



Examen de licență iunie 2015 Specializarea Informatică Română

Subiectul 1

Scrieți un program într-unul din limbajele de programare Python, C++, Java, C# care:

- Definiște o clasă** *Produce* având un atribut privat *preț* de tip întreg, un constructor public pentru inițializarea prețului și o metodă publică *prețDeVânzare()* care returnează *prețul* produsului.
- Definiște o clasă** *ProduceAmbalat* derivată din *Produce* având un atribut privat *prețAmbalaj* de tip întreg, un constructor public pentru inițializarea *prețului* produsului și a *prețului ambalajului* și de asemenea o metodă publică *prețDeVânzare()* suprascrisă, returnând suma dintre *prețul* produsului și *prețAmbalaj*.
- Definiște o funcție** care inserează un obiect de tip *Produce* într-o listă de produse ordonată descrescător după *prețul de vânzare* al acestora (astfel încât lista să rămână ordonată).
- Folosind funcția definită la punctul (c), **definiște o funcție** care construiește o listă de produse ordonată după *prețul de vânzare* al acestora, conținând următoarele: un obiect de tip *Produce* având *preț* egal cu 100; un obiect de tip *ProduceAmbalat* având *preț* egal cu 70 și *prețAmbalaj* egal cu 10; un obiect de tip *ProduceAmbalat* având *preț* egal cu 90 și *prețAmbalaj* egal cu 15.
- Definiște o funcție** care primește ca parametru o *listă* de produse de tipul celei descrise la d) și returnează media prețurilor de vânzare ale produselor din listă.
- Construiește în **funcția principală** a programului lista indicată la punctul (d) și afișează pe ieșirea standard rezultatul returnat de funcției de la punctul (e).
- Pentru tipul de dată *listă* utilizat în program, scrieți specificațiile operațiilor folosite.

Notă

- Nu se vor folosi containere sortate.
- Nu se vor defini alte metode decât cele specificate în enunț.
- Nu se vor folosi operații de sortare.

Pentru tipul de dată *Listă* puteți folosi biblioteci existente (Python, C++, Java, C#). În cazul în care nu folosiți biblioteci existente, specificați toate operațiile folosite.

Subiectul 2

a. Creați o bază de date relațională, cu toate tabelele în a treia formă normală, pe baza următoarelor informații despre rețete de prăjituri:

- **ingrediente:** nume ingredient, categorie ingredient (lichid, aromatizant, îndulcitor etc), unități de măsură (grame, litri, etc), număr de calorii pe unitate;
- **prăjituri:** cod prăjitură, denumire prăjitură, descriere prăjitură, listă de ingrediente (cu cantități corespunzătoare pentru fiecare ingredient);

Justificați că tabelele identificate sunt în a treia formă normală.

b. Având baza de date creată la punctul a., scrieți următoarele interogări folosind SQL și cel puțin o dată algebra relațională:

b1. Lista tuturor ingredientelor (nume, categorie ingredient, unitate de măsură, cantitate) pentru «Tiramisu »

b2. Codul și numele prăjiturilor care **conțin** orice tip de îndulcitor dar **nu** conțin aromatizantul vanilie.

b3. Codul și numele prăjiturilor care au mai puțin de 500 calorii.

Subiectul 3

3.1 Presupunând că în fragmentul de program de mai jos, toate instrucțiunile se execută fără eroare și că pipe-urile sunt închise corect, răspundeți la următoarele întrebări.

<pre>1 #define N 6 2 #define K 1 3 4 int main() { 5 int i, n=1, p[N][2]; 6 for(i=0; i<N; i++) { pipe(p[i]); } 7 write(p[0][1], &n, sizeof(int)); 8 9 for(i=0; i<N; i++) { 10 if(fork() == 0) { 11 read(p[i][0], &n, sizeof(int)); 12 n++; 13 write(p[(i+K)%N][1], &n, sizeof(int)); 14 exit(0); 15 } 16 } 17 18 for(i=0; i<N; i++) { wait(0); } 19 read(p[0][0], &n, sizeof(int)); 20 printf("%d\n", n); 21 return 0; 22 }</pre>	<p>(a) Câte procese fiu se vor crea?</p> <p>(b) Ce se va afișa? Explicați funcționarea fragmentului de program.</p> <p>(c) Explicați funcționarea fragmentului de program dacă valoarea lui K este 2?</p>
--	---

3.2 Considerați fragmentul de script UNIX Shell de mai jos.

(a) Explicați funcționarea scriptului

(b) Ce se va afișa?

