

# Arhiva pentru proba scrisa licenta

## Model 2021

- Operating systems (model): M1. M2.
- Databases (model): M3. M4. M5. M6.
- Programming (model): M7. M8. M9. M10.

## Interview 2020

- Operating systems (model)
  - Regex: I1. I2. I3. I4. I5.
  - Threads: I6. I7.
- Databases: I8. I9. I10.
- Programming
  - Algorithms: I11. I12. I13. I14. I15. I16. I17. I18. I19.
  - OOP: I20. I21. I22. I23. I24. I25. I26. I27.

## Archive

- Operating systems (model)
  - Regex:

|                          |                         |                         |                          |                          |                          |
|--------------------------|-------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
| <a href="#">↓ AP 1;</a>  | <a href="#">↓ AP 2;</a> | <a href="#">↓ AP 3;</a> | <a href="#">↓ AP 4;</a>  | <a href="#">↓ AP 5;</a>  | <a href="#">↓ AP 6;</a>  |
| <a href="#">↓ AP 7;</a>  | <a href="#">↓ AP 8;</a> | <a href="#">↓ AP 9;</a> | <a href="#">↓ AP 10;</a> | <a href="#">↓ AP 11;</a> | <a href="#">↓ AP 12;</a> |
| <a href="#">↓ AP 13;</a> |                         |                         |                          |                          |                          |
  - Threads:

|   |                          |                          |                          |                          |                          |
|---|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <a href="#">↓ AP 14;</a>                          | <a href="#">↓ AP 15;</a> | <a href="#">↓ AP 16;</a> | <a href="#">↓ AP 17;</a> | <a href="#">↓ AP 18;</a> | <a href="#">↓ AP 19;</a> |
| <a href="#">↓ AP 20;</a>                          | <a href="#">↓ AP 21;</a> | <a href="#">↓ AP 22;</a> | <a href="#">↓ AP 23;</a> | <a href="#">↓ AP 24;</a> | <a href="#">↓ AP 25;</a> |
| <a href="#">↓ AP 26;</a> <a href="#">↓ AP 27;</a> |                          |                          |                          |                          |                          |
- Databases:

|                          |                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <a href="#">↓ AP 28;</a> | <a href="#">↓ AP 29;</a> | <a href="#">↓ AP 30;</a> | <a href="#">↓ AP 31;</a> | <a href="#">↓ AP 32;</a> | <a href="#">↓ AP 33;</a> |
| <a href="#">↓ AP 34;</a> | <a href="#">↓ AP 35;</a> | <a href="#">↓ AP 36;</a> | <a href="#">↓ AP 37;</a> | <a href="#">↓ AP 38;</a> | <a href="#">↓ AP 39;</a> |
| <a href="#">↓ AP 40;</a> | <a href="#">↓ AP 41;</a> | <a href="#">↓ AP 42;</a> | <a href="#">↓ AP 43;</a> | <a href="#">↓ AP 44;</a> | <a href="#">↓ AP 45</a>  |
| <a href="#">↓ AP 46;</a> | <a href="#">↓ AP 47;</a> | <a href="#">↓ AP 48;</a> |                          |                          |                          |
- Programming
  - Algorithms:

|                          |                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <a href="#">↓ AP 49;</a> | <a href="#">↓ AP 50;</a> | <a href="#">↓ AP 51;</a> | <a href="#">↓ AP 52;</a> | <a href="#">↓ AP 53;</a> | <a href="#">↓ AP 54;</a> |
| <a href="#">↓ AP 55;</a> | <a href="#">↓ AP 56;</a> | <a href="#">↓ AP 57;</a> |                          |                          |                          |
  - OOP:

|                          |                          |                          |                          |                          |                         |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------|
| <a href="#">↓ AP 58;</a> | <a href="#">↓ AP 59;</a> | <a href="#">↓ AP 60;</a> | <a href="#">↓ AP 61;</a> | <a href="#">↓ AP 62;</a> | <a href="#">↓ AP 63</a> |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------|

## Model POCU Licenta

- |                           |                           |                           |                           |                           |                           |                           |
|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| <a href="#">↓POCU 1;</a>  | <a href="#">↓POCU 2;</a>  | <a href="#">↓POCU 3;</a>  | <a href="#">↓POCU 4;</a>  | <a href="#">↓POCU 5;</a>  | <a href="#">↓POCU 6;</a>  | <a href="#">↓POCU 7;</a>  |
| <a href="#">↓POCU 8;</a>  | <a href="#">↓POCU 9;</a>  | <a href="#">↓POCU 10;</a> | <a href="#">↓POCU 11;</a> | <a href="#">↓POCU 12;</a> | <a href="#">↓POCU 13;</a> | <a href="#">↓POCU 14;</a> |
| <a href="#">↓POCU 15;</a> | <a href="#">↓POCU 16;</a> | <a href="#">↓POCU 17;</a> | <a href="#">↓POCU 18;</a> | <a href="#">↓POCU 19;</a> | <a href="#">↓POCU 20;</a> | <a href="#">↓POCU 21;</a> |

# Model 2021

The evaluation of your fundamental and specialty knowledge will be done through an oral exam which will consist of:

- **6 problems (with open-ended answers):**
  - 2 from the field of **Operating Systems**
  - 2 from the field of **Databases**
  - 2 from the field of **Algorithms and Programming**
    - 1 pertaining to **Algorithms**
    - 1 pertaining to **Programming**
- The time for solving the 6 problems: **20 minutes**
- The time for presenting your answer: **10 minutes**
- **Scoring** will be done as follows:
  - one score (1-10) for each field (including 1 point by default)
  - the final score will be the arithmetic mean of the scores for each of the 3 fields.

## Note

1. For each question there will be a score for the correct answer and a score for justifying the answer.
2. The code samples for **Algorithms and Programming** are given and required to be in the C++ programming language.

## Operating Systems (2021)

### M1.

Establish the truth value of the statements below regarding the command given. Justify your answer.

`grep -E -i '^[aeiou]{2,3}.*[^0-9]$' a.txt`

- i. Displays from a.txt only the lines that start with “a” or “A”.
- ii. It will not display from a.txt any line shorter than 3 characters.

### M2.

Establish the truth value of the statements below regarding the given code fragment. Justify your answer.

```
1 if (fork() == 0) {  
2     fork();  
3     fork();  
}
```

- i. Line 2 is executed by 2 processes.
- ii. The IF condition is only evaluated by the parent process.

## Databases (2021)

### M3.

Consider a relation with schema R[Cod1, Cod2, Cod3, C1, C2, C3, C4, C5] and the instance below. {Cod1, Cod2, Cod3} is the primary key.

| Cod1 | Cod2 | Cod3 | C1 | C2 | C3 | C4 | C5 |
|------|------|------|----|----|----|----|----|
| 1    | 1    | 1    | 10 | 5  | 2  | 10 | 1  |
| 1    | 1    | 2    | 10 | 6  | 3  | 20 | 1  |
| 1    | 1    | 3    | 20 | 7  | 2  | 15 | 2  |
| 2    | 1    | 1    | 20 | 1  | 4  | 20 | 1  |
| 2    | 1    | 2    | 30 | 2  | 5  | 10 | 1  |
| 2    | 2    | 1    | 30 | 3  | 5  | 30 | 2  |
| 3    | 1    | 1    | 40 | 4  | 5  | 25 | 2  |
| 3    | 1    | 2    | 40 | 4  | 5  | 20 | 1  |

Compute the difference between the cardinality of the first query's result set and the cardinality of the second query's result set. Justify your answer.

```
SELECT C1, COUNT(*), MIN(C4)
FROM R
WHERE C3 IN (2, 5)
GROUP BY C1, C2
```

```
SELECT Cod1, COUNT(*), MAX(C5)
FROM R
GROUP BY Cod1
HAVING COUNT(*) > 2
```

### M4.

Let  $\{A, B\} \rightarrow \{D, E\}$  and  $\{CodP, B\} \rightarrow \{D\}$  be two functional dependencies. For each dependency, state whether or not it's satisfied by the data in the relation below. Justify your answer.

| CodP | CodQ | A | B | C | D | E |
|------|------|---|---|---|---|---|
| 1    | 1    | 1 | 1 | 9 | 2 | 2 |
| 1    | 2    | 2 | 2 | 6 | 3 | 1 |
| 1    | 3    | 2 | 2 | 4 | 3 | 1 |
| 2    | 1    | 3 | 3 | 9 | 4 | 5 |
| 2    | 2    | 4 | 4 | 3 | 2 | 3 |
| 3    | 1    | 1 | 1 | 5 | 2 | 3 |
| 3    | 2    | 4 | 3 | 3 | 1 | 2 |

## M5.

[Model 2020] Se dă mai jos instanța unei relații cu schema S[FK1, FK2, A, B, C, D, E].

{FK1, FK2} este cheie primară.

| <b>FK1</b> | <b>FK2</b> | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |
|------------|------------|----------|----------|----------|----------|----------|
| 1          | 1          | a1       | b1       | c1       | 7        | 2        |
| 1          | 2          | a        | b3       | c1       | 5        | 2        |
| 1          | 3          | a2       | b1       | c2       | 1        | 2        |
| 2          | 1          | a3       | b3       | c2       | 7        | 100      |
| 2          | 2          | a3       | b3       | c3       | 5        | 100      |

Cât este diferența între cardinalitatea rezultatului primei interogări și cardinalitatea rezultatului celei de a doua interogări?

```
SELECT FK2, FK1, COUNT(DISTINCT B)
FROM S
GROUP BY FK2, FK1
HAVING FK1 = 1
```

```
SELECT FK2, FK1, COUNT(C)
FROM S
GROUP BY FK2, FK1
HAVING FK1 = 2
```

## M6.

[Model 2020] Se dau următoarele relații. Cheile primare sunt subliniate și cheile străine sunt scrise cursiv și au aceeași denumire:

Persoane (PersoanId, Nume, Strada, CodPostal, Oras)

Masini (MasinaId, An, Model)

Proprietar (PersoanId, MasinaId)

Accident (ProcesVerbalId, Daune, Detalii)

ParticipantiAccident (PersoanId, ProcesVerbalId, MasinaId)

Care din următoarele interogări returnează toate perechile (**Nume**, **ProcesVerbalId**) cu proprietatea ca persoana respectivă (**Nume**) a făcut parte din accidentul descris în **ProcesVerbalId**, dar cu o mașină care are un alt proprietar?

a)

```
SELECT P.Nume, T.ProcesVerbalId  
FROM Persoane P INNER JOIN ParticipantAccident T  
ON P.PersoanaId = T.PersoanaId  
INNER JOIN Proprietar A ON A.PersoanaId = T.PersoanaId
```

b)

```
SELECT P.Nume, T.ProcesVerbalId  
FROM Persoane P INNER JOIN ParticipantAccident T  
    ON P.PersoanaId = T.PersoanaId  
WHERE P.PersoanaId NOT IN  
    (SELECT A.PersoanaId FROM Proprietar A)
```

c)

```
SELECT P.Nume, T.ProcesVerbalId  
FROM Persoane P INNER JOIN ParticipantAccident T  
    ON P.PersoanaId = T.PersoanaId  
INNER JOIN Proprietar A ON A.MasinaId = T.MasinaId  
WHERE A.PersoanaId != P.PersoanaId
```

d)

```
SELECT P.Nume, T.ProcesVerbalId  
FROM Persoane P INNER JOIN ParticipantAccident T  
    ON P.PersoanaId = T.PersoanaId  
WHERE EXISTS (SELECT A.PersoanaId  
    FROM Proprietar A  
    WHERE A.MasinaId = T.MasinaId  
    AND A.PersoanaId != T.PersoanaId)
```

e) nicio variantă de mai sus nu este corectă.

