

Nușa Dumitriu-Lupan, Carmen Mincă, Daniela Bejan, Alina-Gabriela Boca,
Maria Niță, Adrian Niță, Diana Ghinea, Corina Elena Vinț

INFORMATICĂ ȘI TIC

- Programare în limbajul **Python**
- BBC Micro Bit **Micro:bit**



PREFAȚĂ

*O persoană cu adevărat educată este una care
a învățat cum să învețe și să se schimbe.*

CARL ROGERS

Omul modern trăiește într-un mediu aflat într-o schimbare continuă de mare viteză. Toți elevii trebuie să se pregătească pentru o lume computerizată, o lume a tehnologiei. Aceștia trebuie motivați să își dezvolte creativitatea, capacitatea de comunicare, relaționarea și spiritul de echipă și trebuie să perceapă tehnologia ca fiind parte din viața lor.

Agenda Digitală UE subliniază nevoia ca statele membre să îmbunătățească oferta educațională în domeniul IT în vederea dezvoltării abilităților digitale ale viitorilor absolvenți, cu un impact pozitiv asupra angajabilității acestora.

Programa școlară pentru disciplina INFORMATICĂ ȘI TIC încurajează elevii de gimnaziu să folosească tehnologia într-un mod responsabil și creativ și răspunde nevoilor de continuare a eforturilor de alfabetizare digitală și de reconsiderare a acestui concept din perspectiva noilor cerințe socio-profesionale. Mai mult decât atât, aceștia deprind cu spirit critic modalități de rezolvare a unor probleme elementare prin simularea unor algoritmi de prelucrare a informației și elaborează creativ produse informatice care să valorifice conexiunile dintre disciplina INFORMATICĂ ȘI TIC și societate.

Începând din clasa a V-a, traseul de pregătire a elevilor la disciplina INFORMATICĂ ȘI TIC conduce progresiv, printr-o abordare constructivistă, modernă, la dezvoltarea unor competențe digitale, utile atât în eficientizarea activității curente a elevului, cât și în tranziția către învățământul liceal.

În acest context, această carte se dorește a fi un material accesibil și consistent recomandat profesorilor care predau disciplina INFORMATICĂ ȘI TIC, elevilor din gimnaziu și tuturor celor care doresc să înțeleagă toate "de ce?"-urile de inițiere în lucrul creativ cu această unealtă minunată, calculatorul.

Cu inspirație și imaginație, autorii au elaborat o lucrare adaptată vârstei și aspirațiilor elevilor de clasa a VII-a, un adevărat ghid în dirijarea învățării care să conducă, pe termen mediu și lung, la reușita corelației între abilitățile pe care elevii le dobândesc pe parcursul studiilor cu abilitățile digitale solicitate pe piața muncii.

Urăm cititorilor succes în studiul acestei lucrări!

prof. NUȘA DUMITRIU-LUPAN



Cu Editura Corint
viitorul este mai aproape de tine!

În acest auxiliar ai posibilitatea de a accesa diverse link-uri cu ajutorul codurilor QR. Nu ai nevoie decât de un dispozitiv mobil dotat cu cameră video și acces la internet, de exemplu un smartphone, și de instalarea unei aplicații pentru scanarea codurilor QR. Ai astfel la dispoziție o metodă plăcută și captivantă de transmitere a informației, care te va ajuta să înțelegi și să înveți mai ușor.

Mult succes!

32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```
self.file = None
self.fingerprints = set()
self.logdupes = True
self.debug = debug
self.logger = logging.getLogger(__name__)
if path:
    self.file = open(os.path.join(path, 'request_fingerprints.txt'), 'w')
    self.file.seek(0)
    self.fingerprints.update(self.request_fingerprint(request))

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DEBUG', False)
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```



LIMBAJUL PYTHON

LIMBAJUL PYTHON

Python este un limbaj interpretat, de scripting. El este ușor de utilizat și este independent de sistemul de operare utilizat. Aplicațiile Python sunt foarte ușor de depanat, codul putând fi cu ușurință inspectat în timpul rulării. Extensia programelor scrise în Python este **.py**.

Știați că...



