OBIECTIVE

O1 Înlocuirea repetiției cu **test inițial** prin repetiție cu **test final**, astfel încât să rezulte **algoritm echivalent**

O2 Înlocuirea repetiției cu **test final** prin repetiție cu **test inițial**, astfel încât să rezulte **algoritm echivalent**

O3 Înlocuirea repetiției cu **contor** prin repetiție cu **test inițial**, astfel încât să rezulte **algoritm echivalent**

Prof. DARIE SIMONA
Colegiul Național „Nicolae Grigorescu”, Câmpina

INFORMATICĂ, clasa a IX-a

DEFINIȚIE

DEFINIȚIE

Doi algoritmi se numesc

echivalenți

dacă, oricare ar fi

un set de date de intrare,

introdus în ambii algoritmi rezultă,

același set de date de ieșire.

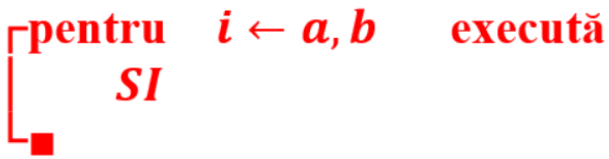
SINTEZĂ

Structura repetitivă cu TEST INIȚIAL

SINTEZĂ

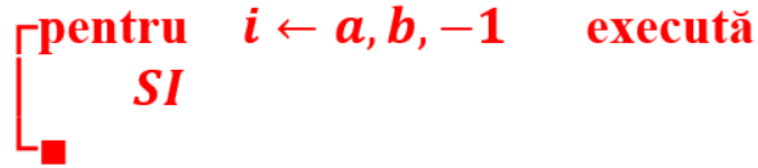
Structura repetitivă cu TEST FINAL

SINTEZĂ



Structura repetitivă CU CONTOR, de pas 1

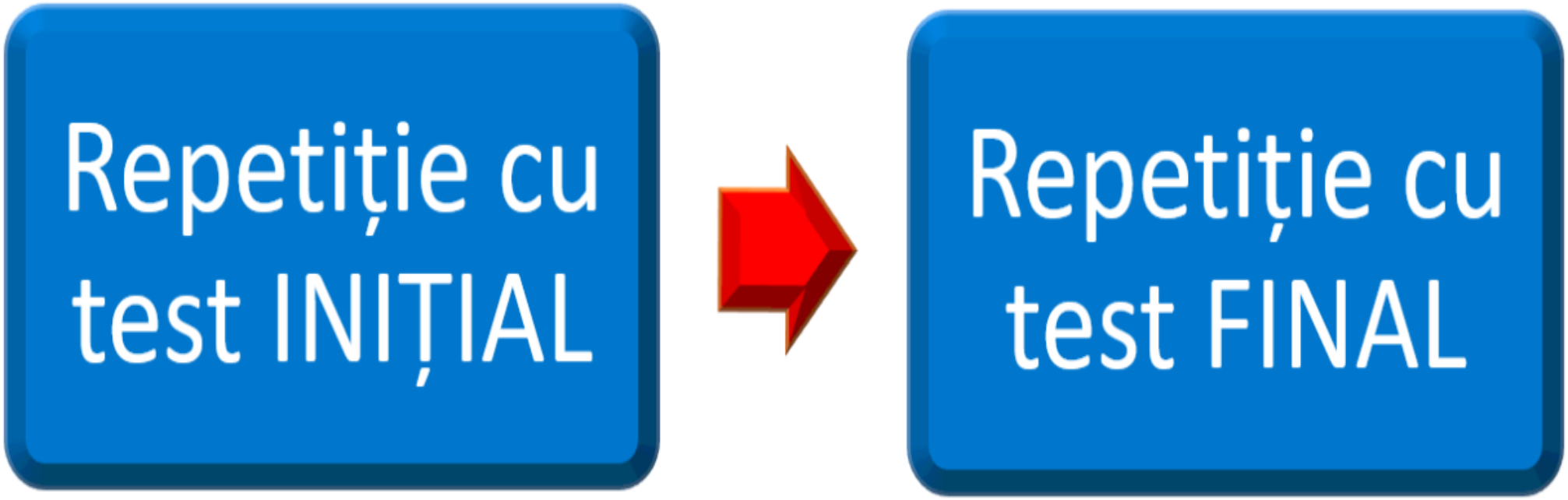
SINTEZĂ



Structura repetitivă cu CONTOR, de pas -1

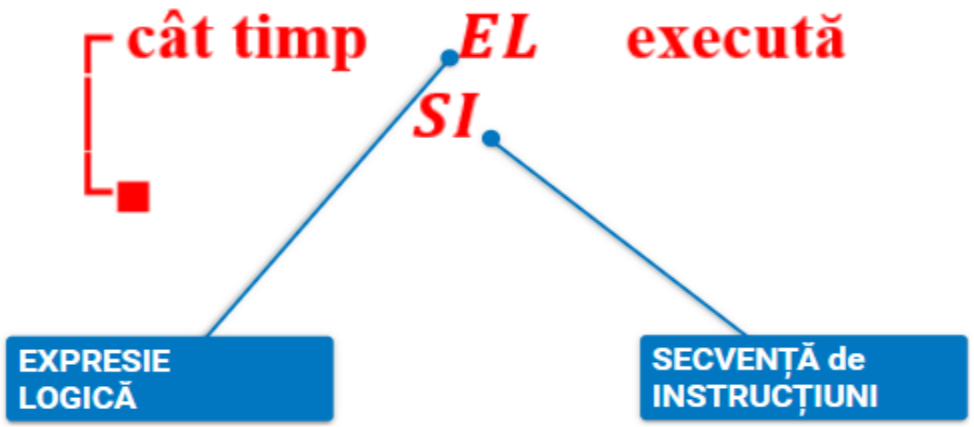
Repetiție cu test INIȚIAL ==> test FINAL

Echivalarea repetiției cu test INIȚIAL



IPOTEZĂ: Algoritmul dat conține structură repetitivă cu test INIȚIAL
CERINȚĂ: Algoritm echivalent înlocuind repetiția cu alt tip de repetiție

IPOTEZĂ

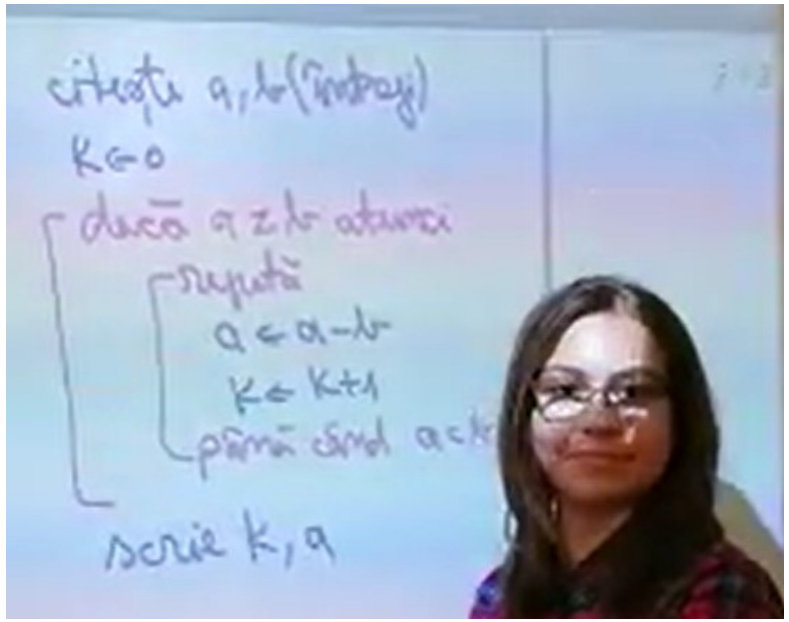


ECHIVALARE

dacă *EL* atunci
  repetă
    SI
  până când *EL – negat*

Algoritmul dat

```
citește a, b (întregi)
k←0
cât timp  a ≥ b  execută
    a←a-b
    k←k+1
scrie k, a
```



