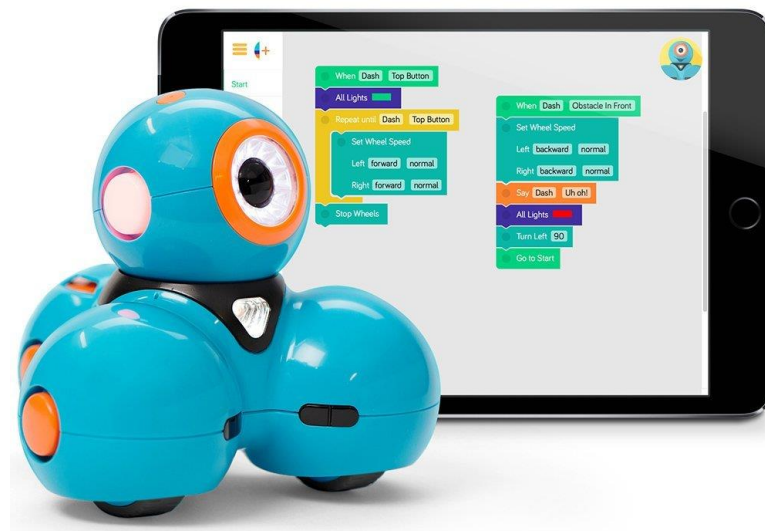


Liceul Teoretic Grigore Moisil



Implementarea unor algoritmi într-un mediu
grafic interactiv – **Blockly**
Structuri repetitive
Clasa a VI-a

Autor: prof. Spătaru Mihaela

CUPRINS

CUPRINS	2
I. Structuri repetitive cu număr necunoscut de pași	3
I.1 Suma mai multor numere	3
I.2 Câte impare.....	4
I.3 Numărul de cifre	4
I.4 Suma cifrelor unui număr	5
I.5 Cifre pare	6
I.6 Cifre impare.....	6
I.7 Cea mai mare cifră	7
I.8 Cea mai mică cifră.....	7
I.9 Prima cifră	8
I.10 Doi prieteni	8
I.11 Cangurul (Blockly și Scratch)	9
II. Structuri repetitive cu număr cunoscut de pași	11
II.1 Afișarea celei mai mari dintre n valori.....	11
II.2 Afișarea celei mai mici dintre n valori	12
II.3 Câte impare/ pare.....	12
II.4 Afișare numere divizibile cu 5.....	12
II.5 Divizori proprii	13
II.6 Divizori proprii eficient	13
II.7 Numerele lui Fibonacci.....	14
II.8 Dărnicie	14
II.9 Moș Crăciun	15
II.10 Joc	16
Bibliografie.....	16

I. Structuri repetitive cu număr necunoscut de pași

I.1 Suma mai multor numere

Enunț: Să se scrie un program care citește de la tastatură mai multe numere naturale, până la introducerea lui 0 și determină suma lor.

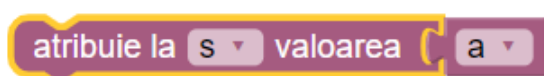
Rezolvare:

Pasul 1. Creez cele o variabilă cu ajutorul căreia rețin valorile numerelor citite, accesând meniul **Variable** → Variabilă nouă...:

Pasul 2. Trag blocul "atribuie la a valoarea ...", în spațiul de lucru și, din meniul **Citiri, afișări** blocul "citește un număr cu mesajul "a= " și le îmbin:



Pasul 3. Creez o variabilă s și îi atribui valoarea lui a.



Pasul 4. Din meniul structuri repetitive aleg instrucțiunea cât timp execută:



Pasul 5. Adaug structuri repetitive condiția $a > 0$, deoarece citesc numerele atâta timp sunt strict pozitive. La final, introduc valoarea 0, pentru a opri citirea și, implicit, ciclarea.

Pasul 6. În continuare, trebuie să citesc următoarele numere și să adaug la sumă valoarea lor. Rezultă că, în interiorul structurii repetitive, adaug două instrucțiuni, una de citire și una de atribuire:



Pasul 7. La final, afișez valoarea sumei:



1.2 Câte impare

Enunț: Să se scrie un program care citește de la tastatură mai multe numere naturale, până la introducerea lui 0 și determină câte dintre ele sunt impare.

Pasul 1. Creez o variabilă **a**, cu ajutorul căreia citesc valorile numerelor introduse de la tastatură, și o variabilă **impare**, pe care o inițializez cu 0.

Pasul 2. Citesc primul număr și, dacă este impar, inițializez variabila **impare** cu 1.

Pasul 3. Adaug structura repetitivă **cât timp execută**, cu condiția **a > 0**.

Pasul 4. În interiorul structurii repetitive, adaug două instrucțiuni, una de citire și o instrucțiune alternativă, pentru a verifica paritatea numerelor citite.

Pasul 4. Dacă numărul citit este impar, cresc valoarea variabilei **impar** cu o unitate.

Pasul 5. Afișez valoarea variabilei **impare**.



1.3 Numărul de cifre

Enunț: Să se scrie un program care citește de la tastatură un număr natural **n** și determină și afișează numărul cifrelor lui **n**.

Observație: Dacă pentru **n** se introduce valoarea 0, programul trebuie să afișeze **1 (o cifră)**.

Pasul 1. Creez variabilele **n** și **nrcif**.

Pasul 2. Atribui variabilei **n** o valoare introdusă de la tastatură.

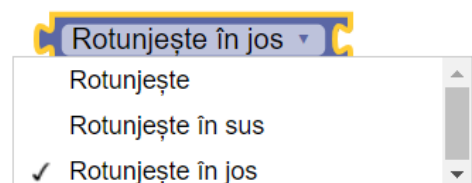
Pasul 3. Dacă **n=0**, atribui variabilei **nrcifre** valoarea 1.

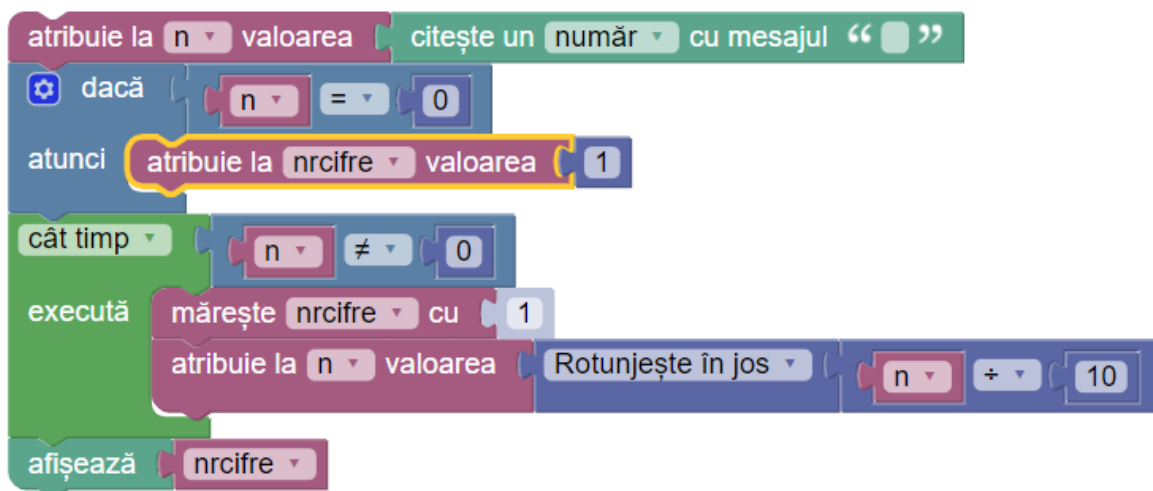
Pasul 4. Adaug structura repetitivă **cât timp execută**, cu condiția **n ≠ 0**.