Limbajul Python a fost dezvoltat în 1991 de către programatorul olandez Guido van Rossum. Versiunea Python 2, lansată în 2000, conține câteva elemente noi și permite dezvoltarea limbajului de o întreagă comunitate de programatori. Python 3 a fost lansat în 2008 și nu este compatibil cu Python 2. "Sfârșitul" lui Python 2 este prevăzut pentru anul 2020.

Fiind un limbaj dinamic, Python permite scrierea unei mari varietăți de programe, cum ar fi: programe specializate pentru procesare de texte, motoare de căutare, aplicații *web*. Python este un limbaj actual folosit de marile companii din domeniul IT (Google, Yahoo!, Amazon etc).

MEDIUL LIMBAJULUI DE PROGRAMARE PYTHON

A. INSTALARE PYTHON

Pentru început, vom instala pe calculator limbajul Python, de la adresa: www.python.org. De asemenea vom instala **Wing IDE** (ediția gratuită **Wing Personal**) de la wingware.com

Știați că...



Un mediu integrat de dezvoltare – **IDE**, este o aplicație software care oferă programatorilor mai multe facilități pentru dezvoltarea de software: scrierea, modificarea, compilarea, depanarea programului.

B. MODALITĂȚI DE LUCRU

B.1. Lucrul în IDLE Python.

După instalare, Python se poate lansa scriind **IDLE** la opțiunea de căutare a interfeței sistemului de operare.

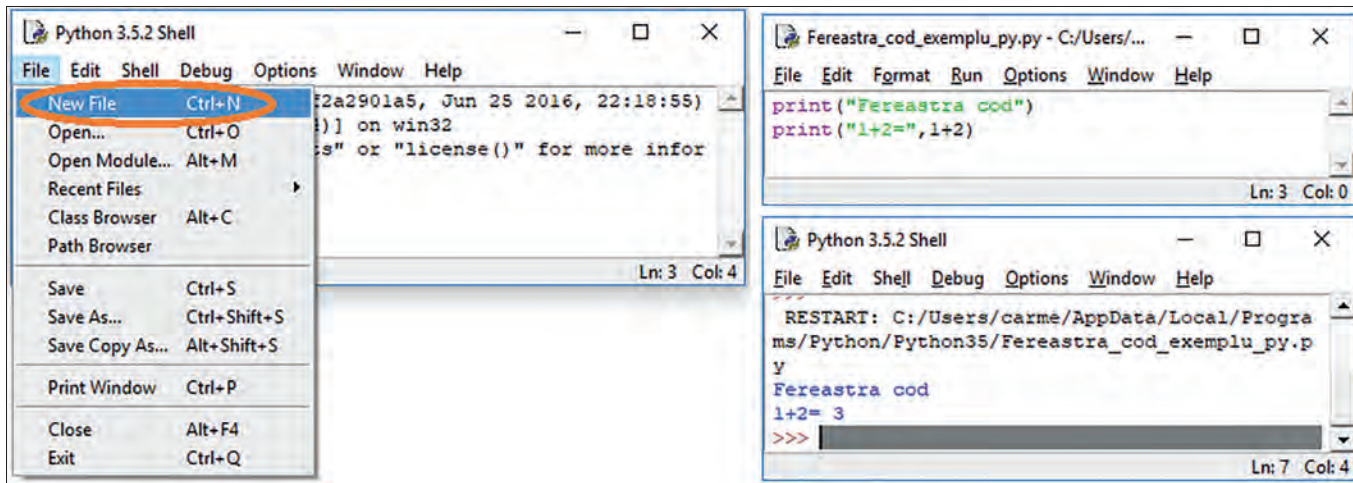
După lansare, se deschide *fereastra de comenzi – Shell* (care interpretează comenzile succesiv) din imaginea alăturată. Fiecare comandă scrisă în *Shell*, este verificată imediat și, dacă nu sunt erori, executată. Astfel, programul este verificat și executat comandă cu comandă.

Interfața ferestrei *Shell* permite, prin intermediul meniului, salvarea, editarea unui program, precum și setări ale interfeței.

IDLE (mediu integrat de dezvoltare și învățare) este un mediu pentru Python, instalat împreună cu limbajul.

