



Bachelor Degree Exam, June 2015 Computer Science - English

Subject 1

Write a program in one of the programming languages Python, C++, Java, C# that:

- Defines a class** *Product* with a private field *price* of type integer, a public constructor that initializes the price, and a public method *sellingPrice()* that returns the *price* of the product.
- Defines a class** *PackagedProduct* derived from *Product* with a private field *packagePrice* of type integer, a public constructor that initializes the product *price* and *packagePrice*, and also an overridden public method *sellingPrice()* that returns the sum of the *packagePrice* and product *price*.
- Defines a function** that inserts an object of type *Product* into a *Product* list sorted descendingly by *selling price* (such that the list remains sorted).
- Using the function defined at (c), **defines a function** that returns a *Product* list sorted descendingly by selling price, containing the following instances: a *Product* having the *price* 100; a *PackagedProduct* having the *price* 70 and the *packagePrice* 10; and a *PackagedProduct* having the *price* 90 and the *packagePrice* 15.
- Defines a function** with a parameter of *Product* list type that returns the average of the selling prices.
- Calls in the **main function** of the program the function indicated at (d) to build the list, then it prints, on the standard output, the result returned by the function indicated at (e).
- For the *list* data type used in the program write the specifications of the used operations.

Remarks

- **Do not use sorted containers.**
- **Do not define other methods than those required in the subject.**
- **Do not use sorting methods.**

You can use existing libraries for lists (Python, C++, Java, C#). In case you do not use existing libraries, please specify only the operations from the interface of the list.

Subject 2

- Create a relational database having all tables in 3NF based on the following information about cake recipes:
 - **ingredients:** ingredient name, ingredient category (liquids, flavors, sweetener etc), measurement units (grams, liters, etc), number of calories per unit;
 - **cakes:** cake code, cake name, cake description, list of ingredients (with specific quantities for each ingredient);

Justify that the identified tables are in 3NF.

b. Given the database created at point a, express the following queries using SQL and at least once relational algebra:

- b1. List of all ingredients (ingredient name, category, measurement unit, quantity) for « Tiramisu »
- b2. Code and name of cakes that **contain** any kind of sweetener but **do not** contain vanilla flavor.
- b3. Code and name of cakes that have less than 500 calories.

Subject 3

3.1 Answer the following questions considering that in the program fragment below all instructions execute without error, and that the pipes are closed correctly.

<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>	<ul style="list-style-type: none"> (a) How many child processes will be created? (b) What will be written to the console? Explain the functioning of the program fragment. (c) Explain the functioning of the program fragment if the value of K is 2?
---	---

3.2 Consider the UNIX Shell script below.

- (a) Explain the script functioning
- (b) What will be written to the console?
- (c) Explain in detail line 8
- (d) Explain why the character „\” is needed on line 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>
--

Remarks: All subjects are compulsory. Each subject will be graded with a mark between 1 and 10 by both evaluators.

Time limit: 3 hours

BAREM
INFORMATICĂ

Subiect 1 (Algoritmică 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

algoritmul 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 "\