

Algoritmi care lucrează pe numere

Problema 1

Se consideră algoritmul $\text{ceFace}(a, b)$, unde a și b sunt numere naturale ($1 \leq a, b \leq 10000$ la momentul apelului).

Algorithm $\text{ceFace}(a, b)$:

ct1 \leftarrow 0

ct2 \leftarrow 0

While $a \bmod 3 = 0$ **AND** $b \bmod 3 = 0$ **do**

$a \leftarrow a \text{ DIV } 3$

$b \leftarrow b \text{ DIV } 3$

EndWhile

While $a \bmod 3 = 0$ **do**

$a \leftarrow a \text{ DIV } 3$

 ct1 \leftarrow ct1 + 1

EndWhile

While $b \bmod 3 = 0$ **do**

$b \leftarrow b \text{ DIV } 3$

 ct2 \leftarrow ct2 + 1

EndWhile

While $a \bmod 5 = 0$ **AND** $b \bmod 5 = 0$ **do**

$a \leftarrow a \text{ DIV } 5$

$b \leftarrow b \text{ DIV } 5$

EndWhile

While $a \bmod 5 = 0$ **do**

$a \leftarrow a \text{ DIV } 5$

 ct1 \leftarrow ct1 + 1

EndWhile

While $b \bmod 5 = 0$ **do**

$b \leftarrow b \text{ DIV } 5$

 ct2 \leftarrow ct2 + 1

EndWhile

If ct1 = ct2 **then**

 Return True

Else

 Return False

EndIf

EndAlgorithm

Care dintre următoarele afirmații sunt adevărate?

- Apelul $\text{findNumber}(90, 135)$ returnează False.
- Dacă puterea maximă a lui 3 la care se divide a este aceeași și pentru b , atunci algoritmul va returna True.
- Dacă numerele a și b nu sunt divizibile cu 3 și nici cu 5, atunci algoritmul va returna True.
- Algoritmul returnează True dacă și numai dacă puterea maximă a lui 3 și puterea maximă a lui 5 la care se divide a este aceeași și pentru b .

Răspuns: a), c)

Observatie. ct1 numără la comun de câte ori a se divide cu 3 și cu 5 în plus față de b , similar ct2 dar pentru b , prin urmare dacă $\text{ct1} = \text{ct2}$ nu înseamnă întotdeauna că cele două numere se divid cu aceeași putere a lui 3 și aceeași putere a lui 5. De exemplu apelul $\text{ceFace}(45, 75)$ va returna True pentru că a se divide cu 3^2 și cu 5^1 , dar b se divide cu 3^1 și cu 5^2 , deci $\text{ct1} = 1$ și $\text{ct2} = 1$, totuși pentru aceste numere d) nu este adevărat.

Problema 2

Un număr se numește narcisist dacă este egal cu suma cifrelor proprii ridicate fiecare la puterea numărului de cifre.

Exemplu. 153 este narcisist: $153 = 1^3 + 5^3 + 3^3$

Precizați care dintre următorii algoritmi returnează True dacă și numai dacă numărul n este narcisist.

<p>a)</p> <pre>Algorithm narcisist(n): t ← n ct ← 0 While t > 0 do ct ← ct + 1 t ← t DIV 10 EndWhile n1 ← n s ← 0 While n1 > 0 do r ← n1 MOD 10 n1 ← n1 DIV 10 For i ← 1, ct execute r ← r * r //Greșit EndFor s ← s + r EndWhile If s = n then Return True Else Return False EndIf EndAlgorithm</pre>	<p>b)</p> <pre>Algorithm narcisist(n): t ← n ct ← 0 While t > 0 do ct ← ct + 1 t ← t DIV 10 EndWhile n1 ← n s ← 0 While n1 > 0 do r ← n1 MOD 10 //bucla infinita pentru //ca nu trecem la urmatoarea //cifra p ← 1 For i ← 1, ct execute p ← p * r EndFor s ← s + p EndWhile If s = n then Return True Else Return False EndIf EndAlgorithm</pre>	<p>c)</p> <pre>Algorithm narcisist(n): t ← n ct ← 0 While t > 0 do ct ← ct + 1 t ← t DIV 10 EndWhile n1 ← n s ← 0 While n1 > 0 do r ← n1 MOD 10 n1 ← n1 DIV 10 p ← 1 For i ← 1, ct execute p ← p * r EndFor s ← s + p EndWhile If s = n then Return True Else Return False EndIf EndAlgorithm</pre>
<p>d)</p> <p>Niciunul din algoritmi nu corespunde cerinței.</p>		

Răspuns: c)

Problema 3

- Care este al șaselea cel mai mic număr format din 4 cifre din mulțimea {0, 1, 2, 3, 4}?
- Care este al unsprezecelea cel mai mic număr format din 4 cifre din mulțimea {0, 1, 2, 3, 4}?