Pentru editarea integrală a unui program, Python permite deschiderea unei ferestre, denumită *ferastră Code*. Selectând opțiunea **New File** din meniul **File** al ferestrei *Shell*, se deschide fereastra *Code*, în care se poate edita întregul program. Interpretarea programului scris se va face la final, iar rezultatul executării programului va fi afișat în fereastra *Shell*. Interfața ferestrei *Code* permite ca programul scris să poată fi:

- salvat – opțiunea **Save** din meniul **File** (Ctrl+S);
- verificat – opțiunea **Check Module** din meniul **Run** (Alt+X);
- lansat în execuție, prin selectarea opțiunii **Run Module** din meniul **Run** (F5).



Exersași

1.1. Tastați în fereastra *Shell*, după cursorul reprezentat prin ">>>", o operație aritmetică oarecare și apoi apăsați tasta **Enter**. Veți obține imediat rezultatul operației, fără a utiliza opțiuni din meniul interfeței.

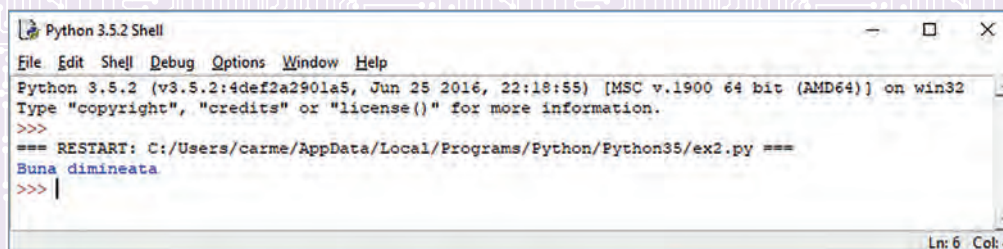
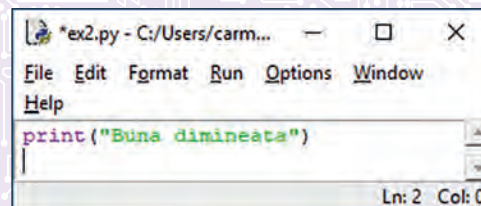
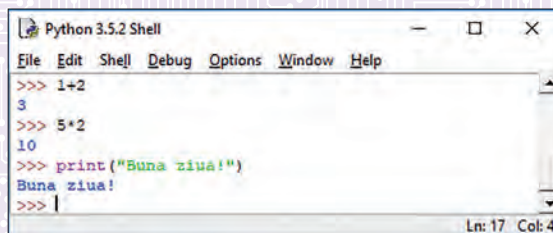
1.2. Pentru a afișa un text oarecare, de exemplu "Buna ziua!", scrieți respectivul text între ghilimele sau apostrofuri, între cele două paranteze ale funcției **print**. Apăsați tasta **Enter**.

2.1. Scrieți textul de la exercițiul anterior în *fereastra Code*. Nu uitați să încadrați textul între ghilimele sau apostrofuri, între cele două paranteze ale funcției **print**.

2.2. Selectați opțiunea **Save** din meniul **File** al *ferestrei Code*. Introduceți numele fișierului, *Ex2.py*, apoi apăsați butonul **Save**.

2.3. Verificați existența erorilor, selectând opțiunea **Check Module** din meniul **Run**. Dacă sunt semnalate erori, corectați-le.

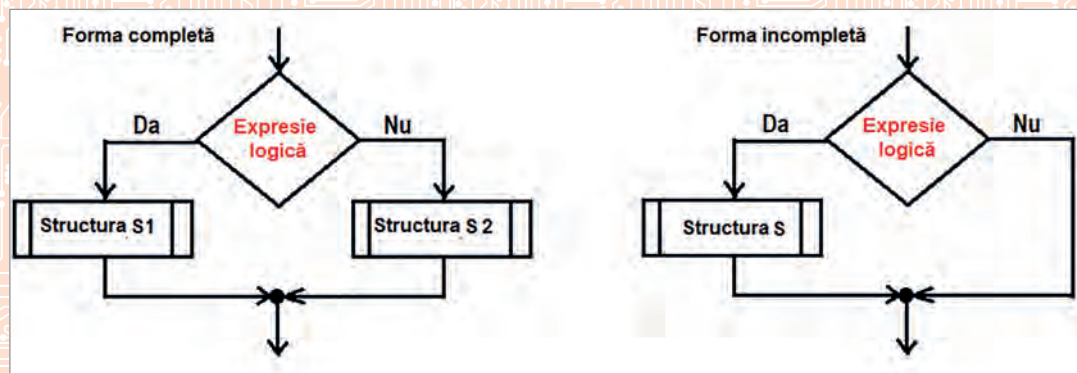
2.4. Lansați în execuție programul, selectând opțiunea **Run Module** din meniul **Run**. În *fereastra Shell* vor fi afișate rezultatele executării programului.



