INFORMATICĂ · CLASA A VI-A · STRUCTURI REPETITIVE · LECȚIA 5

Algoritmi cu repetiție: sume, numărări, cel mai mare număr

Durată: 45 min Nivel: gimnaziu Înainte: lecțiile g6-04

CE VEI ȘTI DUPĂ LECȚIE

- Calculezi suma mai multor numere citite pe rând.
- Numeri câte numere au o proprietate (de exemplu, sunt pare).
- Afli cel mai mare număr dintr-un șir.
- Implementezi acești algoritmi în Scratch cu variabile.

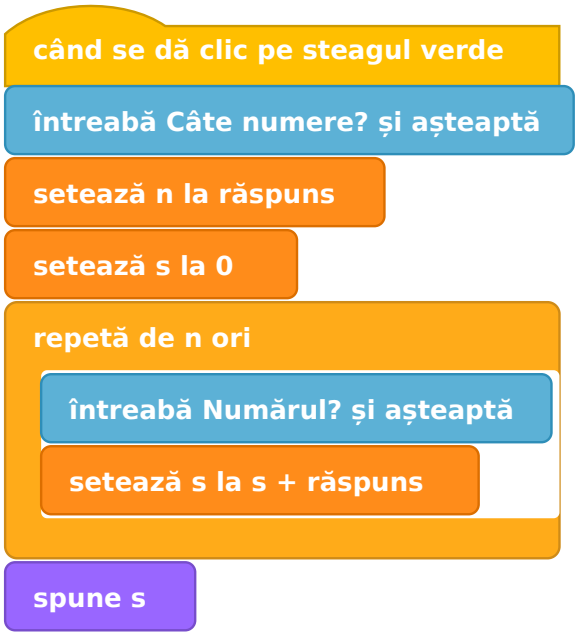
1. Trei modele de algoritm

Aproape orice problemă cu șiruri de numere folosește unul dintre trei modele: **suma**, **numărarea** și **maximul**. Ai deja tot ce trebuie: repetiții și variabile. Le punem acum cap la cap.

2. Suma a n numere

Ideea: o variabilă `s` care începe de la **0**; la fiecare număr citit, îl adunăm la `s`.

```
citește n
s ← 0
pentru i ← 1, n execută
    citește x
    s ← s + x
sfârșit pentru
scrie s
```



Suma în Scratch

Urmărim pentru n = 4 și numerele 2, 5, 1, 4:

	x	s
început		0
i = 1	2	2
i = 2	5	7
i = 3	1	8
i = 4	4	**12**

s începe de la 0 și adună fiecare număr; la final s = 12

Urmărire

i	x	s înainte	s după
1	2	0	2

i	x	s înainte	s după
2	5	2	7
3	1	7	8
4	4	8	12

DE CE AȘA?

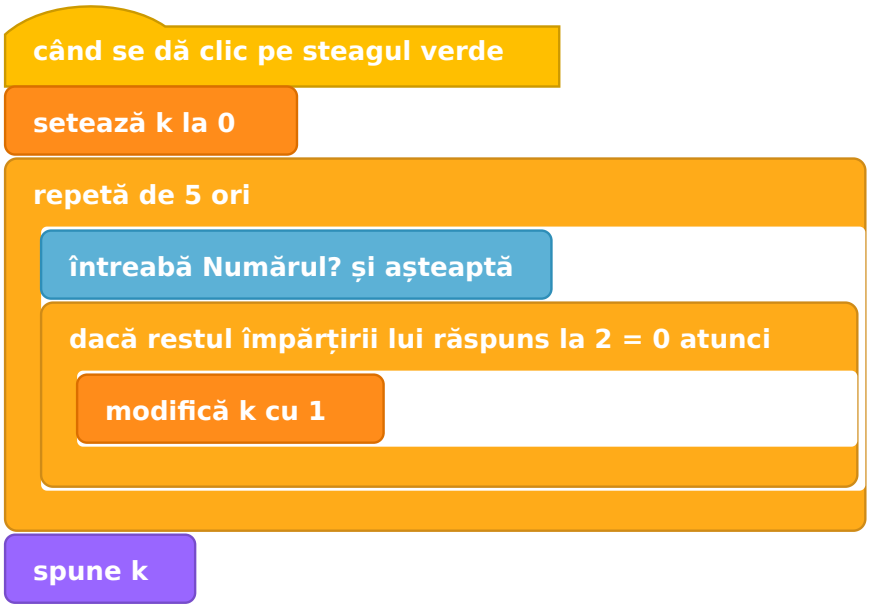
De ce `s` începe de la 0? Zero este **elementul neutru** la adunare: nu schimbă suma. Dacă începi de la alt număr, adaugi acel număr în plus la rezultat. **De ce citim x în interior?** La fiecare repetare avem un număr nou; `x` este mereu „numărul curent”, `s` ține minte totalul.