## Programming (2021)

M7.

[ALGORITHMS] What does the recursive algorithm below compute, if called as  $Computation(x, n, 1, 1, 0)$ , given that  $x[1], x[2], \dots, x[n]$  is a sequence of natural numbers? Justify your answer.

```
Function Computation (x, n, i, y, j) is
    If n < i then
        If j = n then
            Computation ← 0
        else
            Computation ← y * (n - j)
        EndIf else
        If (x[i] ≥ 10) and ((x[i] mod 100) mod 11 = 0) then
            Computation ← Computation(x, n, i + 1, y * x[i], j)
        else
            Computation ← Computation(x, n, i + 1, y, j + 1)
        EndIf
    EndIf
```

M8.

[PROGRAMMING] What is the effect of the compilation and execution of the following program? Justify your answer.

```
#include <iostream>
using namespace std;
class Vehicle{
protected:
    double speed;
public:
    Vehicle(double ms) { speed=ms; }
    virtual void go(){ cout<<"Vehicle going ..."<<endl; }
    void accelerate(){ cout<<"Vehicle accelerating..."<<endl; }
};

class Scooter:public Vehicle{
public:
    Scooter(double s){ speed=s; }
    void go(){ cout<<"Scooter going ..."<<endl; }
    void accelerate(int s) {
        cout<< "Accelerating with "<< s << "..." << endl;
    }
};

int main(){
    Scooter s(1);
    Vehicle& v=s;
    v.go();
    v.accelerate();
    return 0;
}
```

**M9.**

[Model 2020] Ce calculează subalgoritmul recursiv de mai jos, la apelul  $F(x, n, 1, 0, 0)$ , în condițiile în care  $x[1], x[2], \dots, x[n]$  este un sir de numere întregi?

```

Function F(x, n, i, S, k) is
  If (n < i) then
    If k = n then
      F  $\leftarrow$  0
    Else
      F  $\leftarrow S / (n - k)$ 
    EndIf
  else
    If x[i] mod 2 = 0 then
      F  $\leftarrow F(x, n, i+1, S + x[i], k)$ 
    Else
      F  $\leftarrow F(x, n, i+1, S, k + 1)$ 
    EndIf
  EndIf
EndFunction

```

**M10.**

[Model 2020] Ce se afișează în urma execuției următorului program?

|   |  |
|---|--|
| <pre> #include &lt;iostream&gt; using namespace std;  class B { public:   B() {     cout &lt;&lt; "B() ";   }    virtual void f() { cout &lt;&lt; "B.f "; }    void g() {     f();   }    virtual void h() {     g();   } }; </pre> | <pre> class D : public B { public:   D() { cout &lt;&lt; "D() "; }   void f() { cout &lt;&lt; "D.f "; }   void h() {     cout &lt;&lt; "Function h ";     B::h();   } };  int main() {   B b{};   D d{};   B&amp; dd = d;    b.g();   dd.h();   return 0; } </pre> |
|---|--|

## Interview 2020

In 2020, after the first problems there were a few questions asked by the comission. These are **important!**

### Operating systems (interview)

I1.

Ce va afișa comanda de mai jos? Justificați răspunsul

echo "UUID=34" | grep -i "[aeiou]{2,3}[^0-9]"

I2.

Ce trebuie modificat la grepul de mai sus ca să facă match pe toate grupurile de lungime pară de caractere hexadecimale?

`grep -i "[A-F0-9]{2,}\{([A-Z0-9]{2,})\}*\$" b.txt`

I3.

Ce face următoarea comandă?

`exec(/bin/bash/ls, bin/bash/ls, /home, NULL)`

I4.

Ce face \{1, } în sed-ul de la grile?

`sed "/^#\{1,\}/d"`

I5.

Face match la numere binare apoi la linii care contin adunări de numere? Ce rol are \{\} ?

`[01]\{1,\}[+*/-]\{1,\}`

I6.

Câte procese noi (în afara părintelui) creează fragmentul de cod de mai jos? Justificați răspunsul.

```
1 if(fork() == 0) {  
2     fork();  
3     fork();  
4 }
```

I7.

Un for cu fork() și apoi while(wait(NULL)>0) print(ceva); . Câte procese s-au facut și printează ceva?

## Databases (interview)

I8.

Define second normal form.

I9.

Dependenta funcțională  $\{C1, C2\} \rightarrow \{C4\}$  este satisfăcută de datele din relația de mai jos? Argumentați răspunsul.

| CodA | C1 | C2 | C3 | C4 |
|------|----|----|----|----|
| a1   | 1  | 1  | 0  | 4  |
| a2   | 1  | 2  | 2  | 3  |
| a3   | 1  | 2  | 5  | 3  |
| a4   | 1  | 3  | 1  | 3  |
| a5   | 1  | 1  | 2  | 4  |

I10.

Incomplete

- a) Se da o relație, cu o cheie multiplă, și 3 dependințe funcționale. Care trebuie eliminate să fie 3NF?
- b) 3 scheme, care e 3nf?
- c) O tabela cu dependințe - 1NF, 2NF, 3NF, none.
- d) Numar de studenți înrolați la curs (inner join cu count distinct)

## Programming (interview)

Algorithms

I11.

From the algorithms below, which one has the best-case time complexity greater than the best-case time complexity of BubbleSort?

- a) **Merge** sort      b) **Insertion** sort      c) **Selection** sort      d) **Quick** sort

I12.

Dacă se da un vector de numere amestecate care au și dubluri, ce e mai eficient dpdV al timpului pentru a găsi un număr?

I13.

Să faci căutare secvențială sau sortare cu cel mai eficient algoritm și apoi binary search.

I14.

Function which checks palindrome number. Does it return 1 or 0 for a number multiple of 10.

I15.

Functie care returna true daca elementul era egal cu maxima valoare.

I16.

Dati un exemplu pentru care algoritmul QuickSort ajunge la cel mai rau caz din punct de vedere al timpului de executie.

I17.

Care este complexitatea în cel mai rău caz a algoritmului de sortare a unui sir folosind metoda sortării rapide (Quicksort)?

I18.

Give an example vector for which insertion sort will reach its best-case time complexity.

I19.

Care este complexitatea in caz defavorabil a sortarii prin insertie a unui vector [x1, x2... xn]?

OOP

I20.

In C++, what should we use when we assign to an object variable of the base-class type an object variable of a derived-class type, so as to avoid losing what has been added in the derived class?  
~Ce trebuie sa folosim atunci cand atribuim unei variabile obiect de tipul clasei de baza o variabila obiect de tipul clasei derivate pentru a nu pierde partea adaugata de clasa derivata?

I21.

What is an abstract class?

I22.

Ce rol are o clasa friend?

I23.

Ce e un copy constructor?

I24.

Cum se modela o asociere in care avem 1 multiplicate si 0..\* ?

I25.

Care sunt pașii efectuați la crearea unui obiect dintr-o clasă derivată?

I26.

O diagrama UML. Explica elementele (clase, functii, atribute, access modifiers, associations, multipliers)

I27.

O diagrama UML sa zic daca se respecta.

# Archive

## Operating systems

Regex

↓ AP 1

Explain this sequence `\{1,\}` that is part of the sed expression found below.

```
sed '/^#\{1,\}.*/d' d.sh
```

↓ AP 2

grep -i "^[A-F0-9]\{2,\}\{([A-Z0-9]\{2,\})\}\*\$" b.txt (cv de genu)

1. Intrebare: face match pe orice linie care incepe cu "A"?
2. Intrebare: face match pe toate caracterele hexazecimale de lungime 2?

↓ AP 3

Determine the truth value of the following statements about the command below, considering that it executes without errors, the `d.sh` file being a Unix Shell script.

```
sed '/^#\{1,\}.*/d' d.sh
```

1. It displays the content of the `d.sh` file, replacing the lines which are commented with `d`
2. It displays all the lines from `d.sh` that are not commented

↓ AP 4

Cum ați modifica expresia `sed` de mai jos încât să ștergă toate liniile goale din `d.sh`.

```
sed '/^#\{1,\}.*/d' d.sh
```

↓ AP 5

How would you change the following regular expression so that it will accept only increases and decreases between numbers represented in base 2 less than 8?

```
grep "^[01]\{1,\}[+/-][01]\{1,\}$" a.txt
```

## ↓ AP 6

Stabiliti valoarea de adevar a urmatoarelor afirmații despre comanda de mai jos, considerand că se execută fără erori.

```
sed '/^.*#\{1,\}/d' d.sh
```

1. Afisează conținutul fișierului d.sh ștergând toate liniile care conțin caracterul #
2. Afisează conținutul fișierului d.sh ștergând, printre altele, toate liniile care încep cu caracterul #

## ↓ AP 7

SO

1. Sed "/^#\{1,\}/d"
  - a. Daca inlocuieste toate liniile care incep cu # cu d
  - b. Daca sterge toate liniile comentate din fisierul bash

## ↓ AP 8

How would you change the following regular expression so that it will accept only increases and decreases between numbers represented in base 2 less than 8?

```
grep "^[01]\{1,\}[+*/-][01]\{1,\}$" a.txt
```

## ↓ AP 9

Determine the truth value of the following statements about the below command, considering that it executes without errors.

```
grep "^[01]\{1,\}[+*/-][01]\{1,\}$" a.txt
```

1. It displays all the lines from a.txt that contain numbers in base 2 greater than 8
2. It displays only those lines from a.txt that contain multiplications of 2 natural numbers in base 2

## ↓ AP 10

Stabiliți valoarea de adevăr a următoarelor afirmații despre comanda de mai jos, considerând că se execută fără erori, fișierul d.sh fiind un script Shell Unix.

```
sed '/^#\{1,\}.*/d' d.sh
```

1. Afisează conținutul fișierului d.sh înlocuind liniile comentate cu d
2. Afisează toate liniile din d.sh care nu sunt comentate

## ↓ AP 11

Stabiliți valoarea de adevăr a următoarelor afirmații despre comanda de mai jos, considerând că se execută fără erori.

```
grep -i "^[A-F0-9]\{2\}\([A-F0-9]\{2\}\)*$" b.txt
```

1. Afisează toate liniile din b.txt care încep cu A
2. Afisează, printre altele, toate liniile din b.txt care conțin doar 2 cifre în baza 16

## ↓ AP 12

Stabiliți valoarea de adevăr a următoarelor afirmații despre comanda de mai jos, considerând că se execută fără erori.

```
sed 's/-[0-9][0-9]*\.[0-9][0-9][0-9]*/###/g' c.txt
```

1. Afisează conținutul fișierului c.txt înlocuind cu ### toate numerele reale negative având cel puțin două zecimale.
2. Afisează conținutul fișierului c.txt înlocuind cu ### toate liniile care conțin numere reale negative având cel puțin două zecimale

## ↓ AP 13

Cum trebuie modificată expresia regulară încât să accepte orice linie care conține un număr în baza 16 cu număr par de cifre?

```
grep -i "^[A-F0-9]\{2\}\([A-F0-9]\{2\}\)*$" b.txt
```

## Threads

↓ AP 14

Determine the truth value of the following statements, considering that the code snippet below is executed without errors.

```
1 for (i=0;i<4;i++){
2     fork();
3 }
4 while(wait(NULL)>0);
5 printf("The created processes are
finished.\n");
```

1. There will be a total of 8 processes.
2. None of the processes will print anything.

↓ AP 15

```
For (int i = 0; i < 4; i++) {
    fork();
}
```

```
While (wait(NULL)) {}
Printf ("Whatever");
```

1. Intrebare: sunt tot timpul mai putin de 4 procese?
2. Intrebare: se executa linia cu printf in parinte dupa ce se executa in copii?

