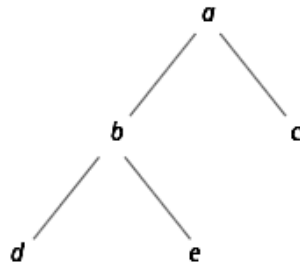


Logică și structuri discrete

1. Definiți demonstrația prin deducție directă și demonstrația prin inducție matematică completă.
2. Definiți noțiunile de *mulțime*, *multimulțime* (*multiset*).
3. Definiți conceptele de *șir*, *limbaj* și *relație*.
4. Arătați modul de reprezentare a grafurilor (direcționate și nedirecționate) din punct de vedere computațional.
5. Într-un graf, să se definească o cale Euler, respectiv un circuit Euler și să se enunțe condiția de existență a acestora.
6. Arătați modul de reprezentare pe calculator și în memorie a arborelui binar din figură:



7. Să se definească noțiunea de funcție calculabilă prin modelul de mașină Turing.
8. Definiți noțiunile de tuplu și listă.
9. Să se descrie procedurile de inferență de tip silogism ipotetic, modus ponens și modus tollens.
10. Să se definească rezoluția și să se enunțe etapele metodei de inferență demonstrativă prin rezoluție.

Tehnici de programare

1. *Vizibilitatea și durata de viață* a unei variabile interne, în limbajul de programare C.
2. *Inițializarea* implicită și explicită a variabilelor, în limbajul de programare C.
3. Comparați un *apel recursiv* cu un *calcul iterativ* echivalent ? Care dintre cele două variante este mai eficientă?
4. Operatorii de *adresare și dereferențiere*, în limbajul de programare C.
5. Ce *operații cu pointeri* sunt permise în limbajul de programare C ?
6. Cum se poate realiza (simula) în C, *transmiterea argumentelor prin adresă* ?
7. Descrieți *modurile de acces la fișiere* permise în limbajul de programare C.
8. Descrieți procesul de rafinare a unui program *în pași succesivi* (Step Wise Refinement).
9. Care este principiul de bază al metodei de rezolvare Divide and Conquer?
10. Enumerați minim 3 avantaje ale utilizării tipurilor de date abstracte.

Programarea orientată pe obiecte

1. Enumerați și descrieți pe scurt cel puțin 3 caracteristici ale orientării pe obiecte.
2. Cum se face transmiterea parametrilor în Java? Precizați efectul asupra tipurilor elementare și asupra tipurilor referință (nu este necesară scrierea de cod).
3. Enumerați și caracterizați pe scurt cei 4 modificatori de acces pentru metode din Java.

4. Ce se înțelege prin polimorfism? Ilustrați printr-un exemplu (nu este necesară scrierea completă a unui program).
5. Prin ce se deosebește o interfață de o clasă abstractă?
6. Ce se poate întâmpla dacă într-o metodă este generată o excepție care nu este declarată în clauza throws a metodei? (Explicați toate cazurile pe care le considerați posibile).
7. Explicați cum se execută metodele sincronizate.
8. Dacă un program conține la un moment dat 10 obiecte de un anumit tip (clasă), câte exemplare ale codului metodelor statice, ale codului metodelor ne-stactice, ale câmpurilor statice și ale câmpurilor ne-stactice ale clasei vor exista în program? De ce?
9. Câte fire de execuție are un program cu interfață grafică în care s-a creat și activat un obiect al unei clase derivate din Thread? Explicați.
10. Câte clase ale unui program Java pot conține o metodă main()? Cum se face lansarea în execuție a programului?

