

Modele de grile pentru proba 1

examenul de evaluare a cunoștințelor fundamentale și de specialitate

Calculatoare

Sesiunea: Iulie 2024

1. Pasul de reprezentare reprezintă:

- a) diferența dintre cel mai mare număr pozitiv reprezentabil și cel mai mic număr pozitiv reprezentabil
- b) diferența dintre cel mai mare număr negativ reprezentabil și cel mai mic număr negativ reprezentabil
- c) distanța între două numere consecutive reprezentabile
- d) distanța între oricare două numere reprezentabile

2. În varianta I a codului complementar, bitul de semn are pondere:

- a) 0
- b) n , unde n reprezintă numărul de ranguri de la partea întreagă
- c) negativă
- d) nu are pondere

3. Care din următoarele afirmații privind reprezentarea în virgula mobilă nu este adevărată:

- a) are avantajul extinderii gamei de reprezentare
- b) eroarea relativă este mai mică decât în cazul reprezentării în virgulă fixă
- c) are dezavantajul algoritmilor complecși pentru operații aritmetice
- d) are dezavantajul unei unități logico-aritmetice cu o structură complexă

4. Care din următoarele afirmații, privind un număr în reprezentare normalizată, este adevărată?

- a) toți biții mantisei sunt 1
- b) cel mai semnificativ bit al mantisei este 1
- c) toți biții exponentului sunt 1
- d) cel mai semnificativ bit al exponentului este 1

5. Care din următoarele nu face parte din structura în 5 unități a calculatorului digital:

- a) unitatea de intrare
- b) unitatea logico-aritmetică
- c) unitatea de control
- d) unitatea de acces direct la memorie (DMA)

6. Care din următoarele afirmații este corectă?

- a) Legătura mulți-la-mulți nu poate fi reprezentată în diagrama Entitate/Legătură (E/R).
- b) Legăturile pot să aibă atribute proprii.
- c) Toate legăturile mulți-la-unu sunt reprezentate prin legături între o entitate slabă și una tare.
- d) Nici una din afirmațiile de mai sus nu este corectă.

7. Se consideră tabela **Distribuitor_tabel** cu următoarele înregistrări:

Distribuitor	MasiniVandute	Profit
FORD	50	55
BMW	NULL	20
Toyota	200	700
Suzuky	100	100

Care este rezultatul pe care îl returnează interogarea **SELECT COUNT(MasiniVandute) from Distribuitor_tabel** ?

- a) 350
- b) 4
- c) 3
- d) NULL

8. Se consideră tabela **Distribuitor_tabel** cu următoarele înregistrări:

Distribuitor	MasiniVandute	Profit
FORD	50	50
BMW	NULL	20
Toyota	200	200
Suzuky	100	100

Care este rezultatul pe care îl returnează interogarea **SELECT COUNT(*) from Distribuitor_tabel**?

- a) 370
- b) 4
- c) 350
- d) NULL

9. Se consideră relația **R(A)** cu **r** tupluri unice în **R**, și relația **S(A)** cu **s** tupluri unice în **S**. Notăm cu **t** numărul de tupluri prezente în relația (R minus S). Care din următoarele perechi de valori (r,s,t) este posibilă?

- a) (5, 3, 1)
- b) (5,2,0)
- c) (10,5,15)
- d) (5,3,2)

10. Se consideră relația **R(A)** cu **r** tupluri unice în **R**, și relația **S(A)** cu **s** tupluri unice în **S**. Notăm cu **t** numărul de tupluri prezente în relația (R natural join S). Care din următoarele perechi de valori (r,s,t) este posibilă?

- a) (5, 0, 5)
- b) (1,1,2)
- c) (3,3,27)
- d) (2,3,6)

11. Numărul zonelor de cunoștințe cheie ale domeniului Inginerie Software, conform „Software Engineering Body of Knowledge” (<http://www.swebok.org>), este de:

- a) 7
- b) 9
- c) 10
- d) 11

12. Prin cerințe non-funcționale înțelegem:

- a) Descrieri ale serviciilor pe care trebuie să le ofere sistemul, cum trebuie să reacționeze și să se comporte sistemul în situații particulare
- b) Constrângeri asupra serviciilor/ funcțiilor-sistem, sau procesului de dezvoltare
- c) Cerințe ce provin din domeniul de aplicație al sistemului
- d) Recomandări ce ajută la luarea deciziilor de proiectare când există opțiuni multiple

13. Primul pas în ingineria cerințelor constă în:

- a) Analiza cerințelor
- b) Extragerea/ clasificarea cerințelor
- c) Specificarea cerințelor
- d) Validarea cerințelor