B. IMPLEMENTAREA STRUCTURII ALTERNATIVE - INSTRUCȚIUNEA IF

! **Să ne reamintim**

Structura selectivă se reprezintă cu ajutorul schemelor logice astfel:



Principiul de executare a acestei structuri este următorul:

Forma completă	Forma incompletă
<p>Pas 1: Se evaluează expresia logică.</p> <p>Pas 2: Dacă rezultatul expresiei logice este ADEVĂRAT (True) atunci se execută Structura S1; altfel, dacă rezultatul este FALS (False) se execută Structura S2.</p>	<p>Pas 1: Se evaluează expresia logică.</p> <p>Pas 2: Dacă rezultatul expresiei logice este ADEVĂRAT (True) atunci se execută Structura S; altfel, algoritmul continuă cu executarea structurii ce succede structura alternativă curentă.</p>

În limbajul Python, sintaxa instrucțiunii care implementează structura alternativă este:

Forma completă	Forma incompletă
<pre>if expresie_logică: instrucțiune_1 else: instrucțiune_2</pre>	<pre>if expresie_logică: instrucțiune_1</pre>

Efectul executării instrucțiunii:

- Pas 1: Se evaluează **expresie**, care are rezultat boolean. Dacă **expresie=true**, atunci se execută **instrucțiune_1**. Altfel se merge la Pas 2.
- Pas 2: Întrucât **expresie=false**, în cazul formei complete se execută **instrucțiune_2**. Apoi se trece la executarea următoarei instrucțiuni din program.

Observație. **instrucțiune_1** și **instrucțiune_2**, prezente în formele instrucțiunii **if**, desemnează orice instrucțiune/instrucțiuni ale limbajului Python.

Exersați

1. Transcrieți în limbajul Python algoritmul de determinare a maximumului dintre două numere întregi (memorate în variabilele **a** și **b**) reprezentat cu ajutorul schemei logice următoare. Observați componența structurii liniare: operația de citire a datelor de intrare (preluate în variabilele **a** și **b**), operația de decizie (reprezentând prelucrarea efectuată de algoritmul, adică obținerea maximumului dintre **a** și **b**) și operația de afișare a datelor de ieșire (afișarea maximumului).

Schema logică	Program Python
<pre> graph TD Start([START]) --> Read[/CITESTE a, b/] Read --> Decision{a > b} Decision -- Da --> AssignA[max ← a] Decision -- Nu --> AssignB[max ← b] AssignA --> Write[/SCRIE max/] AssignB --> Write Write --> Stop([STOP]) </pre>	<pre> a=int(input("primul numar=")) b=int(input("al doilea numar=")) if a>b: max=a else: max=b print("max=",max) </pre> <p>Afișare rezultate</p> <pre> primul numar=25 al doilea numar=109 max=109 </pre>

Atenție! Limbajul Python permite scrierea unei instrucțiuni de decizie "multiplă", asemănătoare instrucțiunii **switch** din limbajul C/C++, și este implementată ca în imaginea alăturată.

Executarea unei astfel de instrucțiuni se realizează astfel:

- Pas 1: Se evaluează **expresie_1**, care are rezultat boolean. Dacă **expresie_1=true**, atunci se execută **instrucțiune_1**. Altfel, mergi la Pas 2.
- Pas 2: Dacă **expresie_2=true**, atunci se execută **instrucțiune_2**. Altfel, mergi la Pas 3.

```

if expresie_1:
    instrucțiune_1
elif expresie_2:
    instrucțiune_2
.....
elif expresie_n:
    instrucțiune_n
else:
    instrucțiune_else
        
```

- Pas n: Dacă **expresie_n=true**, atunci execută **instrucțiune_n**. Altfel, mergi la Pas n+1
- Pas n+1: Se ajunge la acest pas dacă niciuna dintre logice **expresie_1, expresie_2, ..., expresie_n** nu au rezultatul **true**. În acest caz se execută **instrucțiune_else**.