Pasul 5. În interiorul structurii repetitive, adaug instrucțiunea de incrementare a variabilei **nrcifre** cu o unitate.

Pasul 6. În interiorul structurii repetitive, adaug instrucțiunea de tăierea a ultimei cifre a unui număr. Pentru aceasta, vom utiliza o funcție predefinită, **rotunjește în jos**, în locul operatorului div (/), din meniul **Funcții predefinite**. Prin această operație, golesc numărul **n** de cifre. Structura repetitivă se va opri când valoarea variabilei **n** va ajunge la 0.

Pasul 7. Afișez numărul de cifre, **nrcifre**.





I.4 Suma cifrelor unui număr

Enunț: Să se scrie un program care citește de la tastatură un număr natural n și determină și afișează suma cifrelor lui n .

Observație: Pentru variabila n , introduc o valoare cu un număr necunoscut de cifre, maxim 9.

Pasul 1. Creez variabilele n și **sumacifre**.

Pasul 2. Atribui variabilei n o valoare introdusă de la tastatură.

Pasul 3. Inițializez **sumacifre** cu 0.

Pasul 4. Adaug structura repetitivă cât timp execută, cu condiția $n \neq 0$.

Pasul 5. Creez variabila c , în care voi salva valoarea ultimei cifre, obținută prin operația $\text{rest}(n \div 10)$.

Pasul 6. În interiorul structurii repetitive, la variabila **sumacifre** suma de la pasul anterior și ultima cifră, cu ajutorul instrucțiunii de atribuire.

Pasul 7. Tai ultima cifră a numărului, prin operația de rotunjire în jos, Prin aceasta, golesc numărul n de cifre. Structura repetitivă se va opri când valoarea variabilei n va ajunge la 0.

Pasul 8. Afișez suma cifrelor, **sumacifre**.



I.5 Cifre pare

Enunț: Se citește de la tastatură un număr natural n . Să se determine câte cifre pare conține.

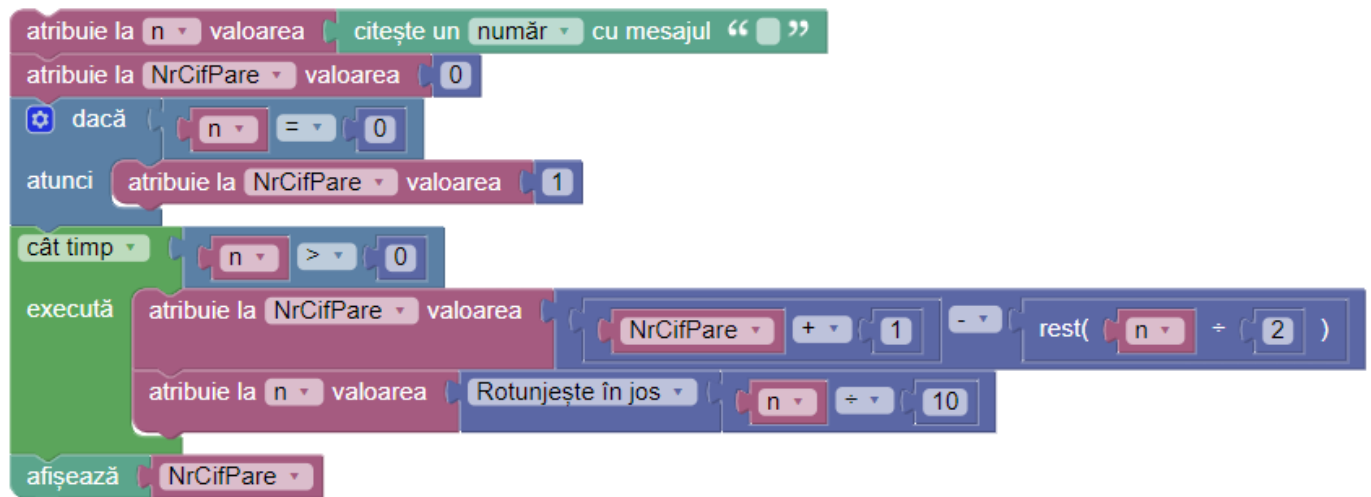
Exemplu: Dacă pentru n se citește **12638**, se va afișa **3**.

Soluție: Pentru a simplifica algoritmul, se pot face câteva observații:

1. $n \% 10 \% 2$ este totuna cu $n \% 2$. Deci, la fiecare pas e suficient să testăm paritatea numărului n , scutind un calcul și o variabilă. Pentru aceasta, trebuie să facem testul înainte de a-l împărți pe n la 10.
2. Ce valori poate lua restul împărțirii la 2? Zero sau unu. Când adunăm 1 la numărul de cifre pare? Atunci când restul este 0. Atunci când restul este unu, trebuie să-l scădem, pentru a nu aduna și numărul de cifre impare.



3. Dacă pentru n se citește valoarea 0, algoritmul trebuie să afișeze 1. Pentru aceasta, vom adăuga o structură alternativă care verifică dacă n este 0 și, în acest caz, atribuie variabilei **NrCifPare** valoarea 1.

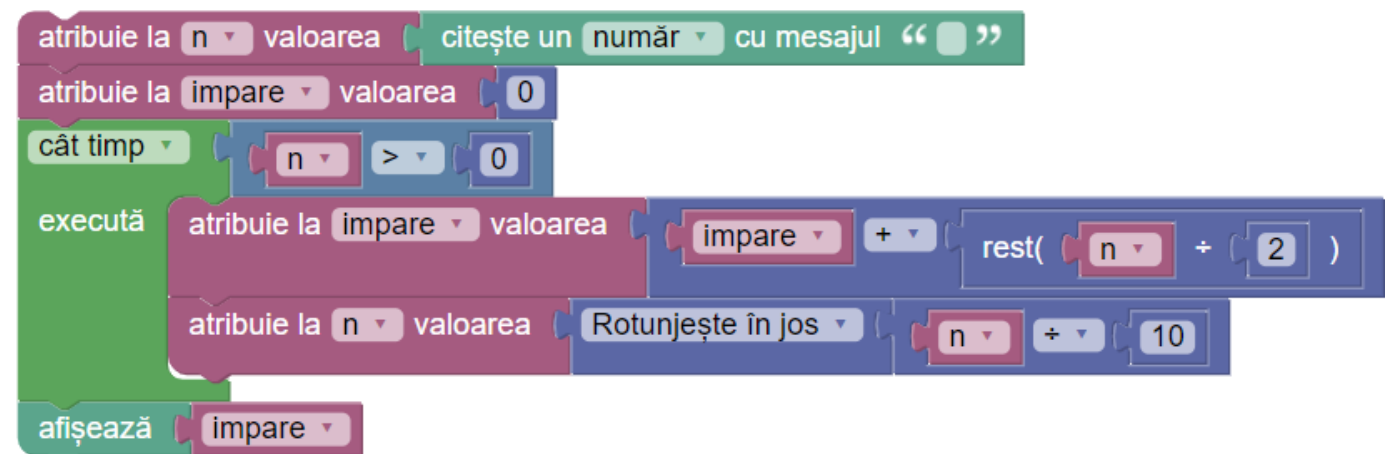


I.6 Cifre impare

Enunț: Se citește de la tastatură un număr natural n . Să se determine câte cifre impare conține.

Exemplu: Dacă pentru n se citește **12638**, se va afișa **2**.

Soluție: La fiecare pas, contorizăm restul împărțirii variabilei n la 2, astfel obținem numărul de cifre impare.



I.7 Cea mai mare cifră

Enunț: Se citește de la tastatură un număr natural n . Să se determine și să se afișeze cea mai mare cifră a numărului.

Exemplu: Dacă pentru n se citește 12638, se va afișa 8.

Pasul 1. Creez variabilele n și **CifraMax**.

