Final Exam 2021 - Computer Science
– Examples of questions –

Note for students:

The written part of the final exam organized in July/September 2021 will consist of 3 questions, with structure and difficulty similar to the ones presented in this collection, but not necessarily identical; from the three main topics (discrete stuctures and algorithms, programming languages and software engineering; and computing systems) will be selected 2 questions; and a question from the interdisciplinary questions proposed at each of the three topics.

If you find any unclear questions or answers please address your comments to the persons responsible for each topic and to flavia.micota@e-uvt.ro.

Topic I: Discrete stuctures and algorithms

- Adrian Crăciun (adrian.craciun@e-uvt.ro)
- Adrian Spătaru (adrian.spataru@e-uvt.ro)

Topic II: Programming languages and software engineering

- Cristina Mîndruţă (cristina.mindruta@e-uvt.ro)
- Flavia Micota (flavia.micota@e-uvt.ro)
- Daniel Pop (daniel.pop@e-uvt.ro)

Topic III: Computing systems

- Ciprian Pungilă (ciprian.pungila@e-uvt.ro)
- Marian Neagul (marian.neagul@e-uvt.ro)
- Gabriel Iuhasz (gabriel.iuhasz@e-uvt.ro)

Problem

- Cristina Mîndruţă (cristina.mindruta@e-uvt.ro)
- Flavia Micota (flavia.micota@e-uvt.ro)
- Florin Fortiş (florin.fortis@e-uvt.ro)

- Marian Neagul (marian.neagul@e-uvt.ro)
- Gabriel Iuhasz (gabriel.iuhasz@e-uvt.ro)
- Adrian Crăciun (adrian.craciun@e-uvt.ro)
- Adrian Spătaru (adrian.spataru@e-uvt.ro)

Topics

Topic I. Discrete stuctures and algorithms

- *Algorithms:* algorithms implementation, complexity analysis of algorithms, search algorithms, sorting algorithms, recursive algorithms, problem solving techniques (divide and conquer, greedy, dynamic programming)
- *Data structures*: stacks, queues, lists, heaps, trees, dictionaries
- *Graph theory and combinatorics:*
  - *Combinatorics:* counting principles; Stirling numbers; symmetry groups, Polya theory.
  - *Graph Theory Elements:* basic notions and definitions; special classes of graphs; connectivity, shortest paths; spanning trees; transport networks and network flows; colorings.
- *Computational Logic:* Parsing expressions in propositional/predicate logic. Semantics, computing the meaning of expressions (propositional/predicate logic). Truth tables (propositional logic). Validity/satisfiability, logical consequence, logic equivalence (propositional/predicate logic). Reasoning, using reasoning (the role of reasoning). Deduction theorem and its variants. Normal forms for propositional formulas. Resolution, DP, DPLL. Natural style reasoning (propositional/predicate logic). Applications of logic: digital circuit design.
- *Formal languages and automata theory:* grammars, languages, regular expressions, finite automata

Topic II. Programming languages and software engineering

- *Programming Languages:* data types, operators, functions, classes, objects, relations (inheritance, aggregation, composition, dependency)
- *Software Engineering:* the activities of the software development process, agile methods for software development, UML diagrams.
- *Databases:* database modeling, normal forms, SQL queries

Topic III. Computing systems

- *Computers Architecture:* representations of numbers in computers (integers, floating point, radix transformations), (combinatorial) circuit design, microarchitecture: data path, microinstructions, pipelines, memory caching, branch prediction, out-of-order execution, ISA (memory models, registers, instructions, data types), addressing modes, Operating System Level (virtual memory: paging, segmentation).
- *Operating Systems:* concurrent access to resources (critical section problem), communication, CPU scheduling algorithms, paging algorithms, deadlock detection and avoidance algorithms
- *Computer Networks:* protocol encapsulation, datagram-oriented communication, retransmission.

Bibliography

- Bruno Buchberger, *Logic for Computer Science, Unpublished Lecture Notes*, Copyright Bruno Buchberger, 1991, http://staff.fmi.uvt.ro/ adrian.craciun/lectures/logica/pdf/buchberger-logic.pdf
- Mordechai Ben-Ari, *Mathematical Logic for Computer Science*, Ed. Springer Verlag London, 2009
- Adrian Crăciun, *Logic for Computer Science (note de curs)*, http://staff.fmi.uvt.ro/ adrian.craciun
- Th. Cormen, Ch. Leiserson, R. Rivest, C. Stein, *Introduction in algorithms*, Ed. MIT Press, 2008; (trad. lb română) Ed. Teora, 1995
- Allen B. Downey, Chris Mayfield, *Think Java - How to Think Like a Computer Scientis*, Copyright Allen Downe, 2012
- H. Garcia-Molina, J. Widom, *Database Systems: The Complete Book*, ED. Pearson Education Inc, 2009
- John M. Harris, Jeffry L. Hirst, Michael J. Mossinghoff, *Combinatorics and Graph Theory*, Ed. Springer Science+Business Media, 2008
- J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*, Addison-Wesley Longman Publishing Co., Inc., 2006
- I. Jacobson, H. Lawson, Pan-Wei Ng, P.E. McMahon, M. Goedicke, *The Essentials of Modern Software Engineering*, ACM Books, 2019
- B. Kernighan, D. Ritchie, *The C Programming Language*, Ed. Addison-Wesley, 2001
- D. C. Kozen, *Automata and Computability*, Ed. Springer, 1997
- Ian Sommerville, *Software Engineering 9-th ed.*, Ed. Pearson Education Limited, 2016
- Ian Sommerville, *Engineering software products*, Pearson Education, 2020

- Bjarne Stroustrup, *The C++ Programming Language*, ED. Pearson Education, 2005
- Andrew S. Tanenbaum, Todd Austin, *Structured Computer Organization*, Ed. Pearson, 2013
- Andrew S. Tanenbaum, *Computer Networks*, Ed. Pearson, 2011
- Andrew S. Tanenbaum, *Modern Operating Systems*, Ed. Pearson, 2009
- http://agilemodeling.com/essays/umlDiagrams.html

Topic I. Discrete stuctures and algorithms

Algorithms and data structures

1. Let $f(n) = n^2, g(n) = n \log n, h(n) = sin(n)$. Which of the following statements is true? There may be more than one correct answer.

   (a) $f = \Theta(g)$.
   (b) $h = o(g)$.
   (c) $f = \Omega(h)$.
   (d) $h = O(g)$.

2. A recursive program satisfies equation $T(n) = 9T(n/3) + \Theta(n^2)$. What can we say about $T(n)$? There may be several correct answers.

   (a) $T(n) = O(n \log n)$.
   (b) $T(n) = \Omega(n^2)$.
   (c) $T(n) = \Theta(n^2)$.
   (d) $T(n) = O(n^3)$.

3. In which of the following data structures does searching an item have *worst-case* complexity $\Theta(\log n)$? There may be more than one.

   (a) Linked lists.
   (b) Heaps.
   (c) Red-black trees.
   (d) Splay trees.

4. Which binary tree traversal can be used to list all numbers in a binary search tree in sorted order? There may be more than one correct answer.

   (a) Breadth-first search.
   (b) Preorder.
   (c) Inorder.
   (d) Postorder.

5. Which of the following sorting methods is *not* a comparison-based sort? There may be more than one right answer.

   (a) Quicksort.
   (b) Radix Sort.
   (c) Insertion Sort.

(d) HeapSort.

6. Which of the following problem has an easy algorithm with complexity $O(n \log n)$? There may be more than one right answer.

   (a) Given a list L of n integers, find three numbers $x, y, z \in L$ (if they exist) such that $x+y = z$.

   (b) Given a list L of n integers and a target value z, find two numbers in $x, y \in L$ (if they exist) such that $x + y = z$.

   (c) Given a list L of n integers and a target value z, find two numbers in $x, y \in L$ (if they exist) such that $x - y = z$.

   (d) Given a list L of n integers, find three numbers $x, y, z \in L$ (if they exist) such that $x+y+z = 0$.

7. Which of the following algorithms correctly computes the *length* of a longest increasing subsequence problem and has complexity $O(n^2)$? There may be more than one right answer.

   (a) Put the numbers in the list in piles of decreasing numbers. A new number is added greedily to the first pile it can be added to, or starts a new pile if it cannot be added to any existing pile. The length of the LIS is the number of piles.

   (b) We run a greedy algorithm, maintaining a list of increasing numbers. When processing a new number we add it to the LIS if possible, we discard it and proceed otherwise.

   (c) We run a backtracking algorithm, maintaining the list of the biggest LIS seen so far. When encountering a new number we add it to the sequence if possible. If not we backtrack and continue with the next sequence.

   (d) We solve the problem by dynamic programming, computing the length of the LIS subsequence ending in a given term a_k of the sequence. We then take the maximum of the so-computed LIS's.

8. We are given a list of courses, each with a start and an end time. We only have one room and want to schedule as many of these courses as possible. Which of the following algorithms finds an optimal solution? There may be more than one correct answer.

   (a) Greedily choose the shortest courses.

   (b) Greedily choose courses that start first.

   (c) Greedily choose courses that end first.

   (d) Sort courses by their endtime. Compute, for each course C_k, a longest sequence of courses that ends with C_k. Take the best such sequence over all k's.

9. A recursive algorithm reduces solving the problem on inputs of size n to solving four subproblems on size $n/2$ and then combining the results. The combining step takes $f(n)$ steps. We want our algorithm to have complexity $O(n^2)$. Which of the following are acceptable complexities for the combining step? There may be several right answers.

(a) $f(n) = O(1)$.
(b) $f(n) = O(n)$.
(c) $f(n) = O(n \log n)$.
(d) $f(n) = \theta(n^2)$.

10. What is the complexity of inserting a new item in a *sorted* linked list of integers?

(a) $O(1)$.
(b) $\Theta(1)$.
(c) $O(n)$.
(d) $\Theta(\log n)$.

11. Which sorting algorithm is *not* recursive ? There may be multiple correct answers.

(a) Mergesort
(b) Quicksort
(c) Heapsort.
(d) Counting Sort.

12. What data structure can we use to implement an iterative tree traversal ?

(a) Queue
(b) Stack
(c) Hash Table
(d) Red-black tree.

13. What is the complexity of computing the median ?

(a) $\Theta(n)$.
(b) $\Theta(n \log n)$.
(c) $\Theta(1)$.
(d) $\Theta(n^2)$.

14. In a red-black tree which of the following is *not* necessarily true ? There may be several correct answers.

(a) The root is red.
(b) The root is black.
(c) Every path from the root to leaves has the same number of red nodes.
(d) A black parent can only have red children.

15. Which of the following is correct? There may be several correct answers.

    (a) The worst-case complexity of insert, delete, search operations in a binary search tree is $O(\log n)$.

    (b) The worst-case complexity of insert, delete, search operations in an AVL tree is $O(\log n)$.

    (c) The worst-case complexity of insert, delete, search operations in a splay tree is $O(\log n)$.

    (d) The worst-case complexity of insert, delete, search operations in a red-black tree is $O(\log n)$.

16. Which of the following sorting algorithm is *not* subject to the $\Omega(n \log n)$ lower bound for sorting?

    (a) Counting sort.

    (b) Heapsort.

    (c) Radix Sort.

    (d) MergeSort.

17. Consider two strings A = "abcca" and B = "abacbcaa". Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B. Then $x + 100y = \ldots$.

    (a) 304.

    (b) 305.

    (c) 405.

    (d) 205.

18. Suppose we sort $n \geq 3$ integers using quicksort, and after the first pivoting there are equally many numbers on the left of the pivot as they are on the right. Which of the following are true? There may be multiple answers.

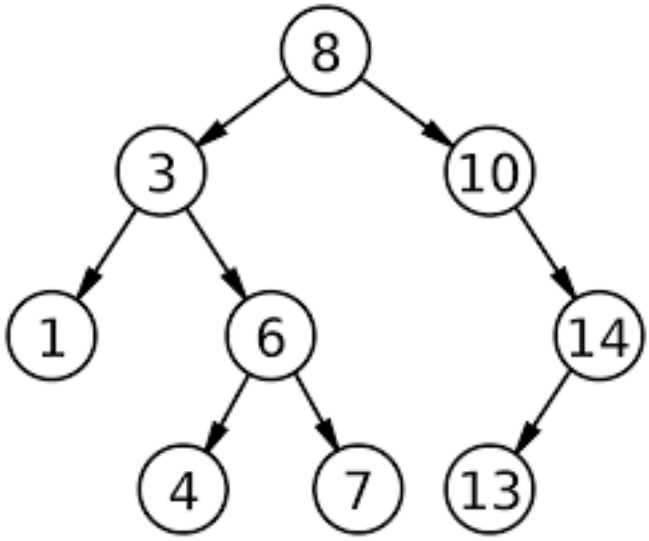
    (a) All the elements could have been the pivot.

    (b) The pivot is the median

    (c) The pivot is **not** the maximum.

    (d) none of the other statements.

19. Which node is certainly colored black in the following BST, if we know it must be a red-black tree? There may be multiple correct answers. Leaf (sentinel) nodes are not represented.



(a) 1
(b) 6
(c) 8
(d) 14

20. What is the **second operation** needed for inserting a value x at the **front** of a singly linked list? We assume the list is represented by one pointer *head* only.

(a) allocate a node with the value x.
(b) $x.next = head$.
(c) $head.prev = x$
(d) $head = x$.
(e) none of the other options.

Graph Theory and Combinatorics

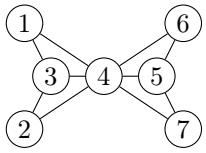
1. Which is the rank of the permutation $\langle 4, 1, 6, 2, 3, 5\rangle$ in lexicographic order?

(a) 376 (b) 378 (c) 380 (d) 720

2. A message is a sequence of two types of signals: of type A which last for 1 second and of type B which last for 2 seconds. E.g., the message ABAAB lasts 7 seconds. How many different messages last 10 seconds?

(a) 68 (b) 32 (c) 89 (d) 144

3. In how many ways can be colored the following configuration by using 2 colors: red and black?



(a) 64 (b) 48 (c) 128 (d) 36

4. In how many ways can we split a group of 50 persons in 49 nonempty groups?

(a) 50 (b) 1225 (c) 49 (d) 2450 (e) 19600

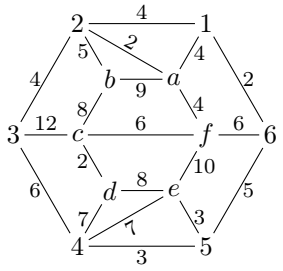
5. In how many ways can we choose 5 fruits from a market stall if on the stall there are: 5 apples, 5 pears and 5 apricots?

