FIȘĂ DE LUCRU · PYTHON · CLASA A VIII-A · FIȘA 03 · LECȚIA PY8G-03

Citirea și afișarea unui șir

Durată: 50 min Nivel: gimnaziu · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici ce se citește și ce se afișează cu append, split și print(*v).
- Repari greșeli frecvente de citire și de afișare.
- Scrii programe care citesc un șir și îl afișează în feluri diferite.

Ce trebuie să știi (o pagină)

Ce vrei	Cum scrii
citești `n`	`n = int(input())`
citești `n` numere, **câte unul pe rând**	`v = []` ↵ `for i in range(n):` ↵ `····v.append(int(input()))`
citești numerele de **pe un rând**	`v = list(map(int, input().split()))`
adaugi la sfârșit	`v.append(x)`
afișezi pe un rând	`print(*v)`
afișezi cu alt separator	`print(*v, sep=", ")`
afișezi un element după altul	`for x in v: print(x, end=" ")`
afișezi invers	`for i in range(len(v) - 1, -1, -1): print(v[i], end=" ")`

ATENȚIE — GREȘELI FRECVENTE

`print(v)` afișează lista **cu paranteze**: `[4, 7, 1]`. Pentru `4 7 1` folosește `print(*v)`.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează?

```
v = []
for i in range(3):
    v.append(i * 2)
print(v)
```

P2. Pentru intrarea de pe două rânduri `3` și `5 1 4`, ce afișează?

```
n = int(input())
v = list(map(int, input().split()))
print(len(v), v[n - 1])
```

P3. Ce afișează?

```
v = [4, 7, 1]
print(*v, sep=" + ")
print(v)
```

P4. Ce afișează?

```
v = [2, 4, 6, 8]
for i in range(len(v) - 1, -1, -1):
    print(v[i], end=",")
```

P5. Ce afișează?

```
v = [1, 2]
v.append(3)
v.append(v[0] + v[2])
print(len(v), v)
```

P6. Pentru intrarea `4`, `10`, `20`, `30`, `40` (fiecare pe rândul lui), ce afișează?

```
n = int(input())
v = []
for i in range(n):
    v.append(int(input()))
for i in range(0, n, 2):
    print(v[i], end=" ")
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P7. Pentru intrarea `3 4` programul afișează `34`, iar noi vrem suma, `7`. De ce se întâmplă asta? Repară-l.

```
v = input().split()
print(v[0] + v[1])
```

P8. Vrem să afișăm șirul în ordine inversă (`1 7 4`), dar programul dă eroare. Găsește greșeala.

```
v = [4, 7, 1]
for i in range(len(v), 0, -1):
    print(v[i], end=" ")
```

P9. Programul citește `n`, apoi `n` numere (câte unul pe rând). Pentru intrarea `2`, `5`, `6` apare o eroare. Care este greșeala?

```
n = int(input())
v = []
for i in range(n):
    x = int(input())
    v.append[x]
print(v)
```

P10. Programul afișează `[5, 8, 2]`, dar noi vrem `5, 8, 2` (fără paranteze). Repară-l cu o singură schimbare.

```
v = [5, 8, 2]
print(v)
```

Nivelul 3 · Scrie cod

★★ scrie cod

P11. Invers. Citește `n` și apoi `n` numere, câte unul pe rând. Afișează-le în ordine inversă, pe un rând, separate prin spațiu.

intrare	ieșire

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

`1` `9`	`9`

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

P12. Din doi în doi. Citește `n`, apoi un rând cu `n` numere. Afișează elementele de pe pozițiile 0, 2, 4, ... (pe un rând).

intrare	ieșire
`5` `10 20 30 40 50`	`10 30 50`
`1` `7`	`7`
`4` `1 2 3 4`	`1 3`

P13. Dublu. Citește `n`, apoi un rând cu `n` numere. Afișează dublul fiecărui număr, separate prin virgulă și spațiu.

intrare	ieșire
`3` `4 5 6`	`8, 10, 12`
`1` `0`	`0`

intrare	ieșire
`2` `-3 3`	`-6, 6`

P14. Mai mari decât primul. Citește `n`, apoi un rând cu `n` numere. Afișează, pe un rând, numerele **mai mari decât primul**. Dacă nu există niciunul, afișează `nimic`.

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

P15. Capete egale. Citește `n`, apoi un rând cu `n` numere. Afișează `da` dacă primul și ultimul număr sunt egale, altfel `nu`.

intrare	ieșire
`3` `4 9 4`	`da`
`4` `1 2 3 4`	`nu`
`1` `5`	`da`

Nivelul 4 · Provocare

★★★ pune mintea la contribuție

P16. Fără n. Pe un singur rând sunt scrise cel puțin două numere, iar numărul lor **nu** este dat. Citește-le și afișează, pe un rând, câte numere sunt și suma dintre primul și ultimul.

intrare	ieșire
`7 2 9`	`3 16`
`4 4 4 4`	`4 8`
`1 5`	`2 6`

P17. Oglinda. Citește `n` și apoi un rând cu `n` numere. Afișează, pe **un singur rând**, șirul, urmat imediat de același șir inversat.

intrare	ieșire
`3` `1 2 3`	`1 2 3 3 2 1`

intrare	ieșire
`1` `5`	`5 5`
`2` `7 8`	`7 8 8 7`

P18. Prenumele din clasă (cu text!). Pe un rând sunt scrise prenume, despărțite prin spațiu. Citește-le **fără** `map(int, ...)` (sunt texte, nu numere) și afișează-le în ordine inversă, pe un rând. Ce observi la citire, față de numere?

intrare	ieșire
`Ana Dan Ema`	`Ema Dan Ana`
`Sergiu`	`Sergiu`