BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 19th 2023

Written Exam for Computer Science

IMPORTANT NOTE:
Without further clarification:

- Assume that all arithmetical operations are performed over boundless data types (no *overflow* / *underflow*).
- Arrays are indexed starting from 1.
- All restrictions apply for the actual parameter values at the time of the initial call.
- A subarray of an array is formed by elements that occupy consecutive positions in the array.

1. Let us consider the algorithm `F(x)`, where $\boldsymbol{x}$ is a natural number ($1 \leq \boldsymbol{x} \leq 10^6$):

```
Algorithm F(x):
    If x = 0 then
        Return 0
    Else
        If x MOD 3 = 0 then
            Return F(x DIV 10) + 1
        Else
            Return F(x DIV 10)
        EndIf
    EndIf
EndAlgorithm
```

Which of the following function calls will return 4?

A. `F(21369)`
B. `F(6933)`
C. `F(4)`
D. `F(16639)`

2. Let us consider the algorithm `ceFace(a, b)`, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are natural numbers ($1 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10^4$) which do not contain the digit 0.

```
Algorithm ceFace(a, b):
    p ← 0
    While a ≠ 0 execute
        c ← a MOD 10
        p ← p * 10 + c
        a ← a DIV 10
    EndWhile
    If p = b then
        Return True
    Else
        Return False
    EndIf
EndAlgoritm
```

The algorithm `ceFace(a, b)` returns *True* if and only if:

A. ***a*** and ***b*** are equal
B. ***a*** and ***b*** are palindromes
C. ***a*** is the reverse number of ***b***
D. the last digit of ***a*** equals the last digit of ***b***

3. Let us consider the algorithm `ceFace(n)`, where $\boldsymbol{n}$ is a natural number ($1 \leq \boldsymbol{n} \leq 10^3$). The operator „/” represents real division, for example: 3 / 2 = 1.5.

```
Algorithm ceFace(n):
    s ← 0
    For i ← 1, n execute
        p ← (i + 1) * (i + 2)
        s ← s + (i / p)
    EndFor
    Return s
EndAlgorithm
```

The value of which expression is returned by the algorithm?

A. $\frac{1}{1} + \frac{1}{1+2} + \cdots + \frac{1}{1+2+\cdots+\boldsymbol{n}}$

B. $\frac{1}{2*3} + \frac{2}{3*4} + \cdots + \frac{n}{(\boldsymbol{n}+1)*(\boldsymbol{n}+2)}$

C. $\frac{1}{1} + \frac{1}{1*2} + \cdots + \frac{1}{1*2*\ldots*\boldsymbol{n}}$

D. $\frac{1}{2*3} + \frac{2}{3*4} + \cdots + \frac{\boldsymbol{n}-1}{\boldsymbol{n}*(\boldsymbol{n}+1)}$

4. Let us consider the algorithm `f(n, x)`, where ***n*** is a natural number ($3 \leq n \leq 10^4$), and ***x*** is an array of ***n*** natural numbers (***x***[1], ***x***[2], ..., ***x***[***n***], $1 \leq x[i] \leq 10^4$, for ***i*** = 1, 2, ..., ***n***).

```
Algorithm f(n, x):
    k ← 0
    For i ← 1, n - 1 execute
        If k = 0 then
            If x[i] = x[i + 1] then
                Return False
            EndIf
            If x[i] < x[i + 1] then
                k ← 1
            EndIf
        EndIf
        If k = 1 then
            If x[i] ≥ x[i + 1] then
                Return False
            EndIf
        EndIf
    EndFor
    If x[n - 1] ≥ x[n] then
        Return False
    EndIf
    Return True
EndAlgorithm
```

Which of the following function calls will return *True*?