Fișă de lucru

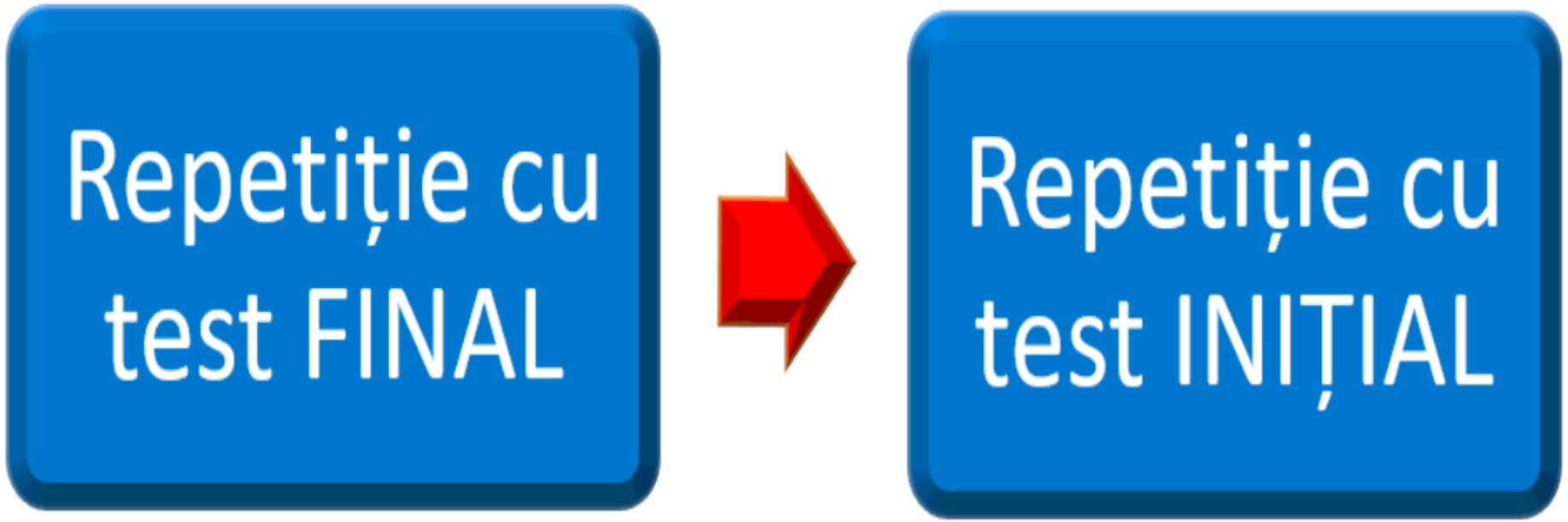
Pentru fiecare dintre algoritmii de mai jos, descrie algoritmul echivalent, înlocuind structura *cât timp - execută* cu o structură *repetă-până când*

1.	2.	3.
citește x,y (numere naturale) ┌dacă x<y atunci │ x←x-y; y←x+y; x←y-x └■ ┌cât timp x≥y execută │ scrie 'A' │ x←x-y │┌dacă x%2=0 atunci scrie 'A' │└altfel scrie 'B' └■	citește x (număr natural) n←0 ┌cât timp x≠0 execută │ y←x; c←0 │ ┌cât timp y>0 execută │ │ ┌dacă y%10>c atunci │ │ │ c←y%10 │ │ └■ │ │ y← [y/10] │ └■ │ n←n*10+c │ citește x (număr natural) └■ scrie n	citește x,y (numere naturale, x≤y) k←0; i←x ┌cât timp i≤y execută │ n←i; c←0 │┌cât timp n>0 și c=0 execută │└┌dacă n%2=1 atunci c←1 └└■ └ n←[n/10] └■ └ k←k+c └ i←i+1 └■ scrie k

Descrie algoritmii echivalenți!