(a) 21 (b) 56 (c) 120 (d) 6 (e) 30

6. Which of the following recurrence relations hold for any $n > k > 0$?

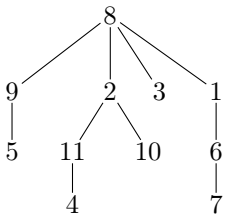
(a) $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$
(b) $\left\{ {n \atop k} \right\} = \left\{ {n-1 \atop k} \right\} + k \cdot \left\{ {n-1 \atop k-1} \right\}$
(c) $\left\{ {n \atop k} \right\} = k \cdot \left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$
(d) $\left[{n \atop k} \right] = \left[{n-1 \atop k} \right] + (n-1) \cdot \left[{n-1 \atop k-1} \right]$

7. Which is the minimum weight spanning tree of the following connected graph? (hint: apply the Kruskal algorithm)



(a) 43
(b) 40
(c) 36
(d) 41

8. Which is the Prüfer sequence of the following tree?



(a) $8, 3, 4, 5, 7, 6, 1, 9, 10, 11, 2$

(b) $3, 4, 5, 7, 6, 1, 8, 2, 10, 11$

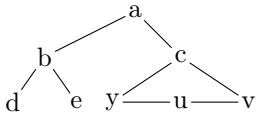
(c) $8, 11, 9, 6, 1, 8, 8, 2, 2$

(d) $8, 11, 9, 6, 1, 8, 2, 2, 8$

9. How many different trees with 5 nodes, labeled with numbers from 1 to 5, there exist?

   (a) 10 (b) 273 (c) 32 (d) 120 (e) 125

10. In how many ways can we color the following graph G with three colors?



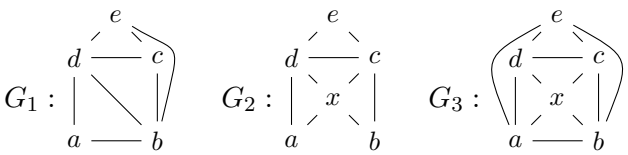
(a) 0

(b) 288

(c) 336

(d) 1874

11. Which of the following graphs are eulerian graphs and which ones are not?



   (a) $G_1 : no$, $G_2, G_3 : yes$
   (b) $G_1 : yes$, $G_2, G_3 : no$
   (c) $G_1, G_2 : yes$, $G_3 : no$
   (d) none
   (e) all of them are eulerian

12. In the Internet, which is made up of interconnected physical networks of computers, each network connection of a computer is assigned an Internet address. In IPv4 Internet protocol, every Internet address is a 32-bit string, made of a network number (*netid*) followed by a host number (*hostid*). There are 3 types of Internet addresses:

   (a) Class A: these are of the form $0\,\underbrace{b_1b_2b_3b_4\ldots b_7}_{\text{netid}}\,\underbrace{b_8b_9\ldots b_{30}b_{31}}_{\text{hostid}}$

   (b) Class B: these are of the form $10\,\underbrace{b_2b_3b_4\ldots b_{15}}_{\text{netid}}\,\underbrace{b_{16}b_{17}\ldots b_{30}b_{31}}_{\text{hostid}}$

   (c) Class C: these are of the form $110\,\underbrace{b_3b_4b_5\ldots b_{23}}_{\text{netid}}\,\underbrace{b_{24}b_{25}\ldots b_{30}b_{31}}_{\text{hostid}}$

   How many different IPv4 addresses are available for the internet network connections?

(a) 2^{29} (b) $2^{30} \cdot 2^{31}$ (c) 2^{31} (d) $7 \cdot 2^{29}$ (e) 2^{32}

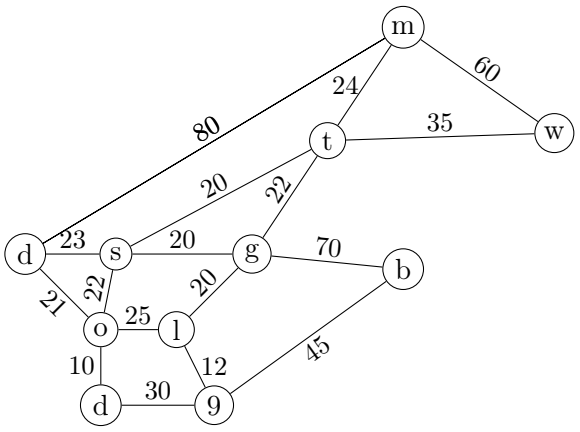
13. In a drawer there are 8 brown socks and 12 black socks. A child picks socks from the drawer at random in the dark. How many socks must he take out to be sure that he picked at least two black socks?

    (a) 3 (b) 10 (c) 14 (d) 9

14. How many integer numbers between 1 and 1000 are divisible with 7, but are not divisible with 3?

    (a) 93 (b) 95 (c) 92 (d) 136

15. Consider the following weighted graph



    What is the total weight of a minimum spanning tree of this graph?

    (a) 229 (b) 216 (c) 230 (d) 234

Computational Logic

1. Consider the predicate logic language that contains the following symbols:

   - variables, indicated with lower case letters
   - function symbols $\mathcal{F}$: $+$ binary infix, $-$ unary prefix, $*$ binary infix.
   - predicate symbols $\mathcal{P}$: $=, <, \leq$ all binary, infix.
   - constant symbols $\mathcal{C}$: $0, 1$.

   Which of the following are terms over this language?

   (a) $(0 * x) - 1$,

(b) $1 + (z * x) < 0$,
(c) $x + ((-1) * 0)$,
(d) $0 * (y + 1)$.

2. For the following propositional formulae, and for the truth valuation $\{P, \neg Q\}$:

(a) $((P \Rightarrow Q) \wedge ((\neg Q) \wedge P))$ evaluates to $\mathbb{T}$,
(b) $((P \Rightarrow Q) \Rightarrow (Q \Rightarrow P)$ evaluates to $\mathbb{T}$,
(c) $((\neg(P \vee Q)) \wedge (\neg Q))$ evaluates to $\mathbb{F}$.

3. Which of the following statements are true:

(a) if a propositional formula is valid then it is satisfiable,
(b) if a propositional formula is not valid then it is satisfiable,
(c) if a propositional formula is not valid then its negation is satisfiable,
(d) if a propositional formula is not valid, then its negation is valid.

4. What is the relation between propositions

$$(F \wedge G) \Rightarrow H$$

and

$$F \Rightarrow (G \Rightarrow H).$$

(a) they are logically equivalent,
(b) the first one is a logical consequence of the second one,
(c) the second one is a logical consequence of the first one,
(d) they are not related in any of the ways above.

5. The formula:

$$P \Leftrightarrow Q$$

is

(a) logically equivalent to the conjunction of,
(b) a logical consequence of,
(c) logically equivalent to the disjunction of

the formulas:

$$Q \Rightarrow R,$$
$$R \Rightarrow (P \wedge Q),$$
$$P \Rightarrow (Q \vee R).$$

6. Which of the following formulae are in Disjunctive Normal Form?

(a) P,
(b) $\neg P \vee Q$,
(c) $P \wedge \neg Q \wedge S$,
(d) $(P \wedge \neg Q \wedge S) \vee \neg S$.

7. What is a resolvent of clauses $\{P, \neg Q, R\}$ and $\{\neg P, Q, S\}$?

(a) $\emptyset$,
(b) $\{P, \neg P, R, S\}$,
(c) $\{R, S\}$.

8. To establish whether a formula G is a logical consequence of formulae $F_1, \ldots, F_n$, which of the following methods can be applied:

(a) check that $(F_1 \wedge \ldots \wedge F_n) \Rightarrow G$ is unsatisfiable,
(b) check that $\neg F_1 \vee \ldots \vee \neg F_n \vee G$ is unsatisfiable,
(c) check that $\neg F_1 \vee \ldots \vee \neg F_n \vee G$ is valid,
(d) check that $F_1 \wedge \ldots \wedge F_n \wedge \neg G$ is unsatisfiable.

9. There is a formula which is logically equivalent to

$$\left(\begin{array}{c} ((P_1 \Rightarrow (P_2 \vee P_3) \wedge (\neg P_1 \Rightarrow (P_3 \vee P_4)))) \\ \wedge \\ ((P_3 \Rightarrow (\neg P_6)) \wedge (\neg P_3 \Rightarrow (P_4 \Rightarrow P_1))) \\ \wedge \\ (\neg(P_2 \wedge P_5)) \wedge (P_2 \Rightarrow P_5) \end{array} \right) \Rightarrow \neg(P_3 \Rightarrow P_6).$$

which contains only propositional connectives taken from:

(a) $\{\neg, \vee\}$,
(b) $\{\vee, \wedge\}$,
(c) $\{|\}$,
(d) $\{\bot, \Rightarrow\}$.

where $|$ is the NAND connective (i.e. $P|Q = \neg(P \wedge Q)$).

10. The clause set corresponding to the formula

$$(\neg P \Rightarrow (Q \wedge R)) \Rightarrow (P \Rightarrow \neg Q)$$

is:

(a) $\{\{\neg P, \neg Q\}\}$,

(b) $\{\{P, \neg Q\}, \{P, R\}, \{\neg Q, R\}\}$,
(c) $\{\{P, \neg Q, \neg R\}, \{P, Q, R\}, \{\neg P, \neg Q, R\}\}$.

11. Consider the clause set containing the following clauses:

(1) $\{P, Q, \neg R\}$,
(2) $\{\neg P, R\}$,
(3) $\{P, \neg Q, S\}$,
(4) $\{\neg P, \neg Q, \neg R\}$,
(5) $\{P, \neg S\}$.

The formula corresponding to this clause set is:

(a) valid,
(b) satisfiable,
(c) unsatisfiable.

12. The Davis-Putnam method returns the answer satisfiable:

(a) when the empty clause is generated,
(b) when the empty clause set is generated,
(c) when no new clauses can be generated, and the empty clause is not in the clause set.

13. To prove a goal G when a disjunction $A \vee B$ is known:

(a) assume A and prove G then assume B and prove G,
(b) assume A and prove G,
(c) assume $\neg A$ and prove B and G.

14. Let P, Q, R be propositional variables. Which of the following propositional formulas correspond to the boolean function with 3 arguments that returns $\mathbb{T}$ when its inputs represent the binary encoding of a prime number:

(a) $P \wedge Q \wedge \neg R$;
(b) $(\neg P \wedge Q) \vee (P \wedge R)$;
(c) $(\neg P \wedge Q \wedge \neg R) \vee (((\neg P \wedge Q) \vee P) \wedge R)$;
(d) $((\neg P \wedge Q) \vee (P \wedge \neg Q) \vee (P \wedge Q))$.

15. Consider the clause set

$$\{\{\neg P, \neg Q, R\}, \{P, \neg Q, \neg R\}, \{\neg P, R\}, \{P, \neg Q, R\}, \{\neg Q, \neg R\}\}.$$

The first step in running the Davis Putnam method is:

(a) applying the splitting rule (using the literal $\neg P$);
(b) applying the pure literal rule (where the pure literal is $\neg Q$);
(c) applying the resolution rule;
(d) applying the one literal rule (using the last clause).

Formal Languages and Authomata Theory

1. For the mobile application you are implementing, you have to store in the database only the Romanian phone numbers (e.g. +40724370012 / 0040724370012) and email addresses with a Romanian domain (e.g. ion@domeniu.ro; popescu.maria@dom.ro). The phone numbers must have exactly 9 digits after the Romanian prefix 0040/+40. The name of the domain and the extension of the email are written in lowercase letters and are case sensitive. Which of the following regular expressions express your needs:

   (a) $(+407|00407)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9);$
   $((a|...|z|A|...|Z)^+.(a|...|z|A|...|Z)^+@(a|...|z)^+.ro)\,|((a|...|z|A|...|Z)^+@(a|...|z)^*.ro)$

   (b) $(+407|00407)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9);$
   $((a|...|z|A|...|Z)^+.(a|...|z|A|...|Z)^+@(a|...|z)^+.ro)\,|\,((a|...|z|A|...|Z)^+@(a|...|z)^+.ro)$

   (c) $(+407|00407)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9);$
   $((a|...|z|A|...|Z)^+.(a|...|z|A|...|Z)^+@(a|...|z)^+.ro)\,((a|...|z|A|...|Z)^+@(a|...|z)^+.ro)$

   (d) $(+407|00407)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9)(0|...|9);$
   $((a|...|z|A|...|Z)^+.(a|...|z|A|...|Z)^+@(a|...|z)^+.ro)\,|\,((a|...|z|A|...|Z)^+@(a|...|z)^+.ro)$

2. Assume you have to build push-down automaton simulating a natural language processor recognizing numerical palindromes of even length. Which of the following grammars $G = (V_N, V_T, S, P)$ help you building the solution:

   (a) $V_N = \{S\}, V_T = \{0, ..., 9\}, S, P = \{S \to 0S0|1S1|...|9S9|\lambda\}$

   (b) $V_N = \{S\}, V_T = \{0, ..., 9\}, S, P = \{S \to 0S0|1S1|...|9S9|0|...|9\}$

   (c) $V_N = \{S\}, V_T = \{0, ..., 9\}, S, P = \{S \to 0S0|1S1|...|9S9|0|...|9|\lambda\}$

3. Consider the set of all strings of balanced parentheses of two types: round and square. An example of where these strings come from is as follows. If we take expressions in C, which use round parentheses for grouping and for arguments of function calls, and use square brackets for array indexes, and drop out everything but the parentheses, we get all strings of balanced parentheses of these two types. For example, $f(a[i] * (b[i][j], c[g(x)]), d[i])$ becomes the balanced parentheses string $([](([][]()))[])$. A grammar $G = \{V_N, V_T, S, P\}$ for generating the strings of round and squared parentheses that are balanced is:

   (a) $V_N = \{S\}$, $V_T = \{(,), [,]\}$, S, $P = \{S \to SS|[S]|(S)|()|[]\}$

   (b) $V_N = \{S\}$, $V_T = \{(,), [,]\}$, S, $P = \{S \to (S)|[S]|()|[]|\lambda\}$

   (c) $V_N = \{S\}$, $V_T = \{(,), [,]\}$, S, $P = \{S \to SS|(S)|[S]|\lambda\}$

4. Are there sets of strings that cannot be described by any regular expression?

   (a) Bitstrings with equal number of 0s and 1s.

   (b) Decimal strings that represent prime numbers.

   (c) DNA strings that are Watson-Crick complemented palindromes.

5. In genomics, a nucleic acid is represented by one of the letters actg. A genome is a string of nucleic acids. A Fragile X Syndrome pattern is a genome having an occurrence of gcg, followed by any number of cgg or agg triplets, followed by ctg (The number of triplets correlates with Fragile X Syndrome, a common cause of mental retardation.). Which of the following genomes contain a Fragile X Syndrome?

   (a) gcggcgtgtgtgcgagagagtgggtttaaagctggcgcggaggcggctggcgcggaggctg
   (b) gcggcgtgtgtgcgagagagtgggtttaaagctggcgcggaggcggctggcgcggaggcte
   (c) gcggcgtgtgtgcgagagagtgggtttaaagctggcgcggaggcggctg
   (d) gcggcgtgtgtgcgagagagtgggtttaaagctggcgcggaggcggctg

6. Which of the following strings the DFA accepts?

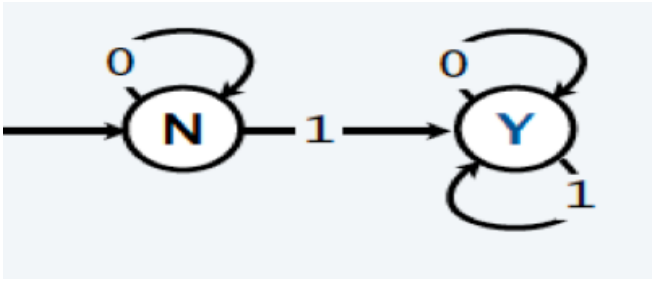


Figure 1: DFA recognizing binary strings with certain pattern

   (a) Bitstrings that end in 1.
   (b) Bitstrings with an equal number of occurrences of 01 and 10.
   (c) Bitstrings with more 1s than 0s.
   (d) Bitstrings with an equal number of occurrences of 0 and 1.
   (e) Bitstrings with at least one 1.
   (f) None.