Arhitectura calculatoarelor

1. Să se analizeze comparativ codurile complementare (complement de unu și complement de doi) folosite la reprezentarea numerelor binare în virgulă fixă.
2. Să se analizeze comparativ codurile zecimale (Binary Coded Decimal-BCD, Excess Three- E3) de reprezentare a numerelor în virgulă fixă.
3. Să se analizeze problemele exponentului și mantisei la reprezentarea numerelor binare în virgulă flotantă.
4. Să se prezinte principalele metode de înmulțire a numerelor binare fără semn.
5. Să se scoată în relief caracteristicile procedurii Robertson pentru înmulțirea numerelor binare întregi.
6. Să se scoată în relief caracteristicile procedurii Booth.
7. Să se prezinte treptele unei ierarhii de memorii tipice, a unităților de informație vehiculate între acestea, precum și a principiilor care stau la baza funcționării unei ierarhii.
8. Să se analizeze impactul performanță/cost corespunzător unei ierarhii de memorii cu două trepte.
9. Să se prezinte elementele de structura ale unui cache și să se specifice problemele caracteristice funcționării cache-ului cu paleta de soluții tehnice aferente.
10. Să se analizeze comparativ tipurile de mapare a adreselor de memorie principală în cache în general și să se particularizeze comparația pe exemplul ipotetic a unei memorii principale de 32 blocuri și a unui cache de 8 slot-uri.

Circuite și semnale numerice

1. Să se reprezinte un semnal numeric (impuls) și să se definească parametrii de bază ai acestuia.
2. Stabiliți punctul de funcționare la o diodă semiconductoră.
3. Care sunt regimurile de funcționare ale unui tranzistor bipolar. În circuitele integrate care din regimurile de lucru sunt preferate și de ce?
4. Enumerați cinci parametri de bază la un circuit numeric integrat.
5. Definiți valorile standard ale nivelelor logice de tensiune la circuitele integrate TTL.
6. Care sunt circuite logice de bază ce pot fi realizate cu diode semiconductoră.
7. Să se definească timpii de propagare la un circuit integrat numeric.
8. Să se definească marginea de zgomot, pentru cazul cel mai defavorabil, la un circuit integrat numeric.

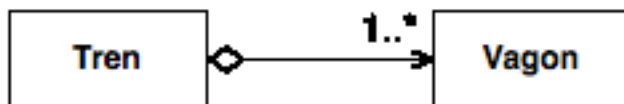
9. Să se definească factorii de încărcare la un circuit intergrat numeric.
10. Care sunt cele trei componente ale puterii consumate de un circuit integrat?

Proiectarea și analiza algoritmilor

1. Să se scrie algoritmul de căutare binară într-un tablou de întregi, prin înjumătățirea intervalului. Se acceptă pseudocod, cod C, C++ sau Java.
2. Să se scrie algoritmul de sortare Quicksort pentru un tablou de întregi prin interschimbarea valorilor de-o parte și de alta a elementului pivot, și aplicarea recursivă a algoritmului independent pe cele două partiții rezultate. Se acceptă pseudocod, cod C, C++ sau Java.
3. Să se scrie algoritmul pentru căutarea unui model într-un șir de caractere. Se acceptă pseudocod, cod C, C++ sau Java.
4. Să se scrie algoritmul de sortare prin interschimbare a unui tablou de întregi. Se acceptă pseudocod, cod C, C++ sau Java.
5. Să se scrie operatorii de inserție și de căutare a unui nod într-o listă simplu înlănțuită cu chei întregi, implementată cu pointeri. Se acceptă pseudocod, cod C, C++ sau Java.
6. Să se scrie operatorii pentru adăugarea sau extragerea de elemente dintr-o stivă. Se acceptă pseudocod, cod C, C++ sau Java.
7. Să se scrie operatorul de inserție a unui arbore binar ordonat ce respectă proprietatea că pentru fiecare nod cheile din stânga sunt mai mici ca și cheia din nod iar cheile din dreapta mai mari decât cheia din nod. Se acceptă pseudocod, cod C, C++ sau Java.
8. Să se scrie structura de date aferentă și algoritmul pentru parcurgerea unui graf în adâncime. Se acceptă pseudocod, cod C, C++ sau Java.
9. Să se scrie algoritmul de determinare al arborelui de acoperire minim pentru un graf folosind metoda lui Prim de selecție a arcului minim, din aproape în aproape, ce leagă mulțimea nodurilor vizitate de mulțimea nodurilor nevizitate. Se acceptă pseudocod, cod C, C++ sau Java.
10. Să se scrie algoritmul de determinare al arborelui de acoperire minim pentru un graf folosind metoda lui Kruskal de inserție a arcelor în ordine crescătoare a ponderilor, însă fără a crea cicluri. Se acceptă pseudocod, cod C, C++ sau Java.