A. `f(6, [1000, 512, 23, 22, 1, 2])`
B. `f(6, [6, 4, 1, 1, 2, 3])`
C. `f(8, [3000, 2538, 799, 424, 255, 256, 299, 1001])`
D. `f(3, [3, 2, 1])`

5. Let us consider the algorithm `calcul(a, b, c, d)`, where ***a***, ***b***, ***c***, ***d*** are natural numbers ($1 \leq a, b, c, d \leq 100$).

```
Algorithm calcul(a, b, c, d):
    x ← a * b
    y ← c * d
    While y ≠ 0 execute
        z ← x MOD y
        x ← y
        y ← z
    EndWhile
    Return x
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns the greatest common divisor of the numbers ***a***, ***b***, ***c***, ***d***.
B. The algorithm returns the greatest common divisor of the numbers ***a*** * ***b*** and ***c*** * ***d***.
C. The algorithm returns the least common multiple of the numbers ***a***, ***b***, ***c***, ***d***.
D. The algorithm returns the least common multiple of the numbers ***a*** * ***b*** and ***c*** * ***d***.

6. Let us consider the algorithm `p(na, a, nb, b)`, where ***na*** and ***nb*** are natural numbers ($0 \leq na, nb \leq 10^4$), ***a*** and ***b*** are arrays of ***na***, respectively ***nb*** natural numbers (***a***[1], ***a***[2], ..., ***a***[***na***], $1 \leq a[i] \leq 10^4$, for ***i*** = 1, 2, ..., ***na*** and ***b***[1], ***b***[2], ..., ***b***[***nb***], $1 \leq b[i] \leq 10^4$, for ***i*** = 1, 2, ..., ***nb***). The local variable ***c*** is an array.

```
Algorithm p(na, a, nb, b):
    i ← 1
    j ← 1
    nc ← 0
    While i ≤ na AND j ≤ nb execute
        nc ← nc + 1
        If a[i] < b[j] then
            c[nc] ← a[i]
            i ← i + 1
        Else
            c[nc] ← b[j]
            j ← j + 1
        EndIf
    EndWhile
    Return nc
EndAlgorithm
```

Which of the following statements are true?

A. If ***na*** = 0 and ***nb*** = 0, then the value returned by ***nc*** is equal to 0.
B. If the elements from ***a*** and ***b*** are in ascending order, then the elements stored in ***c*** are in ascending order.
C. The value returned through ***nc*** is always equal to ***na*** + ***nb***.
D. If ***na***, ***nb*** > 0 and the greatest element of ***a*** is smaller than all elements of ***b***, then ***c*** will have the same elements as ***a***.

7. Let us consider the algorithm `suma(n, a, m, b)`, where ***n*** and ***m*** are natural numbers ($1 \le \boldsymbol{n}, \boldsymbol{m} \le 10^5$), ***a*** and ***b*** are two arrays in ascending order having as elements ***n***, respectively ***m*** natural numbers (***a***[1], ***a***[2], …, ***a***[***n***] and ***b***[1], ***b***[2], …, ***b***[***m***]):

```
Algorithm suma(n, a, m, b):
    s ← 0
    For i ← 1, n, 2 execute
        j ← 1
        While j ≤ a[i] AND j ≤ m execute
            s ← s + b[j]
            j ← j + 1
        EndWhile
    EndFor
    Return s
EndAlgorithm
```

What value will the algorithm return, if ***n*** = 4, ***a*** = [1, 3, 4, 7], ***m*** = 6 and ***b*** = [2, 4, 6, 8, 10, 12]?

A. 42
B. 22
C. 20
D. It is not possible to determine the value that the algorithm will return.

8. Let us consider the algorithm `verifica(n, p1, p2)`, where ***n***, ***p1*** and ***p2*** are natural numbers ($1 \le \boldsymbol{n}, \boldsymbol{p1}, \boldsymbol{p2} \le 10^6$):