↓ AP 16

```
For x
Do
    echo $x
Done
```

```
For x in $*
Do
    echo x
Done
```

1. Afiseaza ac lucru
2. Al 2 lea afiseaza fisiere si directoarele din doc curent?

↓ AP 17

```
fork()
exec(/bin/bash/, bin/bash, -c, ls, NULL)
1. Copilu nu face nmk
2. Parintele afiseaza de 2 ori ls
```

## ↓ AP 18

What will the code snippet below display, assuming it runs without errors? Explain your answer.

```
1 if (fork() < 0){  
2     execl("/bin/bash", "/bin/bash", "-c", "ls", NULL);  
3 }
```

## ↓ AP 19

Determine the truth value of the following statements, considering that the code snippet below is executed without errors.

```
1 for (i=0;i<3;i++){  
2     fork();  
3 }  
4 wait(NULL);  
5 printf("The created processes are  
finished.\n");
```

1. Line 5 will be executed by all processes.
2. Each process will enter the “wait” state at the execution of line 4.

## ↓ AP 20

2. for(i = 0; i < 3; i++)  
 fork()  
 wait(NULL)  
 printf("ceva")  
 a. Daca toate procesele intra in waiting state  
 b. Daca toate procesele executa linia cu printf

## ↓ AP 21

Determine the truth value of the following statements, considering that the code snippet below is executed without errors.

```
1 for (i=0;i<3;i++){  
2     fork();  
3 }  
4 while(wait(NULL));  
5 printf("The created processes are  
finished.\n");
```

1. Only the parent process will execute line 5.
2. Each process will enter the “wait” state at the execution of line 4.

## ↓ AP 22

Stabilită valoarea de adevăr a următoarelor afirmații, considerând că fragmentul de cod de mai jos se execută fără erori.

|  |   |
|--|---|
| <pre> 1 for (i=0;i&lt;3;i++) { 2     fork(); 3 } 4 wait(NULL); 5 printf("S-au terminat procesele create.\n"); </pre> | 1. Vor fi în total 3 procese<br><br>2. Linia 5 se va executa în procesul inițial după ce se termină oricare din procesele fiu |
|--|---|

## ↓ AP 23

Stabilită valoarea de adevăr a următoarelor afirmații, considerând că fragmentul de cod de mai jos se execută fără erori.

|  |  |
|--|--|
| <pre> 1 for (i=0;i&lt;3;i++) { 2     fork(); 3 } 4 wait(NULL); 5 printf("S-au terminat procesele create.\n"); </pre> | 1. Vor fi în total 8 procese<br><br>2. Linia 5 se va executa în procesul inițial după ce s-au terminat toate procesele create. |
|--|--|

## ↓ AP 24

Stabilită valoarea de adevăr a următoarelor afirmații, considerând că fragmentul de cod de mai jos se execută fără erori.

|  |   |
|--|---|
| <pre> 1 for (i=0;i&lt;4;i++) { 2     fork(); 3 } 4 while(wait(NULL)&gt;0); 5 printf("S-au terminat procesele create.\n"); </pre> | 1. Vor fi în total 8 procese.<br><br>2. Niciun proces nu va tipări nimic. |
|--|---|

## ↓ AP 25

Stabilită valoarea de adevăr a următoarelor afirmații, considerând că fragmentul de cod de mai jos se execută fără erori.

|  |  |
|--|--|
| <pre> 1 for (i=0;i&lt;4;i++) { 2     fork(); 3 } 4 while(wait(NULL)&gt;0); 5 printf("S-au terminat procesele create.\n"); </pre> | 1. Vor fi în total 16 procese.<br><br>2. Linia 5 se va executa în procesul inițial după ce s-au terminat toate procesele create. |
|--|--|

↓ AP 26

Explicați condițiile din scriptul Shell Unix de mai jos

```
if [ $# -ge 1 ]; then
    if [ -e $1 ]; then
        file $1
    fi
fi
```

↓ AP 27

|   |  |   |
|---|--|---|
| 1 | for (i=0;i<3;i++) {                          | 1. 4 procese nu vor executa linia 2.  |
| 2 | fork();                                      |   |
| 3 | }  | 2. Procesul părinte se poate termina<br>înainte de terminarea tuturor proceselor<br>fii create. |
| 4 | while(wait(NULL));                           |   |
| 5 | printf("S-au terminat procesele create.\n"); |   |

## Databases

↓ AP 28

Consider an instance of the following relation:

Magazine(CodM, NumeMagazin, Adresa, Localitate).

The primary key is underlined.

| CodM | NumeMagazin | Adresa          | Localitate  |
|------|-------------|-----------------|-------------|
| 1    | Florelia    | Strada 1 nr. 5  | Oradea      |
| 2    | ABC         | Strada 5 nr. 4  | Cluj-Napoca |
| 3    | Kaufland    | Strada 6 nr. 10 | Craiova     |
| 4    | Lidl        | Strada 7 nr. 7  | Sibiu       |

How many tuples and columns contains the table obtained from running the query below?

```
SELECT NumeMagazin
FROM Magazine
WHERE Localitate IN ('Cluj-Napoca', 'Craiova')
INTERSECT
SELECT NumeMagazin
FROM Magazine
WHERE Localitate NOT IN ('Oradea', 'Craiova')
```

**↓ AP 29****VIII.**

Se dă mai jos instanța unei relații cu schema S[FK<sub>1</sub>, FK<sub>2</sub>, A, B, C, D, E] și cheia {FK<sub>1</sub>, FK<sub>2</sub>}.

Răspundeți întrebărilor următoare:

| FK1 | FK2 | A  | B  | C  | D    | E   |
|-----|-----|----|----|----|------|-----|
| 1   | 1   | a1 | b1 | c1 | 7    | 2   |
| 1   | 2   | a_ | b3 | c1 | 5    | 2   |
| 1   | 3   | a2 | b1 | c2 | null | 2   |
| 2   | 1   | a3 | b3 | c2 | null | 100 |
| 2   | 2   | a3 | b3 | c3 | null | 100 |

A. Câte înregistrări va furniza interogarea:

```
SELECT *
  FROM S
 WHERE A LIKE 'a_'
```

## ↓ AP 30

Following you have the instances for two relations with the schemas:

- Profesori[Marca, Nume, Departament], {Marca} is the primary key.
- Cursuri[Cod, Marca, Denumire, AnStudiu], {Cod} is primary key; column Marca has a NOT NULL constraint; {Marca} is external key to {Marca} from the relation Profesori.

### Profesori

| Marca | Nume           | Departament |
|-------|----------------|-------------|
| C102  | Pop Ion        | Matematica  |
| C103  | Ionescu Mircea | Informatica |
| C104  | Lucescu Ana    | Informatica |

### Cursuri

| Cod   | Marca | Denumire        | AnStudiu |
|-------|-------|-----------------|----------|
| AM011 | C102  | Algebra         | 1        |
| AM021 | C102  | Analiza         | 2        |
| PI011 | C103  | Programare      | 1        |
| BI021 | C103  | Baze de Date    | 2        |
| SI012 | C103  | Sisteme Operare | 1        |

What is the result for the next interrogation?

$$(\Pi_{\{Profesori.Nume, Cursuri.Marcă, Cursuri.Denumire\}}(Profesori \triangleright_{Profesori.Marcă=Cursuri.Marcă} Cursuri)) - (\Pi_{\{Profesori.Nume, Cursuri.Marcă, Cursuri.Denumire\}}(Profesori \otimes_{Profesori.Marcă=Cursuri.Marcă} Cursuri))$$

Nominate also the attributes' names..

Note: consider that the projection operator eliminates duplicates.

The algebraic operators used in the interrogation are:

Projection:  $\Pi_a$

Difference: -

Conditional Join:  $\otimes_c$

External left Join:  $\triangleright_c$

**↓ AP 31**

Se dă instanța unei relații cu schema N[CodN, A, B, C, D, E]. {CodN} este cheie primară.

| CodN | A  | B   | C    | D  | E  |
|------|----|-----|------|----|----|
| 1    | a1 | 10  | 4000 | d3 | e1 |
| 2    | a4 | 70  | 450  | d3 | e4 |
| 3    | a4 | 35  | 900  | d2 | e1 |
| 4    | a2 | 30  | 130  | d2 | e2 |
| 5    | a3 | 100 | 20   | d1 | e3 |

Câte tupluri și câte coloane are tabelul obținut în urma execuției interogării de mai jos?

```
SELECT B
FROM N
WHERE B<=50
EXCEPT
SELECT C
FROM N
WHERE C<300
```

**↓ AP 32**

|   |    |    |    |    |
|---|----|----|----|----|
| 2 | a1 | b1 | c2 | d2 |
| 3 | a2 | b2 | c2 | d2 |
| 4 | a3 | b3 | c2 | d3 |
| 5 | a3 | b4 | c3 | d4 |

Cât este diferența dintre cardinalitatea rezultatului primei interogări și cardinalitatea rezultatului celei de-a doua interogări?

```
SELECT T2.PK, T2.A, T2.B, T2.C, T2.D  
FROM P T1  
    INNER JOIN P T2 ON T1.PK=T2.PK
```

și

```
SELECT T1.PK, T1.A, T1.B, T1.C, T1.D  
FROM P T1  
    LEFT JOIN P T2 ON T1.PK=T2.PK AND T2.D<>'d2'  
    INNER JOIN P T3 ON T1.PK=T3.PK and T2.C<>T3.C
```

- a. 0
- b. 1
- c. 5
- d. 2
- e. Nicio variantă de mai sus nu este corectă.

## ↓ AP 33

Given the relation  $R(A, B, C, D)$  with the following functional dependencies  $F = \{ A \rightarrow BC, B \rightarrow D \}$ . The primary key is underlined, and the  $R$  relation does not contain repetitive attributes.

Which of the following statements is/are true:

- $R$  is in the first normal form.
- $R$  is in the second normal form.
- $R$  is in the third normal form.
- $R$  is in the Boyce Codd normal form.
- None of the above is correct.

| CodR | A | B | C   | D | E |
|------|---|---|-----|---|---|
| 1    | 1 | 1 | c1  | 2 | 2 |
| 2    | 1 | 1 | c2  | 2 | 2 |
| 3    | 2 | 2 | c2  | 2 | 2 |
| 4    | 3 | 4 | c2  | 4 | 5 |
| 5    | 3 | 4 | c6  | 4 | 5 |
| 6    | 4 | 5 | c12 | 5 | 4 |

How many tuples are returned by the following query?

```
SELECT DISTINCT r2.A, COUNT(*)
FROM R r1 INNER JOIN R r2 ON r1.CodR = r2.CodR
    INNER JOIN R r3 ON r2.CodR = r3.CodR
WHERE r1.A <> r2.A
GROUP BY r1.A, r2.A
HAVING r2.A >= 1
```



BD

- Un tabel cu CorD ca și PK și un SELECT care facea inner join la același tabel și după inca un INNER JOIN tot pe sine pe  $A2.CorD = A3.CorD$ , cu un group by și un where  $A2.A <> A3.A$  și trebuia să aleg cate linii se returneaza
- O relație  $\{A, B, C, D\}$  unde A era cheie și dependințele  $A \rightarrow BC$  și  $B \rightarrow D$  și să zic ce NF e

**↓ AP 34**

Consider the instance of a relation with the schema P[PK, A, B, C, D]. {PK} is the primary key.