Media = suma împărțită la câte numere sunt: `m ← s / n`.

3. Numărarea

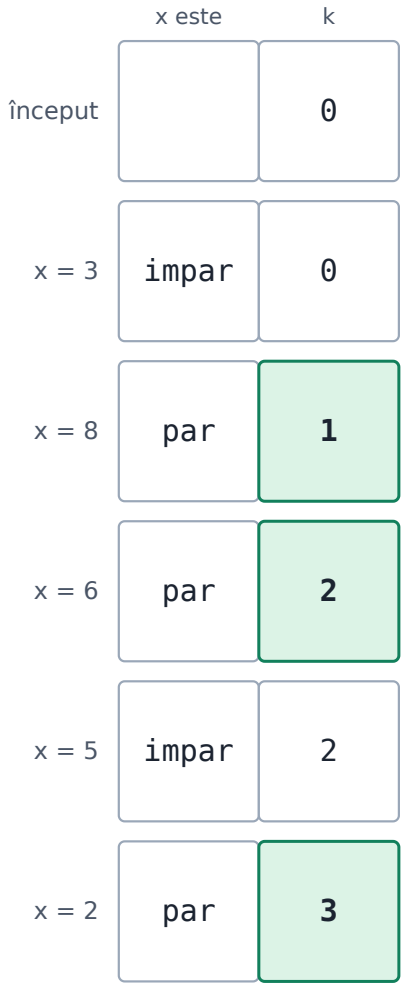
Câte numere pare sunt printre cele n? O variabilă `k` care începe de la **0** și crește cu 1 doar când numărul are proprietatea:

```
k ← 0
pentru i ← 1, n execută
    citește x
    dacă x mod 2 = 0 atunci
        k ← k + 1
    sfârșit dacă
sfârșit pentru
scrie k
```



Numărarea în Scratch

Pentru numerele 3, 8, 6, 5, 2:



k crește doar când x este par; răspuns k = 3

Urmărire

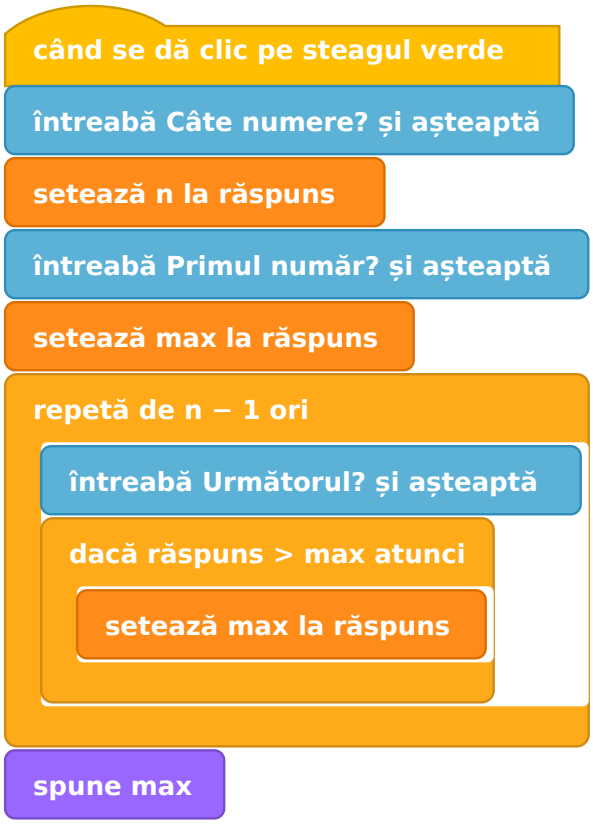
DE CE AȘA?