Pasul 2. Atribui variabilei n o valoare introdusă de la tastatură.

Pasul 3. Inițializez **CifraMax** cu 0.

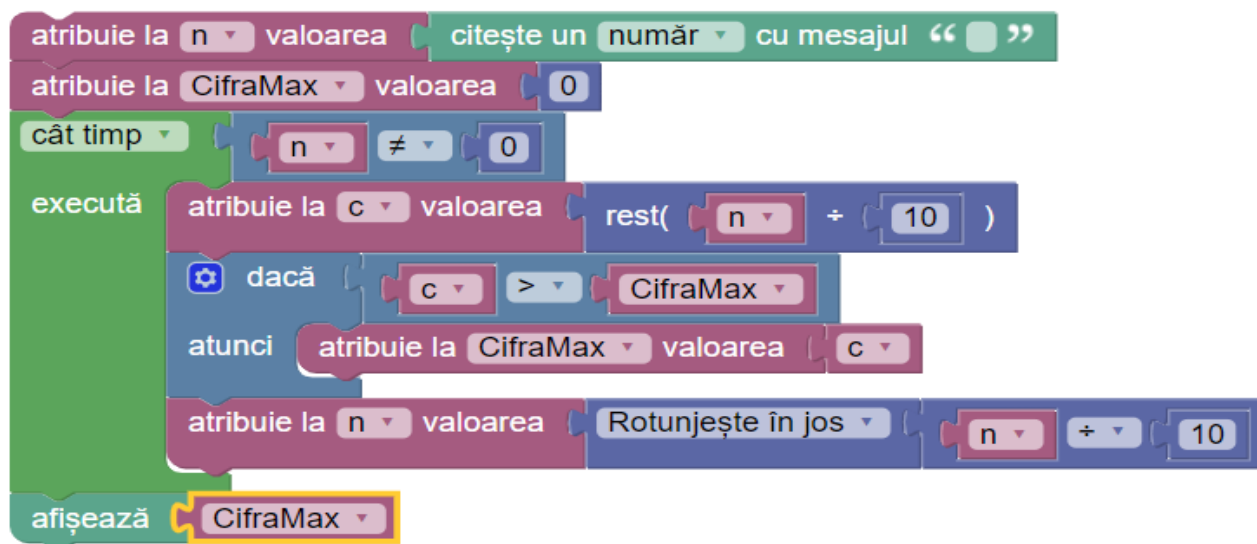
Pasul 4. Adaug structura repetitivă cât timp execută, cu condiția $n \neq 0$.

Pasul 5. Creez variabila c , în care voi salva valoarea ultimei cifre, obținută prin operația $\text{rest}(n \div 10)$.

Pasul 6. În interiorul structurii repetitive, compar cifra curentă cu **CifraMax**. Dacă cifra curentă este mai mare decât **CifraMax**, îi atribui variabilei **CifraMax** valoarea cifrei curente.

Pasul 7. Tai ultima cifră a numărului, prin operația de rotunjire în jos, Prin aceasta, golesc numărul n de cifre. Structura repetitivă se va opri când valoarea variabilei n va ajunge la 0.

Pasul 8. Afișez cea mai mare cifră, **CifraMax**.



I.8 Cea mai mică cifră

Enunț: Se citește de la tastatură un număr natural n . Să se determine și să se afișeze cea mai mică cifră a numărului. Exemplu: Dacă pentru n se citește 12638, se va afișa 1.

Observații:

1. Cea mai mică cifră se inițializează cu 9.



2. În interiorul structurii alternative **dacă atunci** se schimbă sensul operatorului relațional:



I.9 Prima cifră

Enunț: Se citește de la tastatură un număr natural n . Să se determine și să se afișeze prima cifră a numărului.

Exemplu: Dacă pentru n se citește 57634, se va afișa 5.

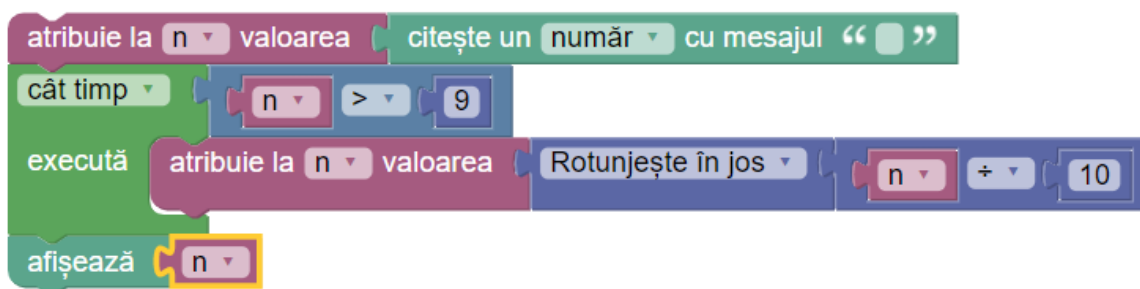
Pasul 1. Creez variabila n .

Pasul 2. Atribui variabilei n o valoare introdusă de la tastatură.

Pasul 3. Aduug structura repetitivă **cât timp ... execută**, cu condiția $n > 9$.

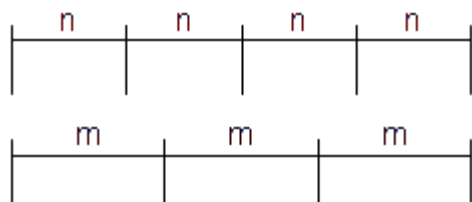
Pasul 4. Cât timp $n > 9$ (n are mai mult de o cifră), tai ultima cifră a numărului, prin operația de rotunjire în jos. Prin aceasta, golesc numărul n de cifre. Structura repetitivă se va opri când valoarea variabilei n va ajunge mai că sau egală cu 9.

Pasul 5. Afișez noua valoare a lui n , care este prima cifră.



I.10 Doi prieteni

Enunț: Doi prieteni, un iepure și o broscuță joacă un joc: pornesc de la o linie de start și încep să sară. Broasca sare n centimetri, iar iepurele m centimetri. Cine este mai în spate vine la rând să sară. Jocul se termină atunci când iepurele și broasca sunt iarăși la egalitate. Câți centimetri au parcurs cei doi prieteni?

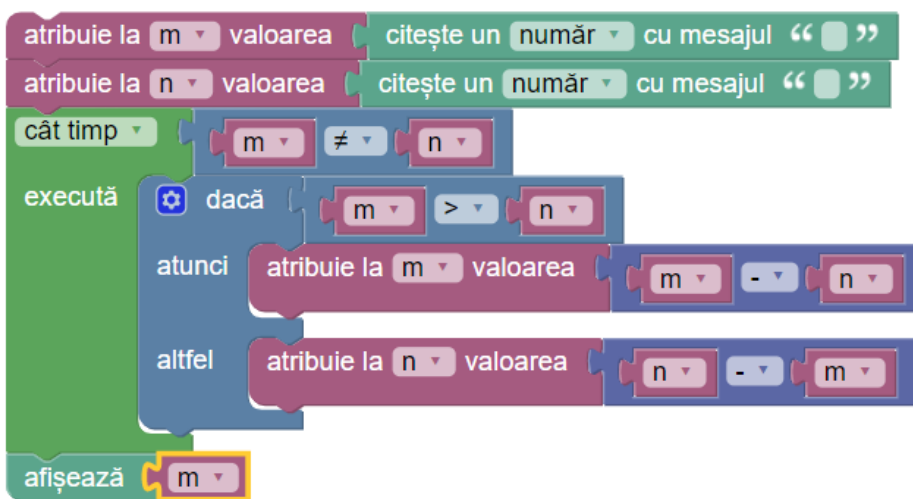


Soluție: după cum se vede și din figură, broscuța și iepurele vor sări o lungime egală cu cel mai mic multiplu comun al numerelor m și n - $\text{cmmmc}(m, n)$.

