



UNIVERSITATEA BABEȘ-BOLYAI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ



Examen de licență iunie 2014
Specializarea Informatică Română

Subiectul 1

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

- Definește o clasă *Student* având un atribut privat *nume* de tip șir de caractere, un constructor public pentru inițializarea numelui și o metodă publică *toString()* care returnează numele studentului.
- Definește o clasă *StudentBursier* derivată din *Student* având un atribut privat *valoareBursă* de tip întreg, un constructor public pentru inițializarea numelui și valorii bursei și de asemenea o metodă publică *toString()* suprascrisă care adaugă *valoareBursă* rezultatului returnat de metoda din clasa de bază.
- Definește o funcție care construiește un dicționar conținând următoarele: un obiect de tip *Student* având *nume* egal cu "Ionescu"; un obiect de tip *StudentBursier* având *nume* egal cu "Popescu" și *valoareBursă* egală cu 200. Cheia unui obiect din dicționar va fi *nume*, iar *valoarea* asociată cheii va fi obiectul.
- Definește o funcție care primește doi parametri, un dicționar de tipul celui descris la c) și un caracter și verifică dacă în dicționar există cel puțin un obiect care satisface proprietatea: *nume* începe cu caracterul dat.
- Construiește în funcția principală a programului dicționarul indicat la punctul c), apoi citește de la intrarea standard un șir de caractere *text* și afișează pe ieșirea standard obiectul din dicționar al cărui cheie este *text*. În cazul în care obiectul nu există în dicționar, se va afișa „nu există”. La final, verifică folosind funcția de la punctul d) dacă în dicționar există cel puțin un obiect al cărui *nume* începe cu primul caracter din *text* și afișează rezultatul.
- Pentru tipul de dată *dicționar* utilizat în program, scrieți specificațiile operațiilor folosite.

Puteți folosi biblioteci existente pentru structuri de date (Python, C++, Java, C#). În cazul în care nu folosiți biblioteci existente, specificați toate operațiile folosite.

Subiectul 2

- Să se evidențieze dependențele funcționale pentru următoarele informații dintr-o facultate:
 - burse oferite studenților:** id (unic), denumire, descriere, firma/instituția care oferă bursa (nume, pagina web), număr locuri oferite, număr luni de bursă, valoare lunară pentru bursă, lista studenților înscriși;
 - studenți:** cod numeric personal (unic), nume, grupa, anul de studiu, specializarea, nota de concurs pentru bursă (aceeași pentru toate bursele oferite), lista burselor la care concurează (în ordinea dorită);

Se cere o bază de date relațională cu tabele în 3NF pentru informațiile de mai sus.

Justificați că tabelele obținute sunt în 3NF.

Handwritten signature

Handwritten signature

b. Pentru baza de date de la punctul a, să se rezolve, folosind algebra relațională sau Select-SQL, următoarele interogări:

b1. Studenții (nume, specializare, nota de concurs) care solicită bursa cu id-ul "BM" și nu solicită bursa cu id-ul "BT".

b2. Pentru bursele la care s-au înscris cei mai mulți studenți se cere denumirea bursei, numele firmei, numărul de studenți înscriși și media notelor de concurs a acestora.

Subiectul 3

a) Se da secvența de cod de mai jos în limbajul C sub Linux. Programul s-a compilat sub numele p.

1	#include <unistd.h>	/home/scs/exam/p s ls
2	#include <stdio.h>	
3	int p[2];	
4	pid_t mypid;	/home/scs/exam/p`c pwd
5	main (int argc, char *argv[]) {	
6	pipe (p);	
7	if (fork () == 0) {	
8	close (p[0]);	
9	if (argv[1][0] == 'c') {	
10	execlp (argv[2], argv[2], 0);	
11	printf ("Exec finished\n");	
12	}	
13	mypid = getpid ();	
14	printf ("p=%d: pp=%d\n", mypid, getppid ());	
15	write (p[1], &mypid, sizeof (int));	
16	exit (0);	
17	}	
18	close (p[1]);	
19	read (p[0], &mypid, sizeof (int));	
20	printf ("Pid parinte %d: pid fiu:%d\n", getpid (), mypid);	
21	}	

a.1 Ce vor afișa pe ecran cele 2 rulări din coloana 3? Explicați rezultatele.

a.2 Ce vor afișa pe ecran cele 2 rulări dacă eliminăm linia 16 (exit(0)). Explicați rezultatele.

b) Se dă programul shell de mai jos:

echo \$#: \$*	./s.sh 1 2 3
p=`echo \$1 grep ^[^0-9]*[0-9]\$`	./s.sh 1 22 3
if ["\$p" != ""]	./s.sh a1 2
then	./s.sh 1 2a 3
shift	
./s.sh \$*	
fi	

b1. Explicați linia 2.

b2. Ce vor afișa pe ecran cele 4 execuții din coloana 2? Explicați rezultatele.

