FI - Přijímací zkouška: Test z informatiky

Jméno a příjmení – pište do okénka	*Číslo přihlášky*	*Číslo zadání*
		2

The computer science test consists of 30 questions, where you choose one of the possible answers. Each question has just one correct answer. Each correctly answered question is valued by 1 point, an incorrectly answered question is valued -0.25. You get 0 points for no answer. Once you have submitted the test (by clicking the Submit button) and confirmed your submission, you will not be able to return to the test.

Algorithms and data structures

1 We have a binary tree with integer keys whose *preorder* traversal is 2, 1, 0, 7, 5, 3, 4, 6, 8, 9 and whose *postorder* traversal is 7, 0, 3, 5, 1, 6, 9, 8, 4, 2. Which keys are stored in the tree's leaves?

A 3, 4, 6, 9
B 0, 3, 6, 9
C 0, 4, 6, 8
***D** 3, 6, 7, 9
E 3, 6, 7, 8

2 Consider the following sequences of expressions; each one denotes a function of one natural argument, *n*. Which of these sequences correctly orders the functions by their asymptotic growth (from the slowest growing to the fastest growing)?

***A** *n + sqrt(n), log (n!), n * log (n) * log (n)*
B *n * log (n) * log (n), n + sqrt(n), n * log (n) * log (n)*
C *log (n!), n + sqrt(n), n * log (n) * log (n)*
D *n + sqrt(n), n * log (n) * log (n), log (n!)*
E *log (n!), n * log (n) * log (n), n + sqrt(n)*

3 Consider a B-tree (not B^+) with the maximal degree of four (i.e., a 2-3-4 tree) with preemptive splitting. The tree is initially empty. We insert the following keys in the given order: 2, 8, 12, 1, 5, 7, 9, 11. How many nodes does the resulting tree have in total?

A 7
***B** 4
C 3
D 5
E 6

4 Consider the following algorithm, where `div` is the integer division operator and `mod` is the modulo operator:

```
input a: integer
x = a
z = 0
while x > 0
        z = z + x mod 11
        x = x div 11
end while
```

The postcondition (output condition) of this algorithm is `z < 21`. Choose a precondition (input condition) that makes this algorithm totally correct.

A `a > 0`
B `a mod 11 = 1`
C `a > 0 and a mod 11 = 1`
***D** `a < 121`
E `a mod 11 + (a div 11) mod 11 < 21`

5 Which statement about the single-source shortest-paths (SSSP) graph problem is true?
V is the number of vertices and *E* is the number of edges of the graph.

***A** If all the edges have positive weight, there is an algorithm that solves the SSSP problem in $O(V^2)$ time.
B If all the edges have the same weight (positive or negative), we can use the breadth-first search (BFS) algorithm to solve the SSSP problem.
C If the graph contains negative-weight edges, we can use Dijkstra's algorithm to solve the SSSP problem in $O(V^2)$ time.
D If the graph is acyclic, we can solve the SSSP problem in $O(V)$ time.
E If the graph contains negative-weight edges, we can use Bellman and Ford's algorithm to solve the SSSP problem in $O(V^2)$ time.

Programming

6 Let us consider the following function:

```
function fun(unsigned integer n) -> unsigned integer
begin
        result = 0
        while n != 0
                digit = XXX
                result = result + digit
                n = YYY
        end while
        return result
end
```

We want the function to compute the sum of the values of all the digits in base-sixteen representation of `n`. For example, the return value for `123` is `18`, since the decimal value `123` is `7B` in base-sixteen.

What should we write instead of XXX and YYY, where `div` is the integer division operator and `mod` is the modulo operator?

A XXX = `(n mod 16) mod 10`; YYY = `n div 16`
B XXX = `n mod 16`; YYY = `n div 10`
C XXX = `n mod 10`; YYY = `n div 10`
D XXX = `n mod 10`; YYY = `n div 16`
***E** XXX = `n mod 16`; YYY = `n div 16`

---

7 Let us consider the following program. The `print` instruction outputs the given number without an end-of-line character, and `mod` is the modulo operator.

```
function foo(integer n)
begin
        if n <= 0 then
                return
        end if
        if n mod 2 == 1 then
                print(n)
        end if
        foo(n - 1)
        if n mod 2 == 0 then
                print(n)
        end if
end

program main()
begin
        foo(5)
end
```

What is going to be printed by the program?