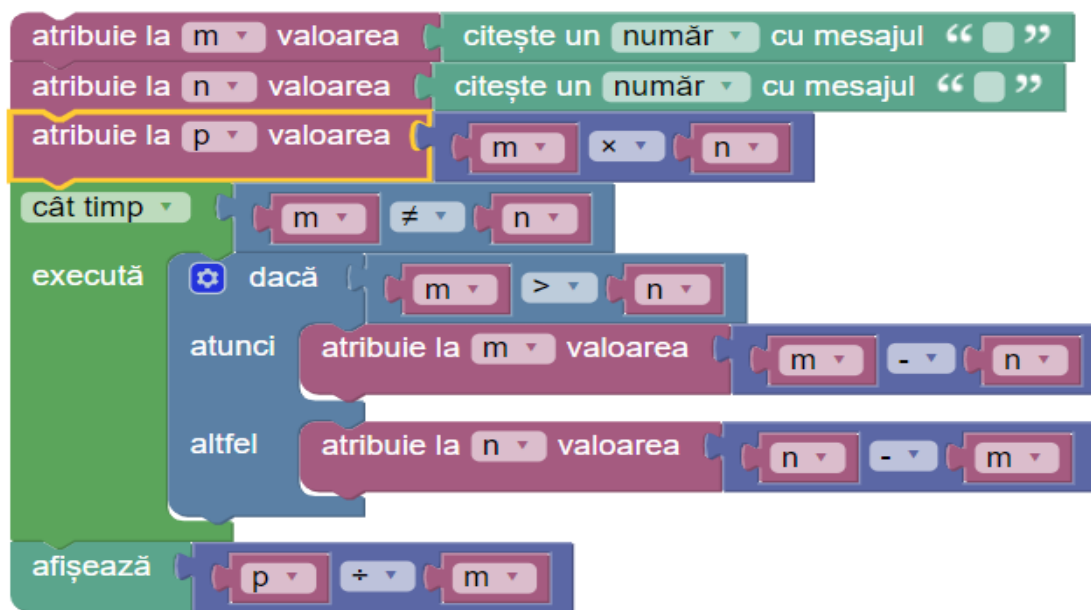
Observație: Pentru a determina cel mai mic multiplu comun a două numere naturale folosim următoarea teoremă: Produsul a două numere naturale este egal cu produsul dintre cel mai

mare divizor comun al lor și cel mai mic multiplu comun al lor: $n * m = \text{cmmdc} * \text{cmmmc}$.

Pentru a afla cel mai mare divizor comun a două numere, scădem din numărul mai mare numărul mai mic, până când cele două numere ajung egale. Această valoare este **cmmdc**.

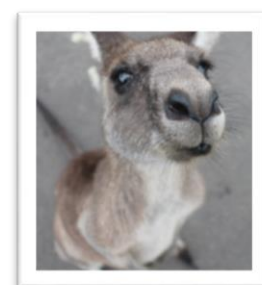


Observație: Deoarece valorile inițiale ale variabilelor n și m se pierd, salvăm valoarea produsului $m \cdot n$ în altă variabilă. Fie aceasta p . La final, afișez rezultatul împărțirii lui p la noua valoare a lui m , care este, de fapt, cel mai mic multiplu comun.



I.11 Cangurul (Blockly și Scratch)

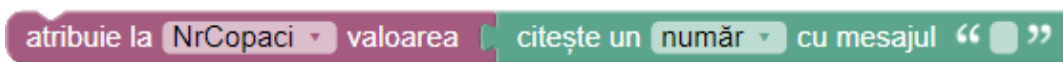
Enunț: Un cangur face salturi pe lângă grupuri de copaci, înainte sau înapoi, până întâlnește un număr de copaci impar, divizibil cu 3. Când ajunge lângă un grup cu număr par de copaci, sare înainte, de atâtea ori cât indică numărul, iar lângă un grup impar de copaci, sare înapoi, tot de atâtea ori cât indică numărul. Stabiliți câte salturi face cangurul înainte. Exemplu: 5, 4, 6, 1, 2, 9 – se va opri la 9, deoarece este impar și divizibil cu 3 și se va afișa 12. (Problemă din manualul digital de clasa a VI-a, Editura Didactică și Pedagogică SA, autori Melinda Emilia Coriteac și Diana Carmen Băican).



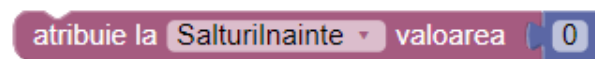
Observație: Deoarece în Blockly nu este predefinită structura repetitivă cu test final, voi rezolva această problemă în ambele limbaje: Blockly și Scratch.

Varianta Blockly

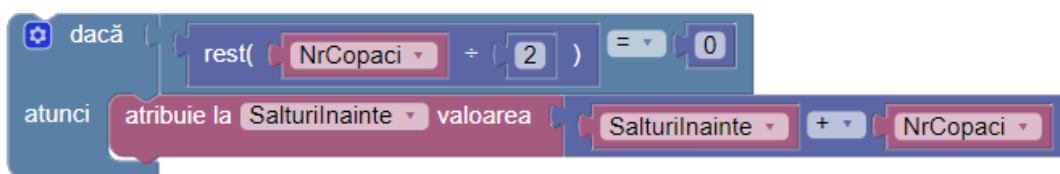
Pasul 1. Citesc prima valoare a numărului de copaci în afara structurii repetitive.



Pasul 2. Inițializez numărul de salturi înainte cu 0.



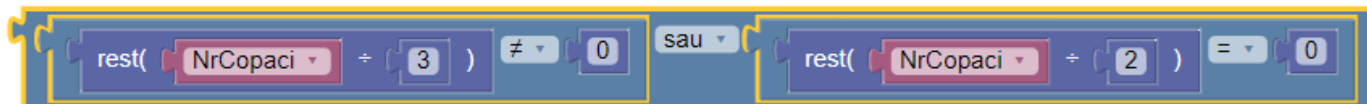
Pasul 3. Selectez structura alternativă: dacă numărul de copaci este divizibil cu 2 îl adaug la numărul de salturi înainte.



Pasul 4. Selectez structura repetitivă cu test inițial cât timp ... execută.

Pasul 5. Deoarece citirea numerelor de copaci trebuie să se oprească atunci când introduc, de la tastatură, un număr impar divizibil cu 3, condiția de continuitate este negarea acestei cerințe.

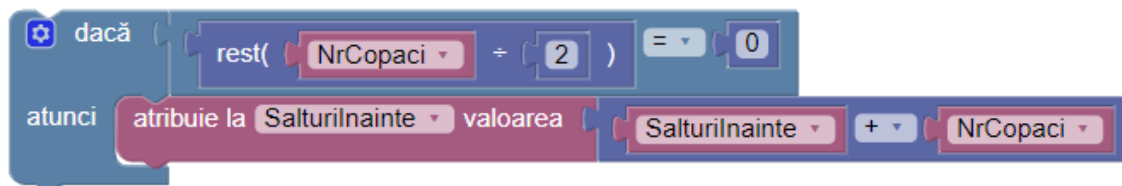
Adică, cât timp nu este îndeplinită condiția (rezultatul testului este fals), algoritmul îmi cere să introduc un nou număr (de copaci). Ciclarea se va opri atunci când condiția va deveni adevărată. Mai jos, am prezentat condiția care trebuie atașată structurii repetitive:



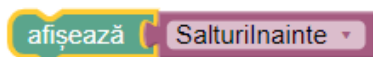
Pasul 6. Trebuie să continui citirea numărului de copaci. Deci, în interiorul blocului repetitiv, adaug o instrucțiune de citire:



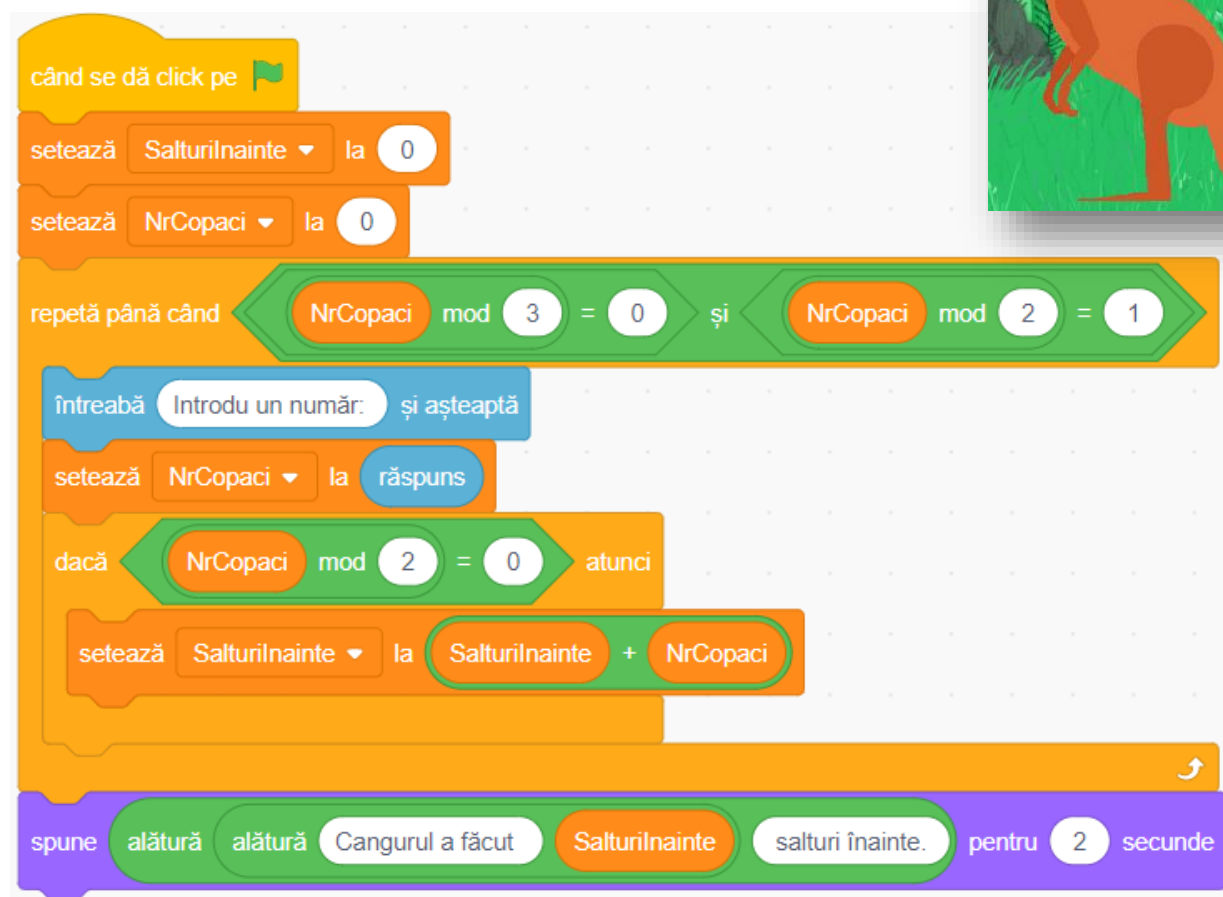
Pasul 7. În interiorul blocului repetitiv, adaug structura alternativă: dacă numărul de copaci este divizibil cu 2 îl adaug la numărul de salturi înainte.



Pasul 8. Afișez numărul total de salturi pe care le face cangurul înainte.



Varianta SCRATCH click



Observație: Deoarece am folosit o structură repetitivă cu test final: repetă până când, condiția este aceeași ca în enunț (numărul de copaci să fie divizibil cu 3 și să fie impar). **Este condiția de oprire!**

II. Structuri repetitive cu număr cunoscut de pași

II.1 Afișarea celei mai mari dintre n valori

Enunț: Să se scrie un program care citește de la tastatură n numere naturale și afișează pe cel mai mare dintre ele.

Rezolvare:

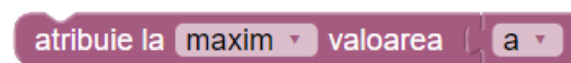
Pasul 1. Creez o variabilă **n**, care reprezintă câte numere urmează să citesc, accesând meniul **Variable** → Variabilă nouă....



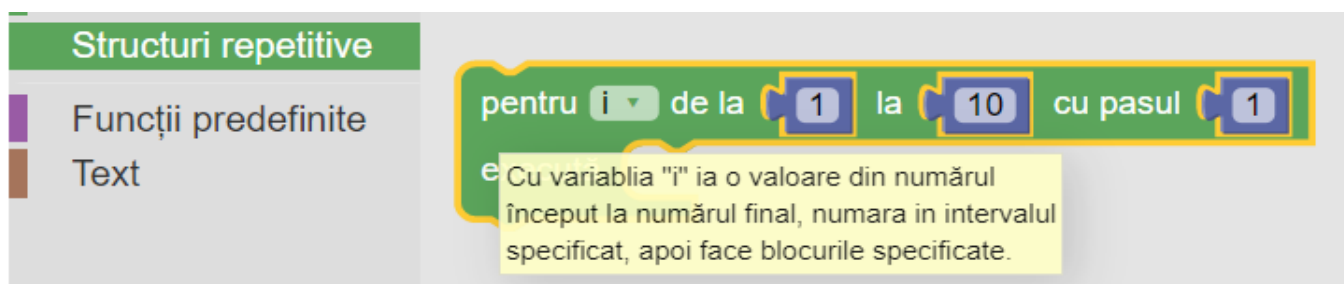
Pasul 2. Creez o variabilă **a** și o inițializez cu următoarea valoare citită.



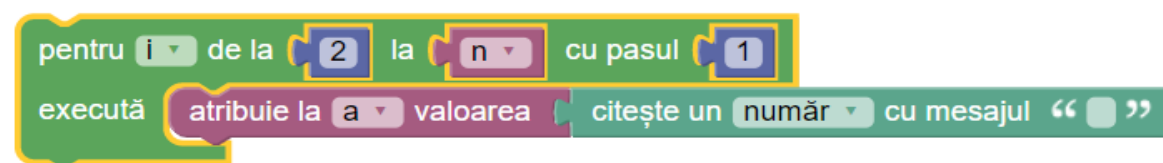
Pasul 3. Creez o variabilă **maxim**, în care salvez cea mai mare valoare citită. Îi atribui variabilei **maxim** valoarea variabilei **a** (presupun că e cea mai mare).



Pasul 4. Selectez, din meniul **Structuri repetitive** → pentru i de la 1 la 10 cu pasul 1 (structura repetitivă cu număr cunoscut de pași) și înlocuiesc 1 cu 2 și 10 cu n.



Pasul 5. În interiorul blocului repetitiv adaug instrucțiunea de citire a următoarelor valori din șir, tot cu ajutorul variabilei **a**.



Pasul 6. Tot în interiorul blocului repetitiv, adaug instrucțiunea alternativă, cu testul **a > maxim**. Dacă rezultatul este **adevărat**, atribui variabilei **maxim** noua valoare a lui **a**.

