

MIHAELA GIURGIULESCU
VALERIU BENEDICTH GIURGIULESCU

Informatică și TIC

Aplicații, indicații de rezolvare și teste
pentru clasa a VI-a

– nivel mediu, nivel avansat –



Cuprins

Argument	5
Testul 1	6
Testul 2	7
Partea I – Utilizarea calculatorului	
Unitatea 1 – Prezentări	
1.1. Test-fulger	10
1.2. Aplicații și indicații de rezolvare. Nivel mediu	11
1.3. Aplicații și indicații de rezolvare. Nivel avansat	13
Unitatea 2 – Animații grafice și modele 3D	
2.1. Test-fulger	17
2.2. Aplicații și indicații de rezolvare. Nivel mediu	18
2.3. Aplicații și indicații de rezolvare. Nivel avansat	19
Unitatea 3 – Internet	
3.1. Test-fulger	22
3.2. Aplicații și indicații de rezolvare. Nivel mediu	23
3.3. Aplicații și indicații de rezolvare. Nivel avansat	25
Autoevaluare	27
Partea a II-a – Algoritmi și programare	
Unitatea 1 – Scratch. Elemente recapitulative	
1.1. Test-fulger	30
1.2. Aplicații și indicații de rezolvare. Nivel mediu	31
1.3. Aplicații și indicații de rezolvare. Nivel avansat	36
Unitatea 2 – Algoritmi și modele de algoritmi	
2.1. Test-fulger	37
2.2. Aplicații și indicații de rezolvare. Nivel mediu	38
2.3. Aplicații și indicații de rezolvare. Nivel avansat	40
Unitatea 3 – Structura repetitivă condiționată anterior	
3.1. Test-fulger	41
3.2. Aplicații și indicații de rezolvare. Nivel mediu	42
3.3. Aplicații și indicații de rezolvare. Nivel avansat	44
Unitatea 4 – Structura repetitivă condiționată posterior	
4.1. Test-fulger	46
4.2. Aplicații și indicații de rezolvare. Nivel mediu	47
4.3. Aplicații și indicații de rezolvare. Nivel avansat	50
Unitatea 5 – Structura repetitivă cu număr cunoscut de pași	
5.1. Test-fulger	51
5.2. Aplicații și indicații de rezolvare. Nivel mediu	52
5.3. Aplicații și indicații de rezolvare. Nivel avansat	53
Unitatea 6 – Aplicabilitatea practică a algoritmilor	
6.1. Test-fulger	60
6.2. Controlul animației din taste	61
6.3. Note muzicale	64
6.4. Detecția video	66
6.5. Scratch și LEGO WeDo – telecomandă pentru un tanc/tractor cu două șenile	68
Autoevaluare	71

3.2. Aplicații și indicații de rezolvare. Nivel mediu

Important



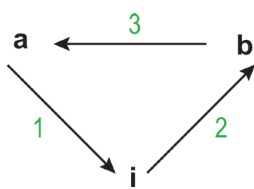
Pentru toate aplicațiile care urmează, consideră că valorile introduse ca date de intrare sunt corecte; nu este nevoie să verifici tipul lor (adică, de exemplu, să verifici dacă sunt numere naturale sau șiruri de caractere, după cum se cere în problemă).

- Se dau trei numere naturale nenule, a , b , c . Construiește un algoritm care le ordonează crescător.

Indicații de rezolvare:

Pentru a ordona (a sorta) crescător sau descrescător mai multe numere vom utiliza metoda Bulelor. Considerăm că cele trei numere a , b , c formează un șir. Metoda Bulelor constă în compararea fiecărui număr din șir cu succesorul său. În cazul nostru, vom compara primul număr cu al doilea (a cu b) și al doilea număr cu al treilea (b cu c). Dacă cele două numere comparate nu sunt în ordinea corectă, atunci ele își schimbă locul între ele (se interschimbă). În cazul nostru, trebuie să ordonăm numerele crescător, deci ar trebui ca primul număr să fie mai mic sau egal cu cel de-al doilea, cu care se compară. Dacă nu este așa, adică dacă primul număr este mai mare decât cel de al doilea, atunci ele își schimbă locul. Șirul de numere se parcurge până la sfârșit, de mai multe ori, până când nu mai este nevoie de nicio interschimbare.

Pentru a vedea dacă s-a mai realizat sau nu vreo interschimbare, definim o variabilă logică (se mai numește Boolean) „ESortat”. Aceasta primește valoarea Fals, dacă există interschimbări și Adevărat, dacă nu există interschimbare.