Observații: - pot exista oricâte linii **elif**;

- **else** nu este obligatoriu să fie prezent într-o construcție ca cea de mai sus (este opțională).

2. Transcrieți în limbajul Python algoritmul care stabilește dacă un număr real este strict pozitiv sau strict negativ sau zero, reprezentat cu ajutorul schemei logice următoare. Observați componența structurii liniare: operația de citire a datelor de intrare (preluată în variabila **x**), o structură de decizie (cu o operație de afișare pe una dintre ramuri și o operație de decizie pe cealaltă ramură).

PROBLEME REZOLVATE

A. Structuri liniare si alternative

1. O eprubetă plină cu apă cântărește **a** grame. În eprubetă se scufundă o monedă de **b** grame. În noua stare, eprubeta cântărește **c** grame. Scrieți un program care să citească numerele **a**, **b**, **c** și care să determine densitatea metalului din care e confecționată moneda.

Exemplu: pentru **a=50**, **b=12** și **c=60.5** se va afișa: 8 (g/cm³)

Rezolvare: O soluție poate fi următoarea:



Limbajul C++	Limbajul Python
<pre>#include <iostream> using namespace std; int main() { double a, c, b, densitate_moneda; cin >> a >> b >> c; densitate_moneda = b / (a + b - c); cout << densitate_moneda; return 0; }</pre>	<pre># se citesc cele trei valori # de la tastatura, de pe aceeasi linie a, b, c = map(float, input().split()) densitate_moneda = b / (a + b - c) print(densitate_moneda)</pre>

2. Scrieți un program care citește un număr natural **n** și verifică dacă este pătrat perfect. Programul va afișa mesajul **DA**, dacă numărul **n** este pătrat perfect, respectiv **NU**, în caz contrar.

Exemplu: pentru **n=25** va afișa **DA**, iar pentru **n=8** va afișa **NU**.

Rezolvare: O soluție poate fi următoarea:



Limbajul C++	Limbajul Python
<pre>#include <iostream> #include <cmath> using namespace std; int main() { int n, radical; cin >> n; radical = sqrt(n); if (radical * radical == n) cout << "DA"; else cout << "NU"; return 0; }</pre>	<pre>import math n = int(input()) radical = int(math.sqrt(n)) if radical * radical == n : print("DA") else: print("NU")</pre>