Fundamente de inginerie software

1. Cum se reprezintă într-o diagrama UML de clasă relația între o clasă "Vagon" și o clasă "Tren" considerând că un tren este compus din unul sau mai multe vagoane și că un obiect "Vagon" poate fi refolosit în relație cu diferite obiecte "Tren". Precizați cum se numește acest tip de relație descris și care au fost indiciile care v-au ajutat să îl identificați.



2. Dacă într-un sistem ar trebui să favorizăm *performanța de timp*, ați opta pentru stilul arhitectural *Layered* (arhitectura stratificată)? Dar dacă ar trebui să favorizăm *securitatea*? Justificați succint răspunsul.

3. Precizati daca urmatoarea afirmatie este adevarata sau falsa, si justificati succint (1-2 fraze) raspunsul: *La testare blackbox avem nevoie si de cod pentru a verifica daca datele de test acopera toate caile din program?*

Nota: raspunsurile “adevarat/fals” neinsotite de frazele explicative nu se puncteaza!

4. Care este diferenta intre urmatoarele tipuri de relatii intre clase: *asociere, agregare, compozitie?*

5. Enuntati legea lui Brooks, referitoare la extinderea echipelor de dezvoltatori in timpul proiectului. De asemenea precizati, argumentand succint, valoarea de adevar a urmatoarei afirmatii: *Intr-un sistem cu o modularitate foarte buna legea lui Brooks nu se aplica pentru ca activitatea de construire a sistemului este o activitate perfect partitionabila.* (**Nota:** precizarea valorii de adevar a afirmatiei, neinsotita de argumentatie nu se puncteaza)

6. Precizati daca urmatoarea afirmatie este adevarata sau falsa, si justificati succint (1-2 fraze) raspunsul: *Testele whitebox nu pot garanta ca toate erorile dintr-o functie vor fi detectate, dar daca sunt dublate de teste blackbox se poate garanta ca vor fi gasite toate erorile din respectiva functie.*

Nota: raspunsurile “adevarat/fals” neinsotite de frazele explicative nu se puncteaza!

7. Precizati daca urmatoarea afirmatie este adevarata sau falsa, si justificati succint (1-2 fraze) raspunsul: *Testarea de regresie este bazata pe ideea ca atunci cand se opereaza schimbari intr-un modul, imediat dupa modificare eforturile de testare trebuie concentrate exclusiv asupra modulului modificat.*

Nota: raspunsurile “adevarat/fals” neinsotite de frazele explicative nu se puncteaza!

8. Precizati, si argumentati succint daca relatia descrisa de diagrama UML de mai jos modeleaza sau nu cazul unui telefon dual-sim (care poate tine simultan doua cartele) cu cartele care pot fi oricand reutilizate si in alt telefon.



9. Considerand urmatoarele doua procese de dezvoltare: *Waterfall* si respectiv *Extreme Programming*, precizati si argumentati succint pe care l-ati alege daca ati fi *pe rand* in urmatoarele cazuri:

- *Cazul 1:* trebuie sa reimplementati de la zero, un sistem complex (cca. 1 milion de linii de cod) pentru gestiunea personalului si a salariilor, care sa inlocuiasca sistemul existent in prezent, fara nici o modificare de cerinte.
- *Cazul 2:* aveti de construit un sistem pentru plata taxelor si impozitelor pentru tara imaginara Ainamor cu o legislatie haotica si in permanenta modificare

Nota: cele 2 cazuri sunt complet independente.

10. Precizati daca urmatoarea afirmatie este adevarata sau falsa, si justificati succint (1-2 fraze) raspunsul: *Procesul de dezvoltare in cascada (waterfall) este recomandat in majoritatea proiectelor intrucat, datorita structurii sale rigide, poate constrange clientul sa descopere, inca de la inceput, toate posibilele cauze de schimbare din sistem.*

Nota: raspunsurile “adevarat/fals” neinsotite de frazele explicative nu se puncteaza!