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

TimP de lucru: 3 ore.

[Signature]

[Signature]
[Signature]

BAREM

Subiect 1 (Algoritmica și Programare):

Oficiu – 1p

Definirea clasei *Student* – 0.75p din care

atribut – 0.25

constructor – 0.25

metoda *toString()* - 0.25

Definirea clasei *StudentBursier* – 1.5p din care

relația de moștenire – 0.25

constructor – 0.5

atribut – 0.25

metoda *toString()* – 0.5

Funcția de la punctul c) – 1p din care

signatura corectă și declarare dicționar - 0.25p

creare obiecte – 0.25p

adăugare obiecte - 0.25p

returnare rezultat - 0.25p

Funcția de la punctul d) – 1.75p din care

signatura corectă - 0.25p

parcursere dicționar – 1.00 p

verificarea condiției - 0.25p

returnare rezultat - 0.25p

Program – 2p din care

apel funcție construire dicționar – 0.25p

citire *text* – 0.25p

căutare obiect din dicționar cu cheie dată – 0.5p

afișare obiect (căutare cu succes) – 0.5p

afișare mesaj (căutare fără succes) – 0.25p

apelul funcției de la punctul d) – 0.25p

Specificațiile operațiilor folosite din tipul de dată *Dicționar* – 2p

Subiect 2 (Baze de date):

1 punct din oficiu

Problema a:

1 punct pentru dependențe,

2 puncte pentru tabele în 3NF;

1 punct pentru justificare.

Problema b:

2 puncte pentru b1;

Nu se punctează varianta de condiție: *bursa*='BM' and *bursa* <> 'BT'

0.5 puncte studenții cu bursa 'BM'

0.5 puncte studenții cu bursa 'BT'

1 punct instrucțiunea finală

3 puncte pentru b2.

1 punct: grupare, număr studenți, valoare medie

1 punct: determinare burse cu numărul maxim de studenți

1 punct: instrucțiunea finală

Subiect 3 (Sisteme de operare):

BAREM

Oficiu 1p

- a) 5p din care
1p identificare ramura executie
1p print dupa exec
1p executie exec + pwd
1p eliminate exit()
1p explicatii
- b) 4p din care
b1 explic grep 2p
b2 0.5 / rulare

Subiect 1 (Algoritmica si Programare):

Oficiu - 1p

Definirea clasei Student - 0.75p din care

atribut - 0.25

constructor - 0.25

metoda toString() - 0.25

Definirea clasei Student - 1.5p din care

relatia de mostenire - 0.25

constructor - 0.2

atribut - 0.25

metoda toString() - 0.2

Funcția de la punctul c) - 1p din care

signatura corectă și declarare dicționar - 0.25p

crearea obiect - 0.25p

adăugarea obiect - 0.25p

returnarea rezultat - 0.25p

Funcția de la punctul d) - 1.75p din care

signatura corectă - 0.25p

parcurgerea dicționar - 1.00p

verificarea condiției - 0.25p

returnarea rezultat - 0.25p

Program - 3p din care

apel funcție constructiv dicționar - 0.25p

citirea text - 0.25p

căutarea obiect din dicționar cu cheia dată - 0.25p

afișarea obiect (căutare cu succes) - 0.25p

afișarea mesaj (căutare fără succes) - 0.25p

apelul funcției de la punctul d) - 0.25p

Specificările operațiilor folosite din tipul de date Dicționar - 3p

Subiect 2 (Baze de date):

1 punct din oficiu

Problemă a:

1 punct pentru dependențe

2 puncte pentru tabel în 3NF

1 punct pentru justificare

Problemă b:

2 puncte pentru b1

Se se punctează varianta de condiție: bursa="BM" and bursa="BT"

0.5 puncte studenții cu bursa "BM"

0.5 puncte studenții cu bursa "BT"

1 punct instrucțiunea finală

3 puncte pentru b2

1 punct: grupare, număr studenți, valoare medie

1 punct: determinarea bursă cu numărul maxim de studenți

1 punct: instrucțiunea finală

[Handwritten signatures and marks]