A None of the other answers is correct.
***B** 53124
C 12345
D 42135
E 54321

---

8 Let us consider the following program in an OOP language with both virtual (late binding) and non-virtual (early binding) methods. Assume that all methods and inheritance are public, and only the methods marked `virtual` are virtual.

```
class A {
        virtual method f() { print("Af"); }
        virtual method g() { print("Ag"); }
        method h() { print("Ah"); }
}

class B inherits from A {
                                virtual   method f() { print("Bf"); g(); }
        method h() { print("Bh"); }
}

class C inherits from B {
                                virtual   method g() { print("Cg"); h(); }
}

function main() {
        ptr: pointer (reference) to A
        ptr = new C;
        ptr.f();  // call method f
}
```

What is going to be printed by the program?

A `BfCgAh`
B `BfCg`
C `Af`
D `BfAg`
***E** `BfCgBh`

---

9 On common platforms, every running program has two kinds of memory available: the **stack** and the **heap**. Consider the following statements:
I. The access to variables on the stack is asymptotically faster than the access to variables on the heap.
II. The heap is used to store local variables.
III. The stack is used to store local variables.

Which option contains all true statements and **no false** ones?

***A** III
B I, II
C I, III
D II
E I

10 Which of the following statements is **false**?

A The lazy evaluation strategy in functional programming allows working with infinite lists (also known as streams).

***B** "Call by name" and "call by value" are two names for the same function evaluation strategy.

C When using the "call by reference" evaluation strategy, all modifications of the formal parameter are actually done to the actual argument.

D Any recursive function can always be equivalently rewritten in an iterative manner.

E Any tail-recursive function can always be equivalently rewritten in an iterative manner.

Computer Networks

11 How many host IP addresses are defined by a class A network (mask 255.0.0.0)?

A 65 534, i.e., 2^{16}-2

***B** 16 777 214, i.e., 2^{24}-2

C Impossible to enumerate because it is a reserved space, which is not used yet.

D Impossible to enumerate because it is a range of multicast addresses.

E 254, i.e., 2^{8}-2

12 A metric of the OSPF protocol (Open Shortest Path First, the mostly used link state protocol) is

A the composite metric defined by CISCO.

B the number of hops multiplied by RTT (Round Trip Time, in msec).

***C** the cost assigned to each router's interface, where cost = 100 000 000 bps / bandwidth (in bps) and might be manually adjusted.

D the number of hops multiplied by MTU (Maximal Transmission Unit, in bytes).

E the cost assigned to each router's interface, where cost = bandwidth (in bps) and might be manually updated.

13 Which option is an example of a network protocol that includes a handshake, e.g., the three-way handshake, and describes why the handshake is important and what the handshake is responsible for within the selected protocol?

A Any routing protocol when routing tables are transferred by UDP. The handshake allows securing the data transfer in TLS by exchanging a shared crypto key.

B TCP and UDP. The receiver sends a negative response during the handshake to terminate the connection. Otherwise, the sender starts sending data with the congestion window set to the maximum value.

***C** A complex transport protocol, e.g., TCP. Connection parameters are negotiated during the handshake. None or negative response during the handshake means the receiver is not able to receive any data.

D UDP but only in combination with RTP and RTCP for the handshake. The handshake sets the initial size of the congestion window to prevent congestion and allows real-time communication.

E Multi Protocol Label Switching (MPLS). The handshake assigns a new label to the communicating nodes, and it ensures the labels are unique within an MPLS cloud.

14 Consider two nodes `Alice` and `Bob` connected as `Alice + ISP A - tranzit - ISP B + Bob`, i.e., three networks connected (two stubs and one transit). `Bob` and `Alice` are not able to ping **each other**. Where is the most probable fault?

A Bob's public key certificate is expired (L5+).

***B** Broken routing between networks (L3).

C Bob uses RJ-11 instead of RJ-45 on his home cabling (L1).

D Bob and Alice should use the command `route` instead of `ping` for connectivity checks (L8).

E Alice's firewall blocks incoming UDP traffic (L4+).

15 Self-correcting codes used within data transfer allow the receiver:

A to inform the sender only of errors incurred during the data transfer.

B to detect and repair all errors incurred during the data transfer.

