.

```
A)  fr[v] += 1
B)  k = int(input())
C)  for v in voturi:
D)  fr = [0] * (k + 1)
E)  print(fr.index(max(fr)), max(fr))
F)  voturi = list(map(int, input().split()))
G)  fr = [0] * k
```

Ce afișează pentru `k = 4` și voturile `2 3 2 4 1 2 3`? La egalitate de voturi, care candidat este afișat?

P14. Trebuie afișate zilele (numerotate de la 1) în care magazinul a avut **numărul minim** de clienți. Pentru `n = 5` și valorile `5 3 7 3 4` afișează `1 2 4` în loc de `2 4`. De ce nu se poate rezolva corect într-o singură parcurgere, fără memorare? Rescrie-l.

```
n = int(input())
mn = 10**9
for zi in range(1, n + 1):
    x = int(input())
    if x <= mn:
        mn = x
        print(zi)
```

Nivelul 3 · Scrie cod

★★ scrie cod

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

P15. Anulează. Pe un rând sunt cuvinte și simbolul `<`. Fiecare cuvânt este „scris”, iar `<` șterge ultimul cuvânt rămas (dacă există). Afișează textul final, cu cuvintele separate prin spațiu, sau `(gol)`.

intrare	ieșire
`Ana are < mere < pere`	`Ana pere`
`< < x`	`x`

intrare	ieșire
`a <`	`(gol)`

P16. Ghișeul. Pe un rând sunt nume de persoane care se așază la coadă și simbolul `*`, care înseamnă „este servită următoarea persoană” (dacă nu e nimeni, `*` nu are efect). Afișează pe primul rând persoanele servite, în ordine, iar pe al doilea cele rămase la coadă. Dacă un rând ar fi gol, afișezi `-`.

intrare	ieșire
`Ana Bob * Cris * *`	`Ana Bob Cris` / `-`
`* Ana`	`-` / `Ana`
`* *`	`-` / `-`

P17. Chestionarul de satisfacție. Pe un rând sunt răspunsuri de la 1 la 5 stele. Afișează, de la 5 la 1, câte un rând de forma `5: **` (atâtea steluțe câte răspunsuri are nota), apoi nota cea mai frecventă. La egalitate alegi nota **mai mare**.

intrare	ieșire
`5 4 4 1 5 4`	`5: **`, `4: ***`, `3:`, `2:`, `1: *`, apoi `4`
`3`	`5:`, `4:`, `3: *`, `2:`, `1:`, apoi `3`
`2 5 2 5`	`5: **`, `4:`, `3:`, `2: **`, `1:`, apoi `5`

P18. Perechi pare. Citește o listă de numere de pe un rând și afișează câte perechi de elemente **aflate pe poziții consecutive** sunt ambele pare.

intrare	ieșire	de ce e util testul
`2 4 5 8 10 3 6`	`2`	caz obișnuit
`4 6 1 3 8 2`	`2`	perechi la ambele capete
`6 6 6`	`2`	perechi care se suprapun
`8`	`0`	un singur element
(rând gol)	`0`	listă vidă
`1 3 5`	`0`	nicio pereche

P19. Zilele slabe. Pe un rând este numărul de clienți din fiecare zi a lunii. Afișează minimul, iar pe rândul următor zilele (numerotate de la 1) în care s-a atins.

intrare	ieșire
`40 25 31 25 60`	`25` / `2 4`
`12`	`12` / `1`
`7 7 7`	`7` / `1 2 3`

P20. Nașteri pe luni. Un cod numeric personal (CNP) are 13 cifre: `S AA LL ZZ JJ NNN C`. Cifrele de pe pozițiile 4 și 5 (numărate de la 1, deci indicii 3 și 4 în text) formează luna nașterii. Pe un rând sunt mai multe CNP-uri (inventate; cifra de control nu se verifică), separate prin spațiu. Construiește lista de frecvențe a lunilor (12 poziții) și afișează-o, apoi luna cu cele mai multe nașteri (la egalitate, cea mai mică) și câte nașteri are.

intrare	ieșire
`1990312350011 2020715350022 5040712350033 6001231350044 1990305350055 2011207350066`	`0 0 2 0 0 0 2 0 0 0 0 2` și `luna 3 cu 2 nașteri`
`1990312350011 2020715350022 5040712350033 2060712350044`	`0 0 1 0 0 0 3 0 0 0 0 0` și `luna 7 cu 3 nașteri`
`1010101350011`	`1 0 0 0 0 0 0 0 0 0 0 0` și `luna 1 cu 1 naștere`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Stivă, coadă sau altceva? Pentru fiecare situație alege modelul și argumentează într-o propoziție: (a) teancul de lucrări corectate, din care profesorul le împarte de sus; (b) apelurile care așteaptă la un call-center; (c) istoricul `Ctrl+Z` dintr-un program de desen; (d) melodiile dintr-un playlist redat în ordinea adăugării; (e) lista elevilor dintr-un catalog, din care vrei imediat al 17-lea; (f) numărul de mesaje de eroare de fiecare tip (codurile 1–8) primite într-o zi.