Repetiție cu test FINAL ==> test INIȚIAL

Echivalarea repetiției cu test FINAL



IPOTEZĂ: Algoritmul dat conține structură repetitivă cu test FINAL
CERINȚĂ: Algoritm echivalent înlocuind repetiția cu alt tip de repetiție

IPOTEZĂ

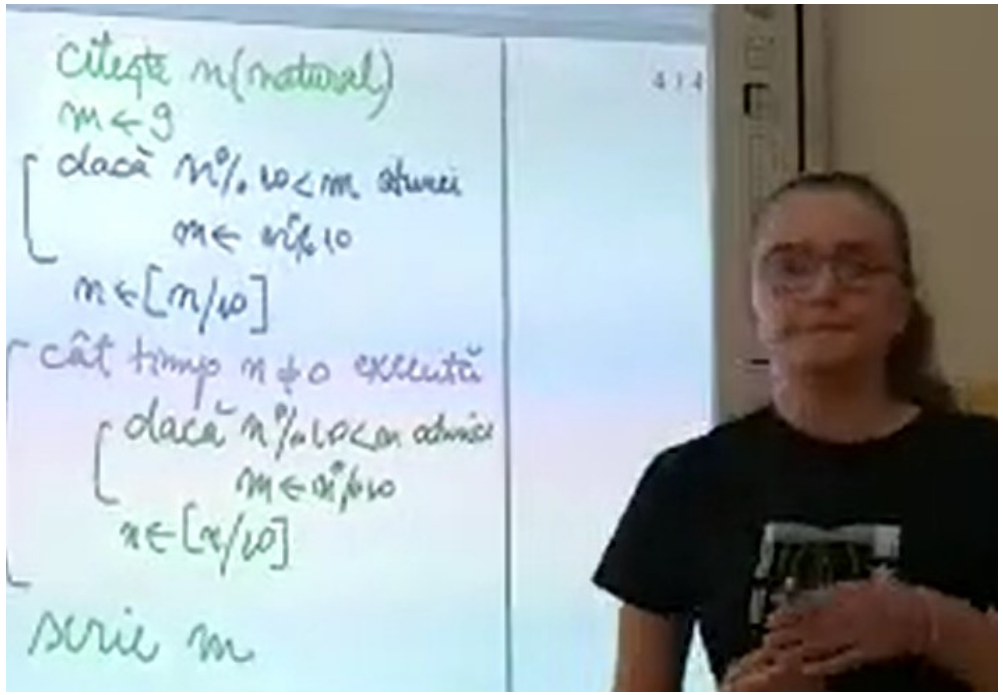
```
repetă
    SI
până când   EL
```

ECHIVALARE

```
SI
cât timp   EL – negat   execută
          SI
```

Algoritmul dat

```
citește n (natural)
m←9
repetă
    dacă   n%10<m   atunci
        m←n%10
    n←[n/10]
până când n=0
scrie m
```



Fișă de lucru

Pentru algoritmii de mai jos, descrie algoritmul echivalent, înlocuind structura *repetă-până când* cu o structură *cât timp - execută*