C to detect all and repair almost all errors incurred during the data transfer.

D to inform the sender of errors incurred during the data transfer and to ask for the resubmission of wrongly received data.

***E** to detect and repair almost all errors incurred during the data transfer.

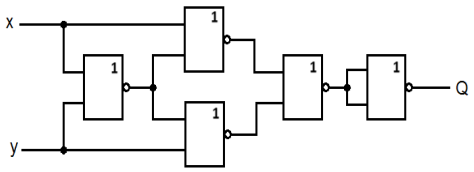
Computer systems

16 Consider an unsigned decimal number 173 stored in 8-bit register. What number will be in this register after performing a left bit rotation (circular shift) **twice**?

A 235
B 181
C 43
D 107
***E** 182

---

17 Consider the following logic circuit composed of NOR gates:



Which statement is true for the value Q at its output?

A Q = x NAND y
***B** Q = x XOR y
C Q = x XNOR y (Q = x NOXOR y)
D Q = x NOR y
E Q = x OR y

---

18 Which statement about multitasking in operating systems is **not** valid?

A Cooperative multitasking is a style of computer multitasking in which the operating system never initiates a context switch from a running process to another process.
B Preemptive multitasking is a style of computer multitasking in which the operating system kernel can initiate a context switch to satisfy the scheduling policy's priority constraint.
C Preemptive multitasking involves the use of an interrupt mechanism.
D Cooperative multitasking is a style of computer multitasking in which processes voluntarily yield control periodically or when the processes idle, which enables multiple applications to be run simultaneously.
***E** Cooperative multitasking allows the computer system to more reliably guarantee each process a regular "slice" of operating time.

---

19 A necessary condition of deadlock is

A using a semaphore.
B the need to use at least three different resources.
***C** keeping an allocated resource and waiting for another one (hold and wait).
D possibility of concurrent use of a single instance of a resource by at least two threads.
E existence of at least two instances of each resource.

---

20 Which RAID (Redundant Array of Independent Disks) type provides no redundancy?

A RAID 5
B any RAID type provides redundancy
C RAID 1
***D** RAID 0
E RAID 10

Database Systems

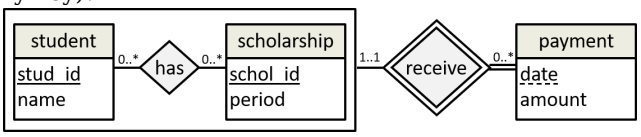
---

21 Let there be two relations: *customer(custid, name, address)* and *account(accno, custid, balance)*. The attribute *account.custid* is the FOREIGN KEY that REFERENCES *customer.custid*, and it is also NOT NULL. Select the only true statement from the following:

A If the relation customer is not empty then the relation account cannot be empty.
B The number of tuples in the relation account is always greater or equal to the number of tuples in the relation customer.
C The relation account cannot be empty.
D The relation customer cannot be empty.
***E** If the relation account is not empty then the relation customer cannot be empty.

---

22 Consider the following E-R diagram with entities *student* and *scholarship* that are connected by a many-to-many relationship *has*. Additionally, there is a weak entity *payment* that is connected by an identifying relationship *receive*. What is the correct transformation of the diagram to database relations (the underlined attributes represent the primary key)?



A *student(stud_id, name), scholarship(schol_id, period), has(stud_id, schol_id), payment(stud_id, schol_id, date, amount)*
B *student(stud_id, name), scholarship(schol_id, period), has(stud_id, schol_id), payment(date, amount), receive(stud_id, schol_id, date)*
***C** *student(stud_id, name), scholarship(schol_id, period), has(stud_id, schol_id), payment(stud_id, schol_id, date, amount)*
D *student(stud_id, name), scholarship(schol_id, period), has(stud_id, schol_id), payment(date, amount), receive(stud_id, schol_id, date)*
E *student(stud_id, name), scholarship(schol_id, period), has(stud_id, schol_id), payment(date, amount), receive(stud_id, schol_id, date)*

---

23 Consider a relation *r(A, B, C, D, E, F, G, H)*, where all attributes are atomic, and the following set of functional dependencies:
A, B → C, D, E
A → G, H
E → F
Which normal form is the most strict one that holds for the decomposition of the relation r to two relations r1(A, B, C, D, E, F) and r2(A, G, H)?