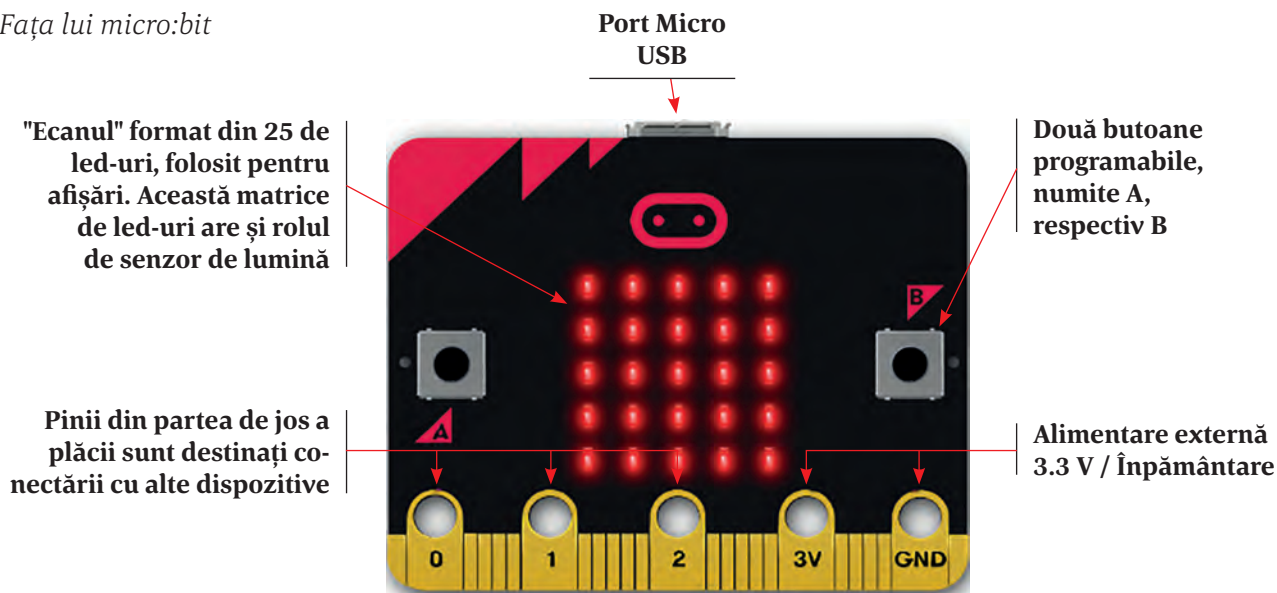


BBC MICRO BIT (MICRO:BIT)

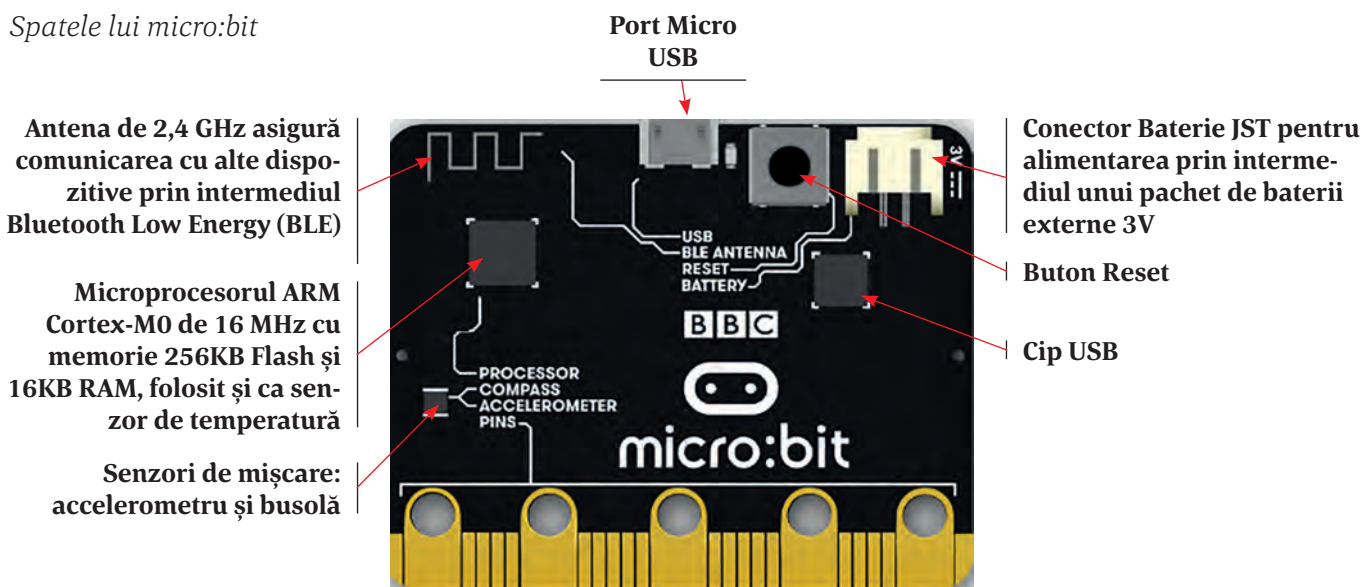
BBC MICRO BIT (MICRO:BIT)

BBC Micro Bit (micro:bit) este un minicalculator (5 cm x 4 cm) special conceput pentru educație. El este asemănător, ca dimensiune, cu un card bancar.