```
Algorithm verifica(n, p1, p2):
    bt ← (p1 + p2) DIV 2
    If p1 > p2 then
        Return False
    EndIf
    If bt * bt = n then
        Return True
    EndIf
    If bt * bt > n then
        Return verifica(n, p1, bt - 1)
    EndIf
    Return verifica(n, bt + 1, p2)
EndAlgorithm
```

Which of the following statements are true?

A. If ***p1***, ***p2*** and ***n*** are relatively prime, then the algorithm `verifica(n, p1, p2)` returns *True*.
B. The algorithm uses the binary search method and if ***n*** is prime, the call `verifica(n, 1, n)` returns *True*.
C. For the call `verifica(n, 1, n)` the algorithm returns *True* if and only if ***n*** is a square number.
D. If $\boldsymbol{p1} \le \boldsymbol{n} \le \boldsymbol{p2}$ and in each of the intervals [***p1***, ***n***] and [***n***, ***p2***] there exists at least one square number, then the call `verifica(n, p1, p2)` returns *True*.

9. Let us consider the algorithm `ceFace(n)`, where ***n*** is a natural number ($1 \le \boldsymbol{n} \le 3000$).

```
Algorithm ceFace(n):
    s ← 0
    i ← 1
    While s < n execute
        s ← s + i
        If s = n then
            Return True
        Else
            i ← i + 2
        EndIf
    EndWhile
    Return False
EndAlgorithm
```

Which of the following statements are true?

A. If ***n*** = 36, the algorithm returns *True*.
B. If ***n*** is equal to a sum of odd consecutive numbers starting from 1, the algorithm returns *True*.
C. If ***n*** is a square number, the algorithm returns *True*, otherwise it returns *False*.
D. If ***n*** = 64, the algorithm returns *False*.

10. Let us consider the algorithm `ceFace(a)`, where ***a*** is a natural number ($1 \le \boldsymbol{a} \le 10^4$).

```
Algorithm ceFace(a):
    ok ← 0
    While ok = 0 execute
        b ← a
        c ← 0
        While b ≠ 0 execute
            c ← c * 10 + b MOD 10
            b ← b DIV 10
        EndWhile
        If c = a then
            ok ← 1
        Else
            a ← a + 1
        EndIf
    EndWhile
    Return a
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns the smallest palindrome greater than or equal to ***a***.
B. The algorithm returns the largest palindrome smaller than or equal to ***a***.
C. The algorithm returns the smallest palindrome greater than ***a***.
D. The algorithm returns the smallest even number greater than ***a***.

11. Let us consider the algorithm `calcul(v, n)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 10^4$), and $\boldsymbol{v}$ is an array of $\boldsymbol{n}$ natural numbers ($\boldsymbol{v}[1], \boldsymbol{v}[2], ..., \boldsymbol{v}[\boldsymbol{n}]$, $1 \le \boldsymbol{v}[\boldsymbol{i}] \le 10^4$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$):

```
Algorithm calcul(v, n):
    i ← 2
    x ← 0
    If v[1] MOD 2 ≠ 0 then
        Return False
    EndIf
    While i ≤ n execute
        If x = 0 AND v[i] MOD 2 = 0 then
            Return False
        Else
            If x = 1 AND v[i] MOD 2 = 1 then
                Return False
            Else
                i ← i + 1
                x ← (x + 1) MOD 2
            EndIf
        EndIf
    EndWhile
    Return True
EndAlgorithm
```

In which of the following situations does the algorithm return *True*?

A. If the array $\boldsymbol{v}$ contains the values [2, 3, 10, 7, 20, 5, 18] and $\boldsymbol{n} = 7$
B. If the array $\boldsymbol{v}$ has values according to the following pattern: odd, even, odd, even...
C. If the array $\boldsymbol{v}$ contains the values [3, 8, 17, 20, 15, 10] and $\boldsymbol{n} = 6$
D. If the array $\boldsymbol{v}$ has values according to the following pattern: even, odd, even, odd...