14. Modelele de procese de dezvoltare software ce folosesc prototipizarea sunt:

- a) Cele mai indicate atunci când cerințele sunt clare și complet definite
- b) Cele mai indicate când clientul nu poate defini clar cerințele de la început
- c) Foarte utile în cazul proiectelor a căror echipă de dezvoltare este mare
- d) Abordări riscante, ce produc rareori un produs complet

15. Etapele esențiale în cadrul modelului de tip cascadă (Waterfall) sunt:

- a) Analiza și proiectarea
- b) Proiectarea și implementarea
- c) Analiza și implementarea
- d) Implementarea și testarea

16. Care este eroarea în implementarea clasei CPunct astfel încât apelul funcțiilor să fie corect?

<pre>class CPunct { public: CPunct(int x = 0, int y = 0) : m_x(x), m_y(y) {} void Translate(int x, int y) const { m_x += x; m_y += x; } private: int m_x, m_y; };</pre>	<pre>int main() { CPunct p1; const CPunct p2(0, 1); p1.Translate(2, 3); p2.Translate(3, 4); return 0; }</pre>
---	---

- a) Lipsa implementării constructorului implicit
- b) Implementarea metodei Translate nu trebuie să fie const
- c) Membrii de tip dată a clasei trebuie să fie membrii publici
- d) Implementarea este corectă

17. Ce valoare va fi afișată?

<pre>#include <iostream> using namespace std; int k; void afisare() { cout << k << endl; } void afisare(int v) { int k = v; k = ::k + 2; cout << k << endl; }</pre>	<pre>int main() { afisare(5); return 0; }</pre>
--	--

- a) 5
- b) 2
- c) 7
- d) 0

18. Care sunt valorile afișate?

<pre>#include <iostream> using namespace std; class C1 { int v; C1(int v) { v = v; } int GetV() { return v; } }; class C2 { void Afisare() { cout << "C2"; } };</pre>	<pre>int main() { cout << sizeof(C1) << endl; cout << sizeof(C2) << endl; return 0; }</pre>
--	---

- a) 4 și 0
- b) 4 și 1
- c) 4 și 4
- d) 4 și 2

19. Ce va afișa următoarea secvență de cod?

<pre>#include <iostream> using namespace std; class Element { public: Element(int) { cout << "Element#"; } ~Element() { cout << "~Element"; } };</pre>	<pre>int main() { Element* p_element = new Element(4); delete p_element; return 0; }</pre>
--	---

- a) Element#~ElementElement#~ElementElement#~ElementElement#~Element
- b) Element#~Element
- c) Element#~ElementElement#~Element
- d) Element#

20. Dacă în cadrul moștenirii pe mai multe niveluri, clasa Urs moștenește clasa AnimalCarnivor, iar clasa AnimalCarnivor moștenește clasa Animal. În ce secvență sunt apelați destructorii dacă a fost declarat un obiect din clasa Urs

- a) ~Urs() ~Animal() ~AnimalCarnivor()
- b) ~Urs() ~AnimalCarnivor() ~Animal()
- c) ~Animal() ~Urs() ~AnimalCarnivor()
- d) ~Animal() ~AnimalCarnivor() ~Urs()

21. Cine este creatorul limbajului de programare C?

- a) James Gosling
- b) Dennis Ritchie
- c) Ken Thompson
- d) Brian Kernighan

22. Estimați vechimea (maturitatea) limbajului de programare C?

- a) 100 ani
- b) Peste 50 de ani
- c) Circa 25 de ani
- d) 18 ani

23. Alegeți tipurile de date care nu sunt de bază ale limbajului C din lista de mai jos:

- a) char
- b) int
- c) float
- d) short

24. De câte ori poate fi aplicat operatorul de dereferențiere în limbajul de programare C?

- a) 3
- b) 128
- c) 256
- d) Fără număr

25. Alegeți un exemplu de structură de iterație în limbajul de programare C?

- a) for
- b) while
- c) do-while
- d) toate de mai sus

26. Care este expresia echivalentă pentru $x \oplus \bar{y}$?

- a) $x \oplus y$
- b) $x \odot y$
- c) $x | y$
- d) $x\bar{y} + \bar{x}y$

27. Se consideră funcția de comutație de 4 variabile:

$$f(a, b, c, d) = \sum(1, 2, 4, 5, 6, 7, 9) + \sum_d(10, 12, 13, 14)$$

Utilizând diagrame KV să se stabilească care din următoarele forme reprezintă forma minimă disjunctivă:

- a) $ab + cd + \bar{c}d$
- b) $ab\bar{c} + \bar{b}d + ac$
- c) $\bar{a}b + \bar{c}d + c\bar{d}$
- d) $ab + \bar{a}d + \bar{c}d$

28. Se dă un MUX 8 cu 8 intrări de date $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$, trei intrări de selecție S_0, S_1, S_2 , unde S_2 este MSB și o ieșire Y în logică pozitivă. Funcția de comutație $f(x, y, z) = \prod(2, 3, 5, 6)$ se implementează cu un singur MUX 8 astfel:

- a) $(I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7) = (0, 0, 1, 1, 0, 1, 1, 0)$ și $(S_0, S_1, S_2) = (x, y, z)$
- b) $(I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7) = (0, 0, 1, 1, 0, 1, 1, 0)$ și $(S_2, S_1, S_0) = (x, y, z)$
- c) $(I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7) = (1, 1, 0, 0, 1, 0, 0, 1)$ și $(S_0, S_1, S_2) = (x, y, z)$
- d) $(I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7) = (1, 1, 0, 0, 1, 0, 0, 1)$ și $(S_2, S_1, S_0) = (x, y, z)$

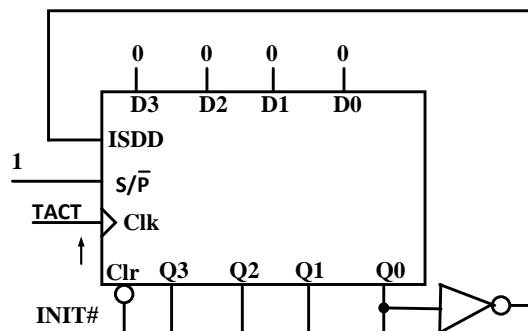
29. Se dă diagram VEM:

$x \backslash y \backslash z$	00	01	11	10
0	0	0	1	\bar{u}
1	u	1	1	u

Forma minimă este:

- a) $xy + yz + zu + x\bar{z}\bar{u}$
- b) $xy + yz + \bar{x}zu$
- c) $xyz + \bar{x}zu + x\bar{z}\bar{u}$
- d) $\bar{y} + \bar{x}\bar{z} + zu + x\bar{z}\bar{u}$

30. Ce implementează schema de mai jos care folosește un registru de deplasare dreapta:



- a) Un numărător în inel care plimbă 0 pe 4 biți
- b) Un numărător modulo 16 pe 4 biți
- c) Un numărător Johnson pe 4 biți
- d) Un numărător în inel care plimbă 1 pe 4 biți

31. Prima regulă care se aplică în crearea de subrețele este:

- a) Ultima grupare din adresa de rețea trebuie să fie 0 sau multiplu de mărimea subrețelei
- b) Mărimea unei subrețele trebuie să fie un număr multiplu al lui 2
- c) Ultima grupare din adresa de rețea trebuie să fie număr par
- d) Mărimea unei subrețele trebuie să fie un număr putere al lui 2

32. A doua regulă care se aplică în formarea de subrețele este:

- a) Ultima grupare din adresa de rețea trebuie să fie 0 sau multiplu de mărimea subrețelei
- b) Mărimea unei subrețele trebuie să fie un număr multiplu al lui 2
- c) Ultima grupare din adresa de rețea trebuie să fie număr par
- d) Mărimea unei subrețele trebuie să fie un număr putere al lui 2

33. A treia regulă care se aplică în formarea de subrețele este:
- Ultima grupare din adresa de rețea este 256 minus mărimea subrețelei
 - Ultima grupare din masca de rețea este 256 minus mărimea subrețelei
 - Ultima grupare din masca de rețea trebuie să fie număr impar
 - Ultima grupare din adresa de rețea trebuie să fie număr par
34. Rezultatul operației AND logic dintre o adresă IPv4 și masca de rețea este:
- Adresa de broadcast a rețelei din care face parte adresa IP
 - Adresa de rețea a rețelei din care face parte adresa IP
 - Întodeauna 1
 - Întodeauna 0
35. Gateway este:
- Adresa MAC a Router-ului de pe interfața din rețeaua curentă
 - Adresa MAC a Router-ului de pe interfața din rețeaua adiacentă rețelei curente
 - Adresa IP a Router-ului de pe interfața din rețeaua curentă
 - Adresa IP a Router-ului de pe interfața din rețeaua adiacentă rețelei curente
36. Cum se numesc procesele în execuție asociate simultan aceluiași program?
- instante;
 - taskuri;
 - joburi;
 - fibre.
37. Preemptiunea înseamnă:
- modalitatea prin care se împiedică acapararea procesorului de către un singur proces sau fir de execuție;
 - alocarea unui set de proces. în mod preemptat;
 - alocarea unui set de fire în mod preemptat;
 - alocarea unor zone de mem. în mod preemptat.
38. Cine NU face parte din contextul local al unui fir de execuție?
- contorul program;
 - stiva proprie;
 - starea registrelor;
 - fișierele deschise de procesul din care face parte firul.
39. Spațiul de adrese ocupat de un program până la încărcarea sa în memorie se numește :
- spatiu fizic de adrese;
 - spatiu disc;
 - spatiu alocabil;
 - spatiu virtual de adrese
40. Care dintre strategiile de încărcare este utilizată mai frecvent?
- prefetch;
 - demandfetch;
 - first-fit;
 - best-fit.