Interschimbarea valorilor a două variabile se realizează utilizând o a treia variabilă intermediară, „i”.

algoritm Ordonare

a , b , c , i numere naturale

ESortat boolean

Citește a , b , c

ESortat \leftarrow Fals

Repetă până când ESortat=Adevărat

Dacă $a > b$

Atunci

$i \leftarrow a$

$a \leftarrow b$

$b \leftarrow i$

ESortat \leftarrow Fals

Altfel

ESortat \leftarrow Adevărat

Dacă $b > c$

Atunci

$i \leftarrow b$

$b \leftarrow c$

$c \leftarrow i$

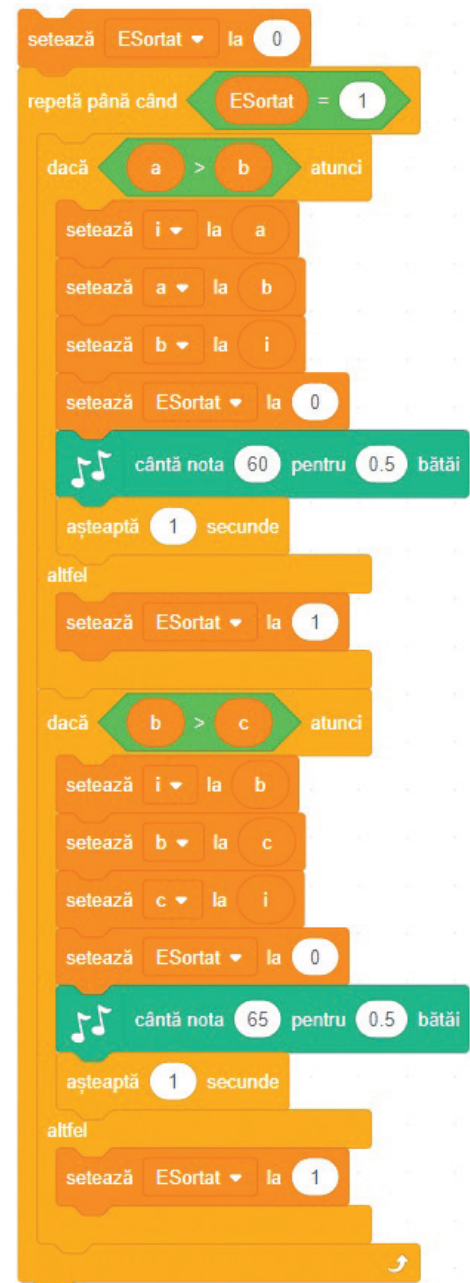
ESortat \leftarrow Fals

Altfel

ESortat \leftarrow Adevărat

Scrie 'Numerele ordonate sunt:', a , b , c

Sfârșit



Poți vedea programul Scratch *Sortare crescătoare 3 nr.sb3* la adresa:

<https://scratch.mit.edu/projects/238930671/#editor>.

În acest program, nu s-a mai realizat validarea variabilelor a , b , c pentru a fi numere naturale nenule. Adaugă tu această validare și salvează programul cu numele *Sortare crescătoare 3 nr_v1.sb3*.

- 2 Scrie algoritmul pentru sortarea (ordonarea) descrescătoare a unui șir de trei numere naturale nenule a , b , c .
- 3 Scrie un program Scratch care să schimbe culoarea unei steluțe timp de 10 secunde, din secundă în secundă.

Indicații de rezolvare:

Pentru a lua în considerare trecerea timpului într-un program Scratch, adăugăm o variabilă *Timp*, căreia îi dăm ca valoare numărul de secunde pe care îl impune problema (10 secunde). Valoarea acestei variabile se va micșora cu 1 la fiecare secundă care trece. Utilizăm structura secvențială condiționată anterior pentru a face acțiuni până când a trecut timpul (adică până când valoarea variabilei *Timp* este 0) și schimbăm culoarea la fiecare secundă, prin modificarea efectului *Culoare*.

Poți găsi programul *Stea lucitoare_10s.sb3* la adresa:

<https://scratch.mit.edu/projects/238930753/#editor>.

- 4 Scrie un algoritm care să o ajute pe mama la cumpărături. Ea introduce inițial în program suma de bani disponibilă, apoi, pe parcurs, cantitatea și prețul unitar, înainte de a mai cumpăra ceva. Programul trebuie să o atenționeze pe mama, dacă este cazul, asupra faptului că nu are suficienți bani pentru acel produs, cât și să îi spună câți bani mai are disponibili.