P22. Vânzările lunare: ai încredere în listă? Un magazin păstrează vânzările anului într-o listă `v`: elementul de pe poziția `i` este luna `i + 1`, iar `0` înseamnă „fără date". a) Scrie `verifica(v)`, care întoarce `ok` pentru o listă cu 12 valori pozitive, `lipsesc k luni` (sau `sunt k valori în plus`) dacă lungimea nu e 12, sau `luna i nu are date` pentru prima valoare nulă. Teste: `[120, 135, 128]` → `lipsesc 9 luni`; `[]` → `lipsesc 12 luni`; lista cu 12 valori și un `0` pe poziția 2 (a treia lună) → `luna 3 nu are date`. b) Media unei liste cu un `0` rămas în ea se calculează greșit cu `sum(v) / len(v)`: cum o corectezi? c) Lista corectă `[120, 135, 128, 150, 160, 172, 180, 175, 160, 150, 140, 210]` a fost scrisă de un coleg începând cu octombrie, nu cu ianuarie. Ce lună spune programul tău că are vânzările maxime? Cum afli din listă în ce ordine este?

P23. Ce poate ieși dintr-o stivă? Numerele `1, 2, 3, 4` intră într-o stivă **în această ordine**, dar poți scoate oricând (de exemplu: pui 1, pui 2, scoți 2, …). Care dintre ordinile de ieșire sunt posibile: (a) `2 1 4 3`; (b) `3 1 4 2`; (c) `4 3 2 1`; (d) `1 4 2 3`; (e) `3 4 2 1`? Pentru cele imposibile, spune unde se blochează.

P24. Frecvențe sau count()? Ai notele a 1000 de elevi (de la 1 la 10). Varianta A construiește lista de frecvențe într-o parcurgere. Varianta B apelează `note.count(v)` pentru fiecare `v` de la 1 la 10. Câte elemente „vizitează" fiecare variantă? Ce se schimbă dacă valorile ar fi coduri de la 1 la 1 000 000? Dar dacă ar fi numere de până la 10^9?

P25. Mini-proiect: centrul de livrări. Un centru folosește **trei structuri într-un singur program**: coletele așteaptă în depozit (**coadă**: primul sosit este primul încărcat), se încarcă în dubă (**stivă**: ultimul încărcat este primul livrat) și se numără livrările pe zone, 1…5 (**listă de frecvențe**). Pe un rând sunt operații: `+Z` (sosește un colet pentru zona `Z`, `Z` între 1 și 5), `L` (mută cel mai vechi colet din

depozit în dubă) și `D` (livrează coletul de deasupra din dubă). `L` pe depozit gol și `D` pe dubă goală nu fac nimic. La final afișează depozitul, duba (de la fund spre vârf) și lista livrărilor pe zonele 1…5.

intrare	ieșire
`+1 +3 +3 L L +2 L D D L D`	`depozit: [], dubă: [1], livrate: [0, 1, 2, 0, 0]`
`L D D +4 +4 +5`	`depozit: [4, 4, 5], dubă: [], livrate: [0, 0, 0, 0, 0]`
`+5 +1 L L L D D D`	`depozit: [], dubă: [], livrate: [1, 0, 0, 0, 1]`
`+2 +2 +2 L L L D`	`depozit: [], dubă: [2, 2], livrate: [0, 1, 0, 0, 0]`

Urmărește pe hârtie primul test (depozit, dubă și frecvențe după fiecare operație) și spune ce s-ar întâmpla dacă duba ar fi o coadă, nu o stivă.

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
- [ ] **#487** **numarare2** — ușoară
  https://www.pbinfo.ro/probleme/487/numarare2
- [ ] **#187** **ciffrecv** — ușoară
  https://www.pbinfo.ro/probleme/187/ciffrecv

Temă pentru acasă (pbinfo)

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

- [ ] **#553** **pozminmax** — ușoară
  https://www.pbinfo.ro/probleme/553/pozminmax
- [ ] **#264** **maxcif** — ușoară
  https://www.pbinfo.ro/probleme/264/maxcif
- [ ] **#267** **unice** — ușoară
  https://www.pbinfo.ro/probleme/267/unice