12. Let us consider the algorithm `ceFace(a, n)`, where $\boldsymbol{n}$ is a natural number ($2 \le \boldsymbol{n} \le 10^4$) and $\boldsymbol{a}$ is an array of $\boldsymbol{n}$ integer numbers ($\boldsymbol{a}[1], \boldsymbol{a}[2], ..., \boldsymbol{a}[\boldsymbol{n}]$, $-100 \le \boldsymbol{a}[i] \le 100$, $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$). In the array $\boldsymbol{a}$ there is at least one positive number.

```
Algorithm ceFace(a, n):
    b ← 0
    c ← b
    For i ← 1, n execute
        b ← b + a[i]
        If b < 0 then
            b ← 0
        EndIf
        If b > c then
            c ← b
        EndIf
    EndFor
    Return c
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns the sum of all elements of the array $\boldsymbol{a}$.
B. The algorithm returns the sum of the elements of the subarray of maximum length that contains only positive elements from array $\boldsymbol{a}$.
C. The algorithm returns the sum of all positive elements in the array $\boldsymbol{a}$.
D. The algorithm returns the sum of a subarray with the maximum sum from array $\boldsymbol{a}$.

13. Let us consider the matrix $\boldsymbol{A}$ of integer numbers with $\boldsymbol{n}$ rows and $\boldsymbol{m}$ columns ($1 \le \boldsymbol{n}, \boldsymbol{m} \le 10^4$). Considering that $\boldsymbol{n} * \boldsymbol{m} = \boldsymbol{p} * \boldsymbol{q}$, we intend to resize this matrix to a matrix $\boldsymbol{B}$ of integer numbers having $\boldsymbol{p}$ rows and $\boldsymbol{q}$ columns ($1 \le \boldsymbol{p}, \boldsymbol{q} \le 10^4$), as in the example below, where $\boldsymbol{n} = 4$, $\boldsymbol{m} = 6$, $\boldsymbol{p} = 3$ and $\boldsymbol{q} = 8$. Rows and columns are indexed starting from 1.

A:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24

B:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24

Which of the following options presents an algorithm that, for the pair of natural numbers $\boldsymbol{i}$ and $\boldsymbol{j}$ ($1 \le \boldsymbol{i} \le \boldsymbol{n}$, $1 \le \boldsymbol{j} \le \boldsymbol{m}$) that represent indexes in matrix $\boldsymbol{A}$, will return the pair of indexes from $\boldsymbol{B}$ corresponding to the value $\boldsymbol{A}[\boldsymbol{i}][\boldsymbol{j}]$?

A.
```
Algorithm reshape(i, j, n, m, p, q):
    Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm
```

B.
```
Algorithm reshape(i, j, n, m, p, q):
    i ← i - 1
    j ← j - 1
    Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm
```

C.
```
Algorithm reshape(i, j, n, m, p, q):
    i ← i - 1
    j ← j - 1
    Return (i * m + j) DIV q + 1,
                                (i * m + j) MOD q + 1
EndAlgorithm
```

D.
```
Algorithm reshape(i, j, n, m, p, q):
    Return (i * m + j - 1) DIV q + 1,
                          (i * m + j - 1) MOD q + 1
EndAlgorithm
```

14. Let us consider the algorithm `ceFace(n, m)`, where ***n*** is a natural number ($1 \leq n \leq 10^4$), and ***m*** is a matrix with ***n*** rows and ***n*** columns, and its elements are natural numbers (***m***[1][1], ..., ***m***[1][***n***], ***m***[2][1], ..., ***m***[2][***n***], ..., ***m***[***n***][1], ..., ***m***[***n***][***n***]). Let us consider that the elements of matrix ***m*** are initially equal to 0.