Fundamente de ingineria calculatoarelor

1. Să se prezinte soluțiile de replasare a blocurilor la accesarea unui cache și metodele de implementare ale acestora.
2. Să se prezinte politicile de scriere în cache-uri cu caracteristicile avantajoase și dezavantajoase specifice diferitelor metode.
3. Să se prezinte metodele de reducere a miss penalty-ului la accesarea cache-lui și să se scoată în relief elementele specifice fiecărei soluții.
4. Să se prezinte metodele de reducere ale miss rate-ului la accesarea cache-lui și să se scoată în relief elementele specifice fiecărei soluții.
5. Să se prezinte metodele de reducere ale miss penalty-ului și miss rate-ului bazate pe paralelism la accesarea cache-lui și să se scoată în relief elementele specifice fiecărei soluții.
6. Să se detalieze modurile de adresare specifice mașinii MIPS (Microprocessor without Interlocked Pipeline Stages) convenționale cu 5 stagii.
7. Să se clasifice și să se definească tipurile de hazarduri de date cu exemplificarea hazardului de tip RAW (Read After Write) la mașina MIPS (Microprocessor without Interlocked Pipeline Stages) convențională cu 5 stagii.
8. Să se prezinte soluțiile de combatere a hazardurilor de date de tip RAW (Read After Write) pentru instrucțiunile load și store la mașina MIPS (Microprocessor without Interlocked Pipeline Stages) convențională cu 5 stagii.
9. Să se analizeze degradarea de performanță prin hazardul de control la mașina MIPS (Microprocessor without Interlocked Pipeline Stages) convențională cu 5 stagii și să se propună o soluție de îmbunătățire bazată pe intervenții în schemele mașinii.
10. Să se urmărească modificarea conținutului registrului de instrucții (Instruction Register – IR), a numărătorului de program (Program Counter – PC) și a indicatorului de stivă (Stack Pointer – SP) la execuția unei instrucții de apelare a unei subrutine, respectiv în cazul unei intreruperi sau a unei devieri (trap).

