INFORMATICĂ · PYTHON · CLASA A VII-A · STRUCTURA REPETITIVĂ · LECȚIA 12

Tehnici: sumă, produs, contor, maxim și cifrele unui număr

Durată: 100 min Nivel: gimnaziu · Python Înainte: lecțiile py7g-10, py7g-11

CE VEI ȘTI DUPĂ LECȚIE

- Folosești tiparele clasice: sumă, produs, contor (numărare), maxim și minim.
- Alegi corect valoarea de pornire a fiecărei variabile.
- Calculezi suma și numărul cifrelor unui număr cu while.

1. Patru tipare care apar peste tot

Aproape toate problemele cu repetiții sunt variații ale acelorași tipare. Fiecare are o variabilă care **pornește de la o valoare potrivită** și **se schimbă la fiecare pas**.

Tiparul	Variabila	Pornește de la	La fiecare pas
suma	`s`	`0`	`s = s + x`
produsul	`p`	`1`	`p = p * x`
contorul (numărătoarea)	`c`	`0`	`c = c + 1` (doar când e îndeplinită o condiție)
maximul / minimul	`mx` / `mn`	**primul număr**	actualizezi dacă găsești unul mai mare / mai mic

DE ȘTIUT

Suma pornește de la **0** (numărul care nu schimbă o adunare), iar produsul de la **1** (numărul care nu schimbă o înmulțire). Dacă produsul ar porni de la 0, ar rămâne mereu 0!

2. Produsul: factorialul

`n! = 1 · 2 · 3 · … · n`. De exemplu, `5! = 120`.

```
n = int(input())
p = 1
for i in range(1, n + 1):
    p = p * i
print(p)
```

Pentru intrarea `5`:

```
120
```

3. Contorul: numărăm ce ne interesează

Se citesc `n` numere. Câte dintre ele sunt **pare**? Contorul crește **doar** când condiția este adevărată, deci avem un `if` în buclă:

```
n = int(input())
c = 0
for i in range(n):
    x = int(input())
    if x % 2 == 0:
        c = c + 1
print(c)
```

Pentru intrările `5`, `4`, `7`, `10`, `3`, `8`:

```
3
```

Cele 5 numere sunt 4, 7, 10, 3, 8; pare sunt 4, 10 și 8.

4. Maximul și minimul

Se citesc `n` numere (`n ≥ 1`). Care este cel mai mare și care este cel mai mic?

Ideea: primul număr este, deocamdată, *și* cel mai mare, *și* cel mai mic. Apoi citim restul și corectăm când găsim ceva mai bun.

```
n = int(input())
x = int(input())
mx = x
mn = x
for i in range(n - 1):       # au mai rămas n - 1 numere
    x = int(input())
    if x > mx:
        mx = x
    if x < mn:
        mn = x
print("Maximul:", mx)
print("Minimul:", mn)
```

Pentru intrările `5` și apoi `7`, `2`, `9`, `5`, `-3`:

```
Maximul: 9
Minimul: -3
```

> **ATENȚIE — GREȘELI FRECVENTE**
>
> **Nu porni maximul de la 0!** Dacă toate numerele sunt negative (de exemplu `-5`, `-2`), maximul real este `-2`, dar un `mx = 0` ar rămâne `0`, pentru că nimic nu l-ar depăși. De aceea pornim de la **primul număr citit**.

5. Cifrele unui număr

Cu `while n > 0`, `n % 10` (ultima cifră) și `n = n // 10` (o tăiem), parcurgi toate cifrele, de la **ultima spre prima**. Programul de mai jos calculează **suma cifrelor** și numără **cifrele pare**:

```
n = int(input())
s = 0
pare = 0
while n > 0:
    cifra = n % 10
    s = s + cifra
    if cifra % 2 == 0:
        pare = pare + 1
    n = n // 10
print("Suma cifrelor:", s)
print("Cifre pare:", pare)
```

Pentru intrarea `4872`:

```
Suma cifrelor: 21
Cifre pare: 3
```

Observă cum se **combină** tiparele: `s` este o sumă, `pare` este un contor, iar bucla este un `while` care taie cifrele.

Numărul oglindit. Dacă „lipești" cifrele în ordinea în care le scoți, obții numărul scris invers:

```
n = int(input())
r = 0
while n > 0:
    r = r * 10 + n % 10
    n = n // 10
print(r)
```

Pentru intrarea `1234`:

```
4321
```

Pașii lui `r`: `0 → 4 → 43 → 432 → 4321`. La fiecare pas, `r * 10` face loc unei noi cifre la final.

6. Cum alegi tiparul?

Enunțul zice...	Folosești
„suma", „adună"	sumă, pornește de la `0`
„produsul", „factorial"	produs, pornește de la `1`
„câte", „numără"	contor + `if`, pornește de la `0`
„cel mai mare", „cel mai mic"	maxim/minim, pornește de la **primul număr**
„cifrele lui n"	`while n > 0` cu `n % 10` și `n // 10`

7. Verifică-te

1. Cu ce valori pornesc `s`, `p` și `c`?
2. Care este valoarea finală a lui `p`?

```
p = 1
for i in range(1, 5):
    p = p * i
```

3. Completează: „Câte numere din 1...20 sunt multipli de 3?"

```
c = 0
for i in range(1, 21):
    if ______:
        c = c + 1
print(c)
```

4. Pentru ce valori ale lui `n` nu se execută niciodată corpul buclei `while n > 0`?
5. Ce afișează programul pentru `n = 305`?

```
n = 305
s = 0
while n > 0:
    s = s + n % 10
    n = n // 10
print(s)
```

Răspunsuri

1. `s = 0`, `p = 1`, `c = 0`.
2. `24` (`1 · 2 · 3 · 4`; `range(1, 5)` se oprește la 4).
3. `i % 3 == 0`; programul afișează `6` (3, 6, 9, 12, 15, 18).
4. Pentru `n = 0` și pentru numerele negative.
5. `8` (5 + 0 + 3).

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] **#46** **sumapare** — ușoară
  https://www.pbinfo.ro/probleme/46/sumapare
- [ ] **#49** **factorial** — ușoară
  https://www.pbinfo.ro/probleme/49/factorial
- [ ] **#354** **n-maxim** — ușoară
  https://www.pbinfo.ro/probleme/354/n-maxim
- [ ] **#10** **suma-cifrelor** — ușoară
  https://www.pbinfo.ro/probleme/10/suma-cifrelor

Temă pentru acasă (pbinfo)

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

- [ ] #66 **numaruldecifre** — ușoară
  https://www.pbinfo.ro/probleme/66/numaruldecifre
- [ ] #68 **ciframaxima** — ușoară
  https://www.pbinfo.ro/probleme/68/ciframaxima
- [ ] #69 **oglindit** — ușoară
  https://www.pbinfo.ro/probleme/69/oglindit