Care e diferența față de sumă? La sumă adunăm **valoarea** numărului (`s ← s + x`) la fiecare pas. La numărare adunăm **1** (`k ← k + 1`), dar numai când e îndeplinită condiția. Aceeași structură, alt scop.

4. Cel mai mare număr

Presupunem că primul număr este cel mai mare; apoi comparăm fiecare număr nou cu „campionul” de până acum.

```
citește n
citește x
max ← x
pentru i ← 2, n execută
    citește x
    dacă x > max atunci
        max ← x
    sfârșit dacă
sfârșit pentru
scrie max
```



Maximul în Scratch

Pentru numerele 4, 9, 2, 9, 7:

	max	x > max?
x = 4 (primul)	4	–
x = 9	9	**da**
x = 2	9	nu
x = 9	9	nu
x = 7	9	nu

campionul se schimbă doar când vine un număr STRICT mai mare; răspuns 9

Urmărire

SFAT

Gândește-te la un concurs: primul concurent e „campion provizoriu”. Fiecare concurent nou îl provoacă; dacă e mai bun, devine el campion.

DE CE AȘA?

De ce nu pornim cu `max ← 0`? Dacă toate numerele sunt negative (de exemplu −5, −2, −9), maximul real este −2, dar cu `max ← 0` am răspunde greșit 0. Pornind cu **primul număr**, `max` este mereu o valoare din șir. **De ce `>` și nu `≥`?** Pentru maxim valoarea e aceeași; `>` evită o atribuire inutilă.

5. Cum alegi modelul?

Întrebarea	Model	Variabila începe de la
„care este suma / totalul?”	sumă	0
„câte numere au proprietatea ...?”	numărare	0
„care este cel mai mare / cel mai mic?”	maxim / minim	primul număr
„care este produsul?”	produs	**1** (element neutru la înmulțire)

6. Greșeli frecvente

Greșeala	Efect	Corect
`s` nu este inițializat	rezultat greșit sau rămas de la rularea veche	`s ← 0` înainte de repetare
`s ← x` în loc de `s ← s + x`	păstrezi doar ultimul număr	adună
`k ← k + 1` în afara lui `dacă`	numeri toate numerele	pune în interiorul deciziei
`max ← 0` cu numere negative	maxim greșit	primul număr
media: `s / i` în loc de `s / n`	împărțire greșită	împarte la câte numere sunt
citești `x` în afara repetării	folosești mereu același număr	citește în interior

7. Exersează pe hârtie

1. Urmărește suma pentru n = 3 și numerele 3, 7, 2. Care e suma? Dar media?

2. Câte numere sunt mai mari decât 10 printre 12, 5, 18, 10, 30? Completează tabelul pentru k.

3. Care este cel mai mare număr dintre 5, 1, 9, 3, 9? Urmărește max.

4. Calculează suma numerelor pare dintre 4, 7, 10, 3, 6 (adună doar numerele pare). Care e răspunsul?

5. Scrie în pseudocod algoritmul pentru cel mai mic număr dintre n numere (minimul).

6. Ce valoare are `max` la final pentru numerele −5, −2, −9 dacă pornești cu `max ← −5`? Dar dacă ai fi pornit cu `max ← 0`?

7. Scrie algoritmul care afișează suma numerelor de la 1 la n. Ce rezultă pentru n = 100? (indiciu: perechi 1 + 100, 2 + 99, ...)

8. Ce afișează `k ← 0`, `pentru i ← 1, 6 execută dacă i mod 3 = 0 atunci k ← k + 1`, `scrie k`?

9. Scrie algoritmul care calculează produsul a n numere. Cu ce valoare începe produsul și de ce?

10. Pentru notele 7, 9, 10, 6: calculează suma, media și cea mai mare notă.

8. Provocări în Scratch

1. Pisica întreabă 5 numere și spune suma și media lor.
2. Pisica întreabă numere până când primește 0 și spune câte numere pozitive au fost.
3. Pisica întreabă 6 note și spune cea mai mare și cea mai mică notă.
4. **Dificil:** pisica întreabă n numere și spune câte sunt pare și câte impare (două variabile de numărare).