```
Algorithm ceFace(n, m):
    a ← 0
    b ← 1
    For j ← 1, n execute
        i ← 1
        While i + j ≤ n − 1 execute
            If (i MOD 2 = 1) AND (j MOD 2 = 1) then
                m[i][j] ← b
                c ← a + b
                a ← b
                b ← c
            EndIf
            i ← i + 1
        EndWhile
    EndFor
EndAlgorithm
```

Which of the following statements are **FALSE**?

A. If ***n*** = 11, the value of ***m***[6][4] is 21
B. If ***n*** = 7, the value of ***m***[3][5] is 4
C. If ***n*** = 10, the value of ***m***[6][4] is 21
D. If ***n*** = 7, the maximum value in the matrix is 8

15. The algorithms below process an ascending sorted array ***x***, having ***n*** natural numbers elements ($1 \leq n \leq 10^4$, ***x***[1], ***x***[2], ..., ***x***[***n***]). Parameters ***first*** and ***last*** are natural numbers ($1 \leq first \leq last \leq n$).

Choose the algorithms that have the lowest time complexity when called in the form of `A(x, 1, n, n)`.

A.
```
Algorithm A(x, first, last, n):
    If first > last then
        Return 0
    EndIf
    m ← (first + last) DIV 2
    If x[m] = n then
        Return m
    Else
        If x[m] > n then
            Return A(x, first, m - 1, n)
        Else
            If x[m] < n then
                Return A(x, m + 1, last, n)
            EndIf
        EndIf
    EndIf
EndAlgorithm
```

B.
```
Algorithm A(x, first, last, n):
    While first < last execute
        m ← (first + last) DIV 2
        If x[m] = n then
            Return m
        Else
            If x[m] > n then
                last ← m - 1
            Else
                If x[m] < n then
                    first ← m + 1
                EndIf
            EndIf
        EndIf
    EndWhile
    Return 0
EndAlgorithm
```

C.
```
Algorithm A(x, first, last, n):
    For i ← first, last execute
        If x[i] = n then
            Return i
        EndIf
    EndFor
    Return 0
EndAlgorithm
```

D.
```
Algorithm A(x, first, last, n):
    For i ← first, last execute
        If x[i] = n then
            x[i] ← 3 * n
        EndIf
    EndFor
EndAlgorithm
```

16. Andrei is playing with the following algorithm, where ***n*** and ***m*** are non-zero natural numbers ($1 \leq n, m \leq 10^4$). The algorithm `abs(x)` returns the absolute value of ***x***.

```
Algorithm problema(n, m):
    b ← abs(m - n)
    c ← n - m
    If b - c = 0 then
        a ← n MOD m
    Else
        a ← (m + 2) MOD n
    EndIf
    Return a
EndAlgorithm
```

He observes that regardless of the value of the variable ***n*** corresponding to the specification, there are at least two values of ***m*** for which the algorithm `problema(n, m)` returns 0. What are these values of ***m***?

A. 1 and ***n***
B. 1 and ***n*** + 2
C. ***n*** and ***n*** + 2
D. 1 and ***n*** – 2

17. A student wants to generate, using the backtracking method, all odd numbers with three digits, with digits taken from the array [4, 3, 8, 5, 7, 6], in the given order. Knowing that the first 5 generated numbers are, in this order: 443, 445, 447, 433, 435, what will be the tenth generated number?

A. 487 B. 453 C. 457 D. 455

18. Let us consider the algorithm `f(k, n, x)`, where ***k***, ***n*** are natural numbers ($1 \leq k, n \leq 10^3$) and ***x*** is an array of ***n*** natural numbers (***x***[1], ***x***[2], …, ***x***[***n***], $1 \leq x[i] \leq 10^4$, for ***i*** = 1, 2, ..., ***n***).

```
Algorithm f(k, n, x):
    If n = 0 then
        Return 0
    Else
        d ← 0
        For i ← 2, x[n] DIV 2 execute
            If (x[n] MOD i) = 0 then
                d ← d + 1
            EndIf
        EndFor
        If d = k then
            Return 1 + f(k, n - 1, x)
        Else
            Return f(k, n - 1, x)
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. For ***x*** = [4, 9, 26, 121] the result of the call `f(1, 4, x)` will be 3.
B. For ***x*** = [4, 8, 6, 144] the result of the call `f(2, 4, x)` will be 3.
C. For ***x*** = [4, 9, 25, 144] the result of the call `f(1, 4, x)` will be 3.
D. For ***x*** = [8, 27, 25, 121] the result of the call `f(2, 4, x)` will be 3.

19. Let us consider the algorithm `check(n)`, where ***n*** is a natural number ($1 \leq n \leq 10^5$).

```
Algorithm check(n):
    While n > 0 execute
        If n MOD 3 > 1 then
            Return False
        EndIf
        n ← n DIV 3
    EndWhile
    Return True
EndAlgorithm
```

Specify the effect of the algorithm.