7. In computational biochemistry, an amino acid is represented by one of the characters CAVLIMCRKHDENQSTYFWP. A protein is a string of amino acids. A C_2H_2-type zinc finger domain signature is:

   (a) C followed by 2, 3, or 4 amino acids, followed by
   (a) C followed by 3 amino acids, followed by
   (a) L, I, V, M, F, Y, W, C, or X followed by 8 amino acids, followed by
   (a) H followed by 3, 4, or 5 amino acids, followed by H.

   Which of the following proteins is in the C2H2-type zinc finger domain?

(a) CAASCGGPYACGGWAGYHAGWH.

(b) CAVLIMCRKHDENQSTYFWP.

(c) CAASCGGPYACGGWAGYHGWH.

8. In computational biochemistry, an amino acid is represented by one of the characters CAVLIMCRKHDENQSTYFWP. A protein is a string of amino acids. A C_2H_2-type zinc finger domain signature is:

   - C followed by 2, 3, or 4 amino acids, followed by
   - C followed by 3 amino acids, followed by
   - L, I, V, M, F, Y, W, C, or X followed by 8 amino acids, followed by
   - H followed by 3, 4, or 5 amino acids, followed by H.

   Which of the following Java code implements a Generalized Regular Expression Pattern (GREP) matcher for the C_2H_2-type zinc finger domain signature?

   (a)
   ```
   String re = "C.{2,4}C...[LIVMFYWC].{8}H.{3,5}H";
   String zincFinger = "CAASCGGPYACGGAAGYHAGAH";
   boolean test = zincFinger.matches(re);
   ```

   (b)
   ```
   String re = "C.{2,4}C...[^LIVMFYWC].{8}H.{3,5}H";
   String zincFinger = "CAASCGGPYACGGAAGYHAGAH";
   boolean test = zincFinger.matches(re);
   ```

   (c)
   ```
   String re = "C.{2,4}C...{LIVMFYWC}.{8}H.{3,5}H";
   String zincFinger = "CAASCGGPYACGGAAGYHAGAH";
   boolean test = zincFinger.matches(re);
   ```

9. Which of the following Java code implements a Generalized Regular Expression Pattern (GREP) matcher for email addresses with the *edu* or *com* domains?

   (a)
   ```
   String re = "[a-z]*@([a-z]+\.)+(edu|com)";
   String zincFinger = "wayne@cs.princeton.edu";
   boolean test = zincFinger.matches(re);
   ```

   (b)
   ```
   String re = "[a-z]+@([a-z]+\.)+(edu|com)";
   String zincFinger = "wayne@cs.princeton.edu";
   boolean test = zincFinger.matches(re);
   ```

   (c)
   ```
   String re = "[^a-z]+@([a-z]+\.)+(edu|com)";
   String zincFinger = "wayne@cs.princeton.edu";
   boolean test = zincFinger.matches(re);
   ```

10. Which of the following strings match the regular expression $a^*bb(ab|ba)^*$

(a) *abb*
(b) *aaba*
(c) *abba*
(d) *bbbaab*
(e) *cbb*
(f) *bbababbab*

Interdisciplinary subjects

1. Let M_n be the set of matrices of dimension $n \times n$ whose elements are 0 or 1. For every two matrices $A, B \in M_n$ and $1 \leq k \leq n$ we define the operations $A \oplus B$ and $A \odot_k B$ as follows: $A \oplus B = C \in M_n$ if $C[i][j] = \max(A[i][j], B[i][j])$ and $A \odot_k B = D \in M_n$ if $D[i][j] = A[i][k] \cdot B[k][j]$ for all $1 \leq i, j \leq n$.

   Let G be a simple undirected graph with n nodes numbered from 1 to n, and $A_G \in M_n$ be its adjacency matrix.

   - The number of simple undirected graphs with n nodes labeled from 1 to n is (1).
   - Let I_n be the identity matrix of dimension $n \times n$, and $1 \leq p \leq n$. Consider the following algorithm to compute the matrix $B \in M_n$:

     $B = A_G \oplus I_n$;
     for $k := 1$ to p **do**
         $C = B \oplus (B \odot_k B)$;
         $B = C$;

     The runtime complexity of this algorithm is (2).
   - Which of the following assertions holds when $B[i][j] = 1$? (3)

     (c1) There is a path of length at most $p + 1$ from node i to node j.
     (c2) There is a path of length p from node i to node j.
     (c3) There is a path from node i to node j which goes through node p.
     (c4) There is a path from node i to node j which goes through all nodes from the set $\{1, 2, \ldots, p\}$.

   Answers

   (1) $2^{n(n-1)/2}$ from $\{2^{n(n-1)/2}, 2^n, n(n-1)/2, n-1, n \cdot (n-1)\}$
   (2) $\Theta(p \cdot n^2)$ from $\{\Theta(p \cdot n^2), \Theta(p \cdot n), \Theta(p \cdot n^3), \Theta(n), \Theta(p \cdot 2^n)\}$
   (3) (c1) from {(c1),(c2),(c3),(c4)}.

   Explanations:

   (a) Question related to combinatorics: the set of edges of G can be any subset of the set $M = \{i-j \mid 1 \leq i \neq j \leq n\}$. M has $n(n-1)/2$ elements and $2^{n(n-1)/2}$ subsets. Therefore, there are $2^{n(n-1)/2}$ simple graphs of this kind.

   (b) Question about algorithms and data structures. The operations $\oplus$ and $\odot_k$ have runtime complexity $\Theta(n^2) \Rightarrow B$ is computed in time $\Theta(p \cdot n^2)$.

(c) Question of logic: $B[i][j] = 1$ if and only if there is a path from i to j whose intermediate nodes are in the set $\{1, 2, \ldots, p\}$. It is easy to conclude that (c1) is the only assertion which must hold in this case.

2. A binary min-heap is a data structure which models an almost complete binary tree which has the *heap property*: For every node N with a parent P, the key of P is smaller than the key of N.

   The array-based implementation of a binary min-heap with n nodes is an array $A[0..2^m - 1]$ with two extra attributes: the capacity $A.length = 2^m - 1$ and the size $A.size = n$, such that $A.size \leq A.length$. The elements in the nodes of the binary min-heap are stored in the first n elements of A: the root is stored in $A[0]$, and if N is the left (resp. right) child of a node P stored in $A[i]$ then N is stored in $A[2 \cdot i + 1]$ (resp. $A[2 \cdot i + 2]$).

   For $0 \leq i, j < n$ we define the relation $grandpa(i, j)$ if $A[j]$ stores the parent of the parent of $A[i]$. The formula which defines the relation $grandpa(i, j)$ in a binary min-heap is (1)

   (a1) $(4 \cdot j + 3 \leq i) \wedge (i \leq 4 \cdot j + 6)$.

   (a2) $(4 \cdot i + 3 \leq j) \wedge (j \leq 4 \cdot i + 6)$.

   (a3) $j = \lfloor i/4 \rfloor$.

   (a4) $(4 \cdot j + 1 \leq i) \wedge (i \leq 4 \cdot j + 2)$.

   (a5) $(4 \cdot i + 1 \leq j) \wedge (j \leq 4 \cdot i + 2)$.

   The maximum number of nodes in a binary min-heap with depth h is (2)

   (b1) $2^{h+1} - 1$.

   (b2) 2^h.

   (b3) $h^2 - 1$.

   (b4) $2^{h-1} + 1$.

   If A is a binary min-heap then A is sorted in the ascending order of the keys from its nodes: (3)

   If A is sorted in the ascending order of the keys from its nodes then A is a binary min-heap: (4)

   The runtime complexity of the deletion of the node with minimum key from a binary min-heap with n nodes is (5)

   ANSWERS

   (1) (a1) from {(a1), (a2), (a3), (a4), (a5)}
   (2) (b1) from {(b1), (b2), (b3), (b4)}
   (3) no from {yes, no}
   (4) yes from {yes, no}
   (5) $O(\log n)$ din $\{O(1), O(\log n), \Theta(n \log n), \Theta(n)\}$.

3. Let L be the language generated by the grammar $G = (V_N, V_T, S, P)$ where $V_N = \{S, A, B, C, X, Y, Z\}$, $V_T = \{a, b, c\}$, and

$$P = \{\ S \to \lambda \mid AX \mid BY \mid CZ,$$
$$A \to \lambda \mid BY \mid CZ,$$
$$B \to \lambda \mid AX \mid CZ,$$
$$C \to \lambda \mid AX \mid BY,$$
$$A \to a, B \to b, C \to c\}.$$

Select the correct answers. L is a regular language: (1)

Which of the following statements is true? (2)

(a1) $L = \{w \in V_T^* \mid w$ does not contain consecutive identical letters$\}$.
(a2) $L = \{w \in V_T^* \mid w$ contains identical consecutive letters$\}$.
(a3) $L = \{w \in V_T^* \mid w$ contains the substring $abc\}$.
(a4) $L = \{\lambda\}$.

Let s_n be the number of strings from L with length n. For every $n > 1$, the following recursive relation holds: (3)

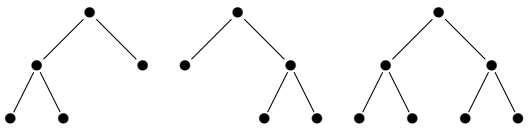
(b1) $s_n = 2 \cdot s_{n-1}$
(b2) $s_n = 3 \cdot s_{n-1}$
(b3) $s_n = s_{n-1} + 2 \cdot s_{n-2}$
(b4) $s_n = 3\, s_{n-1} + s_{n-2}$

L contains (4) strings of length 4.

Answers

(1) yes	from {yes, no}
(2) (a1)	from {(a1), (a2), (a3), (a4)}
(3) (b1)	from {(b1), (b2), (b3), (b4)}
(4) 24	from $\{16, 24, 32, 81, 243\}$

4. A binary tree T is *full* if either T has one node, or the root of T has two children which are roots of full binary trees. For instance, a full binary tree with depth 2 has one of the following three shapes:



Fill in the sentences with the correct missing phrases.

The number of nodes in a full binary tree must be (1). The number of leaves in a full binary tree with n nodes is (2). The worst-case runtime complexity in a binary search tree which is full and has n nodes is (3).

Let s_n be the total number of shapes of full binary trees with depth n. From the previous picture, we see that $s_2 = 3$. Also, it is easy to see that $s_0 = s_1 = 1$. Which of the following recursive formulas holds for all $n > 3$: (4).

(a) $2 \cdot s_{n-1} \sum_{k=0}^{n-2} s_k + (s_{n-1}^2 - s_{n-1})$
(b) s_{n-1}^2
(c) $2 \cdot s_{n-1} \sum_{k=0}^{n-1} s_k$
(d) $s_{n-1} + s_{n-2}$
(e) $s_{n-1} \cdot s_{n-2}$

Answers

(1) odd — from {odd, even, a power of 2, a power of 2 minus 1}
(2) $(n+1)/2$ — from $\{(n+2)/2, n-2, n-3, n(n-1)/2\}$
(3) $\Theta(n)$ — from $\{O(1), \Theta(n), \Theta(\log n), \Theta(n \log n)\}$
(4) (a) — from {(a),(b),(c),(d),(e)}

Explanations. (1) and (2) require logical reasoning by induction. (3) requires logical reasoning and knowledge about the worst-case runtime complexity of search in binary search trees (Data Structures). (4) requires familiarity with the counting principles of combinatorics: the rules of sum, product, and the principle of inclusion and exclusion.

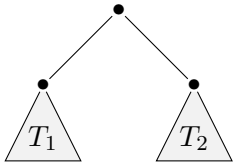
(1) Options 2, 3, 4 can be easily refuted by counter examples. Option 1 can be proved easily by induction.
We prove by induction on n that the number of nodes in a full binary tree is odd. If $n = 1$ then the tree consists of one node, and 1 is odd. If $n > 1$ then the root of the tree has 2 children which are roots of some full subtrees T_1 and T_2. Let n_1 and n_2 be the number of nodes in T_1 and T_2. Then $n = n_1 + n_2 + 1$. By induction hypothesis for T_1 with number of nodes $n_1 < n$, we learn that n_1 is odd. By induction hypothesis for T_2 with number of nodes $n_2 < n$, we learn that n_2 is odd too. It follows that $n = n_1 + n_2 + 1$ is odd too.

(2) We can prove by induction that the total number of leaves is $(n+1)/2$.
The smallest full binary trees has $n = 1$ nodes and 1 leaf, which is $(n+1)/2$. If $n > 1$ then the tree has $n = n_1 + n_2 + 1$ nodes, where n_1 is the number of nodes in the left subtree, and n_2 is the number of nodes in the right subtree. By induction hypotheses, the left subtree has $(n_1 + 1)/2$ leaves, and the the right subtree has $(n_2 + 1)/2$ leaves, thus the total number of leaves of the tree is

$$\frac{n_1 + 1}{2} + \frac{n_2 + 1}{2} = \frac{n_1 + n_2 + 1 + 1}{2} = \frac{n + 1}{2}.$$

(3) We know that the worst-case runtime complexity of search in a binary search tree is proportional to the maximum possible depth of the tree. It is easy to prove by induction that the maximum depth of a full binary tree with $n = 2 \cdot m + 1$ nodes is m, which is $\Theta(n)$. Thus, the correct answer is $\Theta(n)$.

(4) A full binary tree with depth n has the shape



where one or both of T_1, T_2 has depth $n-1$. We distinguish three disjoint cases:

1) T_1 has depth $n-1$ and T_2 has depth smaller than $n-1$. By the rules of sum and product, there are $s_{n-1} \cdot \sum_{k=0}^{n-2} s_k$ such shapes.
2) T_2 has depth $n-1$ and T_2 has depth smaller than $n-1$. By the rules of sum and product, there are $s_{n-1} \cdot \sum_{k=0}^{n-2} s_k$ such shapes.
3) Both T_1 and T_2 have depth $n-1$. By the principle of inclusion and exclusion, there are $s_{n-1}^2 - s_{n-1}$ such shapes.

By the rule of sum, we learn that $s_n = 2 \cdot s_{n-1} \sum_{k=0}^{n-2} s_k + (s_{n-1}^2 - s_{n-1})$. Thus, the correct answer is (a).

4. Let $[a_1, b_1]$, $[a_2, b_2]$, ..., $[a_n, b_n]$ be n intervals on the real axis. We assume that all endpoints are distinct real numbers.

We build a graph G as follows:

- the edges of the graph are the n intervals
- two nodes are connected with an edge if the corresponding intervals do intersect.

Indicate an algorithm with runtime complexity $O(n \log n)$ which computes a maximum clique in graph G. We recall that a clique is a set of nodes C such that there is an edge between every two distinct nodes from C.

Answer

(Step 1) We sort the numbers a_1, b_1, a_2, b_2, ..., a_n, b_n using, for example, Mergesort.

(Step 2) We traverse these numbers from left to right.

(Step 3) For every moment t we count how many intervals are "open" at that moment (left endpoint $\leq t$, right endpoint $> t$). The difference between the number of left endpoints and the number of right endpoints coincides with this number.

(Step 4) We choose a point t from the set $\{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$ which maximizes this parameter. The procedure to compute the maximum is the usual one.

(Step 5) For every interval, we verify if it contains point t. We obtain a clique of maximum size.