Răspuns a): 1010

Răspuns b): 1020

Numerele de 4 cifre vor fi, în ordine crescătoare: 1000, 1001, 1002, 1003, 1004, 1010, 1011, 1012, 1013, 1014, 1020, etc.

Problema 4

- Care este al treilea cel mai mic număr format din 4 cifre distincte din mulțimea {0, 1, 2, 3, 4}?
- Care este al șaselea cel mai mic număr format din 4 cifre distincte din mulțimea {0, 1, 2, 3, 4}?

Răspuns a): 1032

Răspuns b): 1043

Numerele de 4 cifre care respectă cerința vor fi, în ordine crescătoare: 1023, 1024, 1032, 1034, 1042, 1043, 1203, 1204, 1230, 1234, 1240, 1243, 1302, etc.

Problema 5

Se consideră algoritmi `check(n)` și `findNumber(n)`, unde n este un număr natural ($1 \leq n \leq 10000$).

```
Algorithm check(n):
  While n > 0 execute
    c ← n MOD 10
    If c MOD 2 = 0 then
      Return False
    EndIf
    n ← n DIV 10
  EndWhile
  Return True
EndAlgorithm
```

```
Algorithm findNumber(n):
  m ← n + 1
  While check(m) = False execute
    m ← m + 1
  EndWhile
  Return m
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- a) Apelul `findNumber(135)` returnează 135.
- b) Apelul `findNumber(134)` returnează 135.
- c) Pentru orice număr par n apelul `findNumber(n)` returnează $n + 1$.
- d) Apelul `findNumber(n)` returnează cel mai mic număr mai mare decât n format doar din cifre pare.

Răspuns: b)

Observatie. De fapt, algoritmul `findNumber` returnează cel mai mic număr mai mare decât n format doar din cifre impare.

Problema 6

Se dă un număr n natural, ($0 < n \leq 10^9$). Să se scrie un algoritm `afiseazaNumere(n)` care afișează toate numerele naturale x cu proprietatea că suma dintre numărul x și suma cifrelor lui x este exact n .

Exemple:

Pentru $n = 10$ se va afișa 5.

Pentru $n = 21$ se va afișa 15.

Pentru $n = 1001$ se va afișa 982 și 1000.

Metoda I (naivă)

Iterăm de la 1 la n și verificăm condiția pentru fiecare număr.

```

Algorithm sumaCifre(n):
  s ← 0
  While n > 0 execute
    r ← n MOD 10
    s ← s + r
    n ← n DIV 10
  EndWhile
  Return s
EndAlgorithm

```

```

Algorithm afiseazaNumere(n):
  For i ← 1, n execute
    s ← i + sumaCifre(i)
    If s = n then
      Write i
    EndIf
  EndFor
EndAlgorithm

```

Metoda II (mai eficientă)

Pentru un număr $n \leq 10^9$ suma cifrelor nu poate depăși valoarea 100 (chiar mai puțin, pentru că fiecare cifră este maxim 9).

Prin urmare valorile x căutate se pot afla doar în intervalul $[n - 100, n - 1]$.

Astfel vom verifica doar aceste valori.

```

Algorithm sumaCifre(n):
  s ← 0
  While n > 0 execute
    r ← n MOD 10
    s ← s + r
    n ← n DIV 10
  EndWhile
  Return s
EndAlgorithm

```

```

Algorithm afiseazaNumere(n):
  For i ← n-1, n-100, -1 execute
    If i > 0 AND i + sumaCifre(i) = n then
      Write i
    EndIf
  EndFor
EndAlgorithm

```

Observație: $n - 100$ poate fi mai mic decât 0, ar trebui să ne oprim la numărul 1 în acel caz (adică am putea opri și For-ul mai repede).

Problema 7

Se consideră algoritmul $\text{calculeaza}(n)$, unde n este un număr natural ($1 \leq n \leq 10000$).

