FIȘĂ DE LUCRU · PYTHON · CLASA A VIII-A · FIȘA 09 · LECȚIA PY8G-09

Proprietăți ale șirurilor: palindrom, perechi, elemente distincte

Durată: 50 min Nivel: gimnaziu · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici rezultatul verificărilor de palindrom, al parcurgerii perechilor și al celor pentru elemente distincte.
- Repari limitele greșite ale buclelor imbricate și steagurile greșite.
- Scrii programe cu perechi de elemente.

Ce trebuie să știi (o pagină)

Ce vrei	Cod
perechea lui `v[i]` la palindrom	`v[n - 1 - i]`, pentru `i` în `range(n // 2)`
palindrom	`ok = True`; `if v[i] != v[n - 1 - i]: ok = False`
toate perechile `i < j`	`for i in range(n):` ↵ `····for j in range(i + 1, n):`
număr de perechi la `n` elemente	`n · (n - 1) / 2`
elemente distincte	`ok = True`; pentru perechi: `if v[i] == v[j]: ok = False`
valoarea absolută	`abs(x)`

ATENȚIE — GREȘELI FRECVENTE

`j` pornește de la `i + 1`. Dacă pornește de la `i`, compari un element cu el însuși; dacă pornește de la `0`, numeri fiecare pereche de două ori.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează?

```
v = [1, 2, 3, 4, 1]
n = len(v)
ok = True
for i in range(n // 2):
    if v[i] != v[n - 1 - i]:
        ok = False
print(ok)
```

P2. Ce afișează?

```
v = [1, 6, 3, 4, 5, 2]
n = len(v)
c = 0
for i in range(n):
    for j in range(i + 1, n):
        if v[i] + v[j] == 7:
            c += 1
print(c)
```

P3. Ce afișează?

```
for i in range(3):
    for j in range(i + 1, 3):
        print(i, j)
```

P4. Ce afișează?

```
v = [2, 2, 3, 2]
n = len(v)
c = 0
for i in range(n):
    for j in range(i + 1, n):
        if v[i] == v[j]:
            c += 1
print(c)
```

P5. De câte ori se execută instrucțiunea `k += 1`? (Nu o calcula „în cap" pe toate: găsește regula.)

```
n = 6
k = 0
for i in range(n):
    for j in range(i + 1, n):
        k += 1
print(k)
```

P6. Ce afișează?

```
v = [1, 2, 3, 3, 5, 1]
n = len(v)
c = 0
for i in range(n // 2):
    if v[i] != v[n - 1 - i]:
        c += 1
print(c)
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P7. Lista `[1, 2, 9, 2, 5]` **nu** este palindrom, dar programul afișează `True`. Care este greșeala de logică? Repară-l.

```
v = [1, 2, 9, 2, 5]
n = len(v)
ok = False
for i in range(n // 2):
    if v[i] == v[n - 1 - i]:
        ok = True
print(ok)
```

P8. Sunt 3 perechi cu suma 7 (lista este aceeași ca la P2), dar programul afișează `6`. De ce? Repară-l.

```
v = [1, 6, 3, 4, 5, 2]
n = len(v)
c = 0
for i in range(n):
    for j in range(n):
        if v[i] + v[j] == 7:
            c += 1
print(c)
```

P9. Lista `[1, 2, 3]` are elemente distincte, dar programul afișează `False`. De ce? Repară-l.

```
v = [1, 2, 3]
n = len(v)
distincte = True
for i in range(n):
    for j in range(i, n):
        if v[i] == v[j]:
            distincte = False
print(distincte)
```

P10. Vrem un **singur** mesaj, `distincte` sau `se repetă`, dar programul scrie mai multe rânduri. Repară-l cu un steag.

```
v = [4, 9, 2]
n = len(v)
for i in range(n):
    for j in range(i + 1, n):
        if v[i] == v[j]:
            print("se repetă")
        else:
            print("distincte")
```

Nivelul 3 · Scrie cod

★★ scrie cod

P11. Palindrom? Citește `n` și un rând cu `n` numere. Afișează `da` dacă șirul este palindrom, altfel `nu`.

intrare	ieșire
`5` `1 4 7 4 1`	`da`

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

P12. Perechi cu suma S. Citește `n`, un rând cu `n` numere și un număr `S`. Afișează câte perechi de elemente (cu poziții diferite, fiecare pereche o singură dată) au suma `S`.

intrare	ieșire

`6`
`3 7 5 5 2 8`
`10`
	`3`

`3`
`1 2 3`
`10`
	`0`

`4`
`5 5 5 5`
`10`
	`6`

P13. Distincte? Citește `n` și un rând cu `n` numere. Afișează `da` dacă toate sunt distincte, altfel `nu`.

intrare	ieșire
`4` `4 9 2 7`	`da`
`5` `4 9 2 7 9`	`nu`
`1` `5`	`da`

P14. Perechi egale. Citește `n` și un rând cu `n` numere. Afișează câte perechi de poziții `(i, j)` cu `i < j` au `v[i] == v[j]`.

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

P15. Inversiuni. Citește `n` și un rând cu `n` numere. Afișează câte perechi `(i, j)` cu `i < j` au `v[i] > v[j]` (cu cât sunt mai puține, cu atât șirul este „mai ordonat”).

intrare	ieșire
`4` `3 1 2 4`	`2`
`3` `1 2 3`	`0`
`3` `3 2 1`	`3`

Nivelul 4 · Provocare

★★★ pune mintea la contribuție

P16. Cea mai mică diferență. Citește `n` (cel puțin 2) și un rând cu `n` numere. Afișează cea mai mică valoare a lui `abs(v[i] - v[j])` pentru toate perechile `i < j`.

intrare	ieșire
`4` `10 3 8 15`	`2`
`2` `5 5`	`0`
`3` `1 100 50`	`49`

P17. Prima pereche. Citește `n`, un rând cu `n` numere și un număr `S`. Afișează pozițiile `i` și `j` (cu `i < j`) ale **primei** perechi cu suma `S`, în ordinea în care sunt găsite de cele două bucle. Dacă nu există, afișează `nu există`.

intrare	ieșire

`5`
`8 3 5 9 2`
`11`
	`0 1`

`3`
`1 2 3`
`10`
	`nu există`

`4`
`4 6 5 5`
`10`
	`0 1`

P18. Triplete. Citește `n` și un rând cu `n` numere. Afișează câte **triplete** de poziții `i < j < k` au suma `0`. Câte bucle imbricate ai nevoie?

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