A The third normal form (3NF).
***B** The second normal form (2NF).
C Boyce-Codd normal form (BCNF).
D The first normal form (1NF).
E The fourth normal form (4NF).

---

24 Let there be relations `employee(empid, name, salary)`, `project(projid, topic, length)`, and `worker(empid, projid, fte)` where the attributes of the relation `worker` are foreign keys to employee and project relation, respectively. The attribute `fte` is the employee's workload percentage (0.0-1.0) in the project. The attribute `length` is the length of the project in months, and `salary` corresponds to the employee's monthly paycheck.

What is the correct SQL command that returns the title and total personal costs of each project?

A `SELECT topic, SUM(salary * fte * length) FROM employee, project, worker WHERE employee.empid = worker.empid AND project.projid = worker.projid`
***B** `SELECT topic, SUM(salary * fte * length) FROM (employee NATURAL INNER JOIN project) NATURAL INNER JOIN worker GROUP BY projid, topic`
C `SELECT topic, SUM(salary * fte * length) FROM (employee LEFT OUTER JOIN worker) LEFT OUTER JOIN project GROUP BY projid`
D `SELECT topic, salary * fte * length FROM employee, project, worker WHERE employee.empid = worker.empid AND project.projid = worker.projid GROUP BY topic`
E `SELECT topic, SUM(salary * fte) * SUM(length) FROM (employee NATURAL INNER JOIN worker) NATURAL INNER JOIN project GROUP BY projid, topic`

---

25 Let there be relations `employee(empid, name, salary)`, `project(projid, topic, length)`, and `worker(empid, projid, fte)` where the attributes of the relation `worker` are foreign keys to employee and project relation, respectively. The attribute `fte` is the employee's workload percentage (0.0-1.0) in the project. The attribute `length` is the length of the project in months, and `salary` corresponds to the employee's monthly paycheck.

Which relational algebra expression retrieves the names of employees who **exclusively** work on short-term projects (shorter than 6 months)?

***A** $\pi_{name}(\pi_{empid}(worker) - \pi_{empid}(\sigma_{length \geq 6}(project \bowtie worker)) \bowtie employee)$

B $\pi_{name}(employee \bowtie (\sigma_{length<6}(project) \bowtie worker))$

C $\pi_{name}(employee \bowtie (\pi_{projid}(\sigma_{length<6}(project)) - \pi_{projid}(worker)))$

D $\pi_{name}(employee) - \pi_{name}(\sigma_{length \geq 6}(project \bowtie worker \bowtie employee))$

E $\pi_{name}(\sigma_{length<6}(project \bowtie worker \bowtie employee)) - \pi_{name}(employee \bowtie worker)$