```
Algorithm calculeaza(n):
  If n < 10 then
    Return n
  EndIf
  m ← n DIV 10
  a ← n MOD 10
  b ← m MOD 10
  If b < a then
    a ← b
  EndIf
  m ← (n DIV 100) * 10 + a
  Return calculeaza(m)
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- Apelul $\text{calculeaza}(1747)$ returnează 1.
- Dacă n este un număr natural care nu e divizibil cu 10, iar $n1$ reprezintă oglinditul numărului n , atunci apelurile $\text{calculeaza}(n)$ și $\text{calculeaza}(n1)$ vor returna aceeași valoare indiferent de valoarea lui n .
- Dacă numărul este divizibil în același timp cu 3, 5 și 7, atunci algoritmul va returna întotdeauna 0.
- Dacă numărul este divizibil în același timp cu 2, 3 și 5, atunci algoritmul va returna întotdeauna 0.
- Algoritmul returnează 0 dacă și numai dacă numărul n este divizibil cu 10.

Răspuns: a), b), d)

Observație. De fapt algoritmul returnează cifra minimă a numărului n .

Răspunsul b este adevărat pentru că în acel caz oglinditul numărului conține exact aceleași cifre, eventual în altă ordine, deci cifra minimă va fi aceeași.

Răspunsul d este adevărat pentru că din divizibilitatea cu 2 și cu 5 rezultă ca ultima cifră a numărului este 0, deci cifra minimă va fi 0.

Contraexemplu pentru răspunsul c este $n = 315$, care e divizibil cu 3, 5 și 7, dar algoritmul va returna 1.

Contraexemplu pentru răspunsul c este $n = 102$, care nu e divizibil cu 10, dar conține o cifră 0, deci algoritmul va returna 0.

Problema 8

Se consideră următoarele instrucțiuni:

1. $x \leftarrow x \text{ MOD } 15$
2. $z \leftarrow x \text{ DIV } y$
3. $x \leftarrow 100$
4. $y \leftarrow x \text{ DIV } 10$

În ce ordine pot fi executate instrucțiunile astfel încât la final variabilele x , y și z să aibă valori egale, indiferent de valorile avute anterior:

- a) 2, 4, 1, 3
- b) 3, 4, 2, 1
- c) 1, 2, 4, 3
- d) 3, 1, 2, 4

Răspuns: b

Observație. Este singurul răspuns în care valorile finale ale lui x , y și z nu depind de valori inițiale necunoscute.

Problema 9

Care dintre variabilele din secvența următoare de cod își vor păstra valoarea inițială la sfârșitul secvenței?

```
a ← b + c
c ← a - c
b ← c
c ← a - b
```

- a) a și c
- b) b și c
- c) doar c
- d) a, b și c

Răspuns: b

Dacă ne uităm la secvența de cod observăm următoarele:

$$a = b + c$$

$$c = a - c = b + c - c = b \quad // \text{la pasul 2}$$

$$b = c = b \quad // \text{deci } b \text{ nu își schimbă valoarea la acest pas}$$

$$c = a - b = (b + c) - b = c \quad // \text{deci } c \text{ revine la valoarea lui inițială}$$

Problema 10

Scrieți un algoritm recursiv care verifică dacă un număr natural n este palindrom.

Idee 1: Avem un algoritm cu 3 parametri:

- primul parametru reprezintă numărul n pe care nu îl schimbăm pentru a-l putea compara cu oglinditul său
- al doilea parametru este inițializat cu numărul n , dar apoi tot eliminăm ultima lui cifră pentru a forma oglinditul
- al treilea parametru este cel unde se formează oglinditul

Când al doilea parametru devine 0 înseamnă că s-au procesat toate cifrele numărului n , deci s-a obținut oglinditul numărului n în al treilea parametru.

```
Algorithm palindrom(n, n1, m):
  If n1 = 0 then
    If n = m then
      Return True
    Else
      Return False
    EndIf
  EndIf
  m ← m * 10 + n1 MOD 10
  palindrom(n, n1 DIV 10, m)
EndAlgorithm
```

Apel inițial: `palindrom(n, n, 0)`

Idee 2: Avem un singur parametru n reprezentând numărul, din care eliminăm pe rând prima și ultima cifră după ce le comparăm.

- Calculăm numărul de cifre nr a numărului n
- Obținem prima cifră ca $n \text{ DIV } 10^{nr-1}$
- Comparăm prima cifră cu ultima cifră, adică $n \text{ MOD } 10$
- Dacă nu sunt egale returnăm fals
- Dacă sunt egale eliminăm prima și ultima cifră, apelând recursiv pentru $(n \text{ MOD } 10^{nr-1}) \text{ DIV } 10$
- Condiție de oprire: dacă $n < 10$ atunci returnăm true