Answers Topic I

Algorithms and Data Structures

1. (b),(c),(d)
2. (b),(d)
3. (c)
4. (c)
5. (b)
6. (b),(c)
7. (a),(d)
8. (c),(d)
9. (a),(b),(c)
10. (c)
11. (c),(d)
12. (b)
13. (a)
14. (a),(c),(d)
15. (b),(d)
16. (a),(c)
17. (d)
18. (b), (c)
19. (a), (c), (d)
20. (b)

Graph Theory and Combinatorics

1. (b)
2. (c)
3. (b)
4. (b)
5. (a)
6. (a),(c)
7. (b)
8. (c)
9. (e)
10. (b)
11. (a)
12. (d)
13. (b)
14. (b)
15. (a)

Computational Logic

1. (c),(d)
2. (b),(c)
3. (a),(c)
4. (a),(b),(c)
5. (b)
6. (a),(b),(c),(d)
7. (b)
8. (c),(d)
9. (a),(c),(d)
10. (a)
11. (b)
12. (b),(c)
13. (a)
14. (b), (c)
15. (b)

Formal Languages and Automata Theory

1. (b)
2. (a)
3. (b)
4. (a), (b), (c)
5. (a)
6. (f)
7. (a)s
8. (a)
9. (b)
10. (a), (d)

Topic II. Programming languages and software engineering

Python Language

1. Analyze the following code:

```
class A:
  def __init__(self, s):
      self.s = s
  def print(self):
      print(s)
a = A("Welcome")
a.print()
```

   (a) The program has an error because `class A` does not have a constructor.

   (b) The program has an error because `class A` should have a `print` method with signature `print(self, s)`.

   (c) The program has an error because `class A` should have a `print` method with signature `print(s)`.

   (d) The program would run if the line `print(s)` is changed to `print(self.s)`.

2. What is the result of the execution of the following code?

```
x = 2
for i in range(x):
     x += 1
     print (x)
```

   (a) 0 1 2 3 4 ...

   (b) 0 1

   (c) 3 4

   (d) 0 1 2 3

3. Which is the output of the following code?

```
names1 = ['Amir', 'Barry', 'Chales', 'Dao']
names2 = names1
names3 = names1[:]
names2[0] = 'Alice'
names3[1] = 'Bob'
sum = 0
for ls in (names1, names2, names3):
```

```
    if ls[0] == 'Alice':
        sum += 1
    if ls[1] == 'Bob':
        sum += 10

print(sum)
```

(a) 11
(b) 12
(c) 21
(d) 22
(e) 33
(f) 0

4. Which of the following is a features of DocString?

   (a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods
   (b) Provide a convenient way of associating documentation only with Python classes, and methods
   (c) All functions should have a docstring
   (d) Providing documentation to a application is time-waste
   (e) Docstrings can be accessed by the __doc__ attribute on objects

5. Which of the following statements are TRUE about generators:

   (a) Generators are lazy in nature
   (b) Generators are eager in nature
   (c) Generators could make your program to perform faster if the values are generated only once
   (d) Generators could make your program to perform faster if the values are generated multiple times
   (e) Generators could make your program to perform slower

6. Assertion

   (a) Best way to test precondition violation
   (b) Can be disabled in production environments
   (c) An example of good defensive programming
   (d) Are used for debugging in production environments

(e) Assersions and exceptions are used to provide feedback to the client

7. What type of knowledge is used by computers?

   (a) Declarative knowledge for defining programming paradigms
   (b) Declarative knowledge for describing the algorithms
   (c) Imperative knowledge for defining programming paradigms
   (d) Imperative knowledge for describing the algorithms
   (e) No knowledge
   (f) Random knowledge

8. Which is the output of the following code?

```
class A:
  def __init__(self):
     self.calcI(30)
  def calcI(self, i):
    self.i = 2 * i;

class B(A):
  def __init__(self):
    super().__init__()
    print("i from B is", self.i)
  def calcI(self, i):
    self.i = 3 * i;

b = B()
```

   (a) Just the `__init__` method of class `B` gets invoked.
   (b) The `__init__` method of class `A` gets invoked and it displays "`i from B is 0`".
   (c) The `__init__` method of class `A` gets invoked and it displays "`i from B is 60`".
   (d) The `__init__` method of class `A` gets invoked and it displays "`i from B is 90`".

9. Which of the following statements are true regarding the opening modes of a file?

   (a) When you open a file for reading, if the file does not exist, an error occurs.
   (b) When you open a file for writing, if the file does not exist, an error occurs.
   (c) When you open a file for reading, if the file does not exist, the program will open an empty file.
   (d) When you open a file for writing, if the file does not exist, a new file is created.

(e) When you open a file for writing, if the file exists, the existing file is overwritten with the new content.

10. Which of the following statements are true about the following code?

```
s=0
try:
  f = open("test.txt",encoding = 'utf-8')
  for i in f:
      s += len(i)
finally:
  f.close()

print(s)
```

(a) the program will print the size of an existing "test.txt" file, in bytes

(b) the program will crash because i is an integer

(c) the program will crash if the file does not exist

(d) the program will print 0 if the file does not exist

C Language

1. Assuming the prototype (declaration) of `square` is in sight before the call

```
int a=2;
square(a);
...
```

and `square` is defined as:

```
void square(int x){
    x=x*x;
}
```

(a) after the call, a has value 2

(b) after the call, a has value 4

(c) the definition of `square` does not serve the purpose of squaring the actual argument in the caller function

(d) the definition of `square` is formally correct

2. What's wrong with the sequence:

```
int t[N], *low=t, *mid, *high=&t[N-1];
mid = (low+high) / 2;
```

(a) addition of two pointers is an illegal operation

(b) a pointer can't be initialized with an array

(c) the initialization of low and high is incorrect

(d) mid can't be initialized with a real value (when low+high is odd!)

3. Knowing that the . and the -> operator have equal precedence, higher than the precedence of the * operator and assuming the following declarations, which of the expressions bellow are correct?

```
struct point {
    int x, y;
};
struct rectangle{
    struct point p1, p2;
} *r[N];
```

(a) `r[i].p1.x`

(b) `r[i]->p1.x`

(c) `(*r[i]).p1.x`

(d) `*r[i].p1.x`

4. Which of the following storage classes is implicit, due to the place of the variable's declaration:

(a) `static` (b) `auto` (c) `register` (d) `extern`

5. Can two functions, neither of whom calls the other, communicate (share data)?

(a) no

(b) yes, through messages

(c) eventually, through global variables

6. Given `float x=2.5;` the value of the expression `3.0*x+10/4` is:

(a) 10.0 (b) 9.5 (c) of type `double` (d) of type `float`

7. Which of the following name predefined integer data types in C

(a) char (b) double (c) short (d) int (e) byte (f) long

8. The type of `p`, used in the expression `p->m` is:

   (a) pointer to some structure
   (b) the type of `m`
   (c) void *

9. Knowing that "0123456789" is the memory address where the string constant is generated, which is the result of evaluating the expression: "0123456789"[i] if i=10?

   (a) the expression is syntactically wrong
   (b) undefined
   (c) $'\backslash 0'$
   (d) of type char

10. What would the value of `q-p` be if:

```
int t[20], *p=t, *q=&t[19];
```

   (a) 19
   (b) 20
   (c) `t[19 ]- t[0]`
   (d) It is illegal to substract a pointer from another pointer!

11. What will be the value of variable a (before return) by running the following C code:

```
int main(){
    char a, a1=200, a2=110;
    a = a1 + a2;
    return 0;
}
```

   (a) $a = 54$
   (b) $a = 310$
   (c) $a = 182$
   (d) $a = -54$

12. What will be the value of variable "b" after execution of the following code:

```
unsigned char a; int b=0;
for (a=0; a<256; a++) {  b++;}
```

(a) there is an infinite loop
(b) 0
(c) 256
(d) compilation error
(e) -1

13. Select the correct statement regarding the following lines:

```
const int a = 12;
#define b 12
```

(a) there is no difference between `const` and `define`
(b) the const definition will allocate space in memory
(c) `b` will be processed by the preprocesor and will not appear at compilation
(d) we cannot see in debug the constant `b` but we can see the constant `a`

14. Whitch of the following statements will generate an memory allocation:

(a) `int x;`
(b) `typedef int integer`
(c) `struct {char x; long y; short z};`
(d) `double *p;`

15. Whitch of the following allocations ocupies more space:

(a) `int *pi;`
(b) `char *pc;`
(c) `long *pl;`
(d) `double **ppd;`
(e) none of them (all pointers have the same ammount of bytes)

16. What will be the output after running of the following code :

```
int compute(int x){
     static int b=20;
    b--;
    return b+x;
}
int main (void){
    int i;
    for (i=3;i>=0;i--)
```

```
    printf("%d ",compute(i));
    return 0;
}
```

(a) 22 20 18 16

(b) 23 22 21 20

(c) 23 21 19 17

(d) 3 2 1 0

17. Let we have int variable `x` and `int*` pointer variable `p`. We have initialized `x` by `2` and `p` to the address of `x`. Which of the following statements are true?

(a) The address of `x` and the value of `p` is the same

(b) The value of `x` and the value of `*p` is the same

(c) The address of `p` and the value of `&x` are equal

(d) `&x == *p`

C++ Language

1. A C++ constructor has the following properties:

(a) It is a member function in a class with the name matching the class name

(b) It is a member function that may return a value

(c) It is a member function without a return value

(d) It is a member function whose role is to initialise the instances of the class

(e) It is a member function whose role is to release allocated memory

(f) A given class cannot have more than one constructor

2. Which of the following statements are true about templates?

(a) Template is a feature of C++ that allows us to write one code for different data types

(b) Template is an example of compile time polymorphism

(c) Templates can not be used for user-defined data types.

(d) It is not possible to have user-defined templates, it is only possible to use those already created.

3. Which of the following statements is true?

(a) A static function cannot throw an exception.

(b) A static function cannot access a non-static member of the class.

(c) A static function cannot access a static member of the class.

(d) A static member cannot be modified in const non-static member functions.

4. Exceptions are:

   (a) errors that appears when a program is compiled
   (b) special situations in programs resolved with the help of tests of type if(variable == NULL)
   (c) errors that appear at runtime
   (d) thrown using the try statement and resolved using the catch statement
   (e) thrown using the throw statement and resolved using the try and catch statements

5. Match the verb with the class relationship it describes

Verb	Relationship
1. is a	a. Association
2. has a	b. Inheritance
3. uses a	c. Composition / Aggregation

   (a) (1, a), (2, b), (3, c)
   (b) (1, b), (2, c), (3, a)
   (c) (1, c), (2, b), (3, a)
   (d) (1, b), (2, a), (3, c)

6. The OCP principle refers to:

   (a) responsibilities that a class must implement
   (b) defining a consistent class hierarchy
   (c) problems that appear because of duplicate code
   (d) the possibility of class extension and avoidance of the modification of existing classes / code

7. Which of the following statements are false with respect to operator overloading in C++:

   (a) It is possible to add new operators to the language
   (b) It is possible to change the arity of the operator (number of accepted operands)
   (c) There are operators that cannot be overloaded
   (d) One can use either a member function or a global function to implement an operator overload.

8. If a class X has pointer variable members, then the class should also contain:

   (a) an empty destructor $X(){}$
   (b) a fiend function that copies an object of class X
   (c) an overload of the assignment operator
   (d) a copy constructor
   (e) an overload of the == operator

(f) a destructor that releases the memory pointed by the members of type pointer

9. Given the following code:

```
class Date {
 public:
  Date(int day=0, int month=0, int year=0);
  Date(const Date& ref);
};

void g(Date d) { }
void f() {
 Date d{7 , 3, 2007};
 Date d1;
 Date d2 = d;
 d2 = d;
 g(d);
}
```

How many times the constructors of class *Date* are invoked?

(a) 2

(b) 3

(c) 4

(d) 5

10. Which of the following statements is true?

```
class Shape { virtual void draw() = 0; };

struct Curve : Shape {
void addControlPoint(int x, int y);
};

int main() { Curve c; return 0; }
```

(a) class *Curve* cannot be instantiated because of the method *addControlPoint* that is not implementatied

(b) class *Curve* cannot be instantiated because it is an abstract class

(c) the code compiles successfully.

Java Language

1. What data encapsulation refers to?

   (a) Encapsulation of specific expertise into a class;
   (b) Hiding access to the private state of an object;
   (c) Making all the attributes of a class private;
   (d) Including no methods in a class but only attributes;
   (e) Hiding access to the entire state of an object.

2. Which of the following statements are false for the Java language?

   (a) The multiple inheritance can be obtained by applying the simple inheritance by a number of times;

(b) The simple inheritance can be simply obtained implementing a single interface;
(c) Only the abstract class can benefit from multiple inheritance;
(d) The language has extended inheritance which can replace multiple inheritance;

3. Which statement from the ones below are true in relation with the Java language?

(a) The final keyword applies to classes, attributes and methods;
(b) A final method can only call other final methods in the same class;
(c) A final method in a class cannot be overriden in subclasses;
(d) A final attribute can be set only once;
(e) Final attributes can only exist in final classes.

4. Which of the following assertions hold in relation with the Java language?

(a) Inheritance can always be replaced by objects' composition;
(b) A class can implement any number of interfaces;
(c) Using objects' composition provide a possibility to avoid access restrictions;
(d) Object composition is also referred as objects aggregation.

5. What is true about the String class in Java?

(a) The content of a String cannot be modified after creation
(b) The == operator is used to compare the content of two String objects
(c) The String class is final
(d) The String class has an attribute called size providing the length of the string

6. Which of the following statements is true about a Java abstract class:

(a) A class which cannot be instantiated
(b) A class having at least an abstract method
(c) A class defined using the abstract keyword

7. Which is the output of the following program?

```
public class MainClass {
  void method(int ... a){
    System.out.print(1 + " ");
  }
  void method(int[] a){
     System.out.print(2+ " ");
  }
  public static void main(String args[]){
```

```
    MainClass a = new MainClass();
    a.methods(new int[]{1,2,3,4});
  }
}
```

(a) 1

(b) 2

(c) 1 2

(d) compilation error

(e) execution error

8. Which of the following statements are false about Collections and Collection?

(a) Collections is a special type of collection which holds a Set of collection

(b) Both Collection and Collections entity belongs to java.util package

(c) Collections is a utility class

(d) Collection is an interface to Set and List

(e) Collection instances do not use erasure when are instantiated

9. Which of the following line of code is suitable to start a thread?

```
class X implements Runnable {
  public void run() {
    System.out.println(“Thread is in Running state”);
  }
  public static void main(String args[]) {
    /* Missing code? */
  }
}
```

(a)
```
X obj = new X();
Thread t=new Thread(obj);
t.start();
```

(b)
```
X obj = new X();
Thread t=new Thread(X);
```

(c)
```
X x= new X();
Thread t=new Thread();
x.run();
```

(d)
```
Thread t=new Thread(X());
t.start();
```

(e) None of these

(f)
```
Thread t=new Thread(new X());
t.start();
```

(g) What is the result of the following statements?

```
List list = new ArrayList();
list.add("one");
list.add("two");
list.add(7);
6: for (String s: list)
7:       System.out.print(s);
```

(a) onetwo
(b) onetwo followed by an exception
(c) Compiler error on line 6
(d) Compiler error on line 7

Software Engineering

1. Software verification can imply

   (a) automatic static analysis
   (b) assesing usefulness and usability of the software in operational situations.
   (c) debugging
   (d) software inspections
   (e) testing to prove error existence
   (f) verify that software meets user requirements

2. State machine diagram shows

   (a) system functions.
   (b) system response to internal events.
   (c) system response to external events.
   (d) interactions between objects in the system.
   (e) data structures.
   (f) interactions between actors and the system.
   (g) data flow in the system.