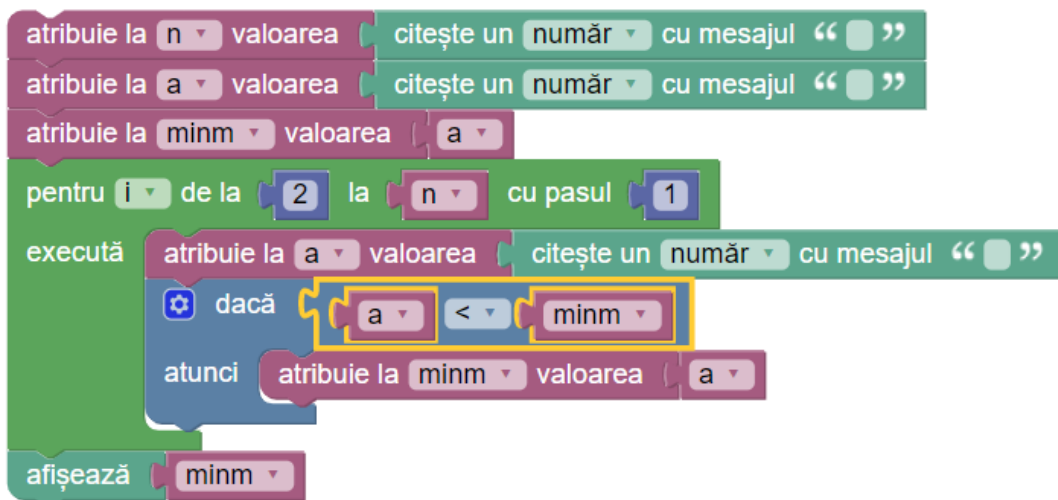


Pasul 7. Afișez valoarea variabilei **maxim**.



II.2 Afișarea celei mai mici dintre n valori

Pentru afișarea celei mai mici dintre un număr de n valori, utilizăm același algoritm de la punctul anterior, schimbând numere variabilei maxim cu **minim** și inversând sensul operatorului relațional din condiția structurii alternative, $a < \text{minim}$.

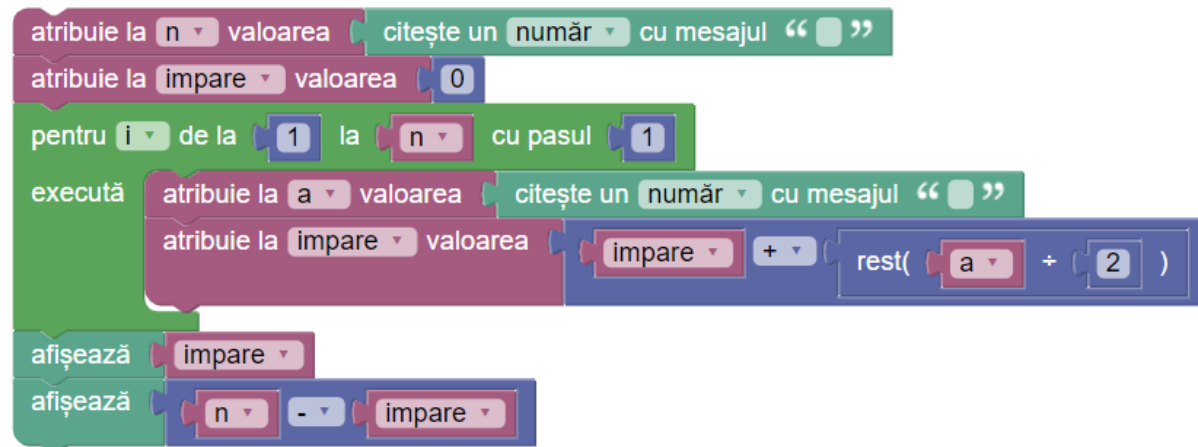


II.3 Câte impare/ pare

Enunț: Se citesc n numere de la tastatură. Să se afișeze câte dintre ele sunt impare și câte sunt pare.

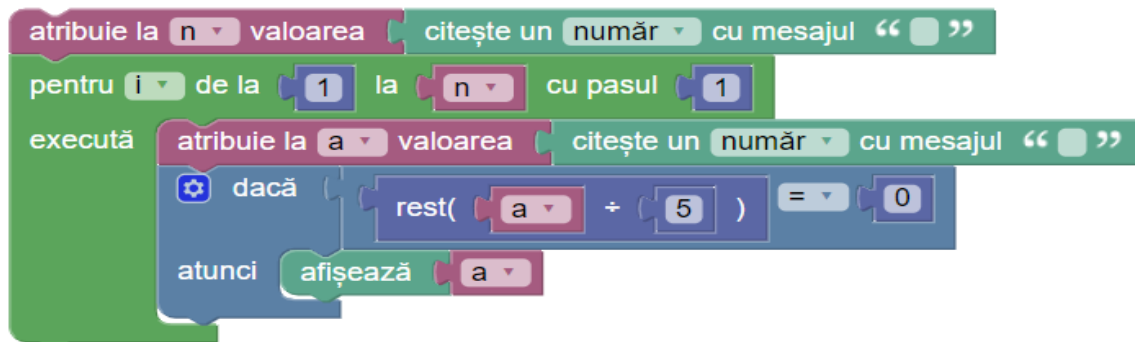
Soluție: Deoarece știm câte numere urmează să citim, la prima parcurgere vom afla câte numere impare sunt. Pe cele pare le vom calcula scăzând din n , numărul celor impare.

Observație: Deoarece numărul de cifre pare se obține printr-o însumare, variabila impare trebuie inițializată cu 0 (prin instrucțiunea de atribuire).



II.4 Afișare numere divizibile cu 5

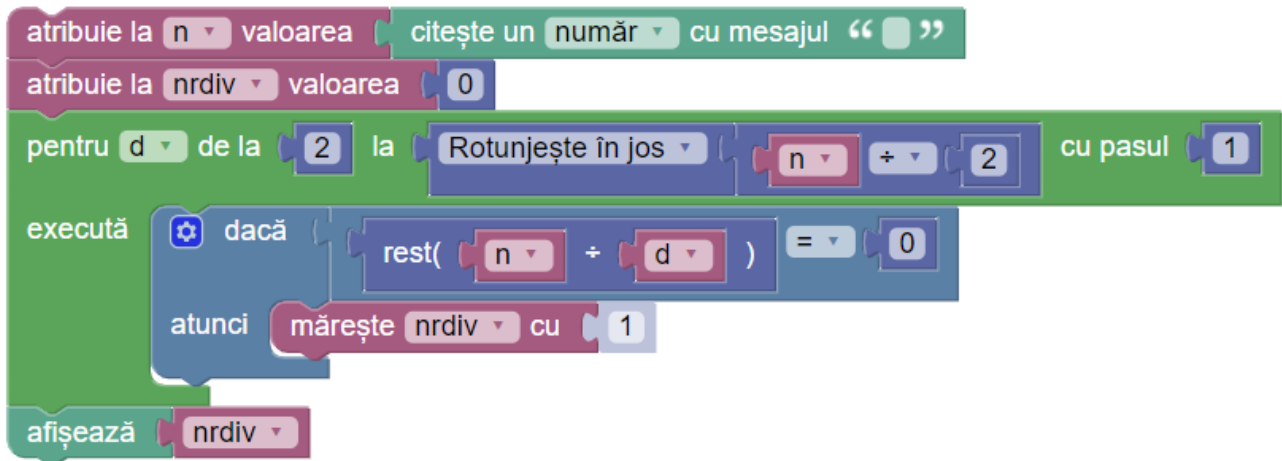
Enunț: Se citește, de la tastatură, un număr natural n și apoi n numere. Să se afișeze numai cele divizibile cu 5.



II.5 Divizori proprii

Marcel și Ioana vor să joace un joc. Fiecare numește un număr și celălalt trebuie să răspundă câți divizori proprii are numărul respectiv. Câștigă cel care a răspuns cel mai repede. De aceea, ei s-au gândit să scrie un program la calculator, pentru a rezolva mai repede cerința. Alegeți pe care dintre cei doi doriți să-l ajutați și scrieți algoritmul în Blockly.

