

CURS PREGATIRE ADMITERE UBB 2023

Curs 6 - Exercitii

BAZE DE NUMERATIE

- Intrucat am ajuns la concluzia ca bazele de numeratie merita o atentie deosebita, am decis sa reluam acest capitol.
- Primele observatii care trebuie facute sunt:
 - Bazele de numeratie reprezinta niste modalitati de reprezentare a numerelor
 - Oamenii folosesc baza 10 deoarece omul are 10 degete
 - Calculatoarele folosesc baza 2 deoarece pot utiliza doar valori 0/1 – True / False in reprezentarea numerelor
 - Informaticienii folosesc de multe ori baza 16 pentru ca reprezinta o scriere prescurtata a numerelor in baza 2.

DATE GENERALE

- Bazele de numeratie cunoscute sunt: B2, B3, B4, B5, B6, B7, B8, ..., B16
- Desi exista o infinitate de baze, cele mai utilizate sunt B2, B4, B8, B10, B16
- Orice numar scris intr-o baza oarecare b , poate contine doar cifre din intervalul $\{0, 1, \dots, b-1\}$
- Transformarile intre baze reprezinta cea mai importanta operatie si poate fi facuta in multiple moduri.

REPREZENTAREA NUMERELEOR

- B10 : $102321 = 1 * 100000 + 0 * 10000 + 2 * 1000 + 3 * 100 + 2 * 10 + 1 * 1$
- Generalizare:
 - În baza 10, $abcd = a * 10^3 + b * 10^2 + c * 10^1 + d * 10^0$
 - În orice alta baza, bz, $abcd = a * bz^3 + b * bz^2 + c * bz^1 + d$
- Astfel, dacă avem în baza 2 numărul 1011001:
$$\begin{aligned}1011001 &= 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 \\&= 64 + 16 + 8 + 1 = 89 \text{ în baza } 10\end{aligned}$$

TRANSFOMAREA INTRE BAZA

- Desi exista modalitatea aceasta prezentata anterior foarte simpla de transformare din orice baza b spre baza 10, din baza 10 spre baza b exista cateva variante diferite, insa cea mai simpla este cea a scaderilor repetate.
 - Fie 153 in baza 10 si vrem sa il obtinem in baza 2:
 - $153 \% 2 = 1, 153 / 2 = 76$
 - $76 \% 2 = 0, 76 / 2 = 38$
 - $38 \% 2 = 0, 38 / 2 = 19$
 - $19 \% 2 = 1, 19 / 2 = 9$
 - $9 \% 2 = 1, 9 / 2 = 4$
 - $4 \% 2 = 0, 4 / 2 = 2$
 - $2 \% 2 = 0, 2 / 2 = 1$
 - $1 \% 2 = 1, 1 / 2 = 0$
- Deci, $153(10) = 10011001(2)$
ATENTIE! RESTURILE SE CITESC DE JOS IN SUS

CONVERSII RAPIDE

Baza 2 -> Baza 16

base 10	base 2=2¹	base 4=2²	base 8=2³	base 16=2⁴
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8
9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F

CUM FUNCTIONEAZA?

- Fie $1110011001(2) \rightarrow x(16)$
- Trebuie sa grupam cate 4 termeni de la dreapta spre stanga si obtinem:
 - 0011, 1001, 1001
- Fiecare grupare, este inlocuita cu cifra corespunzatoare din baza 16:
 - 0011 \rightarrow 3, 1001 \rightarrow 9
- Astfel, inlocuind obtinem ca:
 - $1110011001(2) = 399(16)$

CONCLUZII

- Se observa foarte clar motivul pentru care informaticienii si matematicienii prefera scrierea in baza 16.
- Transformarea se intampla foarte repede si citirea numerelor in baza 16 este mult mai usoara.
- Arhitecturile din ziua de astazi ale calculatoarelor sunt pe 64 de biti (probabil va amintiti cand ati instalat windows-ul), iar acest lucru face imposibila citirea numerelor in baza 2 care au o lungime de pana la 64 de biti.
- Aceste numere, in baza 16 au o lungime de maxim 8 cifre.