P

| PK | A  | B  | C  | D  |
|----|----|----|----|----|
| 1  | a1 | b1 | c1 | d1 |
| 2  | a1 | b1 | c2 | d2 |
| 3  | a2 | b2 | c2 | d2 |
| 4  | a3 | b3 | c2 | d3 |
| 5  | a3 | b4 | c3 | d4 |

How many tuples and how many columns are in the natural join of the projections P1[PK, A, C] and P2[B, C]?

Note: consider that the projection operator eliminates duplicates.

The columns with the same name appear one single time in the results for the natural join.

## ↓ AP 35

Following you have the instances for two relations with the schemas:

- Profesori[Marca, Nume, Departament], {Marca} is the primary key.
- Cursuri[Cod, Marca, Denumire, AnStudiu], {Cod} is primary key; column Marca has a NOT NULL constraint; {Marca} is external key to {Marca} from the relation Profesori.

**Profesori**

| Marca | Nume           | Departament |
|-------|----------------|-------------|
| C102  | Pop Ion        | Matematica  |
| C103  | Ionescu Mircea | Informatica |
| C104  | Lucescu Ana    | Informatica |

**Cursuri**

| Cod   | Marca | Denumire        | AnStudiu |
|-------|-------|-----------------|----------|
| AM011 | C102  | Algebra         | 1        |
| AM021 | C102  | Analiza         | 2        |
| PI011 | C103  | Programare      | 1        |
| BI021 | C103  | Baze de Date    | 2        |
| SI012 | C103  | Sisteme Operare | 1        |

What is the result for the next interrogation?

$$(\Pi_{\{Profesori.Nume, Cursuri.Marca, Cursuri.Denumire\}}(Profesori \triangleright_{Profesori.Marca=Cursuri.Marca} Cursuri)) - (\Pi_{\{Profesori.Nume, Cursuri.Marca, Cursuri.Denumire\}}(Profesori \otimes_{Profesori.Marca=Cursuri.Marca} Cursuri))$$

Nominate also the attributes' names..

Note: consider that the projection operator eliminates duplicates.

The algebraic operators used in the interrogation are:

Projection:  $\Pi_\alpha$

Difference:  $-$

Conditional Join:  $\otimes_c$

External left Join:  $\triangleright_c$

## ↓ AP 36

Consider the relational schemas:

- Profesori[Marca, Nume, Departament], {Marca} is the primary key.
- Cursuri[Cod, Marca, Denumire, AnStudiu], {Cod} is the primary key; column Marca has a NOT NULL constraint; {Marca} is external key to {Marca} from relation Profesori.

Query I:

```
SELECT P.Marcă
FROM Profesori P
    INNER JOIN Cursuri C ON P.Marcă=C.Marcă
GROUP BY P.Marcă
HAVING COUNT(*)=2
```

Select which ones from the below queries returns the same result as Query I:

a. SELECT \* FROM

```
(  
    SELECT DISTINCT Marca  
    FROM Cursuri  
    UNION  
    SELECT DISTINCT Marca  
    FROM Cursuri  
) T
```

b. SELECT \*

```
FROM  
(  
    SELECT DISTINCT Marca  
    FROM Cursuri  
    UNION ALL  
    SELECT DISTINCT Marca  
    FROM Cursuri  
) T
```

c. SELECT DISTINCT Marca

```
FROM Cursuri  
INTERSECT  
SELECT DISTINCT  
Marca FROM Cursuri
```

d. SELECT DISTINCT C1.Marcă

```
FROM Cursuri C1  
WHERE EXISTS  
(  
    SELECT C2.Marcă  
    FROM Cursuri C2  
    WHERE C2.Marcă=C1.Marcă  
    GROUP BY C2.Marcă  
    HAVING COUNT(*)=2  
)
```

e. None of these options returns the same result as Query I.

**↓ AP 37**

The instance of the following relation is given:

Cabinet[MarcaAngajat, NumeDoctor, NumarPacient, NumePacient, DataProgramare, CodInterventie], without repetitive groups (of attributes). {MarcaAngajat, DataProgramare} is the primary key of the relation. The relation has no secondary keys.

| MarcaAngajat | NumeDoctor | NumarPacient | NumePacient   | DataProgramare           | CodInterventie |
|--------------|------------|--------------|---------------|--------------------------|----------------|
| A10          | Dan Ion    | P111         | Maria Dumitru | 1 Iulie 2020 10:00       | I5             |
| A10          | Dan Ion    | P115         | Tudor Rusu    | 2 August 2020 09:15      | I3             |
| A11          | Ana Cuc    | P112         | Lucian Pop    | 12 August 2020 08:20     | I5             |
| A11          | Ana Cuc    | P112         | Lucian Pop    | 11 Septembrie 2020 10:15 | I5             |
| A12          | Mihai Tic  | P115         | Tudor Rusu    | 23 Iulie 2020 14:00      | I3             |
| A12          | Mihai Tic  | P114         | Irina Scurt   | 16 August 2020 08:00     | I7             |

Each doctor has a unique value for MarcaAngajat. Each patient of the cabinet has a unique value for NumarPacient. Each procedure has a unique value for CodInterventie.

Each tuple from the relation is an appointment. For example, first tuple has the following meaning: the patient with the NumarPacient = P111 has an appointment in 1st of July 2020 at 10:00; the doctor that will attend the patient has MarcaAngajat = A10 and his name is Dan Ion, and the procedure has CodInterventie = I5; the patient's name is Maria Dumitru.

Which of the following statements is true/are true:

- a. The schema is the second normal form.
- b. The functional dependency  $\{MarcaAngajat, DataProgramare\} \rightarrow \{\text{NumarPacient}\}$  is satisfied by the relation's data.
- c. The schema is in the third normal form.
- d. The schema is in the first normal form.
- e. None of the above statements is true.

## ↓ AP 38

Se dau relațiile de mai jos. Cheile primare sunt subliniate. Cheile străine sunt scrise cursiv și au aceeași denumire cu a cheilor primare corespunzătoare.

Magazine(CodM, NumeMagazin, Adresa, Localitate)

Categorii(CodC, NumeCategoria)

Produse(CodP, NumeProdus, Descriere, Marca, CodC)

Distributie(CodM, CodP, pret).

Fie TR tabelul rezultat cu schema TR(NumeMagazin, NumeProdus, pret). D este categoria cu NumeCategoria='Dulciuri', iar H este categoria cu NumeCategoria='Haine'.

Pentru fiecare magazin M, TR conține câte un rând pentru fiecare produs P din categoriile D sau H aflat în distribuția magazinului. Fiecare rând conține: numele magazinului M, numele produsului P și prețul produsului P în magazinul M.

Care dintre următoarele interogări returnează tabelul rezultat TR?

- a) 

```
SELECT M.NumeMagazin, P.NumeProdus
FROM Magazine M INNER JOIN Distributie D ON M.CodM=D.CodM
INNER JOIN Produse P ON D.CodP=P.CodP
INNER JOIN Categorii C ON P.CodC=C.CodC
WHERE C.NumeCategoria='Dulciuri' OR C.NumeCategoria='Haine'
```
  
- b) 

```
SELECT M.NumeMagazin, P.NumeProdus, D.pret
FROM Magazine M
INNER JOIN Distributie D ON M.CodM=D.CodM
INNER JOIN Produse P ON D.CodP=P.CodP
INNER JOIN Categorii C ON P.CodC=C.CodC
WHERE C.NumeCategoria IN ('Dulciuri', 'Haine')
```
  
- c) 

```
SELECT M.NumeMagazin, P.NumeProdus
FROM Magazine M
LEFT JOIN Distributie D ON M.CodM=D.CodM
LEFT JOIN Produse P ON D.CodP=P.CodP
LEFT JOIN Categorii C ON P.CodC=C.CodC
WHERE C.NumeCategoria='Dulciuri' OR C.NumeCategoria='Haine'
```
  

---

  
- d) 

```
SELECT M.NumeMagazin, P.NumeProdus, C.NumeCategoria
FROM Magazine M
INNER JOIN Distributie D ON M.CodM=D.CodM
INNER JOIN Produse P ON D.CodP=P.CodP
INNER JOIN Categorii C ON P.CodC=C.CodC
WHERE C.NumeCategoria='Dulciuri' OR C.NumeCategoria='Haine'
```
  
- e) Nicio variantă de mai sus nu este corectă.

## ↓ AP 39

Considerăm următoarele relații:

Teatre (TID, Nume, Oras)

Spectacole (SpID, Titlu, Autor)

Repertoriu (TID, SpID, DataPremierii).

Care dintre următoarele expresii de algebră relațională ne furnizează "toate teatrele care au avut în repertoriu cel puțin un spectacol bazat pe una dintre piesele lui Shakespeare". Expresiile pot conține erori.

- a.  $\pi_{\{Nume\}}(Teatre \times Repertoriu \times \sigma_{Autor='Shakespeare'}(Spectacole))$
- b.  $(\pi_{\{Nume\}}(Teatre * Repertoriu)) * (\sigma_{Autor='Shakespeare'}(Spectacole))$
- c.  $\pi_{\{Nume\}}(Teatre * Repertoriu * (\sigma_{Autor='Shakespeare'}(Spectacole)))$
- d.  $(\pi_{\{Nume\}}(\sigma_{Autor='Shakespeare'}(Teatre * Repertoriu))) \cap (\sigma_{Autor='Shakespeare'}(Spectacole))$
- e. Niciunul dintre răspunsuri nu este corect.

**Operatorii algebrici utilizati:**

- Selecție:  $\sigma_c$
- Produs cartezian:  $\times$
- Proiecție:  $\Pi_a$
- Join natural:  $*$
- intersecție:  $\cap$

## ↓ AP 40

Se dă instanța unei relații cu schema N[CodN, A, B, C, D, E]. {CodN} este cheie primară.

| CodN | A  | B   | C    | D  | E  |
|------|----|-----|------|----|----|
| 1    | a1 | 10  | 4000 | d3 | e1 |
| 2    | a4 | 70  | 450  | d3 | e4 |
| 3    | a4 | 35  | 900  | d2 | e1 |
| 4    | a2 | 30  | 130  | d2 | e2 |
| 5    | a3 | 100 | 20   | d1 | e3 |

Cât este diferența între cardinalitatea rezultatului primei interogări și cardinalitatea rezultatului celei de a doua interogări?

```
SELECT A, E, COUNT(B)
FROM N
GROUP BY A, E
HAVING COUNT(B) IN (1,3,4,5)
```

```
SELECT D, A, SUM(C)
FROM N
GROUP BY D, A
HAVING SUM(C) NOT BETWEEN 120 AND 300
```

Răspuns: \_\_\_\_\_

## ↓ AP 41

Un magazin de închirieri video ține evidența filmelor închiriate într-o bază de date cu următoarea schemă. Cheile primare sunt scrise cu bold, iar cheile străine sunt scrise cursiv și au același nume cu coloanele spre care fac referire.