A. The algorithm returns *True* if ***n*** is a power of 3 and *False* otherwise.
B. The algorithm returns *True* if the representation of ***n*** in base 3 contains only digits 0 and 1 and *False* otherwise.
C. The algorithm returns *True* if ***n*** can be written as a power of 3 or as a sum of distinct powers of 3 and *False* otherwise.
D. The algorithm returns *True* if the representation of ***n*** in base 3 contains only digit 2 and *False* otherwise.

20. One event was supposed to take place in a certain room I, but must be moved to room II, where the numbering of the seats is different. In both rooms there are ***L*** rows of chairs ($2 \leq L \leq 50$), each row is divided halfway by an aisle and has ***K*** chairs ($2 \leq K \leq 50$) on each side of the aisle (hence, the room has a total of 2 * ***K*** * ***L*** chairs).

In room I, each seat is identified by a single number. The seats on the left of the aisle have even numbers, and the numbering of seats begins with the row in front of the scene. Therefore, the chairs in the front row have numbers (starting from the aisle toward the edge of the room) 2, 4, 6 etc. After all the chairs from a row were numbered, the numbering on the following row begins with the chair next to the aisle using the next even number. The seats on the right of the aisle are numbered similarly but using odd numbers. Hence, the chairs in the first row have the numbers (starting from the aisle toward the edge of the room) 1, 3, 5 etc.

In room II each seat is identified by three values. Row number (a value between 1 and ***L***, row 1 being the one in front of the scene), the direction of the seat related to the aisle (value "stânga" (left) or "dreapta" (right)) and the number of the seat in that row (a value between 1 and ***K***, chair 1 being next to the aisle).

Since the event must be relocated, the seats on the tickets for room I (given by a single number) must be transformed to valid seats in room II (given by ***row***, ***seat***, ***direction***).

Which of the following algorithms, given input data ***L, K, nrLoc*** according to the statement, correctly performs the transformation? A transformation is considered correct if each spectator will have a unique seat in room II.

A.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ←  nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

B.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ← nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

C.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    rand ← nrLoc DIV (2 * K) + 1
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

D.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ← nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2 + 1
    Return rand, loc, directie
EndAlgorithm
```

21. Let us consider algorithm `p(x, n, k, final)`, where ***x*** is an array of ***n*** + 1 natural numbers (***x***[0], ***x***[1], ..., ***x***[***n***]). Initially ***x***[***i***] = 0, for ***i*** = 0, 1, 2, ..., ***n***. Variables ***n*** and ***k*** are non-zero natural numbers (1 ≤ ***n***, ***k*** ≤ 20), and ***final*** is of type boolean. The algorithm `Afis(x, 1, n)` displays the elements ***x***[1], ***x***[2], ..., ***x***[***n***].

```
Algorithm p(x, n, k, final):
    While final = False execute
        While x[k] < n execute
            x[k] ← x[k] + 1
            If OK(x, k) = True then
                If k = n then
                    Afis(x, 1, n)
                Else
                    k ← k + 1
                    x[k] ← 0
                EndIf
            EndIf
        EndWhile
        __________ // insert code sequence here
    EndWhile
EndAlgorithm
```

```
Algorithm OK(x, k):
    For i ← 1, k – 1 execute
        If x[k] = x[i] then
            Return False
        EndIf
    EndFor
    Return True
EndAlgorithm
```

What code sequence should be inserted into the algorithm so that, after calling `p(x, n, 1, False)` all permutations of order ***n*** are displayed, each only once?

A.

```
If k > 1 then
    k ← k – 1
Else
    final ← True
EndIf
```

B.

```
If k > 0 then
    k ← k - 1
Else
    final ← True
EndIf
```

C.

```
final ← True
```

D.

```
If k > 1 then
    k ← k - 1
    final ← True
EndIf
```

22. Let us consider the algorithms `problema(n)` and `calcul(a, b)`, where n, a, b are natural numbers ($0 \le n, a, b \le 9$).