EXERCITII

- Convertiti $1101011101_2 \rightarrow x(8)$
- Convertiti $19232_{10} \rightarrow x(16)$
- Calculati $1101011_2 + 1011011_2$
- Calculati $1111111_2 + 1_{10}$
- Decideti daca $111101_2 >/< 200_{10}$
- Care este cea mai mica baza de numeratie b in care pot fi reprezentate urmatoarele valori:
 - 37632
 - 2642
 - 3526118
 - A33D21



GRILE

4. Un tip de date întreg reprezentat pe x biți (x este număr natural strict pozitiv) va putea reține valori întregi din:

- A. $[0, 2^x]$
- B. $[0, 2^{x-1}-1]$
- C. $[-2^{x-1}, 2^{x-1}-1]$
- D. $[-2^x, 2^x-1]$

Raspuns: B, C

6. Se consideră următoarea expresie logică: $(\text{NOT } Y \text{ OR } Z) \text{ OR } (X \text{ AND } Y)$. Alegeti valorile pentru X , Y , Z astfel încât rezultatul evaluării expresiei să fie *adevărat*:

- A. $X \leftarrow \text{fals}; Y \leftarrow \text{fals}; Z \leftarrow \text{fals};$
- B. $X \leftarrow \text{fals}; Y \leftarrow \text{adevărat}; Z \leftarrow \text{fals};$
- C. $X \leftarrow \text{adevărat}; Y \leftarrow \text{fals}; Z \leftarrow \text{adevărat};$
- D. $X \leftarrow \text{fals}; Y \leftarrow \text{adevărat}; Z \leftarrow \text{adevărat};$

Raspuns: A, C, D

7. Precizați care dintre următoarele expresii are valoarea adevărat dacă și numai dacă numărul natural n este divizibil cu 3 și are ultima cifră 4 sau 6:

- A. $n \text{ DIV } 3 = 0$ și $(n \text{ MOD } 10 = 4 \text{ sau } n \text{ MOD } 10 = 6)$
- B. $n \text{ MOD } 3 = 0$ și $(n \text{ MOD } 10 = 4 \text{ sau } n \text{ MOD } 10 = 6)$
- C. $(n \text{ MOD } 3 = 0$ și $n \text{ MOD } 10 = 4)$ sau $(n \text{ MOD } 3 = 0$ și $n \text{ MOD } 10 = 6)$
- D. $(n \text{ MOD } 3 = 0$ și $n \text{ MOD } 10 = 4)$ sau $n \text{ MOD } 10 = 6$

Raspuns: B, C

5. Se dă subalgoritmul $f(a, b)$:

```
Subalgoritm f(a, b):
    Dacă a > 1 atunci
        returnează b * f(a - 1, b)
    altfel
        returnează b * f(a + 1, b)
    SfDacă
SfSubalgoritm
```

Precizați de câte ori se auto apelează subalgoritmul $f(a, b)$ în urma execuției următoarei secvențe de instrucțiuni:

```
a ← 4, b ← 3
c ← f(a, b)
```

- A. de 4 ori
- B. de 3 ori
- C. de o infinitate de ori
- D. niciodată

Raspuns: C

1. Subalgoritmul generare(*n*) prelucrează un număr natural *n* ($0 < n < 100$).

```
Subalgoritm generare(n):
    nr ← 0
    Pentru i ← 1, 1801 execută
        folositi ← fals
    SfPentru
    CâtTimp nu folositn execută
        suma ← 0, folositn ← adevărat
        CâtTimp (n ≠ 0) execută
            cifra ← n MOD 10, n ← n DIV 10
            suma ← suma + cifra * cifra * cifra
        SfCâtTimp
        n ← suma, nr ← nr + 1
    SfCâtTimp
    returnează nr
SfSubalgoritm
```

Precizați care este efectul acestui subalgoritm.

- A. Calculează, în mod repetat, suma cuburilor cifrelor numărului *n* până când suma egalează numărul *n* și returnează numărul repetărilor efectuate
- B. Calculează suma cuburilor cifrelor numărului *n* și returnează această sumă
- C. Calculează suma cuburilor cifrelor numărului *n*, înlocuiește numărul *n* cu suma obținută și returnează această sumă
- D. Calculează numărul înlocuirilor lui *n* cu suma cuburilor cifrelor sale până când se obține o valoare calculată anterior sau numărul însuși și returnează acest număr

Raspuns: D

3. Se consideră subalgoritmul expresie(*n*), unde *n* este un număr natural ($1 \leq n \leq 10000$).

Subalgoritm expresie(*n*):

Dacă $n > 0$ atunci

Dacă $n \bmod 2 = 0$ atunci

 returnează $-n * (n + 1) + \text{expresie}(n - 1)$

 altfel

 returnează $n * (n + 1) + \text{expresie}(n - 1)$

 SfDacă

 altfel

 returnează 0

 SfDacă

SfSubalgoritm

Precizați forma matematică a expresiei $E(n)$ calculată de acest subalgoritm:

- A. $E(n) = 1 * 2 - 2 * 3 + 3 * 4 + \dots + (-1)^{n+1} * n * (n + 1)$
- B. $E(n) = 1 * 2 - 2 * 3 + 3 * 4 + \dots + (-1)^n * n * (n + 1)$
- C. $E(n) = 1 * 2 + 2 * 3 + 3 * 4 + \dots + (-1)^{n+1} * n * (n + 1)$
- D. $E(n) = 1 * 2 - 2 * 3 - 3 * 4 - \dots - (-1)^n * n * (n + 1)$

Raspuns: A

8. Fie următorul subalgoritm:

```
Subalgoritm f(a):
    Dacă a ≠ 0 atunci
        returnează a + f(a - 1)
    altfel
        returnează 0
    SfDacă
    SfSubalgoritm
```

Care din afirmațiile de mai jos sunt false?

- A. daca a este negativ, subalgoritmul întoarce 0
- B. valoarea calculată de f este $a * (a + 1) / 4$
- C. subalgoritmul calculează suma numerelor naturale mai mici sau egale cu a
- D. apelul $f(-5)$ intră în ciclu infinit.

Raspuns: A, B, C

9. Se consideră următorul subalgoritm:

Subalgoritm SA9(a):

Dacă $a < 50$ atunci

Dacă $a \bmod 3 = 0$ atunci

 returnează $SA9(2 * a - 3)$

altfel

 returnează $SA9(2 * a - 1)$

SfDacă

altfel

 returnează a

SfDacă

SfSubalgoritm

Pentru care dintre valorile parametrului de intrare a subalgoritmul va returna valoarea 61?

- A. 16
- B. 61
- C. 4
- D. 31

Raspuns: A, B, D

10. Se consideră subalgoritmul prelucreaza(v , k), unde v este un sir cu k numere naturale ($1 \leq k \leq 1000$).

```
Subalgoritm prelucreaza(v, k)
    i ← 1, n ← 0
    CâtTimp i ≤ k și vi ≠ 0 execută
        y ← vi, c ← 0
        CâtTimp y > 0 execută
            Dacă y MOD 10 > c atunci
                c ← y MOD 10
            SfDacă
                y ← y DIV 10
            SfCâtTimp
            n ← n * 10 + c
            i ← i + 1
        SfCâtTimp
        returnează n
    SfSubalgoritm
```

Precizați pentru care valori ale lui v și k subalgoritmul returnează valoarea 928.

- A. $v = (194, 121, 782, 0)$ și $k = 4$
- B. $v = (928)$ și $k = 1$
- C. $v = (9, 2, 8, 0)$ și $k = 4$
- D. $v = (8, 2, 9)$ și $k = 3$

Raspuns: A, C

11. Se consideră următoarea expresie logică $(X \text{ OR } Z) \text{ AND } (\text{NOT } X \text{ OR } Y)$. Alegeți valorile pentru X , Y , Z astfel încât evaluarea expresiei să dea rezultatul TRUE:

- A. $X \leftarrow \text{FALSE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{TRUE};$
- B. $X \leftarrow \text{TRUE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{FALSE};$
- C. $X \leftarrow \text{FALSE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{FALSE};$
- D. $X \leftarrow \text{TRUE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{TRUE};$

Raspuns: A, D

12. Se consideră următorul program:

Varianta C	Varianta C++	Varianta Pascal
<pre>#include <stdio.h> int prelVector(int v[], int *n) { int s = 0; int i = 2; while (i <= *n) { s = s + v[i] - v[i - 1]; if (v[i] == v[i - 1]) *n = *n - 1; i++; } return s; } int main(){ int v[8]; v[1] = 1; v[2] = 4; v[3] = 2; v[4] = 3; v[5] = 3; v[6] = 10; v[7] = 12; int n = 7; int resultat = prelVector(v, &n); printf("%d;%d", n, resultat); return 0; }</pre>	<pre>#include <iostream> using namespace std; int prelVector(int v[], int&n) { int s = 0; int i = 2; while (i <= n) { s = s + v[i] - v[i - 1]; if (v[i] == v[i - 1]) n--; i++; } return s; } int main(){ int v[8]; v[1] = 1; v[2] = 4; v[3] = 2; v[4] = 3; v[5] = 3; v[6] = 10; v[7] = 12; int n = 7; int resultat = prelVector(v, n); cout << n << ":" << resultat; return 0; }</pre>	<pre>type vector=array [1..10] of integer; function prelVector(v: vector; var n: integer): integer; var s, i: integer; begin s := 0; i := 2; while (i <= n) do begin s := s + v[i] - v[i - 1]; if (v[i] = v[i - 1]) then n := n - 1; i := i + 1; end; prelVector := s; end; var n, rezultat:integer; v:vector; begin n := 7; v[1] := 1; v[2] := 4; v[3] := 2; v[4] := 3; v[5] := 3; v[6] := 10; v[7] := 12; rezultat := prelVector(v,n); write(n, ',', rezultat); end.</pre>

Precizați care este rezultatul afișat în urma executării programului.

