FIȘĂ DE LUCRU · PYTHON · CLASA A VIII-A · FIȘA 07 · LECȚIA PY8G-07

Modificarea unui șir: inserare și ștergere

Durată: 50 min Nivel: gimnaziu · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici efectul unei modificări, al unei inserări sau al unei ștergeri.
- Repari direcția greșită a mutărilor și alte greșeli de indici.
- Scrii programe care modifică, inserează și șterg elemente.

Ce trebuie să știi (o pagină)

Operația	Cod
modifici un element	`v[i] = valoare` (parcurs după **indice**)
ștergi de pe poziția `p`	`for i in range(p, n - 1):` ↵ `····v[i] = v[i + 1]` ↵ `v.pop()`
inserezi `x` pe poziția `p`	`v.append(0)` ↵ `for i in range(n, p, -1):` ↵ `····v[i] = v[i - 1]` ↵ `v[p] = x`
adaugi la sfârșit	`v.append(x)`
scoți ultimul element	`v.pop()`
(scurtături)	`del v[p]`, `v.insert(p, x)`
păstrezi doar o parte	`w = []`, apoi `w.append(x)` pentru cele bune

ATENȚIE — GREȘELI FRECVENTE

La ștergere muți elementele **spre stânga**, începând din față. La inserare le muți **spre dreapta**, începând de la **sfârșit**. Invers, ai suprascrie valori.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează?

```
v = [3, -2, 8, -5, 1]
for i in range(len(v)):
    if v[i] < 0:
        v[i] = -v[i]
print(*v)
```

P2. Ce afișează? De ce?

```
v = [1, 2, 3]
for x in v:
    x += 10
print(v)
```

P3. Ce afișează?

```
v = [10, 20, 30, 40, 50]
n = len(v)
p = 2
for i in range(p, n - 1):
    v[i] = v[i + 1]
v.pop()
print(*v)
```

P4. Ce afișează?

```
v = [1, 2, 3]
n = len(v)
p = 0
x = 9
v.append(0)
for i in range(n, p, -1):
    v[i] = v[i - 1]
v[p] = x
print(*v)
```

P5. Ce afișează?

```
v = [5, 6, 7]
v.pop()
v.append(8)
v.append(9)
print(v, len(v))
```

P6. Ce afișează?

```
v = [4, 0, 7, 0, 0, 3]
w = []
for x in v:
    if x != 0:
        w.append(x * 2)
print(*w)
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P7. Vrem să ștergem elementul de pe poziția 1 (rezultat: `10 30 40 50`), dar apare o eroare. Care este ea? Repară-l.

```
v = [10, 20, 30, 40, 50]
n = len(v)
p = 1
for i in range(p, n):
    v[i] = v[i + 1]
v.pop()
print(*v)
```

P8. Vrem să inserăm `99` pe poziția 1 (rezultat: `10 99 20 30 40`), dar toate elementele după el devin `20`. Care este greșeala? Repară-l.

```
v = [10, 20, 30, 40]
n = len(v)
p = 1
x = 99
v.append(0)
for i in range(p + 1, n + 1):
    v[i] = v[i - 1]
v[p] = x
print(*v)
```

P9. Vrem să ștergem elementul de pe poziția 1 (rezultat: `10 30 40 50`), dar lista are încă 5 elemente. Ce lipsește?

```
v = [10, 20, 30, 40, 50]
n = len(v)
p = 1
for i in range(p, n - 1):
    v[i] = v[i + 1]
print(*v)
```

P10. Vrem să ștergem toate zerourile din listă (rezultat: `5 7`), dar apare o eroare. Explică de ce și repară programul **fără** să ștergi din listă (construiește una nouă).

```
v = [0, 5, 0, 7]
for i in range(len(v)):
    if v[i] == 0:
        del v[i]
print(*v)
```

Nivelul 3 · Scrie cod

★★ scrie cod

P11. Ștergere. Citește `n`, un rând cu `n` numere și o poziție `p`. Șterge elementul de pe poziția `p` (mutând celelalte) și afișează lista.

intrare	ieșire

`5`
`10 20 30 40 50`
`1`
	`10 30 40 50`

intrare	ieșire

`3`
`1 2 3`
`0`
	`2 3`

`3`
`1 2 3`
`2`
	`1 2`

P12. Inserare. Citește `n`, un rând cu `n` numere și, pe al treilea rând, o poziție `p` și o valoare `x` (separate prin spațiu). Inserează `x` pe poziția `p` și afișează lista.

intrare	ieșire

`4`
`10 20 30 40`
`1 99`
	`10 99 20 30 40`

`2`
`5 6`
`0 1`
	`1 5 6`

`2`
`5 6`
`2 9`
	`5 6 9`

P13. Fără minus. Citește `n` și un rând cu `n` numere. Înlocuiește fiecare număr negativ cu `0` și afișează lista.

intrare	ieșire
`5` `3 -2 8 -5 1`	`3 0 8 0 1`
`3` `-1 -2 -3`	`0 0 0`
`2` `4 6`	`4 6`

P14. Fără zerouri. Citește `n` și un rând cu `n` numere (cel puțin unul nenul). Afișează numerele **nenule**, în aceeași ordine.

intrare	ieșire
`6` `4 0 7 0 0 3`	`4 7 3`
`3` `0 0 5`	`5`
`2` `1 2`	`1 2`

P15. Totalul la coadă. Citește `n` și un rând cu `n` numere. Adaugă la sfârșitul listei **suma** tuturor elementelor și afișează lista.

intrare	ieșire
`3` `1 2 3`	`1 2 3 6`
`1` `5`	`5 5`
`2` `-1 1`	`-1 1 0`

Nivelul 4 · Provocare

★★★ pune mintea la contribuție

P16. Zerourile la coadă. Citește `n` și un rând cu `n` numere. Afișează numerele nenule în ordinea inițială, apoi toate zerourile.

intrare	ieșire
`6` `0 3 0 5 0 8`	`3 5 8 0 0 0`
`3` `1 2 3`	`1 2 3`
`4` `0 0 0 1`	`1 0 0 0`

P17. Coada de la cantină. Citește `n` comenzi, câte una pe rând. Comanda `a x` adaugă elevul cu numărul `x` la **sfârșitul** cozii, iar comanda `s` servește (șterge) elevul din **fața** cozii. La final afișează coada, de la primul la ultimul. Nu se dau comenzi `s` când coada e goală.

intrare	ieșire

`4`
`a 5`
`a 7`
`s`
`a 9`
	`7 9`

`3`
`a 1`
`a 2`
`a 3`
	`1 2 3`

`5`
`a 4`
`a 6`
`s`
`s`
`a 8`
	`8`

P18. Ștergere de la coadă la început. Citește `n` și un rând cu `n` numere. Șterge din listă **toate numerele impare**, folosind `del v[i]` în timp ce parcurgi lista. Cum eviți problema din P10? Hint: parcurge lista de la sfârșit spre început.

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