3. Agile methods in software development imply

   (a) Incremental delivery

(b) Customer involvement during development process

(c) Establishing normative processes for team working

(d) Periodic activities to eliminate complexity from the system

(e) Modeling the whole software before writting the code

4. Select the cases in which only concepts are reused:

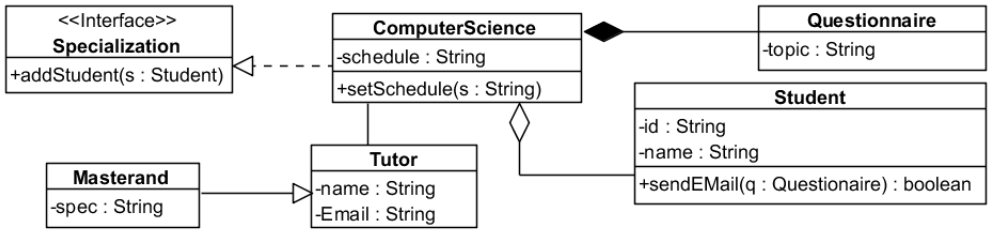
   (a) Software services

   (b) Design patterns

   (c) Program libraries

   (d) Architectural patterns

5. Consider the following class diagram.



   Check the true statements?

   (a) An object of type `ComputeScience` contains a collection of objects of type `Student`.

   (b) Class `Questionnaire` has a public attribute of type `String`.

   (c) Class `ComputerScience` has a public operation `addStudent(s:Student)`.

   (d) Class `ComputerScience` has a private operation `setSchedule(s:String)`.

   (e) Class `Tutor` is superclass for the class `Masterand`.

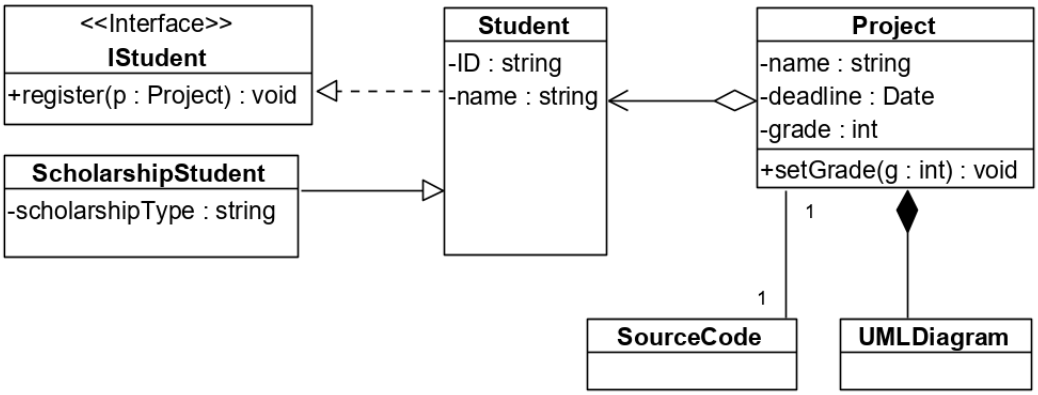
   (f) Class `ComputerScience` defines a composition of objects of type `Questionnaire`.

   (g) A unidirectional association exists between class `ComputerScience` and class `Tutor`.

   (h) Class `ComputerScience` inherits interface `Specialization`.

   (i) Class `Tutor` defines an aggregate of objects of type `Masterand`.

6. Consider the following class diagram.



Which sequence of Java code correctly and completely describes the relations of class `Proiect`?

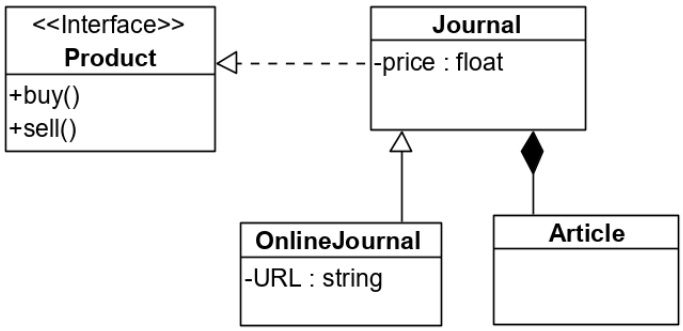
(a)
```
class Project extends Student {
        private Collection <UMLDiagram> diagrams = new LinkedList<>();
        private SourceCode code;
        ...}
```

(b)
```
class Project {
        private Collection <Student> students;
        private Collection <UMLDiagram> diagrams = new LinkedList<>();
        private SourceCode code;
        ...}
```

(c)
```
class Project {
        private Student student;
        private UMLDiagram diagram;
        private SourceCode code;
        ...}
```

(d)
```
class Project {
        private Collection <Student> students = new LinkedList<>();
        private Collection <UMLDiagram> diagrams;
        private SourceCode code;
        ...}
```

7. Consider the following class diagram.



Which sequence of Java code correctly and completely describes the relations of class `Journal`?

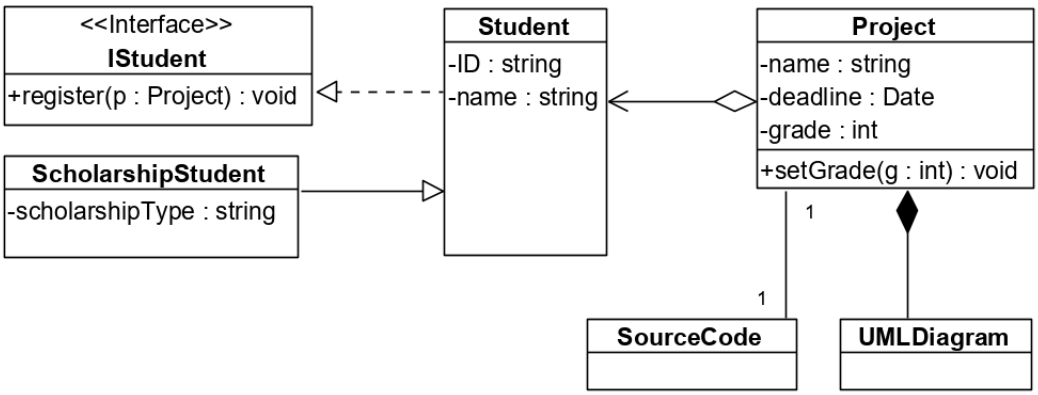
(a)
```
class Journal extends OnlineJurnal implements Product {
    private Collection <Article> articles;
    ...}
```

(b)
```
class Journal implements Product {
    private Collection <Article> articles = new ArrayList<>();
    private OnlineJurnal jurnal;
    ...}
```

(c)
```
class Journal implements Product {
    private Collection <Article> articles = new ArrayList<>();
    ...}
```

(d)
```
class Journal extends OnlineJurnal {
    private Collection <Article> articles = new ArrayList<>();
    private Product product;
    ...}
```

8. Consider the following class diagram.



Check the complete and correct description of the relations represented on the diagram:

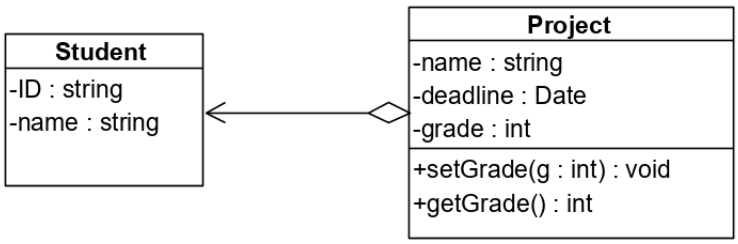
(a) Bidirectional association between classes `Project` and `SourceCode`; composition between classes `Project` (composite) and `UMLDiagram` (component); aggregation between classes `Project` (aggregate) and `Student` (component); generalization between interface `IStudent` (implemented) and class `Student` (implements); realization between classes `Student` (superclass) and class `ScholarshipStudent`(subclass).

(b) Bidirectional association between classes `Project` and `SourceCode`; composition between classes `Project` (composite) and `UMLDiagram` (component); aggregation between classes `Project` (aggregate) and `Student` (component); realization between interface `IStudent` (implemented) and class `Student` (implements); generalization between classes `Student` (superclass) and class `ScholarshipStudent` (subclass).

(c) Bidirectional association between classes `Project` and `SourceCode`; aggregation between classes `Project` (aggregate) and `UMLDiagram` (component); composition between classes `Project` (composite) and `Student` (component); generalization between interface `IStudent` (implemented) and class `Student` (implements); realization between classes `Student` (superclass) and class `ScholarshipStudent` (subclass).

(d) Bidirectional association between classes `Project` and `SourceCode`; composition between classes `UMLDiagram` (composite) and `Project` (component); aggregation between classes `Student` (aggregate) and `Project` (component); realization between interface `IStudent` (implemented) and class `Student` (implements); generalization between classes `Student` (superclass) and class `ScholarshipStudent` (subclass).

9. Consider the following class diagram.



Which sequence of Java code correctly describes the relationship between classes `Student` and `Project`?

(a)
```
class Student extends Project{...}

class Project{...}
```

(b)
```
class Project extends Student{...}

class Student{...}
```

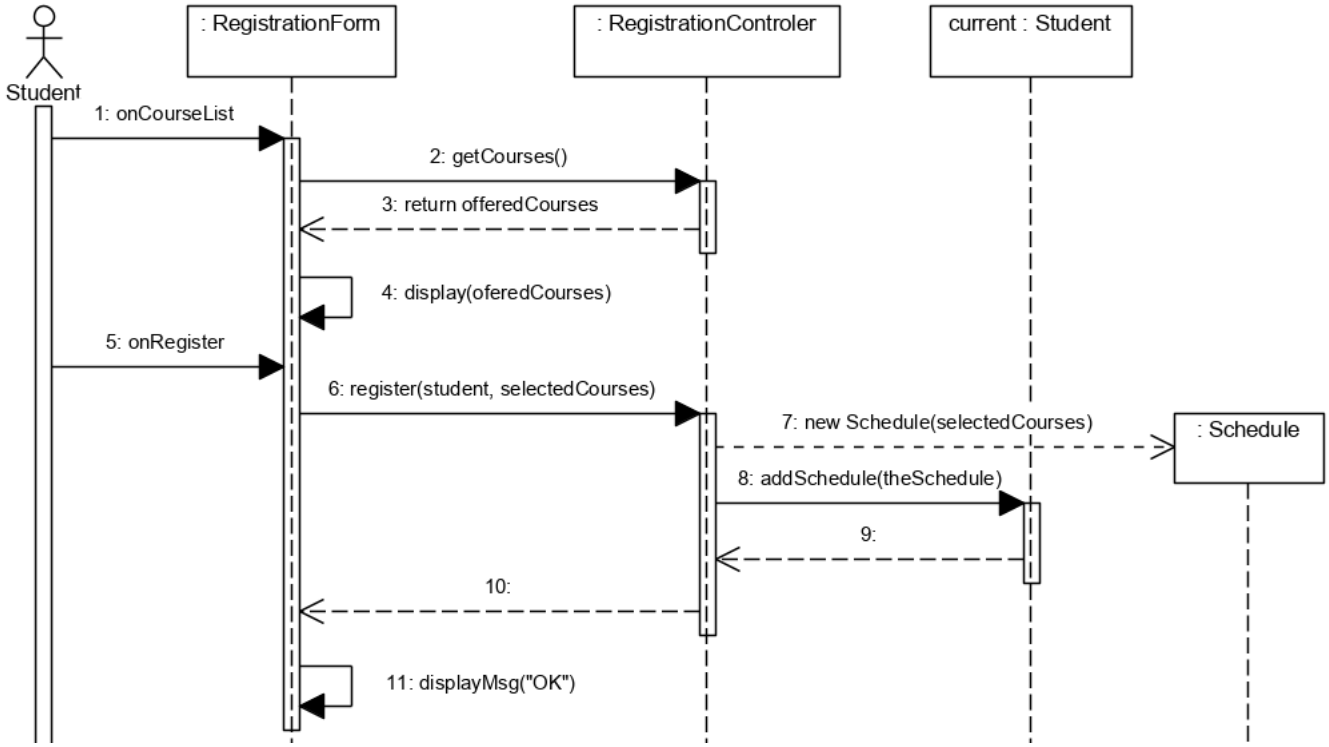
(c)
```
class Project {
```

```
        private Collection <Student> students = new LinkedList<>();...}

    class Student{...}
```

(d)
```
class Project {
    private Collection <Student> students = new LinkedList<>();...}

class Student {
    private Project project;
     ...}
```

(e)
```
class Proiect {
    private Collection <Student> students = new LinkedList<>();...}

class Student {
    private Collection<Proiect> projects;
     ...}
```

10. Consider the following sequence diagram.



Which operations of class `RegistrationControler` result from it?

(a) `getCourses()`

(b) `display(offeredCourses)`

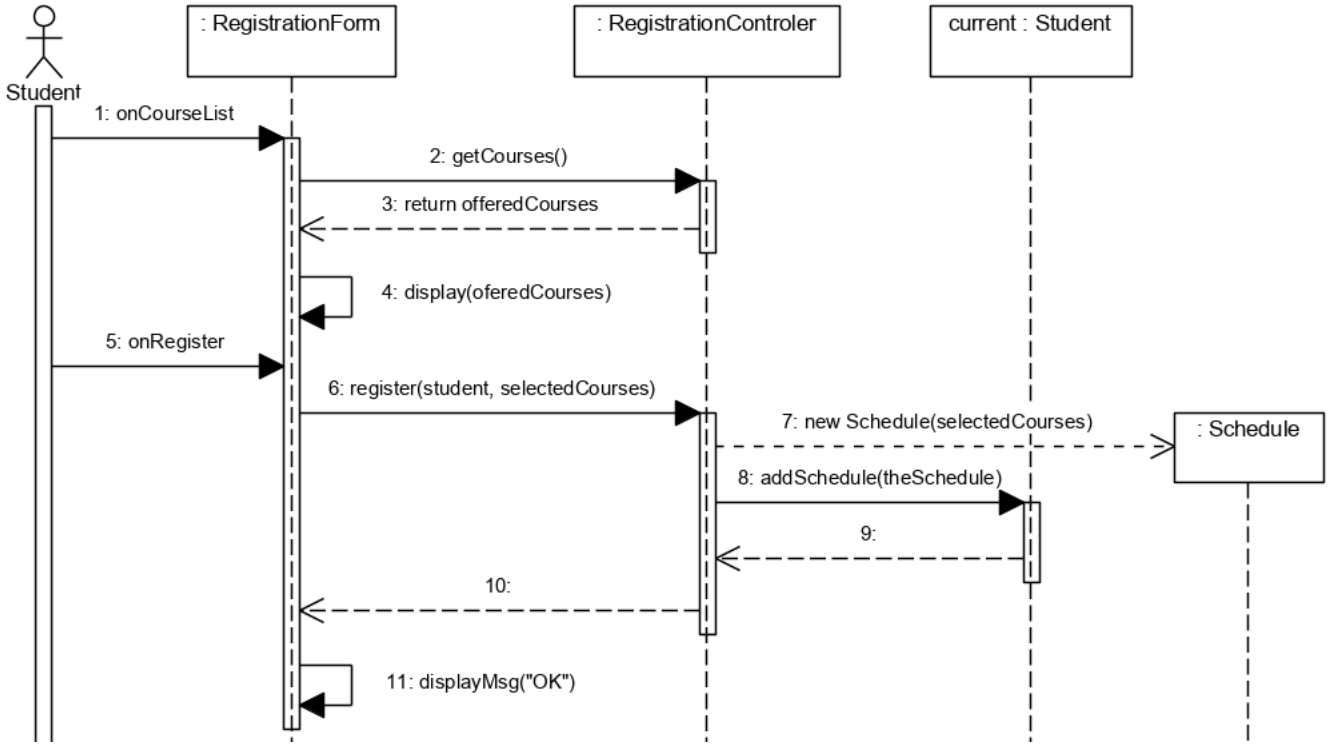
(c) `register(student, selectedCourses)`

(d) `new Schedule(selectedCourses)`

(e) `addSchedule(theSchedule)`

(f) `displayMsg(''OK'')`

11. Consider the following sequence diagram.



Check the classes from which are instantiated the objects implied in the interaction:

(a) RegistrationForm

(b) RegistrtationControler

(c) selectedCourses

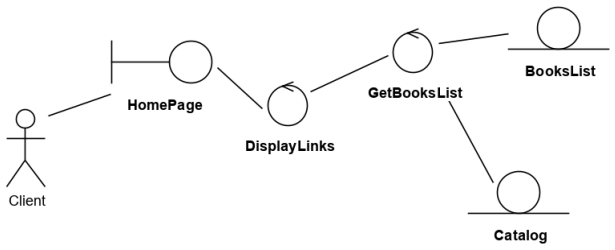
(d) current

(e) Student

(f) Schedule

(g) theSchedule

12. Consider the following robustness diagram.



Which statements are true?