- A. 7;11
- B. 6;9
- C. 7;9
- D. 7;12

Raspuns: B

15. Se consideră toate șirurile de lungime $l \in \{1, 2, 3\}$ formate din litere din mulțimea $\{a, b, c, d, e\}$. Câte dintre aceste șiruri au elementele ordonate strict descrescător și un număr impar de vocale? (a și e sunt vocale)

- A. 14
- B. 7
- C. 81
- D. 78

Raspuns: A

16. Se consideră dat subalgoritmul $\text{apartine}(x, a, n)$ care verifică dacă un număr natural x aparține mulțimii a cu n elemente; a este un sir cu n elemente și reprezintă o mulțime de numere naturale ($1 \leq n \leq 200, 1 \leq x \leq 1000$).

Fie subalgoritmii $\text{reuniune}(a, n, b, m, c, p)$ și $\text{calcul}(a, n, b, m, c, p)$, descriși mai jos, unde a, b și c sunt siruri care reprezintă mulțimi de numere naturale cu n, m și respectiv p elemente ($1 \leq n \leq 200, 1 \leq m \leq 200, 1 \leq p \leq 400$). Parametrii de intrare sunt a, n, b, m și p , iar parametrii de ieșire sunt c și p .

<p>1. Subalgoritm $\text{reuniune}(a, n, b, m, c, p)$:</p> <p>2. Dacă $n = 0$ atunci</p> <p>3. Pentru $i \leftarrow 1, m$ execută</p> <p>4. $p \leftarrow p + 1, c_p \leftarrow b_i$</p> <p>5. SfPentru</p> <p>6. altfel</p> <p>7. Dacă nu $\text{apartine}(a_n, b, m)$ atunci</p> <p>8. $p \leftarrow p + 1, c_p \leftarrow a_n$</p> <p>9. SfDacă</p> <p>10. $\text{reuniune}(a, n - 1, b, m, c, p)$</p> <p>11. SfDacă</p> <p>12. SfSubalgoritm</p>	<p>1. Subalgoritm $\text{calcul}(a, n, b, m, c, p)$:</p> <p>2. $p \leftarrow 0$</p> <p>3. $\text{reuniune}(a, n, b, m, c, p)$</p> <p>4. SfSubalgoritm</p>
--	--

Precizați care dintre afirmațiile de mai jos sunt întotdeauna adevărate:

- A. când mulțimea a conține un singur element, apelul subalgoritmului $\text{calcul}(a, n, b, m, c, p)$ provoacă apariția unui ciclu infinit
- B. când mulțimea a conține 4 elemente, apelul subalgoritmului $\text{calcul}(a, n, b, m, c, p)$ provoacă executarea instrucțiunii de pe linia 10 a subalgoritmului reuniune de 4 ori
- C. când mulțimea a conține 5 elemente, apelul subalgoritmului $\text{calcul}(a, n, b, m, c, p)$ provoacă executarea instrucțiunii de pe linia 2 a subalgoritmului reuniune de 5 ori
- D. când mulțimea a are aceleași elemente ca și mulțimea b , în urma execuției subalgoritmului $\text{calcul}(a, n, b, m, c, p)$ mulțimea c va avea același număr de elemente ca și mulțimea a

Raspuns: B, D

20. Fie subalgoritmul $\text{factoriPrimi}(n, d, k, x)$ care determină cei k factori primi ai unui număr natural n , începând căutarea factorilor primi de la valoarea d . Parametrii de intrare sunt numerele naturale n, d și k , iar parametrii de ieșire sunt sirul x cu cei k factori primi ($1 \leq n \leq 10000, 2 \leq d \leq 10000, 0 \leq k \leq 10000$).

```
Subalgoritm factoriPrimi(n, d, k, x):
    Dacă n MOD d = 0 atunci
        k ← k + 1
        x[k] ← d
    SfDacă
    CâtTimp n MOD d ≠ 0 execută
        n ← n DIV d
    SfCâtTimp
    Dacă n > 1 atunci
        factoriPrimi(n, d + 1, k, x)
    SfDacă
SfSubalgoritm
```