Observații: Am redenumit contorul cu d , de la divizor. O metoda eficientă de a calcula numărul divizorilor proprii este de a parcurge divizorii numărului de la 2 până la jumătatea acestuia.



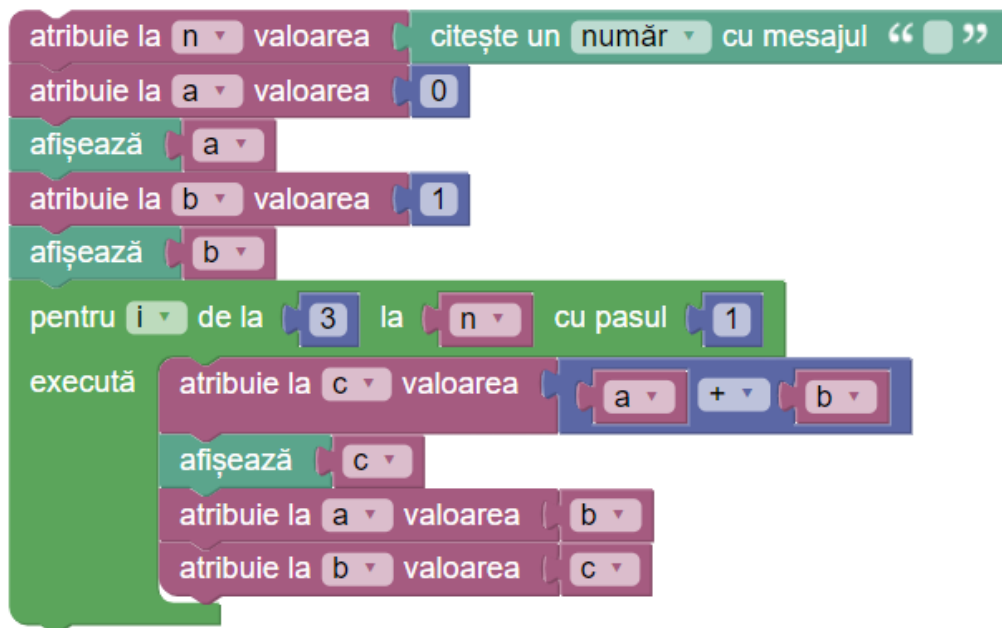
II.6 Divizori proprii eficient

Deoarece Ioana este înscrisă și la cercul de informatică, ea a găsit o metodă mai eficientă pentru calculul divizorilor unui număr, pe care dorește să o împărtășească cu voi. Deci, ea a verificat divizorii până la rădăcina pătrată a numărului, mai puțin cu o unitate și a incrementat variabila **nrdiv** cu 2, de fiecare dată când a găsit un divizor. La final, a verificat dacă numărul este pătrat perfect și dacă este adevărat, a mai adăugat 1 la numărul de divizori proprii. Astfel, ea a câștigat întrecerea.



II.7 Numerele lui Fibonacci

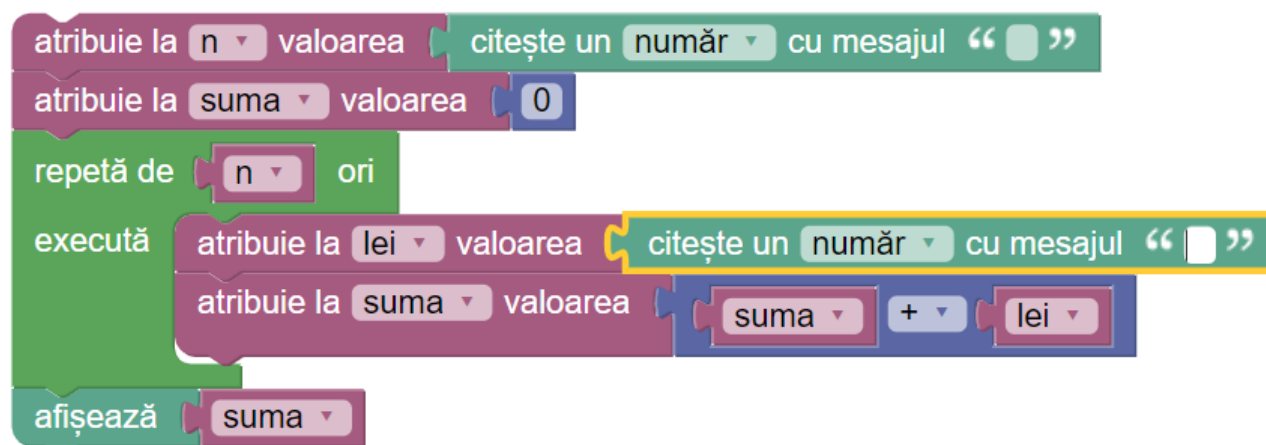
Numerele Fibonacci sunt definite prin următoarea relație: Astfel, fiecare număr Fibonacci este suma celor două numere Fibonacci anterioare. Primele 22 de numere din șir sunt: 0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181,6765,10946.



II.8 Dărnicie

Oana dorește să facă cadou câte o carte celor n prieteni ai ei. Dar ea ar vrea să știe dacă îi ajung banii. Astfel, știind cât costă o carte, ajutați-o pe Oana să construiască un algoritm, în Blockly, care să afișeze cât ar cheltui Oana pentru cele n cărți.

Observație: De această dată, am utilizat varianta simplificată a structurii repetitive cu număr cunoscut de pași – repetă de ... ori. Valoarea implicită a contorului este 1 și pasul este tot 1.



II.9 Moș Crăciun

Cei n copii de la grădiniță stau cuminiți în șir și așteaptă cadourile de la Moș Crăciun. Ei sunt numerotați de la 1 la n (se consideră că n este par) și în prima jumătate sunt copiii de la grupa mare iar în a doua jumătate sunt cei de la grupa mică (numărul de copii din fiecare grupă este același, $n/2$). După ce trece Moș Crăciun se știe câte cadouri a primit fiecare copil și noi dorim să aflăm câți copii din grupa mică au primit mai multe cadouri decât orice copil din grupa mare.