(a) `Home page` is a *boundary* object.

(b) `GetBooksList` can be a persistent object.

(c) `DisplayLinks` is an object for interfacing with the system.

(d) `GetBooksList` could be implemented as a method of an *entity* class.

(e) `GetBooksList` can be a persistent object.

(f) `BooksList` is an object for interfacing with the system.

(g) `Catalog` is an *entity* object.

13. Select the pair of terms to be filled in the spaces of the next statement:

"In Essence 'Checkpoint construction' game, a checkpoint is a set of _____ to be achieved at a specific point in time in a software engineering endeavor AND is expressed in terms of _____ ."

(a) criteria / activities

(b) components / roles

(c) alphas / activity spaces

(d) criteria / alpha states

(e) alpha states / criteria

14. Realize the correct mapping between the concept and its definition:

(a) Test case	[1] re-running an existing set of tests.
(b) Regression testing	[2] the process of testing individual components in isolation.
(c) Stress testing	[3] a series of tests where the load is steadily increased.
(d) Performance testing	[4] exercising the system beyond its maximum design load.
(e) Unit testing	[5] specifications of the inputs of the test and of the expected results, also indicating the tested entity.

(a) a – 5, b – 1, c – 4, d – 3, e – 2

(b) a – 5, b – 3, c – 4, d – 1, e – 2

(c) a – 2, b – 1, c – 3, d – 4, e – 5

(d) a – 1, b – 2, c – 3, d – 4, e – 5

(e) a – 3, b – 2, c – 4, d – 5, e – 1

15. Check the methods to avoid introducing faults into the programs.

    (a) Every class should have a single responsibility

    (b) Use regular expressions to validate input values

    (c) Avoid deeply nested conditional statements

    (d) Logging the interactions of the users with the program.

    (e) Replace the body of a method with a more clear algorithm that returns the same result.

    (f) Minimize the depth of inheritance hierarchies

    (g) Automatically save the user's data at set intervals

    (h) Identify the aspects of the application that vary and separate them from what stays the same.

    (i) Check input values against the range defined by input rules

    (j) Using assertions to check results from an external service

Databases

1. Which of the following are properties of a candidate key?

    (a) Uniqueness

    (b) Irreducibility

    (c) Completeness

    (d) Domain segregation

2. Which are database languages?

    (a) Data Definition Language (DDL)

    (b) Database Distribution Language (DDL)

    (c) Data Manipulation Language (DML)

    (d) Data Maintenance Language (DML)

    (e) Data Query Language (DQL)

    (f) Data Analysis Language (DAL)

3. Given the relation R(A) with two tuples in it {(1), (2)} and the following two transactions

```
(T1) UPDATE R SET A = A*2
(T2) SELECT AVG(A) FROM R; SELECT MAX(A) FROM R
```

If T2 is executed under *repeatable read* isolation level, which statements are true?

(a) IF AVG(A) = 1.5 then MAX(A) equals to 2 or 4

(b) AVG(A) = 1.5 and MAX(A) = 2 , always

(c) Possible values for AVG(A) are 1.5 or 3, and possible values for MAX(A) are 2 or 4

(d) If MAX(A) = 4 then AVG(A) = 1.5

4. Which of the following are issues addressed by normalization?

(a) Deletion anomalies

(b) Insertion anomalies

(c) Modification anomalies

(d) Data redundancy

(e) Data volume

5. Three of the following four expressions finds the names of all students who did not apply to major in CS or PH. Which one finds something different? (Hint: You should not assume student names are unique.)

(a) $\pi_{sName}(Student \bowtie (\pi_{sID}(Student) - (\pi_{sID}(\sigma_{major='CS'}Apply) \cup \pi_{sID}(\sigma_{major='PH'}Apply))))$

(b) $\pi_{sName}(Student \bowtie (\pi_{sID}Student - \pi_{sID}(\sigma_{major='CS' \vee major='PH'}Apply)))$

(c) $\pi_{sName}(\pi_{sID,sName}Student - \pi_{sID,sName}(Student \bowtie \pi_{sID}(\sigma_{major='CS' \vee major='PH'}Apply)))$

(d) $\pi_{sName}Student - \pi_{sName}(Student \bowtie \pi_{sID}(\sigma_{major='CS' \vee major='PH'}Apply))$

6. Consider the following query:

```
SELECT * FROM Student, Apply, College
WHERE Student.sID = Apply.sID
AND Apply.cName = College.cName
AND Student.Bac > 5
AND College.cName = 'UVT'
```

Suppose we can create two indexes, and assume all indexes are tree-based. Which two indexes do you think would be most useful for speeding up query execution?

(a) Student.sID and Student.Bac

(b) Apply.cName and College.cName

(c) Apply.sID and College.cName

(d) Apply.sID and Student.Bac

7. What is the result returned by the following SQL query?

```
SELECT b.Name
FROM Employee a, Employee b, SalaryLevel c, SalaryLevel d
WHERE a.ManagerId = b.Id
AND b.Salary BETWEEN c.MinSalary AND c.MaxSalary
AND b.Salary*1.25 BETWEEN d.MinSalary AND d.MaxSalary
AND c.Level+1 = d.Level
```

(a) Returns employees' names who would move to the next salary level after a 25% wage increase.

(b) Returns managers' names who would move to the next salary level after a 25% wage increase.

(c) Returns the employees who would move to the next salary level after a 25% wage increase if their salary is between the minimum and maximum wage.

(d) None of the above

8. The query engine executes the clauses of a SELECT statement in a specific order.

(1) SELECT
(2) FROM
(3) WHERE
(4) ORDER BY
(5) GROUP BY
(6) HAVING
(7) TOP / OFFSET

Indicate SELECT clauses order.

(a) 5, 1, 7, 6, 3, 4, 2
(b) 5, 1, 2, 6, 3, 4, 7
(c) 5, 1, 2, 3, 6, 4, 7
(d) 5, 2, 1, 6, 4, 3, 7

9. The table below contains a representative data sample for all the functional dependencies in a real-life data set. Which are the functional dependencies that hold for the data below?

A	B	C	D	E	F
100	Timişoara, str. Revolutiei	TM	1234	PA	Descriere A
100	Timişoara, str. Revolutiei	TM	123	PB	Descriere B
100	Timişoara, str. Revolutiei	TM	10	A	Descriere A
101	Timişoara, str. Revolutiei	TM	10	A	Descriere A
151	Deta, str. Principală	TM	1234	PA	Descriere A
200	Arad, str. Republicii	AR	1234	PB	Descriere B

(a) A $\rightarrow$ {B, C}, E $\rightarrow$ {F}

(b) D $\rightarrow$ {A, B, C, E, F}, F $\rightarrow$ {E}

(c) {A, E} $\rightarrow$ {D}

(d) {D, E} $\rightarrow$ {A, B}

10. Considering the relation $R(A, B, C, D, E, F)$ with attributes and functional dependencies from the previous question, which of the following attributes set are candidate keys in R?

    (a) {B, F}

    (b) {A, E}

    (c) {A, E, D}

    (d) {A}

11. Given R defined in the previous question, which of the below statements are false?

    (a) The relation R is in the first normal form.

    (b) The relation R is in the second normal form.

    (c) In the relation R the values in column B determines the values in column A.

    (d) The relation R can not contain two tuples with different values in column D when the values in column A match.

12. Consider the relation $R(A, B, C, D, E)$ with the following functional dependencies: $D \rightarrow C, \{C, E\} \rightarrow A, D \rightarrow A, \{B, E\} \rightarrow D$. Which of the following attribute sets is a superkey of R?

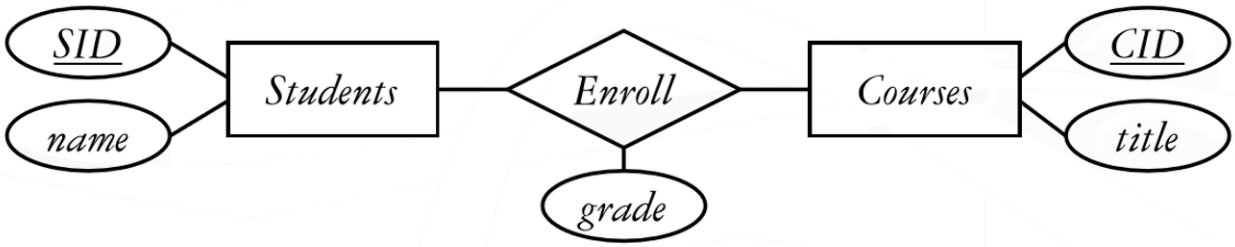
    (a) {B, E}

    (b) {C, D}

    (c) {A, C}

    (d) {A, C, E}

13. Given the E/R diagram below



Which of the following relations appear in the relational model representation of the diagram?

(a) Courses(CID, title)
(b) Students(SID, name)
(c) Enroll(SID,CID)
(d) Enroll(SID, name, CID, title, grade)
(e) Students(SID, name, CID)
(f) Enroll(SID, CID, grade)

14. The relations R(A,B) and S(B,C) contain the following records: R=(1,2), (2,2), (5,1) and S=(2, 2), (2, 1). Let the view V defined by the following statement

```
SELECT A, C FROM R JOIN S ON R.B = S.B
```

Which of the following records are included in V?

(a) (1,1)
(b) (1,2)
(c) (2,1)
(d) (2,2)
(e) (5,1)
(f) (5,2)
(g) (1,5)
(h) (2,5)

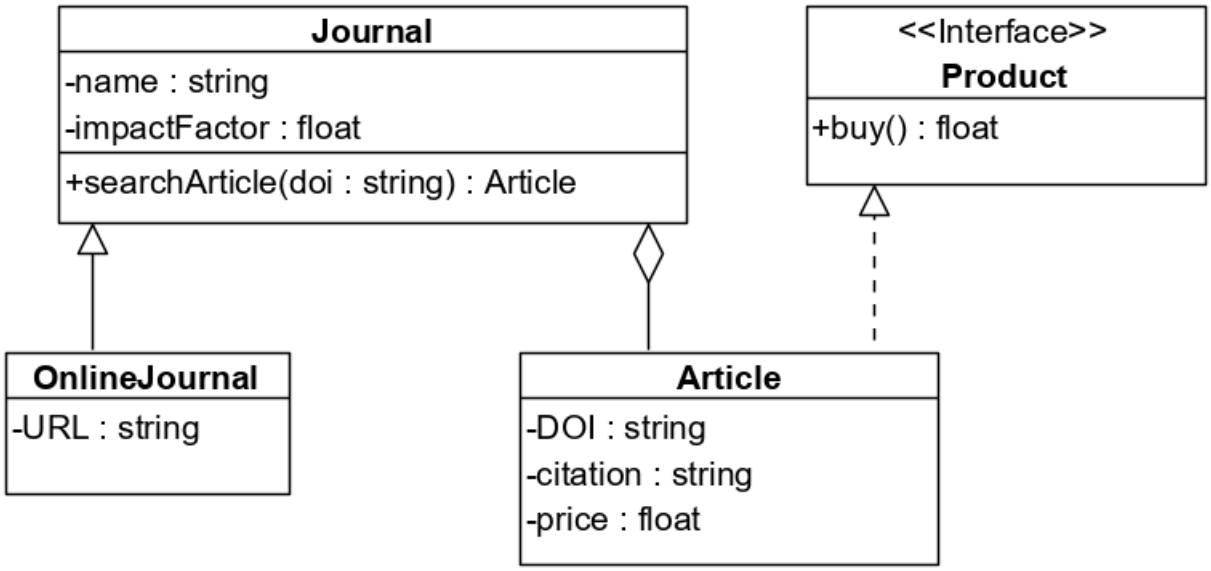
15. Select the statements that can be used to complete the following SQL code block

```
BEGIN TRANSACTION
SELECT COUNT(*) FROM Enrollments WHERE Accepted = 1
UPDATE Enrollments SET Accepted = 1 WHERE CourseTitle = 'Data Base'
```

(a) END TRANSACTION
(b) END
(c) COMMIT
(d) ROLLBACK TRANSACTION

Interdisciplinary subjects

1. Considering the following class diagram:



   (a) Represent in an Object Oriented programming language (Java/Python) the code resulting from the above diagram, including constructors and when is the case the methods that express the relation between the classes.

   (b) Starting from the above diagram propose a normalized relational database schema.

   (c) Based on proposed database schema define a query to find ... .. E.g.: Select the articles from Journal x having the price less that a value Y.

Answers

   (a) The resulting code from the UML represented in Python/Java programming language.