Fața lui micro:bit



Spatele lui micro:bit

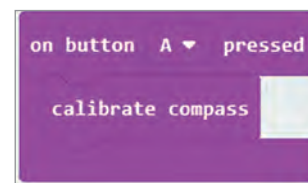


Datele tehnice: <https://tech.microbit.org/hardware/>

Microsoft MakeCode pentru micro:bit este o platformă gratuită care oferă oricui posibilitatea de a programa acest calculator folosind codificarea bazată pe blocuri (asemănătoare cu tehnica de la SCRATCH) sau prin JavaScript. Programarea se realizează în întregime în browserul web, nefiind necesară instalarea niciunui soft suplimentar. Adresa platformei este: <https://makecode.microbit.org/#editor>.

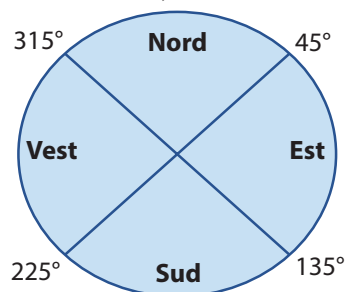
BUSOLA (COMPASS)

micro:bit posedă un cip specializat, numit **magnetometru**, cu ajutorul căruia se poate determina orientarea minicalculatorului față de punctele cardinale. Orientarea se face în grade, de la **0** la **359**. Pentru folosirea busolei, aceasta trebuie, inițial, calibrată. De exemplu, dacă atribuim această acțiune butonului **A**:



Să considerăm cadranul busolei împărțit, schematic, în patru zone, corespunzătoare principalelor puncte cardinale. Facem următoarea convenție:

Punctul cardinal	Interval
Nord	315° - 360° și 0° - 45°
Est	45° - 135°
Sud	135° - 225°
Vest	225° - 315°



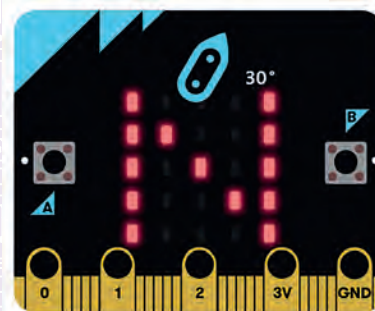
Exersați !

1. În funcție de datele de mai sus, transformați minicalculatorul **micro:bit** într-o busolă digitală. În funcție de direcția în care este orientat minicalculatorul, pe ecranul cu led-uri să se afișeze inițiala punctului cardinal.

Indicație: variabilei **unghi**, definită de utilizator, i se atribuie valoarea, exprimată în grade, dată de orientarea minicalculatorului, folosind **compass heading (°)**

```

forever
  set unghi to compass heading (°)
  if unghi > 315 or unghi < 45 then
    show string "N"
  else if unghi > 45 and unghi < 135 then
    show string "E"
  else if unghi > 135 and unghi < 225 then
    show string "S"
  else if unghi > 225 and unghi < 315 then
    show string "V"
  
```



Simularea rotirii minicalculatorului se realizează prin rotirea (cu ajutorul mouse-ului) logo-ului lui **micro:bit**.

Se observă că se afișează numărul de grade corespunzătoare rotirii într-o direcție sau alta.

2. Refaceți programul folosind doar trei condiții pentru determinarea celor patru puncte cardinale.

CUPRINS

Prefață.....	3
--------------	---

Limbajul Python

Limbajul Python.....	6
Mediul limbajului de programare Python	6
Elemente de bază ale limbajului Python	12
Structura unui program în limbajul Python	14
Date în limbajul Python.....	14
A. Tipuri de date standard.....	14
B. Constante și variabile	15
C. Expresii	17
Operații de citire și afișare/scriere a datelor în Python	24
Instrucțiuni/comenzi pentru implementarea structurilor de control în limbajul Python	28
Modulul MATH	38
Șiruri de caractere în Python – String-uri	38
Liste în Python.....	43
Jocuri în Python.....	47
Ghicește numărul (Guess number).....	47
Urmărește (Follow me)	48
Probleme rezolvate.....	51

BBC Micro Bit (micro:bit)

BBC Micro Bit (micro:bit).....	70
Interfața	71
Afișarea unui număr.....	71
Afișarea șirurilor de caractere	72
Afișarea unor simboluri predefinite (icon)	73
Afișarea unor caractere definite de utilizator	73
Butoanele lui micro:bit	74
Senzorul de temperatură.....	74
Busola (Compass).....	75
Luminozitate	76