41. Care este numărul maxim de noduri dintr-un arbore de căutare binar cu înălțimea = 5?
Obs. Înălțimea unui arbore este adâncimea maximă a oricărui nod frunză de la nodul rădăcină sau lungimea celei mai lungi căi (adică, numărul de muchii) de la nodul rădăcină la orice nod frunză.
- 2^6-1
 - 2^5-1
 - 2^5
 - 2^6
42. Care dintre următoarele metode de parcurgere a arborelui este folosită pentru a obține cheile unui arbore binar de căutare în ordine crescătoare?
- Pre-Ordine
 - In-Ordine
 - Post-Ordine
 - Parcurgere în adâncime (en., Depth-First)
43. Care este complexitatea maximă de timp pentru operațiile de căutare, inserare și ștergere într-un arbore binar de căutare?
- $O(n)$ pentru toate operațiile
 - $O(\log n)$ pentru toate operațiile
 - $O(\log n)$ pentru căutare și inserare și $O(n)$ pentru ștergere
 - $O(\log n)$ pentru căutare și $O(n)$ pentru inserare și ștergere
44. Ce este un arbore AVL?
- un arbore binar de căutare care este perfect echilibrat
 - un arbore binar de căutare care este aproape echilibrat (adică, diferența de înălțime este de cel mult 1)
 - un arbore binar care este perfect echilibrat
 - un arbore cu mai multe căi care este aproape echilibrat (adică, factor de echilibru în mulțimea $\{-1, 0, 1\}$)
45. Care este înălțimea maximă a unui arbore AVL cu p noduri?
- p
 - $\log(p)$
 - $\log(p)/2$
 - p^2
46. Se notează cu $T(n)$ timpul de execuție al unui algoritm pentru o intrare de dimensiune n . Să presupunem că $T(n)$ satisface următoarea recurență :
- $$T(n) = \begin{cases} T_0 & \text{dacă } n = 1 \\ 2T(\lfloor n/2 \rfloor) + \log n & \text{dacă } n > 1 \end{cases}$$
- Atunci timpul de execuție al algoritmului este:
- $\Theta(n)$
 - $\Theta(n \log n)$
 - $\Theta(n^2)$
 - $\Theta(\log n)$.

47. Se notează cu $T(n)$ timpul de execuție al unui algoritm pentru o intrare de dimensiune n . Să presupunem ca $T(n)$ satisface următoarea recurență :

$$T(n) = \begin{cases} T_0 & \text{dacă } n = 1 \\ 2T(\lfloor n/2 \rfloor) + n & \text{dacă } n > 1 \end{cases}$$

Atunci timpul de execuție al algoritmului este:

- a) $\Theta(n)$
- b) $\Theta(n \log n)$
- c) $\Theta(n^2)$
- d) $\Theta(\log n)$.

48. Se notează cu $T(n)$ timpul de execuție al unui algoritm pentru o intrare de dimensiune n . Să presupunem că $T(n)$ satisface următoarea recurență:

$$T(n) = \begin{cases} T_0 & \text{dacă } n = 1 \\ T(\lfloor n/2 \rfloor) + n & \text{dacă } n > 1 \end{cases}$$

Atunci timpul de execuție al algoritmului este:

- a) $\Theta(n)$
- b) $\Theta(n \log n)$
- c) $\Theta(n^2)$
- d) $\Theta(\log n)$.

49. Se consideră următoarele afirmații: i) $\log_2 n = O(3^n)$; ii) $\sqrt{n} = O(\log_2 n)$; iii) $n^2 = O(1000 \cdot n \cdot \log_2 n)$; iv) $3^n = \Theta(2^{n+100})$? Care dintre următoarele afirmații este adevărată :

- a) i) și ii) adevărate, iii) și iv) false;
- b) i) adevărată, ii), iii) și iv) false;
- c) i) și iv) adevărate, ii) și iii) false;
- d) ii), iii) și iv) adevărate, i) falsă.

50. Se consideră următorul algoritm ce primește două numere naturale $m, n \in \mathbb{N}$ și returnează un număr natural:

ALG(m, n)

1. **if** $m = 0$ **then**
2. **return** 1
3. **else if** $n = 0$ **then**
4. **return** 1
5. **else**
6. **return** ALG($m - 1, n$) + ALG($m, n - 1$)

Se notează cu x numărul de adunări, cu y numărul de apeluri ale ALG (inclusiv apelul inițial) și cu z valoarea returnată, pentru apelul ALG(3,3). Care dintre următoarele afirmații este adevărată:

- a) $x = 19, y = 19, z = 20$;
- b) $x = 10, y = 20, z = 30$;
- c) $x = 14, y = 29, z = 15$;
- d) $x = 19, y = 39, z = 20$.