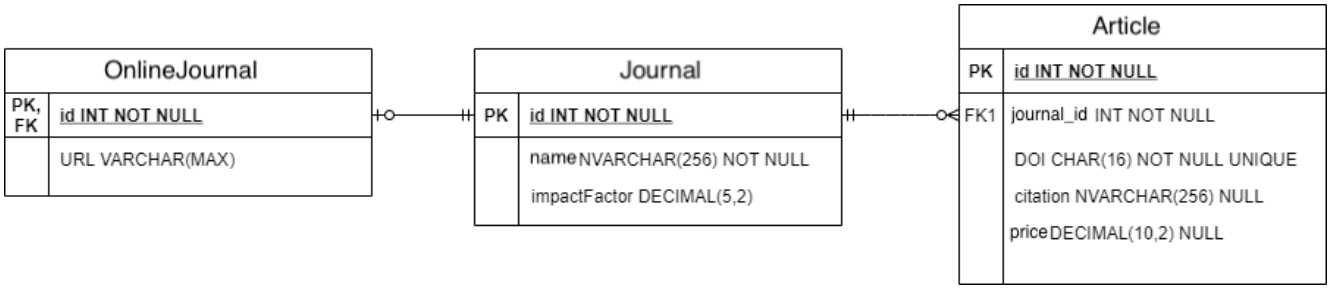
JAVA CODE

```
public interface Product {
  public float buy();
}

public class Journal {
  private String name;
  private float impactFactor;
  private Collection<Article> articles;

  public Journal(final String name,
        final float impFact){
    this.name=name;
    this.impactFactor = impFact;
    articles = new ArrayList<>();
  }
  public Article searchArticle(String doi)
  { ... }
  public void addArticle(Articol a)
  { ... }
  public void removeArticle(Articol a)
  { ... }
}

public class OnlineJuornal
    extends Juornal {
  private String URL;

  public OnlineJuornal(String name,
      float impFac, String url){
    this.name = name;
    this.impactFactor = impFac;
    this.URL = url;
  }
}

public class Article implements Product {
  private String DOI;
  private String citation;
  private float price;

  // define  full constructor for the class

  public float buy() {...}
}
```

PYTHON CODE

```
class Product:
  def buy():
    pass;

class Juornal:
  def __init__(self, name, impfac):
    self.name = name
    self.impactFactor = impfac
    self.articles = []

  def findArticle(self, doi):
    ...
  def addArticle(self, a):
    if isinstance(a, Article):
      ...
  def removeArticle(self, a):
    if isinstance(a, Article):
      ...

class OnlineJuornal(Journal):
  def __init__(self, name, impfac, url):
    Revista.__init__(self, name, impfac)
    self.url = url

class Article(Prodct):
  def __init__(self, doi, cit, price):
    self.DOI = doi
    self.citation = cit
    self.price = price

  def buy():
    ...
```

(b) The database schema usig E/R diagams based on the Crow's foot notation (https://vertabelo.com/blog/crow-s-foot-notation):



Remarks:

- A journal can have zero ore more articles; each article is attached to a journal, there are no articles that do not belong to a journal.
- A journal can be or not online; each of the online journals is related to a journal.
- In order to represent real values in SQL langue it is recommended to avoid the usage of predefined real or float data type. Instead is recommended to use the representation in fix precision DECIMAL/NUMERIC that provides more accurate computation. The `impactFactor` field is represent using 5 digits, from which 2 are decimals values (the maximum value that can be stored in this field is 999.99); the `price` field is represented on 10 digits, from which 2 are used for decimals values (the maximum value that can be stored in this field is 99999999.99).
- In order to design the inheritance relation other variants can be used; because in the presented example the derived class has a low number of distinct proprieties from base class these proprieties can be represent like columns in the journal table. In this case in the `Journal` table it is possible to add a discriminator column (e.g. `type` int NOT NULL) which will indicate the object type. The Object Relational Mapping (ORM) libraries support both mapping models and the user will chose the appropriate variant based on application requirements.
- The journal name can be stored using NVARCHAR data type (a string of Unicode characters with variable length); the `DOI` attribute is represented as CHAR(16) (exact 16 ASCII characters); The attribute `citation` is represented as NVARCHAR(256) (maximum 265 Unicode characters with variable length) etc.

(c) Queries examples:

- Select all the articles from journal X with the pricess less than Y

```
SELECT *
FROM Article a
INNER JOIN Journal r ON a.journal_id = r.id
WHERE r.name='X' AND a.price < Y
```

- Selelect all the articles from online journal

```
SELECT *
FROM Article a
```

```
INNER JOIN OnlineJournal ro ON a.journal_id = ro.id
```

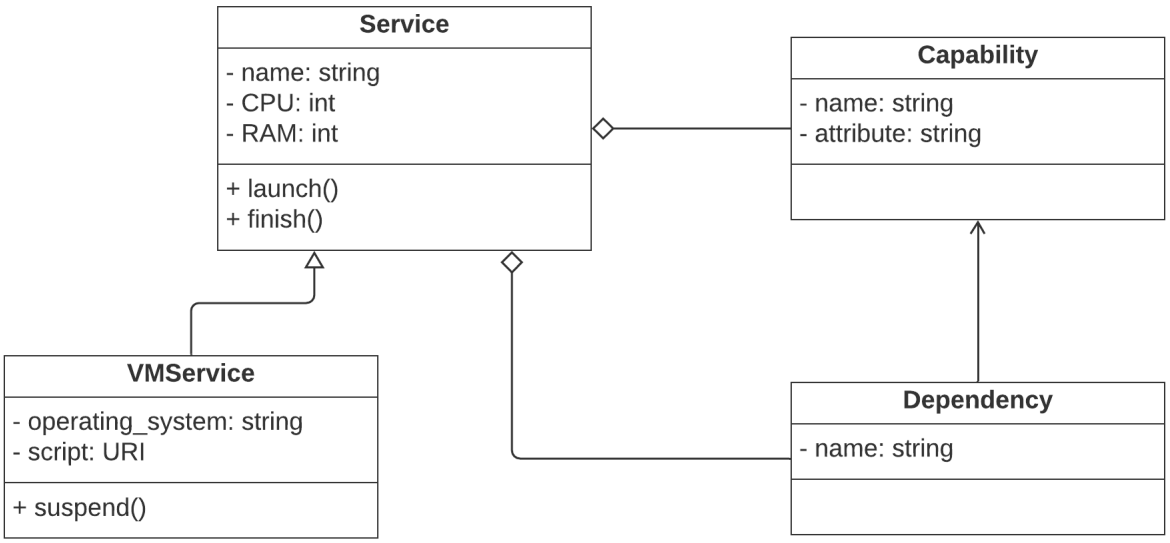
- Select all journals that contain at least one article

```
SELECT R.*
FROM Journal R
INNER JOIN Article A ON A.journal_id = R.id
```

- Select all journals that do not have any articles

```
SELECT R.*
FROM Journal R
LEFT JOIN Article A ON A.journal_id = R.id
WHERE A.id IS NULL
```

2. Considering the following class diagram:

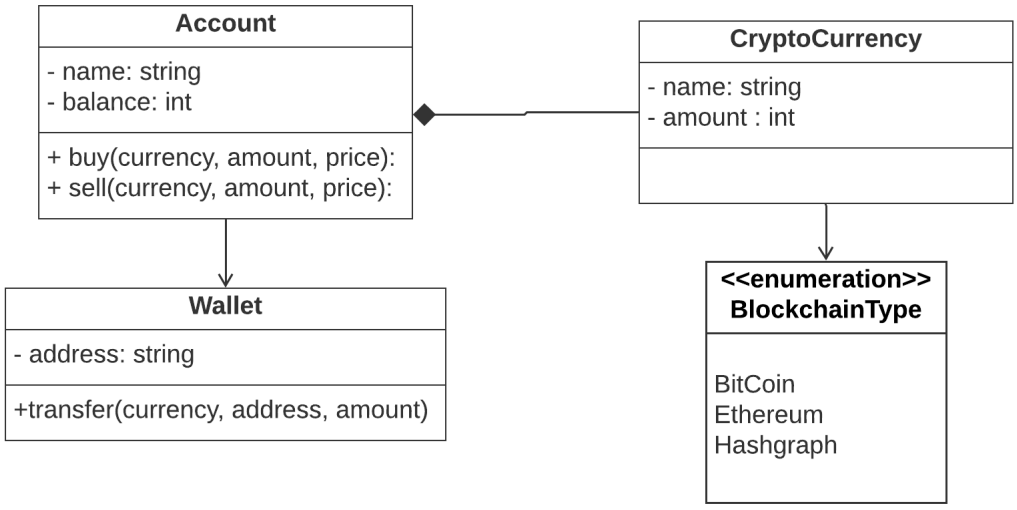


(1) Represent in an Object Oriented programming language (Java/Python) the code resulting from the above diagram, including constructors and when is the case the methods that express the relation between the classes.

(2) Starting from the above diagram propose a normalized relational database schema.

(3) Based on proposed database schema define a query to find ... . E.g.: Select all services which expose capabilities from the list of dependencies of the "RayTracingEngine" service.

3. Considering the following class diagram:

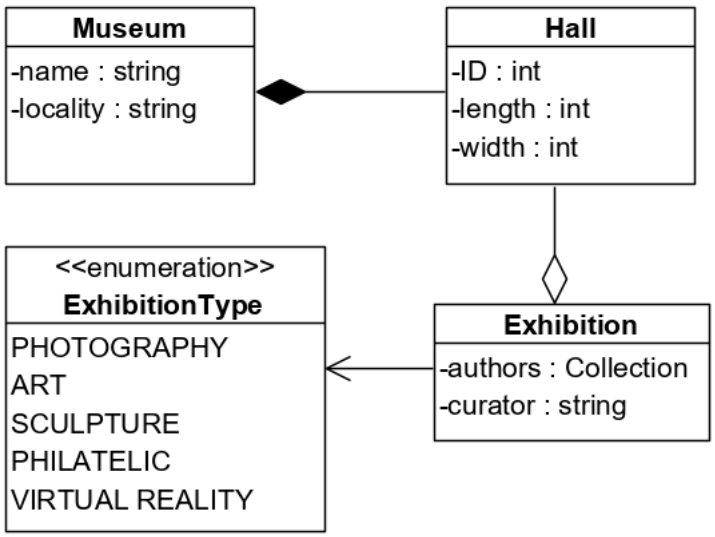


(a) Represent in an Object Oriented programming language (Java/Python) the code resulting from the above diagram, including constructors and when is the case the methods that express the relation between the classes.

(b) Starting from the above diagram propose a normalized relational database schema.

(c) Based on proposed database schema define a query to find ... . E.g. Select all coins of type T with wallet address equal with X.

4. Considering the following class diagram:



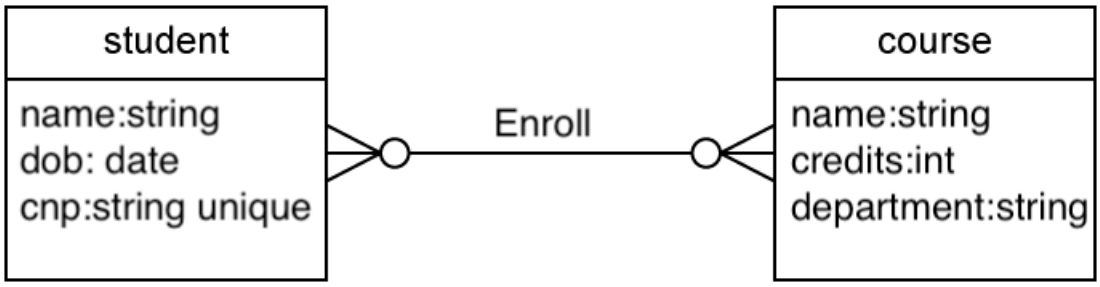
(a) Represent in an Object Oriented programming language (Java/Python) the code resulting

from the above diagram, including constructors and when is the case the methods that express the relation between the classes.

(b) Starting from the above diagram propose a normalized relational database schema.

(c) Based on proposed database schema define a query to find ... . E.g.: Add into the data base a virtual reality exhibition that will be hold in the halls 3, 7 and 8.

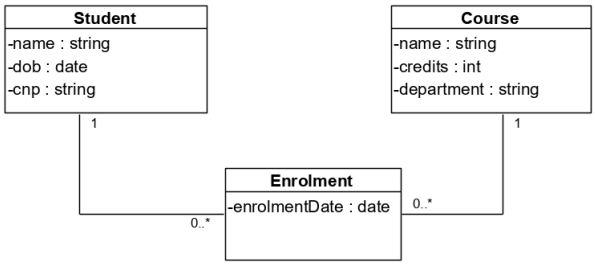
5. Considering the following conceptual model:



Requirements:

- a student can be enrolled to zero or more courses
- zero ore multiple students can be enrolled into a course
- the system has to log the enrolment date of the student to the course
- a student can enroll to a course only once
- CNP is unique

(a) Create a class diagram appropriate to these requirements.
Indication



(b) Represent in an Object Oriented programming language (Java/Python) the code resulting from the above diagram, including constructors and when is the case the methods that express the relation between the classes.

(c) Starting from the above diagram propose a normalized relational database schema.

(d) Based on proposed database schema define a query to find ... . E.g.: Select all courses that a student is enrolled; select all courses that have at least N students enrolled.

Answers topic II

Python Language

1. (d)
2. (c)
3. (b)
4. (a), (c),(e)
5. (a), (c), (d)
6. (a), (b),(c)
7. (a), (d)
8. (d)
9. (a), (d), (e)
10. (a), (c)

C Language

1. (a),(c),(d)
2. (a)
3. (b),(c)
4. (a),(b)
5. (c)
6. (b), (c)
7. (a), (c), (d), (f)
8. (a)
9. (c), (d)
10. (a)
11. (a)
12. (a)
13. (b), (c), (d)
14. (a), (d)
15. (e)
16. (a)
17. (a), (b)

C++ Language

1. (a),(c),(d)
2. (a),(b)
3. (b)
4. (c),(e)
5. (b)
6. (d)
7. (a), (b)
8. (c),(d),(f)
9. (c)
10. (b)

Java Language

1. (b)
2. (a),(b),(c),(d)
3. (a),(c),(d)
4. (b)
5. (a),(c)
6. (a),(c)
7. (a), (c)
8. (d)
9. (a), (f)
10. (c)

Software Engineering

1. (a),(d),(e)
2. (b),(c)
3. (a),(b),(d)
4. (b), (d)
5. (a),(c),(e),(f)
6. (b)
7. (c)
8. (b)
9. (c)
10. (a),(c)
11. (a),(b),(e),(f)
12. (a),(d),(g)
13. (d)
14. (a)
15. (a), (c), (e), (f), (h)

Databases

1. (a),(b)
2. (a), (c), (e)
3. (c)
4. (a),(b),(c),(d)
5. (d)
6. (c)
7. (b)
8. (b)
9. (a), (c)
10. (b)
11. (b), (c), (d)
12. (a)
13. (a), (b), (f)
14. (a),(b),(c),(d)
15. (c),(d)

Topic III. Computing systems

Computer Architecture

1. By translation,
   (a) a program P that written for a high-level machine M is transformed by a translator into a program P_1 that can then be executed on a lower level of that machine, without further need for P,
   (b) each instruction of a program P written for a high-level machine M is transformed into a sequence of instructions for a lower level of that machine, then immediately executed, before the same process is applied to the next instruction from P,
   (c) a program P that is written for a high-level virtual machine VM is transformed into a program P_1 which can be executed on a physical machine M.
2. The data path is defined as:
   (a) the connection between input-output devices and the processor,
   (b) the representation of data types in the memory of a computing system,
   (c) the connection trough buses between registers arithmetic-logical unit and back to registers, used for instruction execution,
   (d) networks interconnecting computing systems (like the Internet).
3. The program counter:
   (a) is a global variable in a program that is used to count the number of instructions executed, and measure program complexity,
   (b) is a register that holds an address to the next instruction that has to be executed,
   (c) is a register that holds the instruction that is being executed in a processor,
   (d) is an operating system variable that counts the number of processes in execution at a given time.
4. Microprogramming is
   (a) the activity of writing programs that run on microprocessors,
   (b) the use of a set of instructions (microinstructions), stored in a memory, that are used to control the data path, for the purpose of executing instructions in one or several data path cycles,
   (c) the interaction with input/output devices by issuing commands to controllers (e.g commands for the hard drive controller to read a word from an address),
   (d) automated generation of small programs for embedded devices.
5. A kilobyte is

(a) 1000 bits,

(b) 1024 bytes,

(c) 2^{13} bits.

6. A typical use for a demultiplexer circuit is the implementation of

   (a) a 1 bit full adder circuit,

   (b) a serial-to-parallel convertor,

   (c) a parallel-to-serial convertor,

   (d) an integer multiplication circuit.