Filme (***idFilm***, titlu, anAparitie)

Clienti (***idClient***, nume, telefon)

Inchirieri (***idFilm***, ***idClient***, ***dataInchiriere***, dataReturnare)

Ce returneaza urmatoarea interogare?

- $\Pi_{idClient,idFilm}(Inchirieri) \div \Pi_{idFilm}(Filme)$ 
  - a. Filmele care au fost închiriate de toți clienții.
  - b. Filmele care au fost închiriate de cel puțin un client.
  - c. Clienții care au închiriat toate filmele.
  - d. Clienții care au închiriat cel puțin un film.
  - e. Nicio variantă de mai sus nu este corectă.

### Operatori algebrici utilizati în interogare

- Proiecție:  $\Pi_a$
- Cât:  $\div$

## ↓ AP 42

Se dă instanța unei relații cu schema R[CodR, A, B, C, D, E]. {CodR} este cheie primară.

| CodR | A | B | C   | D | E |
|------|---|---|-----|---|---|
| 1    | 1 | 1 | c1  | 2 | 2 |
| 2    | 1 | 1 | c2  | 2 | 2 |
| 3    | 2 | 2 | c2  | 2 | 2 |
| 4    | 3 | 4 | c2  | 4 | 5 |
| 5    | 3 | 4 | c6  | 4 | 5 |
| 6    | 4 | 5 | c12 | 5 | 4 |

Câte tupluri returnează interogarea de mai jos?

```
SELECT DISTINCT r2.A, COUNT(*)
FROM R r1 INNER JOIN R r2 ON r1.CodR = r2.CodR
    INNER JOIN R r3 ON r2.CodR = r3.CodR
WHERE r1.A <> r2.A
GROUP BY r1.A, r2.A
HAVING r2.A >= 1
```

- a. 0
- b. 6
- c. 12
- d. 11
- e. Nicio variantă de mai sus nu este corectă.

## ↓ AP 43

Se dă relația  $R(A, B, C, D)$  cu următoarele dependențe funcționale  $F = \{A \rightarrow BC, B \rightarrow D\}$ . Cheia primară este subliniată, iar relația  $R$  nu conține attribute repetitive.

Care din afirmațiile următoare este / sunt adevărată / adevărate:

- a.  $R$  este în prima formă normală.
- b.  $R$  este în a doua formă normală.
- c.  $R$  este în a treia formă normală.
- d.  $R$  este în forma normală Boyce Codd.
- e. Nicio variantă de mai sus nu este corectă.

### ↓ AP 44

Considerăm următoarea relație:  $Z (\underline{U}, \underline{V}, X, Y)$  cu dependențele funcționale:

$$F = \{V \rightarrow Y, X \rightarrow Y, X \rightarrow U\}.$$

Care dependențe funcționale trebuie să fie eliminate ca relația  $Z$  să fie în a doua formă normală (2NF)?

- a.  $V \rightarrow Y$
- b.  $V \rightarrow Y, X \rightarrow Y, X \rightarrow U$
- c.  $V \rightarrow Y, X \rightarrow Y$
- d. Toate dependențele funcționale trebuie eliminate.

### ↓ AP 45

Se consideră următoarele relații (cheile primare sunt subliniate, iar cheile externe sunt scrise cursiv și au aceeași denumire cu cheile primare pe care le referă):

Masini (MasinaID, Model, AnFabricatie)

Politisti (PolitistID, PNumar)

Candidati (CandidatID, Nume, Email, Varsta)

Examene (ExamenID, CandidatID, MasinaID, DataExamenului, PolitistID).

Alegeți cheia / cheile candidat în cazul relației Examene (care nu conțin ExamenID), dacă știm că un candidat poate să se prezinte la mai multe examene cu aceeași mașină și să fie examinat de către același polițist. La un examen un candidat va folosi o singură mașină și va fi examinat de către un singur polițist. Un candidat poate să aibă un singur examen pe aceeași dată. Justificați răspunsul!

↓ AP 46

Se dă instanța unei relații cu schema R [A, B, C, D]. A este cheie primară și este scrisă cu bold.

| A | B | C        | D |
|---|---|----------|---|
| 1 | 3 | Nul<br>1 | 2 |
| 2 | 4 | 4        | 3 |
| 3 | 4 | 1        | 2 |

În această instanță e satisfăcută dependența funcțională  $B \rightarrow D$  ? Justificați.

↓ AP 47

Is the functional dependency  $\{A, B\} \rightarrow \{D, E\}$  satisfied by the data from the following relation? Please explain yours answer.

| CodR | A | B | C       | D | E |
|------|---|---|---------|---|---|
| 1    | 1 | 1 | c1<br>↳ | 2 | 2 |
| 2    | 1 | 1 | c2      | 2 | 2 |
| 3    | 2 | 2 | c2      | 2 | 2 |
| 4    | 3 | 4 | c2      | 4 | 5 |
| 5    | 3 | 4 | c6      | 4 | 5 |

## ↓ AP 48

Care din următoarele scheme oferă o modelare corectă în a treia formă normală pentru un magazin care ține evidența recenziilor date de clienți pentru produse? Cheile primare sunt scrise cu bold, iar cheile străine sunt scrise cursiv și au același nume cu coloanele spre care fac referire. Se cunoaște faptul că un produs poate primi mai multe recenzi. Trebuie stocat și autorul recenziei care este unul din clienți. Un client poate să dea mai multe recenzi aceluiași produs. Alegeți toate opțiunile care sunt corecte.

- a. Clienti (***idClient***, numeClient, email)  
Produse (***idProdus***, numeProdus, pretProdus, recenzi, *idClient*),  
unde, {recenzi} este atribut repetitiv.
  
- b. Produse (***idProdus***, numeProdus, pretProdus)  
Clienti (***idClient***, numeClient, email)  
Recenzi (***idRecenzie***, *idClient*, *idProdus*, dataOra, descriere)
  
- c. Produse (***idProdus***, numeProdus, pretProdus)  
Clienti (***idClient***, numeClient, email)  
Recenzi (***idProdus***, *idClient*, dataOra, descriere)
  
- d. Nicio variantă de mai sus nu este corectă.

## Programming

## Algorithms

↓ AP 49

Consider the sequence  $x[1], x[2] \dots x[n]$  of natural numbers ordered descendingly, a natural number  $a$  such that  $x[1] > a > x[n]$  and the function **determine** written below. Which of the options below describe the value that should be assigned to the variable  $k$  (in the space marked with ..... ) such that the call **determine(a, x, n)** will return the position on which  $a$  should be inserted in the sequence  $x$  so that the elements remain sorted descendingly? **Note:** inserting an element on a position  $k$  in the sequence entails shifting the elements on positions  $k, k+1, \dots, n$  one position to the right.

```
Function determine(a, x, n) is
    i ← 1
    j ← n
    While j - i > 1 do
        k ← .....
        If x[k] ≤ a then j ← k
        Else i ← k
        EndIf
    EndWhile
    determine ← i + 1
EndFunction
```

- A.  $(j - i) \text{ div } 2$
- B.  $(i + j) \text{ div } 2$
- C.  $i + (j \text{ div } 2)$
- D.  $i + (j - i) \text{ div } 2$
- E.  $j + (j - i) \text{ div } 2$

### ↓ AP 50

```
for (number1 = i = 0; i < n; i += 1) {  
    for (j = i; j <= i + 4; j++) {  
        number1 += 1;  
    }  
}  
for (number2 = i = 1; i <= n; i *= 2){  
    for (j = 0; j <= i; j++){  
        number2 += 1;  
    }  
}
```

Ce se executa pentru  $n = 16$  in ambele cazuri (cu variante) ?

### ↓ AP 51

The function *verify* given below receives as a parameter a natural number  $nr$ ,  $1 \leq nr \leq 1000000$ . Choose the true statements about the function, knowing that the instruction marked with (\*) removes the first element from the array *v*.

```
#include <vector>  
using namespace std;  
bool verify(int nr) {  
    vector<int> v;  
    int nrC = nr;  
    while (nrC > 0) {  
        v.push_back(nrC % 10);  
        nrC = nrC / 10;  
    }  
    int nr2 = 0;  
    while (v.empty() == false) {  
        nr2 = nr2 * 10 + v[0];  
        v.erase(v.begin()); // (*)  
    }  
    return nr == nr2;  
}
```

- A. The function verifies if a number is a palindrome
- B. The function returns true for any number
- C. The function computes the number formed from the digits of the number  $nr$ , but in reverse order (the mirror of  $nr$ )
- D. The function computes the sum of the digits of the number  $nr$
- E. The function returns false if the number  $nr$  is a multiple of 10

## ↓ AP 52

Consider the subprogram **process** defined below, which takes as parameter a sequence  $x$  with  $n$  non-zero real numbers and changes these numbers. In what way are the numbers in  $x$  changed when calling **process**?

```
void process(float* x, int n){  
    int p = 0;  
    for (int k = 0; k < n; k++){  
        p = p + 1;  
        for (int i = 0; i < n - 1; i++){  
            if (x[i] > x[i+1]) {  
                x[i] = x[i] * x[i+1];  
                x[i+1] = x[i] / x[i+1];  
                x[i] = x[i] / x[i+1];  
            }  
        }  
        n = p;  
    }  
}
```

### ↓ AP 53

Care este rezultatul returnat de funcția de mai jos, dacă este apelată ca **prelucrare(n, m, 0)**, în condițiile în care  $1 \leq n, m \leq 10000$ ?

- A. Returnează fals dacă scrierea în baza 3 a numerelor **n** și **m** conține cifra 2 de număr egal ori, adevărat în caz contrar.
- B. Returnează adevărat dacă scrierea în baza 3 a numerelor **n** și **m** conține cel puțin o dată cifra 2, fals în caz contrar.
- C. Returnează adevărat dacă scrierea în baza 3 a numerelor **n** și **m** conține cifra 2 de număr egal ori, fals în caz contrar.
- D. Returnează adevărat dacă numerele **n** și **m** au aceeași reprezentare în baza 3.

```
bool prelucrare(int n, int m, int k) {
    if (n == 0 && m == 0)
        return k == 0;
    else if (n % 3 == 2 && m % 3 == 2) {
        return prelucrare(n / 3, m / 3, k); }
    else if (n % 3 == 2) {
        return prelucrare(n / 3, m / 3, k + 1);}
    else if (m % 3 == 2) {
        return prelucrare(n / 3, m / 3, k - 1);}
    else {
        return prelucrare(n / 3, m / 3, k); }
}
```

## ↓ AP 54

Fie subalgoritmul recursiv **prelucrare** care are ca date de intrare două numere naturale  $s$  și  $d$  și ca date de ieșire un sir  $F$  de valori logice (adevărat, fals). Care dintre următoarele afirmații indică elementele sirului  $F$  obținut în urma apelului **prelucrare**(1, 7,  $F$ ) ?

A. Elementele  $F_1, F_2, \dots, F_7$  sunt:

**fals, adevărat, fals, adevărat, fals, adevărat, fals**

B. Elementele  $F_1, F_2, \dots, F_7$  sunt:

**adevărat, fals, adevărat, fals, adevărat, fals, adevărat**

C. Elementele  $F_1, F_2, \dots, F_7$  sunt:

**adevărat, fals, adevărat, fals, fals, adevărat, fals**

D. Elementele  $F_1, F_2, \dots, F_7$  sunt:

**adevărat, fals, adevărat, adevărat, adevărat, fals, adevărat**

**Subalgoritmul prelucrare( $s, d, F$ ) este**  
**Dacă  $s=d$  atunci  $F[d] \leftarrow$  adevărat**  
**altfel**  
    **Dacă  $(s+d) \bmod 2=0$  atunci**  
         $F[(s+d) \bmod 2] \leftarrow$  fals  
        prelucrare( $s, (s+d) \bmod 2-1, F$ )  
        prelucrare( $(s+d) \bmod 2+1, d, F$ )  
    **altfel**  
        prelucrare( $s, (s+d) \bmod 2, F$ )  
        prelucrare( $(s+d) \bmod 2+1, d, F$ )  
    **SfDacă**  
    **SfDacă**  
**SfSubalgoritm**

## ↓ AP 55

Ce face funcția de mai jos, care primește ca parametru un vector  $x[1], x[2]...x[n]$  care conține  $n$  numere distincte și care are aspect de munte (începe cu o secvență nevidă de numere în ordine crescătoare, urmată de o secvență în ordine descrescătoare nevidă)?

**Funcția ceFace( $x, n$ ) este**

$s \leftarrow 1$

$e \leftarrow n$

**Câttimp  $s < e$  execută**

**Dacă  $x[s] < x[e]$  atunci**

$s \leftarrow s + 1$

**altfel**

$e \leftarrow e - 1$

**SfDacă**

**SfCâttimp**

$ceFace \leftarrow x[e]$

**SfFunctie**

## ↓ AP 56

[Start conversație](#)

```

string getName();
int getHeight();
double getPrice();
};

int c = 0;
for (int i = 0; i < this->plants.size(); i++) {
    if (this->plants[i].getPrice() <= limit)
        c++;
}
return c;
}

```

- 1.** (2p) În clasele de mai sus există:

|   |                         |   |                         |   |                 |
|---|-------------------------|---|-------------------------|---|-----------------|
| a | Numai attribute publice | b | Numai attribute private | c | Nici un atribut |
|---|-------------------------|---|-------------------------|---|-----------------|

- 2.** (2p) Care este efectul executării următoarei sechente de instrucțiuni, știind că metoda *addPlant* adaugă un element la finalul vectorului *plants*?

```

Plant p1 = Plant("Lavender", 25, 70.0);
Plant p2 = Plant("Berberis", 50, 90.0);
Plant p3 = Plant("Euonymus", 35, 50.0);
PlantShop shop = PlantShop("Art");
shop.addPlant(p1);
shop.addPlant(p2);
shop.addPlant(p3);
cout << shop.countPlants(50.0);

```

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| a | 3 | b | 1 | c | 2 |
|---|---|---|---|---|---|

- 3.** (2p) Cu ce trebuie înlocuite ... din următoarea funcție pentru a sorta lista de plante *plants* crescător după atributul *height*?

```

void PlantShop::sortPlants() {
    for(int i = 0; i < this->plants.size()-1; i++) {
        int ind = i;
        for(int j = i + 1; j < this->plants.size(); j++) {
            ...
            if (i < ind) {
                Plant aux = this->plants[i];
                this->plants[i] = this->plants[ind];
                this->plants[ind] = aux;
            }
        }
    }
}

```

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| a | if (this->plants[i].getHeight() < this->plants[ind].getHeight())         ind = j; | b | if (this->plants[i].getHeight() < this->plants[j].getHeight())         ind = j; | c | if (this->plants[j].getHeight() < this->plants[ind].getHeight())         ind = i; |
|---|---|---|---|---|---|

- 4.** (3p) Ce se înțelege prin encapsularea datelor?

↓ AP 57

```
class A {  
public:  
    A(){cout << "A";}  
    virtual void f(){cout << "A::f";}  
    void g(){f();}  
};  
  
class B : public A{  
public:  
    virtual void f(){cout << "B::f";}  
};  
  
int main()  
{  
    B b;  
    b.g();  
}
```

Ce afiseaza executia?

## OOP

↓ AP 58

Which is the result of running the program below?

|  |  |
|--|--|
| <pre>#include &lt;string&gt; #include &lt;iostream&gt; using namespace std; class Vehicle{ public:     virtual void printModel() {         std::cout &lt;&lt; "From Vehicle";         printSpecificModel();     }     virtual void printSpecificModel() {     cout &lt;&lt; " base features"; } };</pre> | <pre>class BMW : public Vehicle {  public:     void printModel() {         Vehicle::printModel();         cout &lt;&lt; " features";     }     void printSpecificModel() {         cout &lt;&lt; "-&gt;BMW";     } };  int main() {     Vehicle* myCar1 = new BMW;     myCar1-&gt;printModel();     // deallocations     return 0; }</pre> |
| a) From Vehicle base features<br>features  | c) From Vehicle->BMW features  |
| b) From Vehicle base feature   | d) From Vehicle features->BMW  |

## ↓ AP 59

Considerăm următorul program scris în limbajul C++:

```
#include <iostream>
using namespace std;
class B {
public:
    virtual void f() { cout << "b"; }
};
class D : public B {
private:
    B* b;
public:
    D(B* b = 0) { this->b = b; }
    void f() { b->f(); cout << "d"; }
};
int main() {
    B* b = . . . //linia 1
    b->f();
    //deallocari
    return 0;
}
```

Inițializați variabila b din **linia 1**, astfel încât programul să ruleze corect și să afișeze: **bd**

## ↓ AP 60

What is the result of executing the program below?

```
#include <iostream>
using namespace std;

class A {
private:
    int x;
public:
    static int noOfInstances;
    A(int _x = 5) : x(_x) {
        this->x = _x;
        noOfInstances++;
    }
    A(const A& other) {
        this->x = other.x;
        noOfInstances++;
    }
    void inc(){
        x++;
    }
};

int A::noOfInstances = 0;

void functionA(A a) {
    A b(a);
    b.inc();
}

int main() {
    A a1, a2(2);
    cout << a1.noOfInstances << " ";
    A a3(a1);
    cout << A::noOfInstances << " ";
    functionA(a3);
    cout << A::noOfInstances << " ";
    return 0;
}
```

A. 012  
B. 235  
C. 234  
D. 225

## ↓ AP 61

Consider the following code snippet:

```
#include <string>
#include <iostream>
using namespace std;

class Shape {
public:
    virtual double area() = 0;
    virtual string toString() = 0;
};

class Polygon : public Shape {
public:
    string toString() { return "Polygon"; }
};

class Square : public Polygon {
private:
    double length;
public:
    Square(double l) {
        this->length = l;
    }
    double area() { return length * length; }
    string toString() { return "Square"; }
};

int main()
{
    Shape* s = new Shape(); // line 1
    Shape* p = new Polygon(); // line 2
    Shape* sq = new Square(5); // line 3
    cout << sq->toString() << " " << sq->area(); // line 4
    // deallocations
    return 0;
}
```

Which of the statements below are true:

- A. Objects of type *Shape* cannot be created (line 1 will give a compilation error), however we can create objects of any of the other classes.
- B. Objects of types *Shape* and *Polygon* cannot be created (lines 1 and 2 will have compilation errors).
- C. Objects of neither class can be created (lines 1, 2 and 3 will have compilation errors).
- D. The program compiles without errors and prints “Square 25”.

## ↓ AP 62

Care este rezultatul execuției programului de mai jos?

|   |  |
|---|--|
| <pre>#include &lt;iostream&gt; #include &lt;string&gt; using namespace std;  class Pizza { private:     int id; public:     Pizza(int id) {         this-&gt;id = id;     }     Pizza() { }     virtual void cook() {         cout &lt;&lt; "cook pizza ";     } };</pre> | <pre>class PizzaWithCheese : public Pizza { private:     string topping; public:     PizzaWithCheese(int id, string topping) {         this-&gt;topping = topping;     }     void cook() {         Pizza::cook();         cout &lt;&lt; "with cheese";     } };  int main() {     Pizza x[]={ Pizza(1),                 PizzaWithCheese(1, "burduf")};     x[0].cook();     x[1].cook();     return 0; }</pre> |
|---|--|

a) cook pizza with burduf

c) cook pizza cook pizza

b) cook pizza

d) cook pizza with cheese

## ↓ AP 63

Considerăm următorul program scris în limbajul C++:

```
#include <iostream>
using namespace std;
class B {
public:
    virtual void f() { cout << "b"; }
};
class D : public B {
private:
    B* b;
public:
    D(B* b = 0) { this->b = b; }
    void f() { b->f(); cout << "d"; }
};
int main() {
    B* b = . . . //linia 1
    b->f();
    //deallocari
    return 0;
}
```

Inițializăți variabila b din **linia 1**, astfel încât programul să ruleze corect și să afișeze: **bd**

# Model POCU Licenta

## **↓POCU 1**

**Question 1**Not yet  
answeredMarked out of  
10.00 Flag question

Considerăm următoarele instanțe de relații cu schemele  $R(\underline{A}, B, C, D)$ ,  $S(\underline{D}, E, F)$ . Cheile primare sunt subliniate, cheile externe sunt scrise cursiv și au aceeași denumire cu cheile primare pe care le referă.

| A | B  | C  | D |
|---|----|----|---|
| 1 | 20 | 12 | 5 |
| 2 | 21 | 11 | 4 |
| 3 | 26 | 9  | 5 |

| D | E  | F |
|---|----|---|
| 1 | 20 | 5 |
| 4 | 23 | 7 |
| 5 | 29 | 9 |
| 6 | 28 | 8 |
| 7 | 30 | 5 |

Cât este diferența între cardinalitatea rezultatului primei interogări și cardinalitatea rezultatului celei de a doua interogări?

```
SELECT *
  FROM R RIGHT JOIN S ON R.D=S.D
```

```
SELECT R.D
  FROM R INNER JOIN S ON R.D=S.D
 GROUP BY R.D
 HAVING COUNT(*) > 1
```

Select one or more:

- A. 5
- B. 4
- C. 2
- D. 3
- E. 1

## ↓POCU 2

Question **2**

Not yet  
answered

Marked out of  
10.00

 Flag question

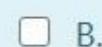
Fie sirul  $x=(5, 3, 2, 1, 1, 1)$ . Ce va afisa urmatorul algoritm?

```
for i=1 to n
    c=x[i];
    x[i]=x[n-i+1];
    x[n-i+1]=c;
end for
for i=1 to n
    print (x[i],",")
end for
```

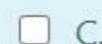
Select one or more:



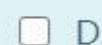
A.  
5,3,2,1,1,1



B.  
1,1,1,2,3,5



C.  
Nicio varianta nu este corecta.



D.  
1,1,2,1,3,5

**↓POCU 3**Question **3**Not yet  
answeredMarked out of  
10.00 Flag question

Se dă relația  $R( A, B, C, D, E)$  cu următoarele dependențe funcționale  $F = \{ C \rightarrow AD, AC \rightarrow E, E \rightarrow B \}$ .  $R$  nu conține atrbute repetitive.

