FIȘĂ DE LUCRU · PYTHON · CLASA A VIII-A · FIȘA 08 · LECȚIA PY8G-08

Interschimbarea, inversarea și rotirea unui șir

Durată: 50 min Nivel: gimnaziu · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici rezultatul unei interschimbări, al unei inversări și al unei rotiri.
- Repari greșeli de perechi, de limite și de ordinea mutărilor.
- Scrii programe de interschimbare, inversare și rotire.

Ce trebuie să știi (o pagină)

Operația	Cod
schimbi două variabile	`aux = a` ↵ `a = b` ↵ `b = aux` (sau `a, b = b, a`)
schimbi două elemente	`v[i], v[j] = v[j], v[i]`
inversezi șirul	`for i in range(n // 2):` ↵ `····v[i], v[n - 1 - i] = v[n - 1 - i], v[i]`
rotești la stânga cu 1	`prim = v[0]` ↵ `for i in range(n - 1): v[i] = v[i + 1]` ↵ `v[n - 1] = prim`
rotești la dreapta cu 1	`ultim = v[n - 1]` ↵ `for i in range(n - 1, 0, -1): v[i] = v[i - 1]` ↵ `v[0] = ultim`
perechea lui `v[i]` la inversare	`v[n - 1 - i]`

ATENȚIE — GREȘELI FRECVENTE

`a = b` urmat de `b = a` **nu** schimbă valorile: după prima atribuire, valoarea veche a lui `a` s-a pierdut.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează?

```
a = 3
b = 5
aux = a
a = b
b = aux
print(a, b)
```

P2. Ce afișează? Ce s-a întâmplat cu valoarea `3`?

```
a = 3
b = 5
a = b
b = a
print(a, b)
```

P3. Ce afișează?

```
v = [4, 9, 1, 7]
v[0], v[3] = v[3], v[0]
v[1], v[2] = v[2], v[1]
print(*v)
```

P4. Ce afișează? Câte interschimbări se fac?

```
v = [1, 2, 3, 4, 5, 6, 7]
n = len(v)
k = 0
for i in range(n // 2):
    v[i], v[n - 1 - i] = v[n - 1 - i], v[i]
    k += 1
print(*v)
print(k)
```

P5. Ce afișează?

```
v = [1, 2, 3, 4]
n = len(v)
prim = v[0]
for i in range(n - 1):
    v[i] = v[i + 1]
v[n - 1] = prim
print(*v)
```

P6. Ce afișează? Cu ce operație din lecție seamănă rezultatul?

```
v = [1, 2, 3, 4, 5]
for i in range(len(v) - 1):
    v[i], v[i + 1] = v[i + 1], v[i]
print(*v)
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P7. Vrem să inversăm lista `[1, 2, 3, 4]`, dar la final ea arată exact ca la început. De ce? Repară-l.

```
v = [1, 2, 3, 4]
n = len(v)
for i in range(n):
    v[i], v[n - 1 - i] = v[n - 1 - i], v[i]
print(*v)
```

P8. Programul de inversare se oprește cu o eroare. Care este ea și cum o repari?

```
v = [1, 2, 3, 4]
n = len(v)
for i in range(n // 2):
    v[i], v[n - i] = v[n - i], v[i]
print(*v)
```

P9. Vrem rotirea la stânga cu o poziție (`2 3 4 5 1`), dar ultimul element iese greșit. Ce a mers prost? Repară-l.

```
v = [1, 2, 3, 4, 5]
n = len(v)
for i in range(n - 1):
    v[i] = v[i + 1]
v[n - 1] = v[0]
print(*v)
```

P10. Vrem rotirea la dreapta cu o poziție (`5 1 2 3 4`), dar toate elementele devin egale. De ce? Repară-l.

```
v = [1, 2, 3, 4, 5]
n = len(v)
ultim = v[n - 1]
for i in range(1, n):
    v[i] = v[i - 1]
v[0] = ultim
print(*v)
```

Nivelul 3 · Scrie cod

★★ scrie cod

P11. Invers. Citește `n` și un rând cu `n` numere. Inversează șirul **în listă** (cu interschimbări) și afișează-l.

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

P12. Perechi vecine. Citește `n` și un rând cu `n` numere. Interschimbă elementele din două în două: pe 0 cu 1, pe 2 cu 3 și așa mai departe. Dacă `n` este impar, ultimul rămâne pe loc.

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

intrare	ieșire
`1` `9`	`9`

P13. Rotire. Citește `n` și un rând cu `n` numere. Rotește șirul la **stânga** cu o poziție și afișează-l.

intrare	ieșire
`5` `1 2 3 4 5`	`2 3 4 5 1`
`1` `7`	`7`
`3` `9 8 7`	`8 7 9`

P14. Rotire de k ori. Citește `n`, un rând cu `n` numere și un număr `k`. Rotește șirul la **dreapta** de `k` ori, câte o poziție, și afișează-l.

intrare	ieșire

`5`
`1 2 3 4 5`
`2`
	`4 5 1 2 3`

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

`4`
`1 2 3 4`
`1`
	`4 1 2 3`

P15. Minim cu maxim. Citește `n` și un rând cu `n` numere **distincte**. Interschimbă cel mai mic element cu cel mai mare și afișează lista.

intrare	ieșire
`5` `4 9 1 7 3`	`4 1 9 7 3`
`3` `1 2 3`	`3 2 1`
`2` `5 2`	`2 5`

Nivelul 4 · Provocare

★★★ pune mintea la contribuție

P16. Palindrom prin oglindă. Citește `n` și un rând cu `n` numere. Construiește o listă nouă `w` cu elementele în ordine inversă (cu `append`, parcurgând de la sfârșit). Afișează `da` dacă `v == w` (listele sunt egale), altfel `nu`. De ce înseamnă asta că șirul este palindrom?

intrare	ieșire
`5` `1 4 7 4 1`	`da`
`4` `1 2 3 4`	`nu`
`1` `5`	`da`

P17. Rotire cu formulă. Citește `n`, un rând cu `n` numere și `k`. Construiește o listă nouă `w` în care `w[i] = v[(i + k) % n]`. Afișează `w`. Ce observi? (Ce rotire obții? Ce se întâmplă dacă `k` este mai mare decât `n`?)

intrare	ieșire

`5`
`1 2 3 4 5`
`2`
	`3 4 5 1 2`

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

`4`
`1 2 3 4`
`5`
	`2 3 4 1`

P18. Inversare parțială. Citește `n`, un rând cu `n` numere și două poziții `a` și `b` (`a ≤ b`), pe același rând. Inversează doar secvența dintre pozițiile `a` și `b`, inclusiv. Hint: folosește două variabile `i = a` și `j = b`, apropie-le cu un `while i < j`.

intrare	ieșire

`6`
`1 2 3 4 5 6`
`1 4`
	`1 5 4 3 2 6`

`5`
`1 2 3 4 5`
`0 4`
	`5 4 3 2 1`

`4`
`1 2 3 4`
`2 2`
	`1 2 3 4`