7. For cache memories, the locality principle refers to:

   (a) the fact that memory locations close to the previously used one are likely to be used in the near future,

   (b) the fact that memory locations that were accessed recently will likely be accessed again,

   (c) the position of cache levels with respect to the processor (on processor, or very near the processor).

8. Which type of branch instruction causes most delay in a typical five stage pipeline:

   (a) unconditional branches,

   (b) conditional branches,

   (c) none of the branch prediction causes any delay in the pipeline,

   (d) both types of branch prediction cause the same amount of delay, on average.

9. Which of the following are necessary to describe the Instruction Set Architecture level:

   (a) the memory model,

   (b) the configuration of the cache,

   (c) the configuration of registers,

   (d) data types,

   (e) instructions.

10. The result of the following addition

$$\begin{array}{r} 11110111 \\ +11110111 \end{array}$$

    in 2's complement representation on 8 bits is:

    (a) 11101110 and the result is correct,

    (b) 11101110 but there is an overflow,

(c) 11101111 and the result is correct,

(d) 11101111 but there is an overflow.

11. The sequence $3D800000$ represents the following number in IEEE 754 single precision floating point representation:

(a) 0.0625,

(b) 16.25,

(c) 41,

(d) 221.

12. [In genera] the Instruction Sets are growing larger and larger over time . Why is that?:

(a) because Moore's law dictates that we double the amount of computation every 2 years;

(b) because the RAM (and memory in general) is growing larger and larger each year;

(c) because with each new generation of CPU's more instructions are added; thus growing the list;

(d) because there is an increase in the number of users and the applications that they use;

(e) because old OPT codes must be kept for backwards compatibility.

13. In the Cache, the role of the Dirty bit is to :

(a) signal that the data in the cache is old;

(b) There is no such thing as a "dirty bit".

(c) signal that the cache line might contain a different value than the corresponding memory location in RAM;

(d) keep track if there was a Cache Hit or Cache Miss in the previous instruction;

(e) None of the above.

14. Select the correct statements in regard to the Central Processing Unit

(a) Its role is to execute instructions;

(b) it contains multiple registers;

(c) Each computer has one CPU in it;

(d) Uses multiple phases to process each instruction;

(e) It controls how the data is written into the RAM memory.

15. Upon completing the Execute Phase the Control Unit...

(a) transfers control to the memory to store the result of the execution;

(b) increments the instruction address;

(c) checks if the next memory address contains and instruction or data;

(d) clears all registers to prepare the CPU for the next instruction;

(e) none of the above.

16. What is the role of the Clock inside a computer?

(a) It is just unit of measure that helps rank the CPUs by speed.

(b) Its signal is used by the Control Unit to advance the internal operation of the CPU.

(c) Its main role is to allow overclocking and/or underclocking of CPUs.

(d) Keeps track not only of time but also the date and other useful timing variables.

(e) Triggers an electrical signal at a precise and regular interval.

Operating Systems

1. What causes a running thread to change its state, from Running to Blocked?

(a) any I/O event in the system

(b) starting a new process in the system

(c) terminating the running process

(d) any context switch that happens in the system

2. When designing an operating system for smartphones and other mobile devices, you should

(a) use bigger page sizes and more states for processes

(b) use smaller page sizes and less states for processes

(c) allow apps to run without interruption from the system, for maximizing user interaction

(d) terminate apps if they run for too long, to avoid starvation of other processes

3. It is required to use defragmentation in a file system in order to

(a) improve the speed of the context switching in the system

(b) allow smaller seek times for mechanic drives

(c) allow files to be read faster sequentially

(d) reduce boot times for the operating system

4. Suppose process A has a burst of 5, and starts at moment 0. Process B has a burst of 20, and starts at moment 5. A new process C is created at moment 3, with a burst of 3. Which of the following is true?

(a) in FIFO scheduling, process B runs before process C

(b) in SJF scheduling, process A finishes before C starts

(c) in Round-Robin scheduling (quantum = 3), the first process to execute after A is B

(d) in SRTN (quantum = 3), processes execute as follows: A, C, B, A, B

5. A 16-bit Von Neumann architecture has a page size of 4,096 bytes and 12 KB of RAM. Access to the pages of the system happen as follows: 5,15,15,5,10. Which of the following is true?

   (a) there are 4 physical pages in the system

   (b) there are 3 physical pages in the system

   (c) there are 16 virtual pages in the system

   (d) the FIFO algorithm causes 2 page faults to be issued

   (e) in the FIFO algorithm, address 1782 belongs to the first virtual page

6. Which of the following are true about memory management?

   (a) smaller page sizes are favorable when accessing small memory data chunks

   (b) larger page sizes are favorable when accessing small memory data chunks

   (c) physical addresses usually have a significantly smaller range than virtual addresses

   (d) internal memory fragmentation can be resolved by the operating system

   (e) external memory fragmentation can be resolved by the operating system

7. Assume you have a browser tab open in Windows, Linux or MacOS X, where you are playing an YouTube video in 4K (UltraHD) resolution. Which of the following could be true?

   (a) the browser tab's associated thread is more CPU-bound than I/O-bound

   (b) the browser tab's associated thread is more I/O-bound than CPU-bound

   (c) the operating system will not schedule any other processes until playback is complete

   (d) new processes that are trying to execute during playback are placed in a FIFO waiting queue

   (e) interrupts are not allowed during YouTube playback

8. Assume you are using a Von Neumann system with a CPU that has multiple physical cores. Which of the following is true?

   (a) true parallelism can never be achieved through the kernel scheduler, unless the operating system is Linux or MacOS X

   (b) true parallelism can never be achieved through the kernel scheduler, unless the operating system is Windows and RAM memory is limited to 4 GB

   (c) the kernel can implement true parallelism in a multi-core CPU only if there are no bus limitations

   (d) in an interactive system, a pre-emptive scheduling algorithm is required to ensure responsiveness

9. In a system with a single type of resources, processes A, B and C have a maximum requirement of 10, 20 and 30 resources of type R. An initial allocation is made, of 8, 18, and 28 resources is made for these three processes. Which of the following statements is true?

   (a) the minimum number of resources required, so that the state of the system is considered still safe, is 6

   (b) the minimum number of resources required, so that the state of the system is considered still safe, is 2

   (c) if only 2 resources are available for an additional allocation, the entire system is in a deadlock

   (d) an additional allocation of 1 resource to process A will leave the system in a deadlock

10. You are asked to design a process scheduling mechanism which favors short processes that perform I/O activities, while retaining the system's interactive requirement, so that the user experience is satisfactory. Which of the following is true?

    (a) a preemptive scheduling mechanism is preferred for all interactive systems

    (b) a non-preemptive scheduling mechanism is preferred in a situation such as the above

    (c) starvation of high-burst processes is possible in certain scenarios

    (d) statistically, processes offer the best response times when using the optimum quantum value, instead of higher quantum values

11. On an interactive operating system, which of the following statements are true:

    (a) An user-level process can voluntarily deny pre-emption, to disallow other processes to run.

    (b) Page faults happen less often on systems with a high amount of external memory.

    (c) It is required to protect critical regions for read accesses, in order to prevent data corruption.

    (d) Setting a thread's priority to a lower level does not impose a guarantee of that thread's stall in the next scheduler execution cycle.

    (e) None of the above.

12. A system with 1 CPU and 2 cores runs the following 2 threads:

```
Thread 0: M  = N-2
Thread 1: N -= M/2
```

    If each instruction is atomic, which of the following are possible resulting values for the variables N and M, assuming that initially $N = 1$, $M = 2$?

    (a) ($N = 1.5$, $M = -1$)

    (b) ($N = 1.5$, $M = -1$) or ($N = 0.5$, $M = -1$)

    (c) ($N = 0$, $M = -1$)

(d) ($N = 0$, $M = -2$)

(e) none of the above

13. Assuming you have a system with 1 CPU and 4 cores, and 7 running threads, which of the following statements is true in an interactive operating system:

    (a) There will never be more than 4 running threads at once.

    (b) The 7 threads are always scheduled in groups of 4 threads running at once.

    (c) In a CPU with a superscalar architecture, execution of the threads should be faster overall than on a CPU that is both non-superscalar and non-out-of-order.

    (d) You cannot play more than 4 YouTube videos at once on this CPU in this operating system.

    (e) All of the above.

14. A 16-bit system with 16 KB of RAM and a page size of 4 KB has the following page accesses happening in the following order: 0, 1, 1, 0, 2, 1, 2, 0, 1. Which of the following statements are true about this system:

    (a) This system has 16 page frames and 4 virtual pages.

    (b) This system has 16 virtual pages and 4 page frames.

    (c) The sequence of accesses to the pages above causes 3 page faults.

    (d) A pointer to an array of 128 characters that points to the address $0x0011$ in memory, resides in virtual page number 11.

    (e) A pointer to an array of 32 characters that points to the address $0x0016$ in memory, resides in virtual page number 0.

15. You are asked to design an interactive operating system for users, whose main purpose is playing back YouTube videos. Which of the following choices should you make?

    (a) To ensure videos will play back smoothly, you should consider scheduling computationally-intensive processes with higher priority as opposed to I/O-bound processes.

    (b) To ensure network buffering happens fast enough, you should consider scheduling I/O-bound processes with higher priority as opposed to CPU-bound processes.

    (c) If the CPU is too slow to render the video smoothly, your OS must focus heavily on I/O to improve playback.

    (d) If the CPU is capable of rendering the video at 60fps smoothly, your OS no longer needs to do any type of I/O at all.

    (e) None of the above.

Computer Networks

1. A process socket local address is equal to:
   (a) port number + IP address;
   (b) IP address;
   (c) port number;
   (d) IP address + hostname + port number;
2. Network layer protocol that reports on the success or failure of data delivery:
   (a) IP;
   (b) TCP;
   (c) ARP;
   (d) ICMP;
3. Which of the following services use UDP?
   (a) SMTP;
   (b) SNMP;
   (c) FTP;
   (d) TFTP;
   (e) DHCP;
   (f) HTTP;
4. Which of the following devices are layer 2 devices?
   (a) bridge;
   (b) repeater;
   (c) router;
   (d) switch;
   (e) hub;
5. Which of the following IP addresses fall into the CIDR block of 115.64.4.0/22? (Choose two.)
   (a) 115.64.8.32;
   (b) 115.64.6.255;
   (c) 115.64.8.30;
   (d) 115.64.5.128;
6. What is the network topology where each node is connected to the two nearest nodes so that the entire network forms a circle?

(a) bus;
(b) ring;
(c) star;
(d) bus-star;

7. Which fields are contained within an Ethernet frame header?

(a) source and destination hardware addresses;
(b) source and destination network addresses;
(c) error correction code;
(d) authentication code;

8. Which of the following statements is true regarding a switch?

(a) it creates a single collision domain and a single broadcast domain;
(b) it creates separate collision domains but one broadcast domain;
(c) it creates separate collision domains and separate broadcast domains;

9. What is a valid IPv4 address example?

(a) 144.92.254.253;
(b) 144-92-43-178;
(c) 144.92.256.176;
(d) 144,92,43,178;

10. What does CSMA/CD stand for?

(a) Carrier Service Multiple Access with Collision Detection;
(b) Carrier Sense Multiple Access with Collision Avoidance;
(c) Carrier Sense Multiple Access with Collision Detection;
(d) Control Sense Multiple Access with Collision Direction;

Interdisciplinary subjects

1. What happens when you press a key in an online game. Place the correct answers in order. Answers: Not all suggestions are correct. (c, a, d, e, g)

(a) An event in placed in a queue in kernel space
(b) The key-code is sent to the cloud where the gameserver interprets its meaning
(c) Device driver picks up messages from the keyboard
(d) Processes in the user space read the messages from the queue

(e) User space processes use the application level in TCP/IP to package the message

(f) The operating system connect the player to the game server(s)

(g) Network drivers in the kernel space use TCP/IP to send the message to online servers that run the game's backend

2. Rank the importance of the following features when trying to maximize the transfer speed of very large volumes of data (files) between 2 identical computers over the internet

   (a) Network latency

   (b) RAM size

   (c) CPU speed

   (d) Network Bandwidth

   (e) Disk seek speed

   (f) OS type

   (g) Disk size

   (h) File transfer application

   (i) Network protocol

3. Assume your computer is running Windows 10 and has 4 GB of RAM. An application that you just started uses 6 GB of RAM to read data from a file representing a large binary tree, which has to be fully loaded in memory in order for searches in the binary tree to work correctly.

   (a) Explain how the virtual memory mechanism works in this particular context, assuming that your application is the only user-level application running in the OS.

   (b) If you add another 2 GB of RAM (getting to 6 GB RAM), will your computer run faster in any way, and if so, in what ways exactly?

   (c) If you now have 6 GB of RAM, does it mean the application that also uses 6 GB of RAM will no longer require any virtual memory at all? Explain your answer.

   (d) How and why would it impact the system's performance, if your storage is done on a SSD drive instead of a classic HDD drive?

4. You are required to design a network-based file-system, for a local area network.

   (a) Explain whether access to files stored in this file-system is a CPU-bound process or an I/O-bound process.

   (b) Assuming the LAN has fault-tolerance on each node, explain the best ways to increase the performance of the file-system with respect to each network topology.

   (c) Considering network scalability is of primary importance first and foremost, explain how you could build such a file-system to take this aspect into account.

   (d) Is there any way to increase performance of this network file-system using a caching mechanism? Explain your answer.

(e) Devise a software-based mechanism or protocol through which a query (read/write) of a network-based file returns instantly (no matter whether the file exists or not, whether it is available or not, etc.), without involving network I/O in any way.

5. You are required to design a file-system for an interactive operating system that has virtual-memory support.

   (a) Explain whether your virtual memory storage would be better placed in a separate file (e.g. as in Windows and Linux), in a separate partition (e.g. as in Linux), or a combination of both.

   (b) You are requested to design the file-system so that it does not allow any internal fragmentation. How would you conceive such a file-system, both from an algorithmic perspective as well as from a data structure perspective, and what are the best ways to increase its performance? Explain the performance of your approach.

   (c) You are requested to design the file-system so that it does not allow any external fragmentation. How would you conceive such a file-system, from an algorithmic perspective, and what are the best ways to increase its performance? Explain the performance of your approach.

6. Explain the following processes:

   (a) Explain the low-level OS and hardware interactions that take place when you open a YouTube video inside a Chrome browser tab in an interactive operating system such as Windows or MacOS X.

   (b) Explain whether the process associated to this tab during YouTube video playback is CPU-bound or I/O-bound.

   (c) Explain the differences that appear when playing back a YouTube video, and a Twitch live stream, for both questions a) and b) above.

Answers Topic III

Computer Arhitecture

1. (a), (c)
2. (c)
3. (b)
4. (b)
5. (b), (c)
6. (b)
7. (a), (b)
8. (b)
9. (a), (c), (d), (e)
10. (a)
11. (a)
12. (e)
13. (c)
14. (a), (b), (d)
15. (b)
16. (b), (e)

Operating Systems

1. (a)
2. (b)
3. (b), (c), (d)
4. (b)
5. (b), (c)
6. (a), (c), (e)
7. (a), (b)
8. (c), (d)
9. (b), (d)
10. (a), (c), (d)
11. (d)
12. (a), (d)
13. (c)
14. (b), (c), (e)
15. (a), (b)

Computer Networks

1. (a)
2. (d)
3. (b),(d),(e)
4. (a),(d)
5. (b),(d)
6. (b)
7. (a)
8. (b)
9. (a)
10. (c)