Care din afirmațiile următoare este / sunt adevărată / adevărate:

Select one or more:

- A.  $R$  este în a treia formă normală.
- B.  $R$  este în forma normală Boyce Codd.
- C. Nicio variantă de mai sus nu este corectă.
- D.  $R$  este în prima formă normală.
- E.  $R$  este în a doua formă normală.

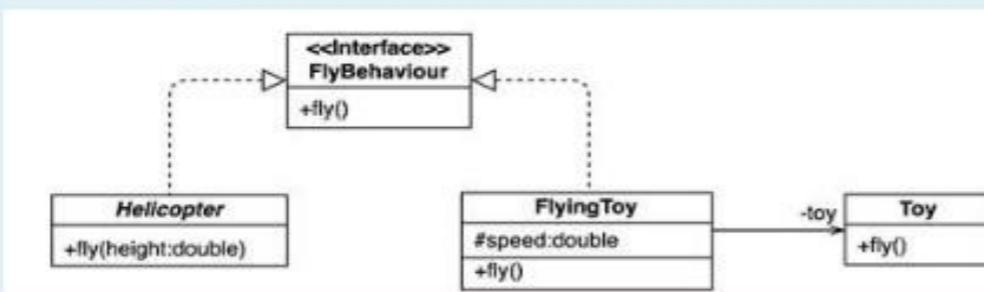
## ↓POCU 4

Question 4

Not yet  
answeredMarked out of  
10.00

Flag question

Fie diagrama UML:



si codul C++ asociat:

```
class FlyBehaviour {
public:
    virtual void fly() = 0;
};
class Helicopter :public FlyBehaviour {
public:
    void fly(double height) {
        cout << "Flying " << height << " meters" << endl;
    }
};
class Toy {
public:
    void fly() {
        cout << "Toy flying ..." << endl;
    }
};
class FlyingToy :public FlyBehaviour {
private:
    Toy toy;
    double speed;
public:
    void fly() {
        toy.fly();
    }
};
```

Care dintre urmatoarele afirmații NU sunt adevărate:

Select one or more:

- A.  
Implementarea clasei **FlyingToy** este conform diagramei.
- B.  
Se pot crea obiecte de tipul clasei **Helicopter**.
- C.  
Implementarea clasei **FlyBehaviour** nu corespunde diagramei.
- D.  
Exista o relatie de asociere intre clasa **FlyingToy** si clasa **Toy**.

**↓POCU 5**

Question 5

Not yet  
answeredMarked out of  
10.00

Flag question

Care este efectul compilării și execuției programului următor:

```
#include <iostream>
using namespace std;

class AClass
{
public:
    virtual void m1() { cout << "AClass::m1() " ; }

class BClass : public AClass
{
public:
    void m1() { cout << "BClass::m1() " ; }

class CClass : public BClass
{
public:
    void m1() { cout << "CClass::m1() " ; }

int main()
{
    BClass* bp = new CClass;
    bp->m1();
    return 0;
}
```

Select one or more:

- A.  
AClass::m10
- B.  
Eroare la compilare
- C.  
CClass::m10
- D.  
BClass::m10

**↓POCU 6**Question **6**Not yet  
answeredMarked out of  
10.00

Flag question

Se dă următoarea schemă de baze de date. Cheile primare sunt subliniate, iar cheile străine sunt scrise cursiv și au același nume cu coloanele spre care fac referire. Nici o coloana nu poate avea valori nule.

Spectacole (*idSpectacol*, nume, dataSpectacol, durata, *idLocatie*)

Locatii (*idLocatie*, nume, capacitate, *idAdresa*)

Adrese (*idAdresa*, strada, numar, oras, judet)

Care din următoarele interogări afișează numărul de spectacole de la fiecare locație în fiecare dată (de ex la locația X pe data 2019-1-1 sunt 10 spectacole, la locația X pe data 2019-2-2 sunt 2 spectacole, etc.) ?

Select one or more:

A.

```
SELECT L.nume, S.dataSpectacol, COUNT(*)  
  
FROM Spectacole S inner join Locatii L on S.idSpectacol=L.idLocatie  
  
GROUP BY S.idLocatie
```

B.

```
SELECT L.nume, S.dataSpectacol, COUNT(*)  
  
FROM Spectacole S inner join Locatii L on S.idLocatie=L.idLocatie  
  
GROUP BY LidLocatie, Lnume, S.dataSpectacol
```

C.

```
SELECT L.nume, S.dataSpectacol, COUNT(*)  
  
FROM Spectacole S inner join Locatii L on S.idLocatie=L.idLocatie  
  
GROUP BY LidLocatie, S.dataSpectacol
```

D.

```
SELECT L.nume, S.dataSpectacol, COUNT(*)  
  
FROM Spectacole S inner join Locatii L on S.idSpectacol=L.idLocatie  
  
GROUP BY LidLocatie,L.nume, S.dataSpectacol
```

## ↓POCU 7

Question 7

Not yet  
answered

Marked out of  
10.00

Flag question

Ce afișează următoarea secvență de cod:

```
import java.util.*;  
  
public class E {  
    public static void main(String[] args) {  
        Integer[] values = {12, 10, 15, 14};  
        LinkedHashSet<Integer> lhSet = new LinkedHashSet<>(Arrays.asList(values));  
        TreeSet<Integer> tSet = new TreeSet<>(lhSet);  
        HashSet<Integer> hSet = new HashSet<>(tSet.descendingSet());  
        System.out.println(tSet);  
        System.out.println(lhSet);  
        System.out.println(hSet);  
    }  
}
```

Select one or more:

- A.
  - [12, 10, 15, 14]
  - [10, 12, 14, 15]
  - [15, 14, 12, 10]
- B.
  - [12, 10, 15, 14]
  - [12, 10, 15, 14]
  - [15, 14, 12, 10]
- C.
  - [10, 12, 14, 15]
  - [12, 10, 15, 14]
  - [10, 12, 14, 15]
- D.
  - [10, 12, 14, 15]
  - [12, 10, 15, 14]
  - [15, 14, 12, 10]

↓POCU 8

Question 8

Not yet  
answeredMarked out of  
10.00

Flag question

Care este efectul compilării și execuției programului următor:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace POCU
{
    class Program
    {
        static void Main(string[] args)
        {
            List<String> students = new List<String> { "Alex", "Ioana", "Mircea" };
            var result = from v in students select v;
            students.Add("Dan");
            result.ToList().ForEach(Console.WriteLine);
        }
    }
}
```

Select one or more:

- A.  
Eroare la compilare
- B.  
DanAlexIoanaMircea
- C.  
AlexIoanaMircea
- D.  
AlexIoanaMirceaDan

↓POCU 9Question **9**Not yet  
answeredMarked out of  
10.00 Flag question

Care dintre urmatoarele afirmatii referitoare la caracteristicile membrilor statici, definiti intr-o clasa A, sunt adevarate (in limbajul C++):

Select one or more:

- A.  
Un atribut static poate fi accesat dintr-o metoda nestatica a clasei A.
- B.  
O metoda statica definita in clasa A poate fi apelata din afara clasei doar de o metoda statică definită de o alta clasa B.
- C.  
O metoda nestatica definita intr-o clasa B poate apela o metoda statica definita in clasa A.
- D.  
Un atribut static sau metoda statică pot fi accesate fără a avea o instanță a clasei A.

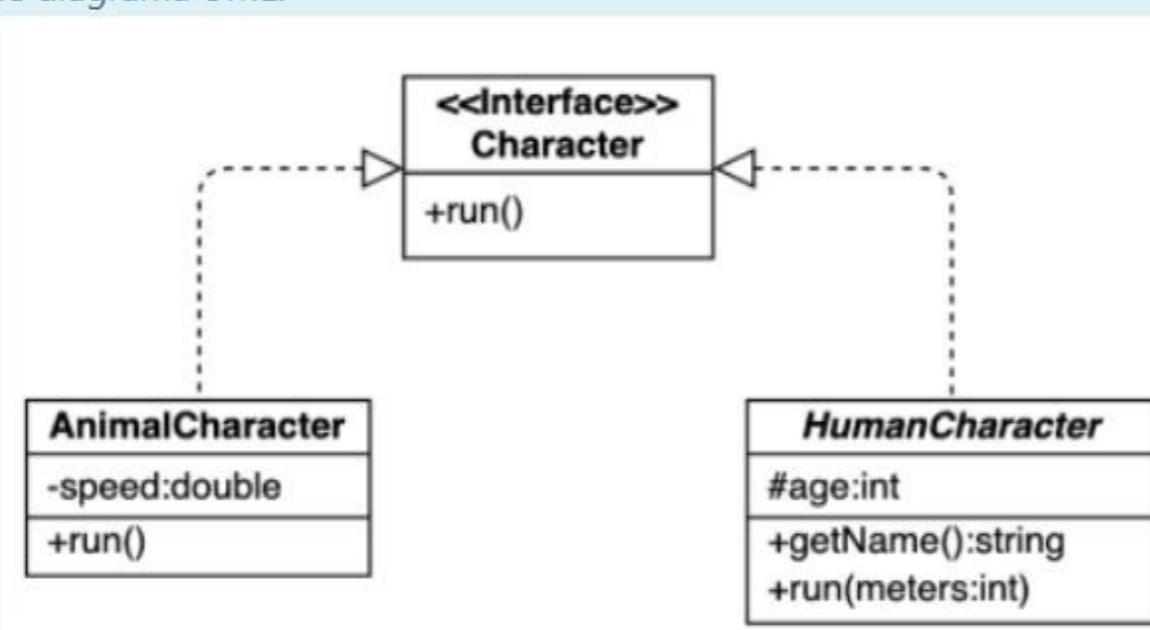
↓POCU 10

Question 10

Not yet  
answeredMarked out of  
10.00

Flag question

Fie diagrama UML:



Care dintre urmatoarele afirmatii sunt adevarate:

Select one or more:

- A.  
Atributul **speed** din clasa **AnimalCharacter** este protejat (protected).
- B.  
Clasa **HumanCharacter** este abstracta.
- C.  
Intre **AnimalCharacter** si **Character** exista o relatie de asociere.
- D.  
Metoda **run(int)** din **HumanCharacter** este abstracta.
- E.  
Metoda **run()** din **Character** este abstracta.

## ↓POCU 11

Question 11

Not yet  
answered

Marked out of  
10.00

Flag question

Care este efectul compilării și execuției programului următor:

```
package POCU;

class A<T> {
    private T entity;
    public A(T entity) {
        this.entity = entity;
    }
    public T getValue(){return entity;}

    @Override
    public String toString() {
        return "A{entity=" + entity + '}';
    }
}
class B<T> extends A<T> {
    public B(T entity) {
        super(entity);
    }
    @Override
    public String toString() {
        return "B{entity=" + getValue() + '}';
    }
}
```

```
public class G11 {
    public static void main(String[] args) {
        A<Integer> a = new A<>(1);
        System.out.println(a.getValue());

        B<A> b = new B<>(a);
        System.out.println(b.getValue());

        B<B> b2 = new B<>(b);
        System.out.println(b2.getValue());
    }
}
```

Select one or more:

- A.  
eroare la compilare
- B.  
eroare la execuție
- C.  
1  
A{entity=1}  
B{entity=B{entity=1}}
- D.  
1  
A{entity=1}  
B{entity=A{entity=1}}

## ↓POCU 12

Question **12**

Not yet  
answered

Marked out of  
10.00

Flag question

Se dă o bază de date cu următoarea schemă. Cheile primare sunt scrise cu bold, iar cheile străine sunt scrise cursiv și au aceeași denumire cu cheile primare pe care le referă. Nici o coloană nu poate avea valori nule.

Mobila (**idMobila**, nume, culoare, pret)

Clienti (**idClient**, nume, prenume, cnp)

Comenzi (**idMobila**, **idClient**, **dataComanda**, nrBucati)

Care din următoarele interogări afișează datele la care s-au comandat cele mai multe bucați de mobilă?

Select one or more:

A.