9. Mini-grile

1. Cu ce valoare începe variabila pentru **sumă**? a) 1 b) 0 c) primul număr d) −1

2. Pentru a număra numerele pare, variabila k: a) crește la fiecare număr b) crește doar la numerele pare c) scade d) se șterge

3. Pentru produsul numerelor, variabila începe de la: a) 0 b) 1 c) primul număr d) 10

4. Pentru numerele 4, 9, 2, 9, 7, valoarea maximă este: a) 4 b) 7 c) 9 d) 31

5. Media a 4 numere cu suma 20 este: a) 4 b) 5 c) 20 d) 80

10. Mai departe (pentru curioși)

Formula lui Gauss. Suma numerelor de la 1 la n este `n * (n + 1) / 2`. Pentru n = 100: 100 · 101 / 2 = 5050. Legenda spune că matematicianul Gauss a descoperit-o la 9 ani, grupând numerele în perechi cu aceeași sumă (1 + 100, 2 + 99, ...). Avantaj: un singur calcul în loc de 100 de adunări.

Minim și maxim împreună. Poți ține două „campioni" în același timp, `min` și `max`, într-o singură parcurgere.

Poziția maximului. Dacă vrei să știi **al câtelea** număr a fost cel mai mare, adaugi o variabilă `poz` care se actualizează odată cu `max`.

În liceu. Aceste trei modele devin primele programe în C++ sau Python (sumă, numărare, maxim), cu `for`, `if` și `cin` / `input`. Lecțiile IX.8–IX.9 le reiau pe larg.

11. Ce am învățat

- **Sumă:** pornește de la 0 și adună fiecare număr.
- **Numărare:** pornește de la 0 și crește cu 1 când e îndeplinită condiția.
- **Maxim:** primul e campion, apoi compari fiecare număr nou.
- Produsul pornește de la 1; media = suma / câte numere sunt.

Citește și exersează în plus

- Scratch — editorul online — aici rulezi exemplele din lecție https://scratch.mit.edu/projects/editor/
- Scratch Wiki — Variables — cum folosești variabilele în programe (engleză) https://en.scratch-wiki.info/wiki/Variables
- Code.org — Hour of Code — puzzle-uri de programare cu blocuri https://code.org/hourofcode/overview

Răspunsuri

Exersează pe hârtie

1. s: 0 → 3 → 10 → 12. Suma **12**, media 12 / 3 = **4**.

2. Numerele mai mari decât 10: 12 și 18 și 30 (10 nu este > 10). k: 1 (12), 1 (5), 2 (18), 2 (10), 3 (30). Răspuns **3**.

3. max: 5 → 5 (1 nu e mai mare) → 9 → 9 (3 nu e) → 9 (9 nu e strict mai mare). Răspuns **9**.

4. Pare: 4, 10, 6. Suma: 4 + 10 + 6 = **20**.

5. `citește n, x`, `min ← x`, `pentru i ← 2, n execută citește x, dacă x < min atunci min ← x`, `scrie min`.

6. Cu `max ← −5`: −2 > −5 → max = −2; −9 nu; răspuns **−2**. Cu `max ← 0`: niciun număr nu e > 0, rămâne **0** (greșit: 0 nici nu este în șir).

7. `citește n`, `s ← 0`, `pentru i ← 1, n execută s ← s + i`, `scrie s`. Pentru n = 100: **5050**.

8. i = 3 și i = 6 sunt multipli de 3: k = **2**.

9. `p ← 1`, `pentru i ← 1, n execută citește x, p ← p * x`, `scrie p`. Începe cu 1 pentru că 1 este elementul neutru la înmulțire (cu 0 produsul ar rămâne mereu 0).

10. Suma: 7 + 9 + 10 + 6 = **32**; media: 32 / 4 = **8**; cea mai mare: **10**.

Mini-grile

1. b. **2.** b. **3.** b. **4.** c. **5.** b.