Indicații de rezolvare:

Date de intrare:

S – număr natural, reprezintă suma de bani pe care o are disponibilă mama;

c – număr natural, reprezintă cantitatea pe care dorește să o cumpere;

p – prețul unitar.

algoritm AjutorMama

S , c , p numere naturale

Citește S

Cât timp $S > 0$

Citește c , p

Dacă $S < p * c$

Atunci

Scrie 'Nu mai ai suficienți bani pentru a realiza cumpărătura aceasta!'

Scrie 'Ți-ar mai trebui', $p * c - S$, 'lei'

Scrie 'Mai ai disponibili doar', S

Altfel

$S \leftarrow S - p * c$

Scrie 'Mai ai', S , 'lei'

Sfârșit

Programul *MamaCumpărături.sb3* poate fi descărcat de la adresa:

<https://scratch.mit.edu/projects/238930503/#editor>.

6.5. Scratch și LEGO WeDo – telecomandă pentru un tanc/tractor cu două șenile



Facem un program Scratch prin care să obținem o telecomandă pentru un tractor/un tanc cu două șenile, făcut din piese Lego. Când se apasă Săgeată *în sus* (↑), tractorul/tancul merge în față, când se apasă Săgeata *în jos* (↓), vehiculul merge în spate, când se apasă Săgeată *la stânga* (←), vehiculul virează spre stânga, iar când se apasă Săgeată *la dreapta* (→), virează la dreapta.

Avem nevoie de:

- Un laptop (este bine să fie cât mai mic, pentru că va trebui să îl ții în brațe și să te plimbi cu el după tanc/tractor). Pe respectivul laptop va trebui să ai instalat programul *Scratch*.
- Un set LEGO WE DO 2.0 (sau cel puțin interfața de conectare din setul LEGO WE DO 2.0).
- Un tractor sau un tanc cu două șenile, făcut din LEGO, la care să cuplăm două motoare (câte unul pe fiecare șenilă), pentru a-l putea pilota.

Acum nu mai avem nevoie decât de un program în Scratch pentru a obține telecomanda.



Important

Pentru a putea comanda cele două motoare ale vehiculului, mergi la categoria **Adaugă o extensie** și adaugă o extensie. Vei alege, din lista care se deschide, „LEGO Education WeDo 2.0”.

Vor fi disponibile blocuri ce apar în imaginea din dreapta:

Pentru a conecta interfața WeDo la laptop trebuie să instalăm pe calculator Scratch Link, de la adresa:

<https://downloads.scratch.mit.edu/link/windows.zip>.

La adăugarea extensiei „Lego Education WeDo 2.0”, se va deschide o fereastră de conectare. Se apasă butonul „Începe căutarea” pentru a căuta interfața WeDo care comunică pe Bluetooth (vezi imaginea de mai jos).



Odată ce am conectat blocul cu interfața WeDo la laptop, apare un semn „bifat” verde în dreptul numelui extensiei, în cadrul panelului blocurilor Scratch pentru WeDo 2.0.

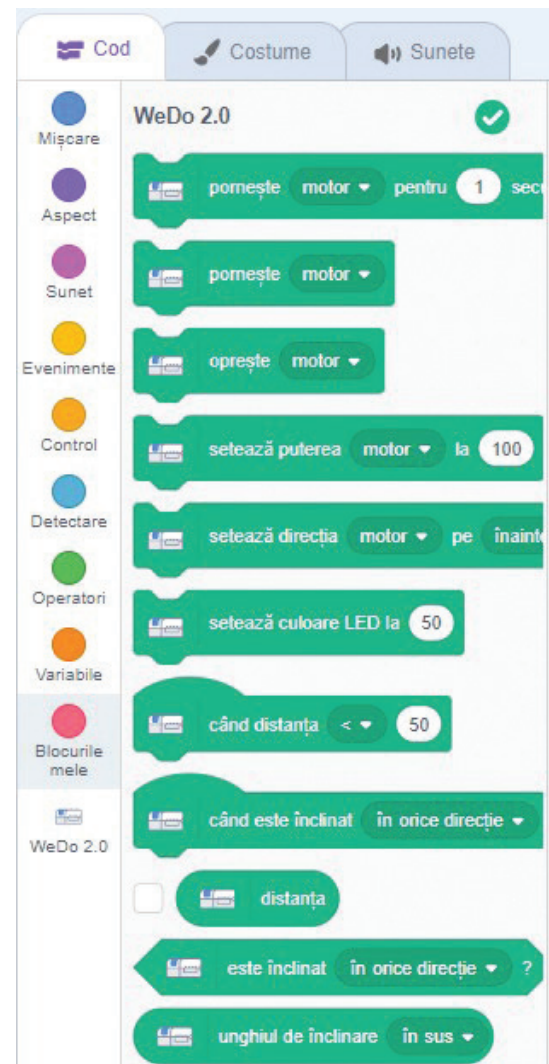
Să recapitulăm ce trebuie să facem înainte de a trece la realizarea programului:

- construim din piese LEGO un tractor sau un tanc;
- atașăm câte un motor electric la fiecare șenilă (motoarele se găsesc în seturile LEGO Technic, sau se cumpără separat);
- conectăm cele două motoare la interfața LEGO WeDo (trebuie să găsim și pentru această piesă LEGO un loc potrivit pe vehiculul nostru);
- conectăm la un laptop interfața LEGO WeDo 2.0.

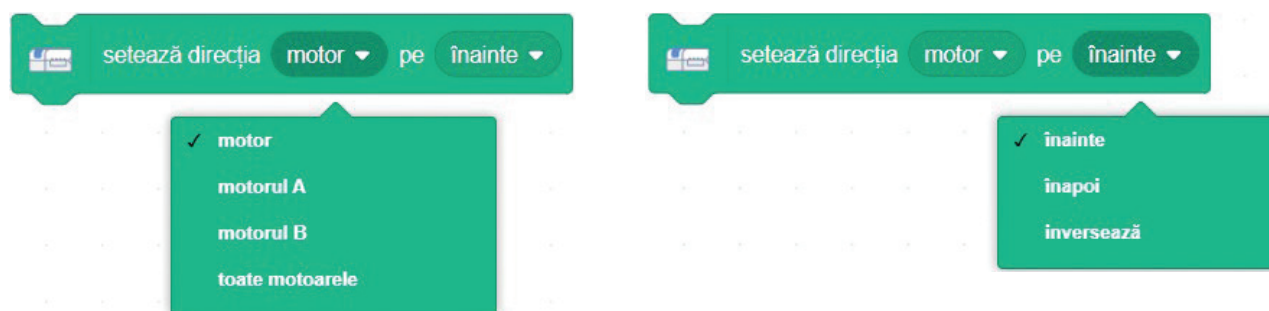
Dacă ai reușit să faci programul „telecomandă”, felicitări! Poți deja să te plimbi cu laptopul în brațe după vehiculul tău, pe care îl teleghidezi cu programul realizat în Scratch.

Dacă ai întâmpinat dificultăți în realizarea programului, îți propunem următoarea variantă de realizare.

- Poți începe prin a face câteva teste, pentru a vedea cum funcționează motoarele, interfața și blocurile din această categorie proaspăt descoperită.



- Încearcă pentru o tastă oarecare (să zicem **Săgeata în jos**): când tasta **Săgeata în jos** este apăsată, pornește motorul A. Astfel vei identifica motorul A și, respectiv, motorul B. A mers? Perfect. Oprește programul și hai să vedem cum putem schimba/controla sensul în care se rotește motorul.
- Pentru aceasta vom folosi blocul următor:



După cum observi, putem să alegem între motorul A și motorul B, dar și sensul în care să meargă vehiculul (înainte/înapoi, însemnând în sens orar/în sens opus acelor de ceas). Putem să inversăm sensul, după cum dorim.

La aceasta se rezumă de fapt tot programul nostru: pentru fiecare dintre cele patru taste cu săgeți, atunci când este apăsată, să alegem un sens de mișcare pentru fiecare motor și să pornim motoarele.

Important



Trebuie să faceți mai multe încercări, pentru că deplasarea depinde de modalitatea de montare a motoarelor pe șenile. Probabil că le-ați montat simetric, așa că, pentru ca vehiculul să meargă în față, va trebui ca cele două motoare să meargă în sensuri opuse, pentru că acestea sunt montate „în oglindă”. Încercați practic și veți înțelege mai bine.

Să ne gândim cum ia tancul o curbă: pentru a vira la stânga, oprește funcționarea șenilei de pe partea stângă și merge înainte doar șenila de pe partea dreaptă. Similar, pentru a vira la dreapta, se oprește șenila dreaptă și merge (spre înainte) doar șenila din stânga.

Ținând cont de toate acestea, programul nostru ar trebui să arate astfel:

