BABEȘ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 17th, 2025

Written exam for Computer Science

IMPORTANT NOTE:

Unless otherwise specified:

- All arithmetic operations are performed on unlimited data types (there is no *overflow* / *underflow*).
- Arrays, matrices and strings are indexed starting from 1.
- All restrictions apply to the values of the actual parameters at the time of the initial call.
- A subarray of an array or a string consists of elements that occupy consecutive positions in the array or in the string.
- If on the same row there are several consecutive assignment statements, they are separated by "; ".

1. Consider the algorithm `ceFace(n, x)`, where n is a natural number ($1 \leq n \leq 10^4$), and x is an array with n integer elements ($x[1], x[2], ..., x[n]$, $-100 \leq x[i] \leq 100$, for $i = 1, 2, ..., n$).

Which of the following implementations of the algorithm `ceFace(n, x)` return the greatest value from array x?

A.
```
Algorithm aux(n, x, i, b):
    If i > n then
        Return b
    EndIf
    If b < x[i] then
        b ← x[i]
    EndIf
    Return aux(n, x, i + 1, b)
EndAlgorithm

Algorithm ceFace(n, x):
    Return aux(n, x, 1, x[1])
EndAlgorithm
```

B.
```
Algorithm ceFace(n, x):
    i ← 1
    While i < n execute
        If x[i] ≥ x[i + 1] then
            Return False
        EndIf
        i ← i + 1
    EndWhile
    Return True
EndAlgorithm
```

C.
```
Algorithm ceFace(n, x):
    i ← 1
    b ← x[i]
    While i < n execute
        If b < x[i + 1] then
            b ← x[i + 1]
        EndIf
        i ← i + 1
    EndWhile
    Return b
EndAlgorithm
```

D.
```
Algorithm ceFace(n, x):
    i ← 1
    b ← x[i]
    While i < n execute
        If b > x[i + 1] then
            b ← x[i + 1]
        EndIf
        i ← i + 1
    EndWhile
    Return b
EndAlgorithm
```

2. Consider the algorithms `f1(a, b)` and `f2(a, b)`, where a and b are natural numbers ($1 \leq a, b \leq 10^4$).

```
Algorithm f1(a, b):
    While b ≠ 0 execute
        temp ← b
        b ← a MOD b
        a ← temp
    EndWhile
    Return a
EndAlgorithm

Algorithm f2(a, b):
    Return f1(a, b) = 1
EndAlgorithm
```

In what cases does the algorithm `f2(a, b)` return *True*?

A. If and only if the numbers a and b are numbers from the *Fibonacci* sequence.
B. If and only if the sum of the digits of the number $a + b$ is equal to the sum of the digits of the number $a – b$.
C. If and only if the number of even digits of the number $a – b$ is equal to the number of odd digits of the number $a + b$.
D. If and only if the numbers a and b are relatively primes.

3. Consider the algorithm `f(n)`, where n is a natural number ($3 \le n \le 10^3$).

```
Algorithm f(n):
    If n = 1 then
        Return 0
    Else
        Return (2 * n - 3) * (2 * n - 1) + f(n - 1)
    EndIf
EndAlgorithm
```

Which of the following sums are computed by the algorithm `f(n)`?

A. $\sum_{k=1}^{n}(2k-3)(2k-1)$

B. $\sum_{k=1}^{n-1}(2k-1)(2k+1)$

C. $\sum_{k=2}^{n-1}(2k-1)(2k+1)$

D. $\sum_{k=2}^{n}(2k-3)(2k-1)$

4. Consider the algorithm `f(x, y)`, where x and y are natural numbers ($3 \le x, y \le 10^3$):

```
Algorithm f(x, y):
    If x = 1 then
        Return y
    EndIf
    If x MOD 2 = 0 then
        If x > 0 then
            Return f(x DIV 2, y * 2)
        Else
            Return 0
        EndIf
    EndIf
    Return y + f(x DIV 2, y * 2)
EndAlgorithm
```

Which of the following algorithms return the same value as the algorithm `f(x, y)` for any values of x and y?