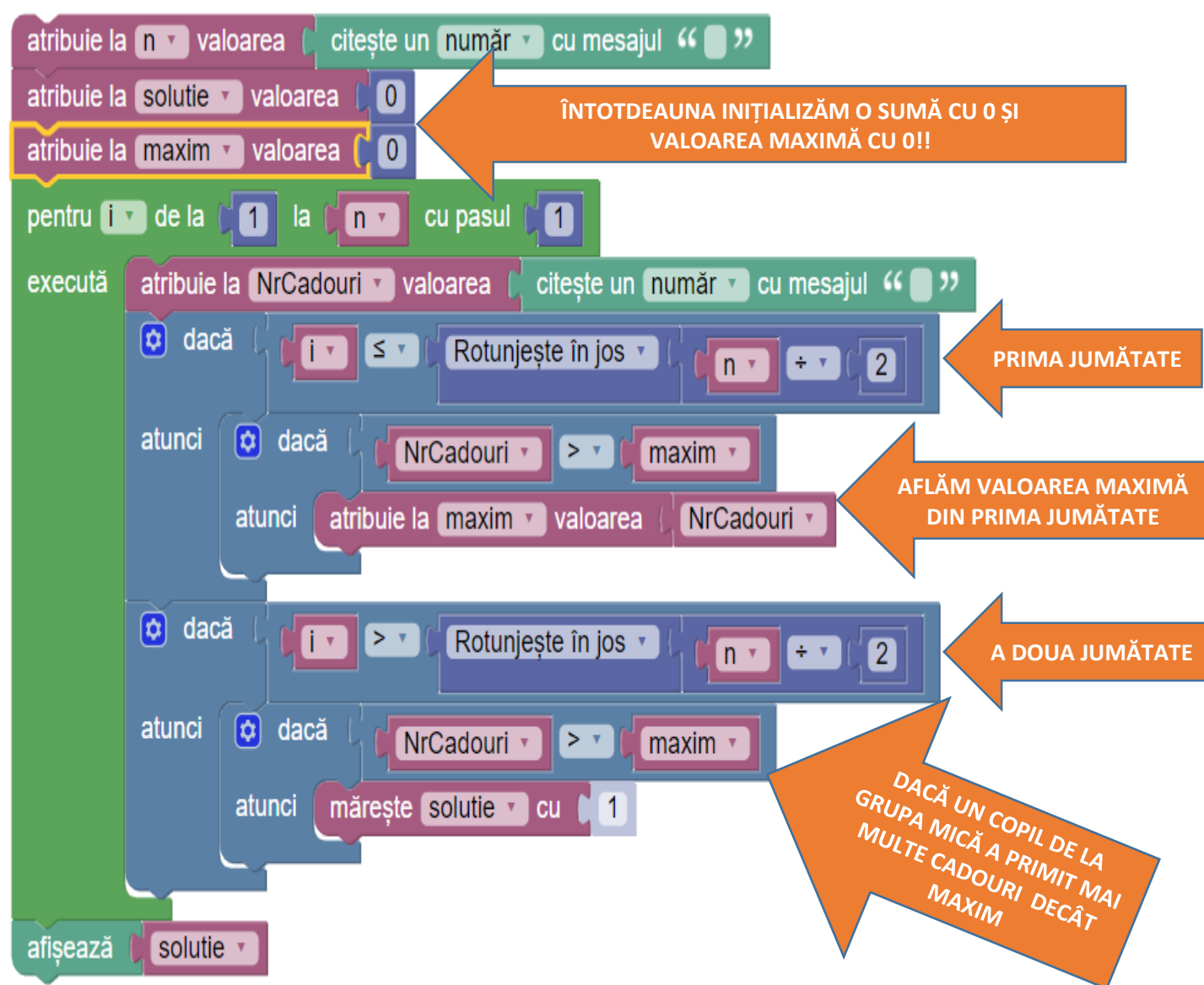


Exemplu: Pentru datele de intrare $n=6$ și 1 6 3 6 8 2, se va afișa 1.

Explicație: Copiii de la grupa mare au primit 1, 6 și respectiv 3 cadouri.

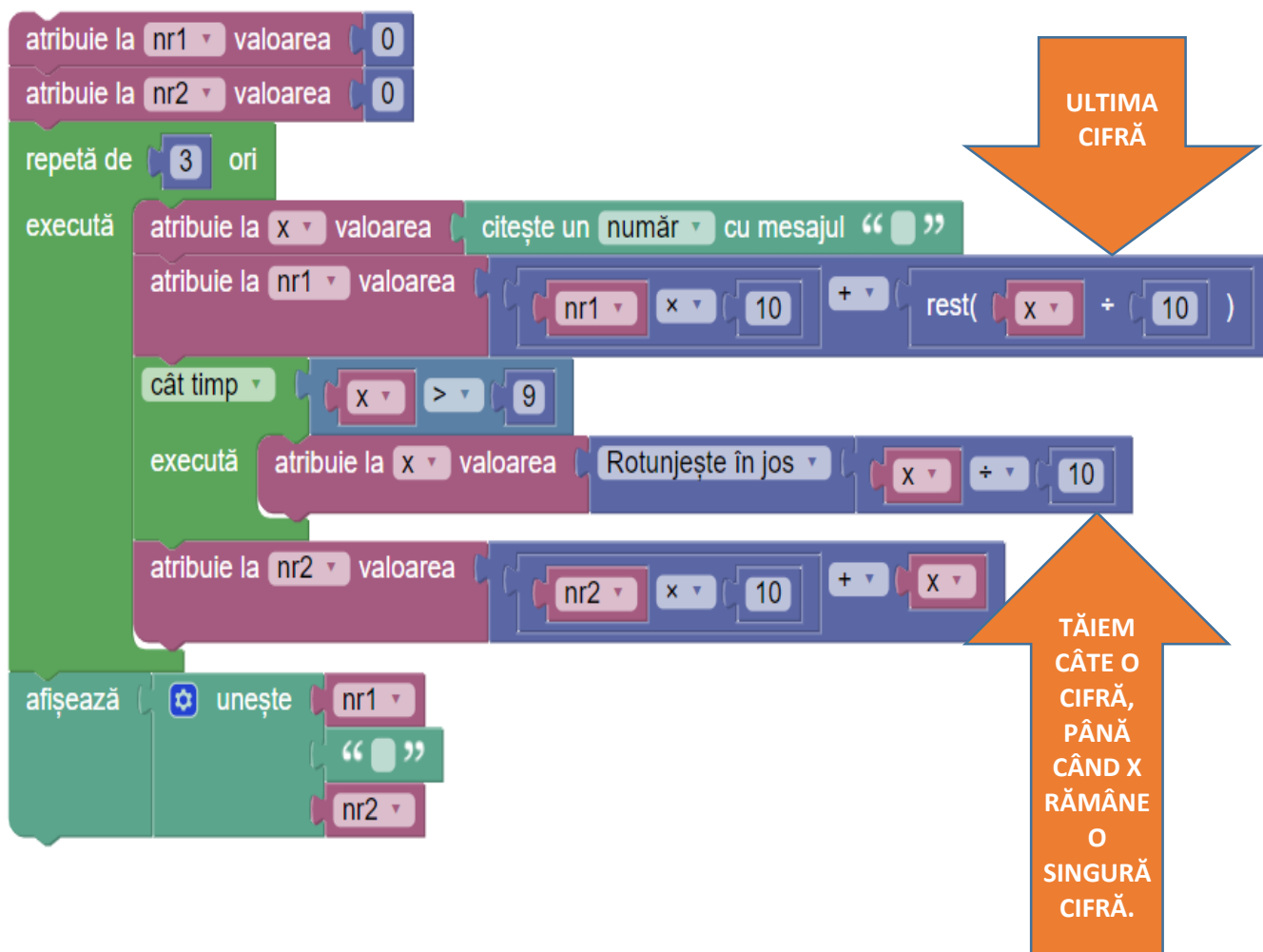
Din grupa mică un singur copil a primit mai multe cadouri decât toți cei de la grupa mare, și anume acela care a primit 8 cadouri.

Soluție: Este suficient să determinăm maximul din prima jumătate a șirului de numere și apoi să testăm fiecare număr din a doua jumătate dacă este mai mare decât această valoare și, dacă este mai mare, incrementăm (creștem) variabila **solutie** cu o unitate.



II.10 Joc

Ema se joacă cu niște cuburi inscripționate cu cifre. Ea are formate trei numere din mai multe cifre și dorește să formeze cu cifrele unităților un număr, iar cu prima cifră a fiecărui număr să formeze un alt număr. Numerele pe care le formează vor lua cifrele exact în ordinea cum apar în cele trei numere. Ajutați-o pe Ema să formeze cele două numere (Exemplu: Dacă cele trei numere inițiale sunt 1234, 745, 6409, numerele formate vor fi 459 și 176). (Problemă din manualul digital de clasa a VI-a, Editura Didactică și Pedagogică SA, autori Melinda Emilia Coriteac și Diana Carmen Băican).



Bibliografie

1. BĂICAN Diana Carmen, CORITEAC Melinda Emilia, Informatică și TIC, Manual pentru clasa a VI-a Editura Didactică și Pedagogică;

Bibliografie web:

www.pbinfo.ro

Programează cu Blockly | www.pbinfo.ro