Stabiliți de câte ori se autoapeleză subalgoritmul $\text{factoriPrimi}(n, d, k, x)$ în următoarea secvență de instrucțiuni:

```
n ← 120
d ← 2
k ← 0
factoriPrimi(n, d, k, x)
```

- A. de 3 ori
- B. de 5 ori
- C. de 6 ori
- D. de același număr de ori ca și în cadrul secvenței de instrucțiuni:

```
n ← 750
d ← 2
k ← 0
factoriPrimi(n, d, k, x)
```

Raspuns: A, D

23. Fie s un sir de numere naturale unde elementele s_i sunt de forma $s_i = \begin{cases} x, & \text{dacă } i = 1 \\ x + 1, & \text{dacă } i = 2 \\ s_{(i-1)} @ s_{(i-2)}, & \text{dacă } i > 2 \end{cases}$

($i = 1, 2, \dots$). Operatorul $@$ concatenează cifrele operandului stâng cu cifrele operandului drept, în această ordine (cifre aferente reprezentării în baza 10), iar x este un număr natural ($1 \leq x \leq 99$). De exemplu, dacă $x = 3$, sirul s va conține valorile $3, 4, 43, 434, 43443, \dots$. Precizați numărul cifrelor aceluia termen din sirul s care precede termenul format din k ($1 \leq k \leq 30$) cifre.

- A. dacă $x = 15$ și $k = 6$, numărul cifrelor termenului aflat în sirul s în fața termenului format din k cifre este 5.
- B. dacă $x = 2$ și $k = 8$, numărul cifrelor termenului aflat în sirul s în fața termenului format din k cifre este 5.
- C. dacă $x = 14$ și $k = 26$, numărul cifrelor termenului aflat în sirul s în fața termenului format din k cifre este 16.
- D. dacă $x = 5$ și $k = 13$, numărul cifrelor termenului aflat în sirul s în fața termenului format din k cifre este 10.

Raspuns: B, C

26. Se consideră subalgoritmul $\text{alg}(x, b)$ cu parametrii de intrare două numere naturale x și b ($1 \leq x \leq 1000$, $1 < b \leq 10$).

```
Subalgoritm alg(x, b):
    s ← 0
    CâtTimp x > 0 execută
        s ← s + x MOD b
        x ← x DIV b
    SfCâtTimp
    returnează s MOD (b - 1) = 0
SfSubalgoritm
```

Precizați efectul acestui subalgoritm.

- A. verifică dacă suma cifrelor reprezentării în baza $b - 1$ a numărului x este divizibilă cu $b - 1$
- B. verifică dacă numărul natural x este divizibil cu $b - 1$
- C. verifică dacă suma cifrelor reprezentării în baza b a numărului x este divizibilă cu $b - 1$
- D. verifică dacă suma cifrelor numărului x este divizibilă cu $b - 1$

Raspuns: B, C

28. Dreptunghiul cu laturile de lungimi m și n (m, n – numere naturale, $0 < m < 101$, $0 < n < 101$) este împărțit în pătrățele cu latura de lungime 1. Se consideră subalgoritmul dreptunghi(m, n):

```
Subalgoritm dreptunghi(m, n)
    d ← m
    c ← n
    CâtTimp d ≠ c execută
        Dacă d > c atunci
            d ← d - c
        altfel
            c ← c - d
        SfDacă
    SfCâtTimp
    returnează m + n - d
SfSubalgoritm
```

Precizați efectul acestui subalgoritm.

- A. Calculează și returnează numărul pătrățelor cu latura de lungime 1 traversate de o diagonală a dreptunghiului.
- B. Determină în d cel mai mare divizor comun al laturilor dreptunghiului și returnează diferența dintre suma laturilor dreptunghiului și d .
- C. Dacă $m = 8$ și $n = 12$, returnează 16.
- D. Dacă $m = 6$ și $n = 11$, returnează 15.

Raspuns: A, C

30. O matrice cu 8 linii, formată doar din elemente 0 și 1, are următoarele trei proprietăți:

- a. prima linie conține un singur element cu valoarea 1,
- b. linia j conține de două ori mai multe elemente nenule decât linia $j - 1$, pentru orice $j \in \{2, 3, \dots, 8\}$,
- c. ultima linie conține un singur element cu valoarea 0.

Care este numărul total de elemente cu valoarea 0 din matrice?

- A. 777
- B. 769
- C. 528
- D. nu există o astfel de matrice

Raspuns: A

VA MULTUMESC!