```
SELECT C.dataComanda, SUM(C.nrBucati)  
  
FROM Comenzi C  
  
GROUP BY C.dataComanda  
  
HAVING SUM(C.nrBucati) >= ALL (SELECT SUM(C.nrBucati)  
  
                                FROM Comenzi C  
  
                                GROUP BY C.dataComanda  
                                )
```

B.

```
SELECT C.dataComanda, SUM(*)  
  
FROM Comenzi C  
  
GROUP BY C.dataComanda  
  
HAVING SUM(*) > ALL ( SELECT SUM(*)  
  
                            FROM Comenzi C  
  
                            GROUP BY C.dataComanda  
                            )
```

C.

```
SELECT C.dataComanda, COUNT(*)  
FROM Comenzi C  
GROUP BY C.dataComanda  
HAVING COUNT(*) >= ANY (  SELECT count(*)  
                           FROM Comenzi C  
                           GROUP BY C.dataComanda  
                         )
```

D.

```
SELECT C.dataComanda, COUNT(*)  
FROM Comenzi C  
GROUP BY C.dataComanda  
HAVING COUNT(*) >= ALL (  SELECT COUNT(*)  
                           FROM Comenzi C  
                           GROUP BY C.dataComanda  
                         )
```

**↓POCU 13**Question **13**Not yet  
answeredMarked out of  
10.00

Flag question

Se consideră subalgoritmul **h(A, n)**, unde **n** este un număr natural (**1 ≤ n ≤ 10000**):

Subalgoritm **h(A, n)**:

Dacă  $n=0$  atunci returnează 0;

SfDacă

Dacă  $n \bmod 2=0$  atunci returnează  $h(A,n-1)+A[n]$ ;

SfDacă

returnează  $h(A,n-1)-A[n]$ ;

SfSubalgoritm

Algoritmul calculează:

Select one or more:

A.

Diferența elementelor de pe pozițiile impare din vectorul **A**.

B.

Numărul elementelor pare din vectorul **A**.

C.

Diferența dintre suma elementelor pare din vector și suma elementelor impare din vectorul **A**.

D.

Diferența dintre suma elementelor de pe poziții pare și suma elementelor de pe pozițiile impare din vectorul **A**.

E.

Numărul elementelor impare din vectorul **A**.

F.

Niciunul din răspunsuri nu este corect.

## ↓POCU 14

Question 14

Not yet  
answered

Marked out of  
10.00

Flag question

Care este rezultatul compilării și execuției următorului program:

```
using System;
namespace POCU
{
    public abstract class Shape
    {
        protected string Name { get; set; }
        public Shape(string Name) { this.Name = Name; }
        public abstract void ComputeArea();
    }
    public class Square : Shape
    {
        protected int Width { get; set; }
        public Square(string Name, int Width) : base(Name) { this.Width = Width; }
        public override void ComputeArea()
        {
            System.Console.WriteLine(Name + " square: " + Math.Pow(Width, 2));
        }
    }
    public class Rectangle : Square
    {
        private int Length { get; set; }
        public Rectangle(string Name, int Width, int Length) : base(Name, Width) { this.Length = Length; }
        public new void ComputeArea()
        {
            System.Console.WriteLine(Name + " rectangle: " + Width * Length);
        }
    }
    public class Program
    {
        public static void Main()
        {
            Shape a;
            a = new Rectangle("yellow", 3, 2);
            a.ComputeArea();

            Shape b = new Square("yellow", 4);
            b.ComputeArea();

            Square c = new Rectangle("yellow", 2, 3);
            c.ComputeArea();

            Rectangle d = new Rectangle("yellow", 2, 3);
            d.ComputeArea();
        }
    }
}
```

Select one or more:

- A.
  - yellow square: 9
  - yellow square: 16
  - yellow square: 4
  - yellow rectangle: 6
- B.
  - yellow square: 16
  - yellow square: 9
  - yellow square: 4
  - yellow rectangle: 6
- C.
  - yellow square: 9
  - yellow square: 16
  - yellow rectangle: 6
  - yellow square: 4
- D.
  - yellow rectangle: 16
  - yellow square: 9
  - yellow square: 4
  - yellow rectangle: 6

**↓POCU 15****Question 15**Not yet  
answeredMarked out of  
10.00

Flag question

Se dă instanța unei relații cu schema R(A, B, C, D, E). Cheia primară este subliniată.

| A | B  | C    | D  | E  |
|---|----|------|----|----|
| 1 | 4  | 3    | 13 | 20 |
| 2 | 20 | 4    | 10 | 5  |
| 3 | 5  | Null | 21 | 6  |
| 4 | 3  | 3    | 15 | 7  |

Care este rezultatul interogării următoare?

```
select AVG(B) * MAX(B) /COUNT(DISTINCT C) from R
```

Select one or more:

- A. 53
- B. 80
- C. Interogarea dă eroare.
- D. Nicio variantă nu este corectă.
- E. 40

## ↓POCU 16

Question 16

Not yet  
answered

Marked out of  
10.00

Flag question

Care este efectul compilării și execuției programului următor:

```
#include<iostream>
using namespace std;

class Base1
{
public:
    char c='a';
};

class Base2
{
public:
    int c=1;
};

class Derived : public Base1, public Base2
{
public:
    void show() { cout << c; }
};

int main(void)
{
    Derived d;
    d.show();
    return 0;
}
```

Select one or more:

- A.  
Programul indica o eroare la compilare
- B.  
Se afiseaza: 1
- C.  
Programul compilează și indica o eroare la executie
- D.  
Se afiseaza: a

**↓POCU 17**

Question 17

Not yet  
answeredMarked out of  
10.00

Flag question

Care este efectul compilării și execuției programului următor:

```
#include <iostream>
using namespace std;

class Shape {
public:
    virtual int getArea() = 0;
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }
protected:
    int width;
    int height;
    string measureUnit;
};

class Rectangle : public Shape {
public:
    int getArea(string _measureUnit) {
        return (width * height);
    }
};

class Triangle : public Shape {
public:
    int getArea() {
        return (width * height) / 2;
    }
};

int main(void) {
    Rectangle rect;
    Shape* tri=new Triangle;

    tri->setWidth(5);
    tri->setHeight(7);
    cout << "triunghi: " << tri->getArea() << endl;

    return 0;
}
```

Select one or more:

- A.  
dreptunghi: 17
- B.  
triunghi: 17
- C.  
triunghi: 35
- D.  
Programul nu este compilat cu succes

**↓POCU 18**

Question 18

Incorrect

Marked out of  
10.00

Flag question

Se dă instanța unei relații cu schema Angajati(Nume, Prenume, DataNasterii, Varsta, Departament, Salar). {Nume, Prenume, DataNasterii} este cheie primară.

| Nume  | Prenume | DataNasterii | Varsta | Departament | Salar |
|-------|---------|--------------|--------|-------------|-------|
| Toma  | Ana     | 1990-5-1     | 30     | Java        | 2200  |
| Matei | Andrei  | 1995-1-1     | 25     | Php         | 2500  |
| Onciu | Ana     | 1996-2-1     | 24     | C#          | 3000  |
| Toma  | Ioan    | 1991-2-1     | 29     | Java        | 2200  |
| Badiu | Anton   | 1990-12-1    | 29     | Php         | 2500  |

Care din următoarele dependențe funcționale sunt satisfăcute de instanța acestei relații?

Select one or more:

- A. DataNasterii → Varsta
- B. Varsta → Salar
- C. Departament → Salar
- 
- D. Nume → Prenume
- 
- E. Nume, Prenume → DataNasterii

## ↓POCU 19

**Question 19**Not yet  
answeredMarked out of  
10.00[Flag question](#)

Care este efectul execuției programului următor:

```
using System;
namespace POCU
{
    class Program
    {
        delegate string CustomDel(string s);
        static string Hello(string s)
        {
            Console.WriteLine("Hello");
            return "Hello" + s;
        }
        static string Goodbye(string s)
        {
            Console.WriteLine("GoodBy");
            return "GoodBy " + s;
        }
        static void Main(string[] args)
        {
            CustomDel del = Hello;
            del += (x) => "How are you, " + x + "?";
            del += Goodbye;
            Console.WriteLine(del.Invoke("John"));
        }
    }
}
```

Select one or more:

- A.  
Hello John  
How are you, John?  
GoodBy John
- B.  
GoodBy John
- C.  
Hello  
GoodBy  
GoodBy John
- D.  
Hello John

**↓POCU 20**Question **20**Not yet  
answeredMarked out of  
10.00

Flag question

Se dă o bază de date cu următoarea schemă. Cheile primare sunt scrise cu bold, iar cheile străine sunt scrise cursiv și au aceeași denumire cu cheile primare pe care le referă.

Clienti (**idClient**, nume, cnp, telefon)

Ticket ( *idClient*, **idAngajat**, **data**, motiv)

Angajati (**idAngajat**, nume, salar)

Care din interogările următoare afișează clientii care au sunat la callcenter atât din motiv de 'plangere', cât și din motiv de 'informatii'.

**Operatorii algebrici utilizati**

- Selecție:  $\sigma_c$
- Proiecție:  $\Pi_a$
- Reuniune, diferență, intersecție:  $U, -, \cap$
- Join natural:  $*$

Select one or more:

 A.
$$\Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'plangere' \wedge Ticket.motiv = 'informatii'} (Clienti * Ticket * Angajati))$$
 B.
$$\Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'plangere' \vee Ticket.motiv = 'informatii'} (Clienti * Ticket * Angajati))$$
 C.
$$\begin{aligned} & \Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'plangere'} (Clienti * Ticket * Angajati)) \cup \\ & \cup \Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'informatii'} (Clienti * Ticket * Angajati)) \end{aligned}$$
 D.
$$\begin{aligned} & \Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'plangere'} (Clienti * Ticket * Angajati)) \cap \\ & \cap \Pi_{Clienti.nume} (\sigma_{Ticket.motiv = 'informatii'} (Clienti * Ticket * Angajati)) \end{aligned}$$

## ↓POCU 21

Question 21

Not yet  
answeredMarked out of  
10.00

Flag question

Ce afiseaza urmatorul program:

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class CC {
6 private:
7     CC* instance = nullptr;
8 public:
9     CC() {}
10    CC* getInstance() {
11        if (instance == nullptr) {
12            instance = new CC();
13        }
14        return instance;
15    }
16    string toString() {
17        return instance ? "CC object: " + instance->toString() : "null";
18    }
19 };
20
21 int main()
22 {
23     CC cc1;
24     CC* cc2 = cc1.getInstance();
25     cout << cc1.toString() << endl;
26     cout << cc2->toString() << endl;
27     return 0;
28 }
```

Select one or more:

- A.  
CC object: null  
CC object: null
- B.  
CC object: null  
CC object: CC object: null
- C.  
La compilare se indică eroare la linia 17.
- D.  
CC object: null  
null