A.

```
Algorithm a(x, y):
    If x = 0 then
        Return 0
    EndIf
    If x MOD 2 = 0 then
        Return 2 * a(x DIV 2, y)
    EndIf
    Return y + a(x - 1, y)
EndAlgorithm
```

B.

```
Algorithm b(x, y):
    If x = 0 then
        Return y
    EndIf
    If x MOD 2 = 0 then
        Return b(x DIV 2, y)
    EndIf
    Return y + b(x - 1, y)
EndAlgorithm
```

C.

```
Algorithm c(x, y):
    If x = 0 then
        Return 0
    EndIf
    Return y + c(x - 1, y)
EndAlgorithm
```

D.

```
Algorithm d(x, y):
    s ← 0
    While x > 0 execute
        s ← s + y
        x ← x - 1
    EndWhile
    Return s
EndAlgorithm
```

5. Consider the algorithm `ceFace(x, n)`, where n is a natural number ($1 \le n \le 10^4$), and x is an array with n integer elements ($x[1], x[2], ..., x[n]$, $-100 \le x[i] \le 100$, for $i = 1, 2, ..., n$):

```
Algorithm ceFace(x, n):
    For i ← 1, 10 execute
        For j ← 1, n - 1 execute
            If x[j] > x[j + 1] then
                tmp ← x[j]
                x[j] ← x[j + 1]
                x[j + 1] ← tmp
            EndIf
        EndFor
    EndFor
EndAlgorithm
```

Which of the following statements will be true after the call `ceFace(x, n)`?

A. The first position of array x will contain the minimum element of the array.

B. The last position of array x will contain the maximum element of the array.

C. The array x is sorted in ascending order.

D. If $n > 10$, then the first 10 elements of array x are sorted in ascending order.

6. Consider the natural number $\boldsymbol{n}$ ($1 \le \boldsymbol{n} \le 10^4$) and array $\boldsymbol{x}$ consisting of $\boldsymbol{n}$ natural numbers ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$, $1 \le \boldsymbol{x}[\boldsymbol{i}] \le 10^3$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$). The algorithm `gcd(a, b)` returns the greatest common divisor of the natural numbers $\boldsymbol{a}$ and $\boldsymbol{b}$.

Which of the following implementations of the algorithm `gcdVector(x, n)` return the greatest common divisor of all the $\boldsymbol{n}$ elements of array $\boldsymbol{x}$?

A.
```
Algorithm gcdVector(x, n):
    If n = 1 then
        Return 1
    EndIf
    Return gcd(x[n], gcdVector(x, n - 1))
EndAlgorithm
```

B.
```
Algorithm gcdVector(x, n):
    result ← x[1]
    For i ← 2, n execute
        result ← gcd(x[i - 1], x[i])
    EndFor
    Return result
EndAlgorithm
```

C.
```
Algorithm gcdVector(x, n):
    result ← gcd(x[1], x[n])
    i ← 2; j ← n - 1
    While i ≤ j execute
        result ← gcd(result, gcd(x[i], x[j]))
        i ← i + 1
        j ← j - 1
    EndWhile
    Return result
EndAlgorithm
```

D.
```
Algorithm gcdVector2(x, n, i):
    If i = n then
        Return x[n]
    EndIf
    Return gcd(x[i], gcdVector2(x, n, i + 1))
EndAlgorithm

Algorithm gcdVector(x, n):
    Return gcdVector2(x, n, 1)
EndAlgorithm
```

7. Consider the algorithm `afla(n, x)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 10^4$), and $\boldsymbol{x}$ is an array of $\boldsymbol{n}$ integer elements ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$, $-100 \le \boldsymbol{x}[\boldsymbol{i}] \le 100$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$):

```
Algorithm afla(n, x):
    M ← x[1]
    For i ← 1, n execute
        For j ← i, n execute
            For k ← j, n execute
                If M < x[i] + x[j] + x[k] then
                    M ← x[i] + x[j] + x[k]
                EndIf
            EndFor
        EndFor
    EndFor
    Return M
EndAlgorithm
```

What is the time complexity of the algorithm?

A. $O(3 * \boldsymbol{n})$
B. $O(\boldsymbol{n}^3)$
C. $O(\boldsymbol{n} / 3)$
D. $O(\log_3 \boldsymbol{n})$

8. Consider the algorithm `ceFace(n)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 100$). The algorithm `min(a, b)` returns the minimum value between the numbers $\boldsymbol{a}$ and $\boldsymbol{b}$.

```
Algorithm ceFace(n):
    c1 ← 0; c2 ← 0
    For i ← 1, n execute
        v ← i
        While v MOD 2 = 0 execute
            c1 ← c1 + 1
            v ← v DIV 2
        EndWhile
        While v MOD 5 = 0 execute
            c2 ← c2 + 1
            v ← v DIV 5
        EndWhile
    EndFor
    Return min(c1, c2)
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns the number of even numbers from the interval $[1, \boldsymbol{n}]$.
B. The algorithm returns the number of numbers that are divisible by 5 from the interval $[1, \boldsymbol{n}]$.
C. The algorithm returns the number of consecutive zero digits found at the end of the sum of the first $\boldsymbol{n}$ natural numbers.
D. None of the previous answers are correct.

9. Consider the array $\boldsymbol{x}$ with $\boldsymbol{n}$ ($2 \le \boldsymbol{n} \le 10^5$) integer elements ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$, $-100 \le \boldsymbol{x}[\boldsymbol{i}] \le 100$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$). The algorithm `max(a, b)` returns the maximum value between the numbers $\boldsymbol{a}$ and $\boldsymbol{b}$.

Which of the following implementations of algorithm `maxPePozitiePara(x, n)` return the greatest element found on an even position in the array?

A.
```
Algorithm maxPePozitiePara(x, n):
    maxVal ← x[2]
    For i ← 4, n, 2 execute
        If x[i] > maxVal then
            maxVal ← x[i]
        EndIf
    EndFor
    Return maxVal
EndAlgorithm
```

B.
```
Algorithm maxPePozitiePara(x, n):
    maxVal ← -100
    For i ← 1, n, 2 execute
        If x[i] > maxVal then
            maxVal ← x[i]
        EndIf
    EndFor
    Return maxVal
EndAlgorithm
```

C.
```
Algorithm maxPePozitiePara2(x, n, i):
    If i > n then
        Return -100
    EndIf
    maxRest ← maxPePozitiePara2(x, n, i + 2)
    Return max(x[i], maxRest)
EndAlgorithm

Algorithm maxPePozitiePara(x, n):
    Return maxPePozitiePara2(x, n, 1)
EndAlgorithm
```

D.
```
Algorithm maxPePozitiePara2(x, n, i):
    If i > n then
        Return -100
    EndIf
    maxRest ← maxPePozitiePara2(x, n, i + 2)
    Return max(x[i], maxRest)
EndAlgorithm

Algorithm maxPePozitiePara(x, n):
    Return maxPePozitiePara2(x, n, 2)
EndAlgorithm
```

10. We consider the following statements to be true:

1. Ana goes for a walk only if it is sunny.
2. Maria goes for a walk only if Ana goes for a walk.
3. If it is sunny, Tudor goes for a walk.

Which of the following conclusions can be deduced from the given statements?

A. If Maria goes for a walk, then Tudor also goes for a walk.
B. If it is not sunny, then Ana does not go for a walk.
C. If it is sunny, then Maria goes for a walk.
D. If it is not sunny, then Tudor does not go for a walk.

11. Consider the number $\boldsymbol{x} = 10000110111_{(2)}$ in base 2 and number $\boldsymbol{y} = 11011_{(4)}$ in base 4.

What is the value of the sum $\boldsymbol{x} + \boldsymbol{y}$ in base 10?

A. 1079 B. 1404 C. 2285 D. None of the options A, B and C

12. Consider the algorithm `ceFace(n, a)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 10^4$), and $\boldsymbol{a}$ is an array of $\boldsymbol{n}$ natural numbers ($\boldsymbol{a}[1], \boldsymbol{a}[2], ..., \boldsymbol{a}[\boldsymbol{n}]$, $1 \le \boldsymbol{a}[\boldsymbol{i}] \le 10^9$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$).

```
1. Algorithm ceFace(n, a):
2.     m ← 0
3.     For i ← 1, n execute
4.         nr ← 1
5.         While a[i] > 9 execute
6.             nr ← nr * 10
7.             a[i] ← a[i] DIV 10
8.         EndWhile
9.         If m < a[i] * nr then
10.            m ← a[i] * nr
11.        EndIf
12.    EndFor
13.    Return m
14. EndAlgorithm
```

Which of the following statements are true?

A. After the call `ceFace(5, [222, 2043, 29, 2, 20035])`, the algorithm returns the value 2000.
B. Regardless of the values of $\boldsymbol{n}$ and $\boldsymbol{a}$, the algorithm `ceFace(n, a)` returns a number that does not belong to array $\boldsymbol{a}$.
C. After the call `ceFace(5, [34, 254, 21, 543, 123])`, the algorithm returns the value 500.
D. If the instruction on line 10 is executed $\boldsymbol{n}$ times, it means that the array $\boldsymbol{a}$, as initially given, was sorted in strictly ascending order.

13. We build an undirected graph as follows: for each natural number n, such that $2 \le n \le 20$, we add a node labeled with that number, and between two nodes labeled with x and y we add an edge if y is a proper divisor of x.

Which of the following statements are true?

A. The built graph has 5 connected components.
B. Only nodes 12, 18 and 20 have the maximum degree.
C. For any 2 nonprime numbers x, y ($2 \le x \le y \le n$), the degree of node x is greater than or equal to the degree of node y.
D. There is a single node with the degree equal to 1.

14. Consider the algorithm `ceva(n, v)`, where v is an array of n ($10 \le n \le 10^5$) integer elements ($v[1], v[2], ..., v[n], -10 \le v[i] \le 10$, for $i = 1, 2, ... n$). The algorithm `zero(k)` returns an array with k elements, all equal to 0.

```
1. Algorithm ceva(n, v):
2.     fr ← zero(21)
3.     s ← 0
4.     For i ← 1, n execute
5.         If fr[v[i] + 11] = 0 then
6.             s ← s + 1
7.         EndIf
8.         fr[v[i] + 11] ← 1
9.     EndFor
10.    Write s
11.    p ← 1
12.    For i ← 1, n execute
13.        If fr[v[i] + 11] = 1 then
14.            p ← p * v[i]
15.            fr[v[i] + 11] ← 0
16.        EndIf
17.    EndFor
18.    Return p
19. EndAlgorithm
```

Which of the following statements are true?

A. If the value displayed on line 10 is greater than 10, the returned result is even.
B. If the value displayed on line 10 is greater than 11, the returned result is a negative number.
C. If the value displayed on line 10 is 21, the returned result is equal to $(10!)^2$.
D. The highest possible value displayed on line 10, for which the returned product does not end with the 0 digit, is 16.

15. Consider the algorithm `f(x)`, where x is a natural number ($0 \le x \le 10^5$).

```
Algorithm f(x):
    If x ≤ 1 then
        Return 1
    EndIf
    Return f(x - 1) + f(x - 2)
EndAlgorithm
```

Which of the following statements are true?

A. After the call `f(10)`, there will be a total of 177 calls of the `f(x)` algorithm, including the initial call.
B. The value returned after the call `f(x)` belongs to the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, …).
C. The value returned after the call `f(10)` is 89.
D. After the call `f(5)` a total of 4 additions will be performed.

16. Consider the algorithm `ceva(A, n, r, c, nr, x)`, where n is a natural number ($1 < n \le 10$), A is a matrix of $n \times n$ natural numbers ($A[1][1], A[1][2], ..., A[n][n]$), r, c, x are natural numbers ($1 \le r, c, x \le 10$), and nr is an integer number ($-10^3 \le nr \le 10^3$). If $n = 4$ and initially $A[3][2] = 5$ and $A[1][4] = 8$, which of the sequences of calls will modify the elements of matrix A such that $A[3][2]$ is equal to 50 and $A[1][4]$ is equal to 16?

```
Algorithm ceva(A, n, r, c, nr, x):
    If (r + x - 1 ≤ n) AND (c + x - 1 ≤ n) then
        For i ← r, r + x - 1 execute
            For j ← c, c + x - 1 execute
                A[i][j] ← A[i][j] * nr
            EndFor
        EndFor
    EndIf
EndAlgorithm
```

A.
```
ceva(A, n, 3, 2, 5, 1)
ceva(A, n, 2, 1, 4, 3)
ceva(A, n, 3, 3, 16, 2)
```

B.
```
ceva(A, n, 1, 3, 2, 2)
ceva(A, n, 3, 2, 2, 3)
ceva(A, n, 2, 1, 10, 3)
```

C.
```
ceva(A, n, 2, 3, 4, 2)
ceva(A, n, 1, 1, 2, 4)
ceva(A, n, 3, 1, 2, 1)
ceva(A, n, 2, 1, 5, 3)
```

D. None of the sequences of calls has the specified result.

17. Consider the algorithm `cauta(x, n, e)`, where x is an array of n ($1 \le n \le 10^4$) natural numbers ($x[1], x[2], ..., x[n]$, for $i = 1, 2, ..., n$, $0 \le x[i] \le 10^4$) and e is a natural number ($1 \le e \le 10^4$).
Which of the following algorithms return *True* if and only if the number e is found in array x?

A.
```
Algorithm cauta(x, n, e):
    g ← False; i ← 1
    While i ≤ n execute
        g ← (x[i] = e)
        i ← i + 1
    EndWhile
    Return g
EndAlgorithm
```

B.
```
Algorithm cauta(x, n, e):
    g ← False; i ← 1
    While NOT g AND i ≤ n execute
        g ← (x[i] = e)
        i ← i + 1
    EndWhile
    Return g
EndAlgorithm
```

C.
```
Algorithm cauta(x, n, e):
    c ← 0
    For i ← 1, n execute
        If x[i] = e then
            c ← c + 1
        Else
            c ← c - 1
        EndIf
    EndFor
    Return n ≠ -c
EndAlgorithm
```

D.
```
Algorithm cauta(x, n, e):
    g ← False; i ← 1
    While i ≤ n execute
        If x[i] < e + 1 AND x[i] MOD e = 0 then
            g ← True
        EndIf
        i ← i + 1
    EndWhile
    Return g
EndAlgorithm
```

18. Consider the natural numbers m and n ($0 \le m, n \le 10$) and the algorithm `Ack(m, n)` which calculates the value of the Ackerman function for the parameters m and n.

```
Algorithm Ack(m, n):
    If m = 0 then
        Return n + 1
    EndIf
    If m > 0 AND n = 0 then
        Return Ack(m - 1, 1)
    EndIf
    If m > 0 AND n > 0 then
        Return Ack(m - 1, Ack(m, n - 1))
    EndIf
EndAlgorithm
```

How many recursive calls will be done after the call `Ack(1, 4)`?

A. There will be 9 recursive calls.
B. The same number of recursive calls as after the call `Ack(1, 2)`
C. 4 recursive calls more than after the call `Ack(1, 2)`
D. There will be 11 recursive calls.

19. Consider the algorithm `P(s, n)`, where s is a string of n characters ($3 < n \le 100$). The algorithm `copy(s, poz, cate)` returns a subarray of array s, consisting of $cate$ characters, starting from position poz, where $0 \le cate \le n$, respectively $1 \le poz \le n - cate + 1$. For strings, the "+" operator represents string concatenation.

```
Algorithm P(s, n):
    i ← n DIV 2
    j ← n DIV i
    If n MOD 2 = 0 then
        s ← copy(s, i, i - 1) +
                  copy(s, j, j - 1)
    Else
        s ← copy(s, i, i - 2) +
                copy(s, j, j - 2) +
                      copy(s, 1, 1)
    EndIf
    Return s
EndAlgorithm
```

Which of the following statements are true?

A. If $s1$ and $s2$ are two strings of length n respectively m, $n \ne m$, after the calls `P(s1, n)` and `P(s2, m)`, the algorithm will return two strings of different lengths.
B. If the string s of length n contains only characters `'a'`, `'b'` and `'c'`, there are 252 distinct values for s, for which, after the call `P(s, n)`, the algorithm will return the string `"ac"`.
C. After the call `P("concurs de admitere", 19)`, the algorithm returns the string `"de admic"`.
D. If the string s of length n contains only characters `'0'` and `'1'`, there are 72 distinct values for s, for which, after the call `P(s, n)`, the algorithm will return the string `"101"`.

20. We color the surfaces covered by two rectangles ***A*** and ***B.*** Each rectangle is defined by its lower left corner and its upper right corner. For rectangle ***A***, the coordinates are (***ax*1**, ***ay*1**) and (***ax*2**, ***ay*2**), and for rectangle ***B***, the coordinates are (***bx*1**, ***by*1**) and (***bx*2**, ***by*2**), $0 \le ax1, ay1, ax2, ay2, bx1, by1, bx2, by2 \le 10^4$, $ax1 < ax2$, $ay1 < ay2$, $bx1 < bx2$, $by1 < by2$. The algorithms `min(a, b)` and `max(a, b)` return the minimum value and the maximum value, respectively between the numbers ***a*** and ***b***.

Which of the following algorithms calculate the area of the colored surface?

A.
```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                                          bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    xSup ← max(0, min(ax2, bx2) - max(ax1, bx1))
    ySup ← max(0, min(ay2, by2) - max(ay1, by1))
    aSup ← xSup * ySup
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

B.
```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                                          bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    If ax2 ≤ bx1 OR bx2 ≤ ax1 OR ay2 ≤ by1 OR
                                  by2 ≤ ay1 then
        aSup ← 0
    Else
        xSup ← min(ax2, bx2) - max(ax1, bx1)
        ySup ← min(ay2, by2) - max(ay1, by1)
        aSup ← xSup * ySup
    EndIf
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

C.
```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                                          bx2, by2):
    arie ← (ax2 - ax1) * (ay2 - ay1)
    If ax2 ≤ bx1 OR bx2 ≤ ax1 OR ay2 ≤ by1
                              OR by2 ≤ ay1 then
        arie ← arie + (bx2 - bx1) * (by2 - by1)
    EndIf
    Return arie
EndAlgorithm
```

D.
```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                                          bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    aSup ← (min(ax2, bx2) - max(ax1, bx1)) +
                 (min(ay2, by2) - max(ay1, by1))
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

21. Consider the algorithm `f(A, B, m, n, k)`, where ***m***, ***n*** and ***k*** are natural numbers ($1 \le m, n, k \le 100$), and ***A*** is an array of ***n*** integers (***A***[1], ***A***[2], ..., ***A***[***n***], where $-100 \le A[i] \le 100$, for ***i*** = 1, 2, ..., ***n***), and ***B*** is an array of ***n*** elements, all equal to zero.

```
Algorithm f(A, B, m, n, k):
    aux ← 0
    For j ← 1, k - 1 execute
        aux ← aux + B[j] * A[j]
    EndFor
    If aux = m then
        Write "Solutie: "
        For j ← 1, k - 1 execute
            If B[j] = 1 then
                Write A[j], " "
            EndIf
        EndFor
        Write new line
    Else
        If k ≠ n + 1 then
            For i ← 0, 1 execute
                B[k] ← i
                f(A, B, m, n, k + 1)
            EndFor
        EndIf
    EndIf
EndAlgorithm
```

If ***A*** = [4, 5, 8, 9, 3, 12, 15, 7], after the call `f(A, B, 12, 8, 1)` the first displayed solution will be `Solutie: 12`.

What will be the second and third displayed solutions?

A.
```
Solutie: 4 8
Solutie: 4 5 3
```

B.
```
Solutie: 4 8
Solutie: 5 7
```

C.
```
Solutie: 9 3
Solutie: 5 7
```

D.
```
Solutie: 9 3
Solutie: 4 8
```

22. Consider the algorithm `oareCe(n1)`, where ***n1*** is a natural number ($0 \leq \boldsymbol{n1} \leq 10^6$).

```
Algorithm oareCe(n1):
    n2 ← 0
    While n1 > n2 execute
        n3 ← n1 MOD 10
        n2 ← n2 * 10 + n3
        n1 ← n1 DIV 10
    EndWhile
    If n1 = n2 then
        Return True // (*)
    EndIf
    Return n1 = (n2 DIV 10)
EndAlgorithm
```

Which of the following statements are true?

A. After the call `oareCe(1210)`, the algorithm will return the value *False*.
B. The call `oareCe(282)` leads to the execution of the line marked with (*)
C. There are 4 distinct values of ***n***, a natural number from the interval [101, 500] for which the call `oareCe(n)` leads to the execution of the line marked with (*).
D. If the algorithm is called with `oareCe(((i * 6) DIV 7) * ((j * 7) DIV (i + j)))`, where ***i*** = 1 and ***j*** = 2, the algorithm returns *True*.

23. Consider the algorithm `interesant(a, b, c, n)`, where ***c*** and ***n*** are natural numbers ($1 \leq n \leq 100$, $1 \leq c \leq 10^4$), and ***a*** and ***b*** are two arrays of length ***n***, with integer elements (***a***[1], ***a***[2], ..., ***a***[***n***] and ***b***[1], ***b***[2], ..., ***b***[***n***], $1 \leq a[i], b[i] \leq 100$, for ***i*** = 1, 2, ..., ***n***). The algorithm `max(x, y)` returns the maximum value between the numbers ***x*** and ***y***.

```
Algorithm interesant(a, b, c, n):
    If n = 0 OR c = 0 then
        Return 0
    EndIf
    If a[n] > c then
        Return interesant(a, b, c, n - 1)
    EndIf
    x ← b[n] + interesant(a, b, c - a[n], n - 1)
    y ← interesant(a, b, c, n - 1)
    Return max(x, y)
EndAlgorithm
```

What value is returned after the call `interesant([2, 3, 1], [6, 10, 3], 5, 3)`?

A. 13
B. 10
C. 16
D. 19

24. Consider the algorithm `divide(x, y)` that computes the quotient of the integer division of number ***x*** to ***y***, where ***x*** and ***y*** are integer numbers ($-10^5 \leq x, y \leq 10^5$, $y \neq 0$). The operation `a << n` shifts the bits of the number ***a*** to the left with ***n*** positions, which is equivalent to multiplying the number with 2^n.

```
Algorithm divide(x, y):
    negativ ← 1
    If x < 0 then
        negativ ← -negativ
        x ← -x
    EndIf
    If y < 0 then
        negativ ← -negativ
        y ← -y
    EndIf
    cat ← 0
    While x ≥ y execute
        temp ← y
        multiplu ← 1
        While x ≥ (temp << 1) execute
            temp ← temp << 1
            ... // (1)
        EndWhile
        x ← x - temp
        ... // (2)
    EndWhile
    ... // (3)
    Return cat
EndAlgorithm
```

What instructions must be added on lines marked with (1), (2) and (3) such that the algorithm `divide(x, y)` returns the correct result?

A.
```
(1): multiplu ← multiplu << 1
(2): cat ← cat + multiplu
(3): If negativ = -1 then
         cat ← -cat
     EndIf
```

B.
```
(1): multiplu ← multiplu << 1
(2): cat ← cat + 1
(3): cat ← -negativ * cat
```

C.
```
(1): multiplu ← multiplu * 2
(2): cat ← cat + 1
(3): cat ← -negativ * cat
```

D.
```
(1): multiplu ← multiplu * 2
(2): cat ← cat + multiplu
(3): cat ← negativ * cat
```

BABEŞ-BOLYAI UNIVERSITY

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 17^{th}, 2025

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

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