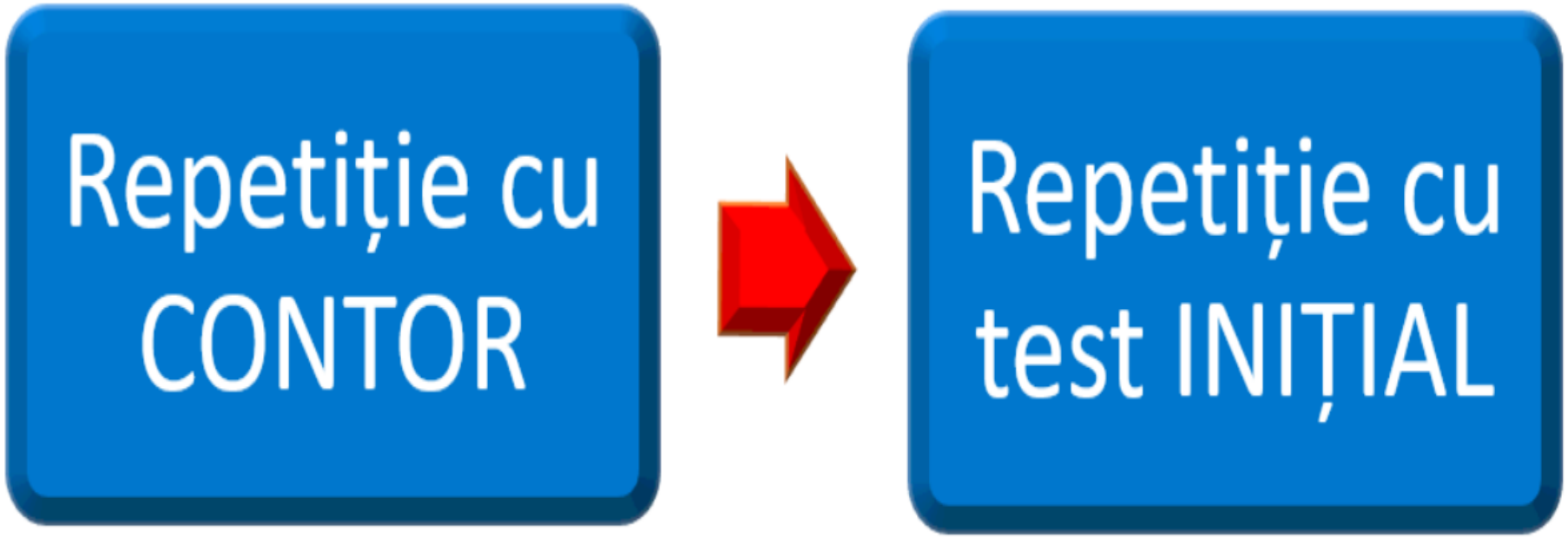
1.	2.	3.

citește n (natural)
a←1
repetă
  scrie a
  a←a+2
până când a>n
	citește x (număr natural)
	
	y←0
	repetă
	y←y*10+9-x%10
	până când x≤y sau y=0
	scrie y
		citește x
		(număr natural nenul)
		y←0
		repetă
		dacă x>9 atunci
		repetă
		x←[x/10]
		până când x≤9
		■
		y←y*10+x
		citește x (număr natural)
		până când x=0
		scrie y

Descrie algoritmii echivalenți!

Repetiția cu CONTOR ==> test INIȚIAL

Echivalarea repetiției cu CONTOR



IPOTEZĂ: Algoritmul dat conține structură repetitivă cu CONTOR
CERINȚĂ: Algoritm echivalent înlocuind repetiția cu alt tip de repetiție

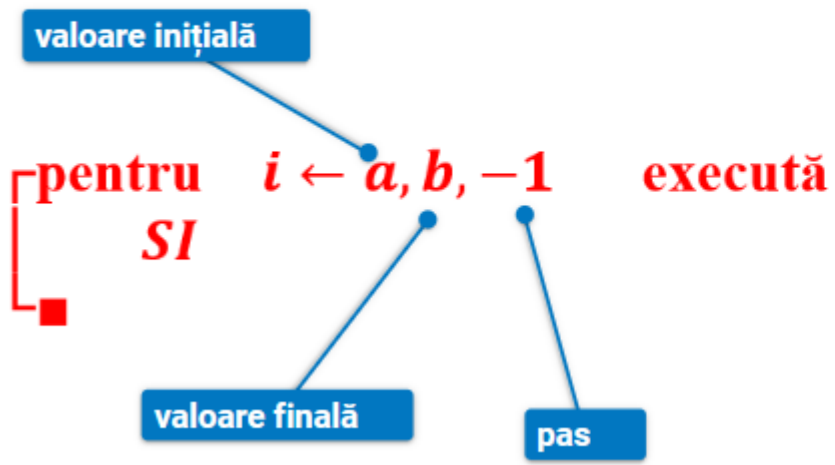
IPOTEZĂ

```
pentru  i ← a, b  execută
    SI
```