(c) Explicați în detaliu linia 8

(d) Explicați de ce este nevoie de caracterul „\” pe linia 11?

<pre>1 #!/bin/bash 2 3 M=0 4 N=0 5 for F in *.log; do 6 A=`grep "\<ERROR\>" \$F wc -l` 7 M=`expr \$M + \$A` 8 B=`grep "\<ERROR\>.*\<segmentation fault\> " \$F wc -l` 9 N=`expr \$N + \$B` 10 done 11 expr 100 * \$N / \$M</pre>

Notă: Toate subiectele sunt obligatorii. Fiecare subiect se notează între 1 și 10 de către ambii corectori.

Timp de lucru: 3 ore.

BAREM

INFORMATICĂ

Subiect 1 (Algoritmă și Programare):

Oficiu – 1p

Definirea clasei *Produce* – 0.75p din care

atribut – 0.25

constructor – 0.25

metoda *prețDeVânzare()* - 0.25

Definirea clasei *ProduceAmbalat* – 1.75p din care

relația de moștenire – 0.25

constructor – 0.5

atribut – 0.25

metoda *prețDeVânzare()* – 0.75

Funcția de la punctul c) – 2.25p din care

signatura corectă - 0.25p

algoritm de inserare în interiorul listei - 1.75p

– parcurgere listă și determinarea poziției de inserare – 1.25p

– adăugare element pe poziția determinată anterior – 0.5

returnare rezultat - 0.25p

Funcția de la punctul d) – 1p din care

signatura corectă și declarare listă- 0.25p

creare obiecte – 0.25p

inserare obiecte în listă - 0.25p

returnare rezultat - 0.25p

Funcția de la punctul e) – 1.25p din care

signatura corectă - 0.25p

parcurgere listă cu calcul preț total de vânzare – 0.75 p

returnare rezultat - 0.25p

Program – 0.5p din care

apel funcție construire listă – 0.25p

afișarea rezultatului cerut – 0.25p

Specificațiile operațiilor folosite din tipul de dată *Listă* – 1.5p

Subiect 2 (Baze de date):

1 punct oficiu

Problema a:

1 punct pentru dependențe funcționale

1 punct pentru tabelele în 3NF;

1 punct pentru justificare.

Problema b:

2 puncte pentru b1

2 puncte pentru b2

0.5 puncte pentru prăjiturile cu îndulcitor

0.5 puncte pentru prăjiturile fără vanilie

1 punct pentru instrucțiunea finală

2 puncte pentru b3

1 punct pentru grupare prăjitori, sumă calorii

1 punct pentru instrucțiunea finală

Subiect 3 (Sisteme de operare):

Oficiu 1p

3.1 5p din care

- (a) 1p: Se creeaza 6 (sau N) procese fiu
- (b) 1p: Se va afisa valoarea 7 (sau N+1)
2p: Procesul parinte si fiii isi transmit in cerc un intreg pe care fiii il incrementeaza
- (c) 1p: Intregul circula doar intre parinte si fiii 0, 2 si 4. Fiilor 1, 3 si 5 nu li se trimite nimic si sunt blocati in apelul sistem read. Parintele este blocat in apelul sistem wait.

3.2 4p din care

- (a) 1p: Explicarea functionarii
- (b) 1p: Se afiseaza procentul de erori "segmentation fault" din numarul total de erori
- (c) 1p: Explicarea liniei 8
- (d) 1p: Explicarea necesitatii "\"