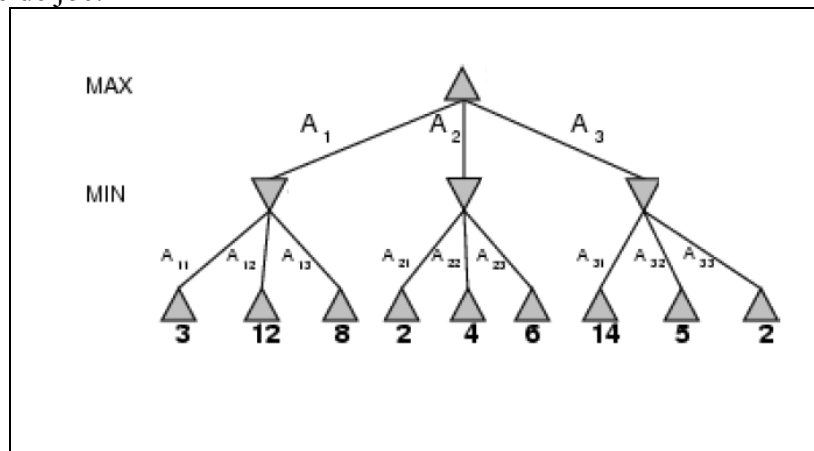
Sisteme de operare

1. Definiți un sistem de operare și enumerați (cu o scurtă caracterizare) cel puțin 4 responsabilități ale unui sistem de operare.
2. Calculați numărul total de blocuri necesar pentru un fișier cu dimensiunea de 1 Moctet într-un sistem de fișiere UNIX cu blocuri de 1 koctet și adrese de bloc pe 4 octeți.
3. Enumerați (cu o scurtă caracterizare) cel puțin 5 directoare de pe primul nivel al unui sistem de fișiere UNIX.
4. Care sunt activitățile efectuate de un interpretor de comenzi (Shell) UNIX pentru stabilirea ambianței de execuție a unei comenzi?
5. Prin ce se deosebește o funcție de bibliotecă de un apel sistem?
6. Care sunt categoriile de utilizatori la stabilirea drepturilor de acces într-un sistem de fișiere UNIX și cum se interpretează drepturile de acces la fișierele obișnuite și la directoare?
7. Explicați cum se execută apelul sistem UNIX cu prototipul:
int fork(void)
8. Care este rolul semnalelor în UNIX? Precizați cum se pot genera semnale și ce acțiuni poate specifica un proces pentru un anumit semnal.

9. Care este rolul unui pipe și în ce condiții se poate utiliza?
10. Ce relație există între un proces și un fir de execuție? În ce moduri se pot implementa firele de execuție?

Bazele inteligenței artificiale

1. Definiți conceptele: cunoaștere, deducție logica.
2. Informațiile despre un fenomen din lumea reală sunt reprezentate în bazele de cunoștințe prin intermediul clauzelor Horn? Cum se definește o clauză Horn? Dați un exemplu.
3. Clauzele Horn utilizate pentru descrierea cunoștințelor conțin constante și variabile. Care este semnificația conceptului de constantă, respectiv variabilă? Care sunt stările în care poate fi prezenta o variabilă și ce reprezintă respectivele stări? Exemplificați aceste concepte.
4. Care este diferența între aspectul declarativ și procedural al mecanismului de inferență în Prolog. Dați un exemplu.
5. În ce constă aspectul de tip determinism și respectiv nedeterminism din bazele de cunoștințe? Care este rolul elementului ! (cut).
6. În ce constă tehnica generează și testează utilizată de mașina de inferență Prolog?
7. Se da arborele de joc:



- a) În ce constă principiul strategiei MinMax?
- b) Care este mutarea ce o va face Max în cadrul strategiei MinMax?
8. De ce se folosesc **why** și **how** în sistemele expert și cum se răspunde la ele?
9. Ce este un estimator euristic?
10. Să se prezinte cel puțin 2 strategii de joc ce se utilizează în prezent în construirea aplicațiilor de tip joc.

Sisteme încorporate

1. Definiți noțiunea de Sistem Încorporat și prezentați diferențele față de calculatoarele de uz general.
2. Ce înseamnă multiplexarea/demultiplexarea terminalelor unui microcontroler? Dați un exemplu.
3. Prezentați structura internă a unui microcontroler.
4. Ce se înțelege prin noțiunea de Registru cu Funcțiuni Speciale?
5. Ce se înțelege prin facilitatea de captare/ comparare?

6. Ce se înțelege prin PWM? Dați un exemplu de modul PWM.
7. Justificați impactul frecvenței și a tensiunii de alimentare asupra consumului sistemelor încorporate. Soluții.
8. Justificați impactul codificării pe magistralele externe asupra consumului sistemelor încorporate. Soluții.
9. Reprezentarea nivelelor logice pe liniile magistralei CAN.
10. Prezentați 2 soluții, principial diferite, pentru conectarea microcontrolerelor la Internet. Prezentați avantajele și dezavantajele lor.

Baze de date

1. Definiți elementele algebrei relationale: operator relational, constituent, domeniu, N-uplet.
2. Definiția relației în algebra relatională.
3. Ce este o baza de date relațională?
4. Definiți operațiile algebrei relationale.
5. Normalizarea bazei de date.
6. Care sunt cerințele formelor normale 1, 2 și 3?
7. Cum se deschid și cum se utilizează mai multe fișiere de date și index în XBase?
8. Cum se face accesul direct la înregistrările dintr-un fișier folosind fișirele index?
9. De ce se deschid mai multe fișiere index pentru un fișier de date? Care este fișierul index după care se fac căutările în acces direct și parcurgerea secvențială în fișierul de date în acest caz?
10. Cum se realizează o legătură automată dintre două tabele într-o bază de date prin cheie simbolică?