ECHIVALARE

```
i←a
cât timp  i ≤ b  execută
    SI
    i←i + 1
```

IPOTEZĂ

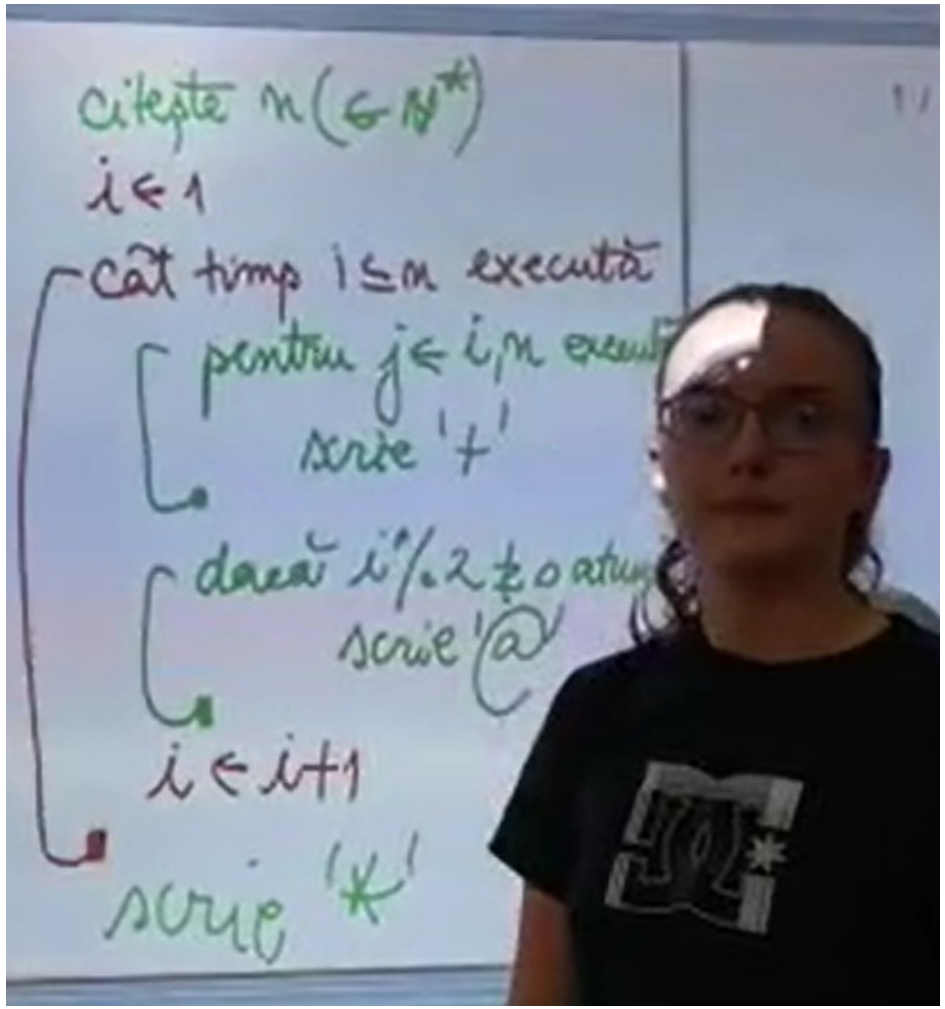


ECHIVALARE

```
i←a
cât timp  i ≥ b  execută
    SI
    i←i − 1
```

Algoritmul dat

```
citeşte n
  (număr natural nenul)
pentru i←1,n execută
  pentru j←i,n execută
    scrie '+'
  dacă i%2≠0 atunci
    scrie '@'
```



Exersare

Fișă de lucru

Pentru algoritmii de mai jos, descrie algoritmul echivalent, înlocuind structura *pentru-execută* cu o structură *cât timp - execută*

1.	2.	3.
citeste n (număr natural) c←0 ┌pentru i←1,n execută │ c←(c+1)%10 │ scrie c └	citeşte a,b (numere întregi) ┌dacă a<b atunci │ a↔b └ ┌pentru x←a,b,-1 execută │ ┌dacă x%2≠0 atunci │ │ scrie x,' ' │ └ └	citeşte n (număr natural nenul) ┌pentru i←1,n-1 execută │ ┌dacă i%2=0 atunci │ │ scrie '#' │ └ │ ┌pentru j←i+1,n execută │ │ scrie '*' │ └ └

Descrie algoritmii echivalenți!

FELICITĂRI!