```
Algorithm problema(n):
    rezultat ← 0
    For k ← 0, n execute
        For p ← 0, n execute
            For j ← 0, n execute
                If p MOD 2 = 0 then
                    rezultat ← rezultat + 1
                EndIf
            EndFor
        EndFor
    EndFor
    Return rezultat
EndAlgorithm
```

```
Algorithm calcul(a, b):
    t ← 0
    For cifra ← a, b execute
        t ← t + problema(cifra)
    EndFor
    Write t
EndAlgorithm
```

Which of the following statements are true?

A. The call `calcul(1, 8)` displays 1095.
B. The call `calcul(1, 8)` displays 1094.
C. The call `calcul(0, 9)` displays 1095.
D. The call `calcul(0, 9)` displays 1595.

23. Let us consider the algorithm `checkAcc(n, f, w, lw)`, where n is a non-zero natural number ($1 \le n \le 10^4$), f is a natural number, w is an array of lw ($1 \le lw \le 10^4$) natural numbers ($w[1], w[2], ..., w[lw]$, where $0 \le w[p] \le 10^4$, for $p = 1, 2, ..., lw$). The algorithm `checkAcc(n, f, w, lw)` calls the algorithm `t(i, j, k)`, where i, j and k are natural numbers. The algorithm `t(i, j, k)` returns a boolean result.

```
Algorithm checkAcc(n, f, w, lw):
    acc ← True
    If lw = 0 AND f ≠ 1 then
        acc ← False
    Else
        index ← 1
        q ← 1
        While (acc = True) AND (index ≤ lw) execute
            crt ← 1
            changed ← False
            While (changed = False) AND (crt ≤ n) execute
                If t(q, w[index], crt) then
                    q ← crt
                    changed ← True
                Else
                    crt ← crt + 1
                EndIf
            EndWhile
            If changed = False then
                acc ← False
            Else
                index ← index + 1
            EndIf
        EndWhile
        If (index > lw) AND (acc = True) AND (q ≠ f) then
            acc ← False
        EndIf
    EndIf
    Return acc
EndAlgorithm
```

In which of the following situations the algorithm `checkAcc(2, f, w, lw)` returns *True*, knowing that the algorithm `t(i, j, k)` returns *True* for the values inside the table, and otherwise returns *False*?

i	***j***	***k***
1	0	1
1	1	2
2	1	2

A. $w = [0, 0, 1, 1]$, $lw = 4$ and $f = 1$
B. $w = [1, 1, 1, 0]$, $lw = 4$ and $f = 2$
C. $w = [0, 0, 1, 1]$, $lw = 4$ and $f = 2$
D. $w = [0, 0, 0, 0]$, $lw = 4$ and $f = 1$

24. Let us consider the array of digits $a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$. To display the elements of array a in a different order, the array b (initially empty) is constructed. At each step, one can choose one of the following operations:

- *Add* – the first element of array a is added to the end of array b and is deleted from array a.
- *Delete* – displays, then deletes the last element of array b.

Notes:

- The elements of array a are processed in the given order.
- The *Add* operation cannot be used if array a is empty and the *Delete* operation cannot be used if array b is empty.
- The processing ends when arrays a and b are both empty.

Which of the following digit orderings **CANNOT** be displayed by considering the rules above?

A. `0 1 2 3 4 5 6 7 8 9`
B. `9 8 7 6 5 4 3 2 1 0`
C. `2 4 6 5 3 7 0 1 9 8`
D. `2 3 1 4 5 0 8 9 7 6`

BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 19th, 2023

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

1.	AB	3.75 points
2.	C	3.75 points
3.	B	3.75 points
4.	AC	3.75 points
5.	B	3.75 points
6.	ABD	3.75 points
7.	B	3.75 points
8.	C	3.75 points
9.	ABC	3.75 points
10.	A	3.75 points
11.	AD	3.75 points
12.	D	3.75 points
13.	C	3.75 points
14.	ABC	3.75 points
15.	AB	3.75 points
16.	A	3.75 points
17.	B	3.75 points
18.	AC	3.75 points
19.	BC	3.75 points
20.	A	3.75 points
21.	A	3.75 points
22.	BD	3.75 points
23.	CD	3.75 points
24.	C	3.75 points