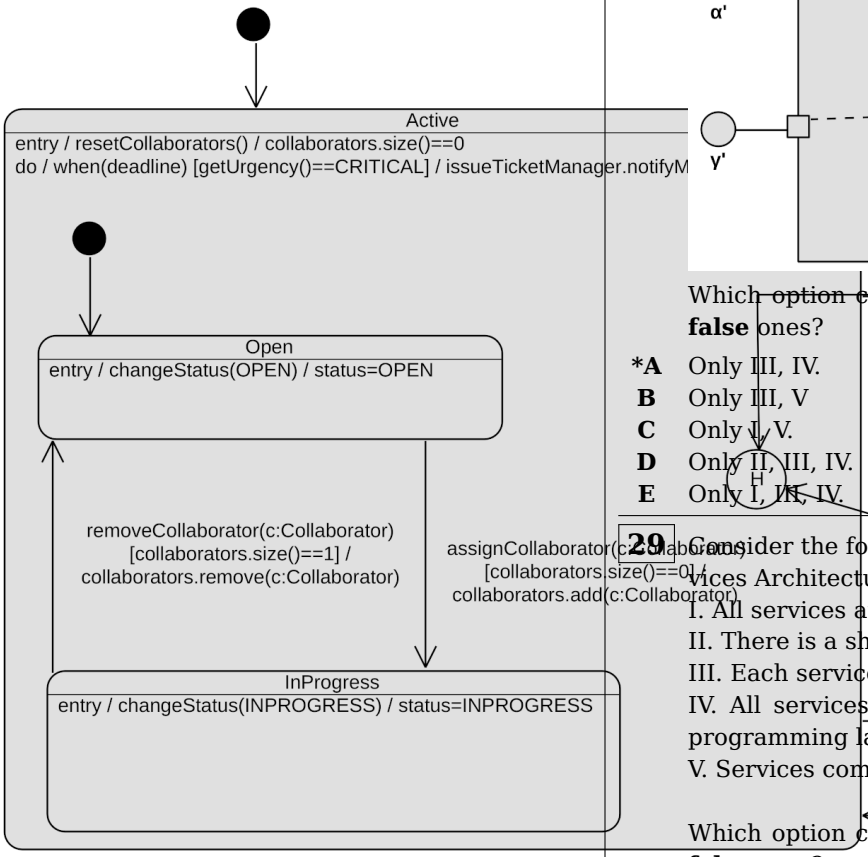
Software engineering

---

26 What do **Cohesion** and **Coupling** measures mean in the context of a software system?

A Coupling: The number of functions in a system's module. Cohesion: The number of external functions communicating with a system's module.
B Coupling and Cohesion are synonyms in Software Engineering: both measuring the internal structure of a module.
C Coupling: The number of external functions communicating with a system's module. Cohesion: The number of functions in a system's module.
***D** Coupling: The level of dependence between modules in a system. Cohesion: The level of dependence between functions within the same module.
E Coupling: The level of dependence between functions within the same module. Cohesion: The level of dependence between modules in a system.

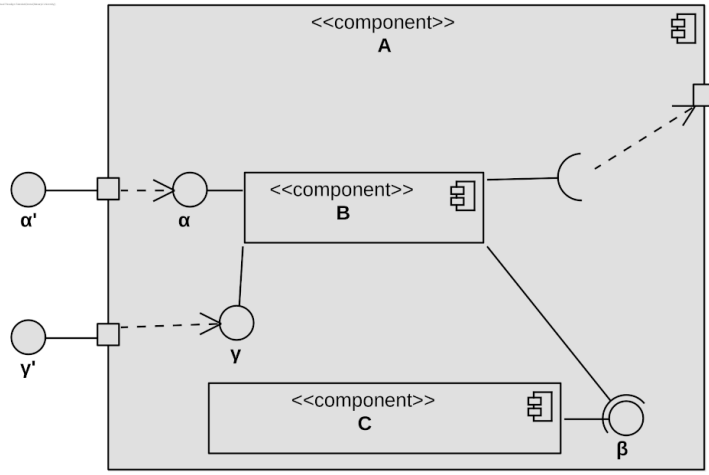
27 Based on the part of the UML State Diagram in the figure that represents an Issue entity with Active state and Active::Open and Active::InProgress substates, consider the following statements:
I. When initializing the system, the initial state will be Active with either the Active::Open or Active::InProgress substates active;
II. When initializing the system, the initial state will be Active with the Active::Open substate active;
III. Entering the Active state through the H node means halting the execution;
IV. Entering the Active state through the H node means re-entering the state from the last sub-state that was set when the Active state was exited;
V. Based on this part of the State Diagram, there can be a maximum of one collaborator assigned to an Issue.



Which option contains all true statements and **no false** ones?

***A** Only II, IV, V.
B Only II, III.
C Only I.
D Only II, III, V.
E Only I, III.

28 Based on the UML Component diagram in the figure, consider the following statements:
I. Component C is a subcomponent of component B;
II. Component B is a subcomponent of component C;
III. Component A provides interfaces α' and γ';
IV. Component A requires the interface δ from component D;
V. Component A provides the interface δ to component D;



Which option contains all true statements and **no false** ones?

***A** Only III, IV.
B Only III, V
C Only I, V.
D Only II, III, IV.
E Only I, III, IV.

29 Consider the following statements about Microservices Architecture (MSA):
I. All services are tightly coupled.
II. There is a shared database for all services.
III. Each service is independently deployable.
IV. All services are implemented using the same programming language.
V. Services communicate through shared memory.

Which option contains all true statements and **no false** ones?

A Only I, II, III.
***B** Only III.
C Only II, IV, V.
D Only I, III.
E Only I, III, V.

30 Which metric is commonly used to measure the productivity of a software development team within Agile methodologies?

A Cyclomatic Complexity
B Number of Classes
***C** Team Velocity
D Lines of Code
E Code Coverage