BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Mate-Info Contest – April 12th 2025

Written test 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 consists of elements occupying consecutive positions in the array.
- If on the same row there are several consecutive assignment statements, they are separated by "; ".

1. Consider the algorithm `calcul(n, c1, c2)`, where ***n*** is a natural number ($1 \le n \le 10^4$), ***c1*** and ***c2*** are digits ($0 \le c1, c2 \le 9$).

```
Algorithm calcul(n, c1, c2):
    If n = 0 then
        Return 0
    EndIf
    If n MOD 10 = c1 then
        Return calcul(n DIV 10, c1, c2) * 10 + c2
    Else
        Return calcul(n DIV 10, c1, c2) * 10 + n MOD 10
    EndIf
EndAlgorithm
```

What will the algorithm return for ***n*** = 1999, ***c1*** = 1 and ***c2*** = 0?

A. 1000
B. 999
C. 1099
D. 1990

2. Consider the algorithm `ceFace(m, n)`, where ***m*** and ***n*** are natural numbers ($1 \le m, n \le 100$):

```
 1. Algorithm ceFace(m, n):
 2.    c ← 1; i ← n
 3.    While i > 0 execute
 4.        If i MOD 2 = 1 then
 5.            c ← c * m
 6.            i ← i - 1
 7.        Else
 8.            m ← m * m
 9.            i ← i DIV 2
10.        EndIf
11.    EndWhile
12.    Return c
13. EndAlgorithm
```

Which of the following statements are true?

A. After the call `ceFace(2, 5)` the algorithm returns 30.
B. If after the call `ceFace(m, n)` the algorithm returns the value ***x***, there is no other pair of numbers ***m1***, ***n1*** (***m1*** ≠ ***m*** and ***n1*** ≠ ***n***) for which the call `ceFace(m1, n1)` will return the same value ***x***.
C. The only value of ***n*** for which line 6 is executed 2 times after the call `ceFace(m, n)` is 5.
D. After the call `ceFace(5, 8)` line 6 is executed exactly once.

3. Consider the algorithm `ceFace(b, n, a)`, where ***b*** and ***n*** are natural numbers ($2 \le b, n \le 100$), and ***a*** is an array of ***n*** natural number elements (***a***[1], ***a***[2], ..., ***a***[***n***], $0 \le a[i] < b$, for ***i*** = 2, 3, ..., ***n*** and $0 < a[1] < b$):

```
Algorithm ceFace(b, n, a):
    v ← a[1]
    For i ← 2, n execute
        v ← v * b + a[i]
    EndFor
    Return v
EndAlgorithm
```

Which of the following statements are true?

A. The call `ceFace(2,6,[1,0,1,0,1,1])` returns the value 43.
B. The call `ceFace(9,3,[7,6,5])` returns the value 626.
C. If ***a***[***n***] = 0, the call `ceFace(b, n, a)` returns an even number.
D. If ***b1*** > ***b2***, then the call `ceFace(b1, n, a)` returns a greater number than the call `ceFace(b2, n, a)`.

4. Consider the integer number ***n***, ($-100 \le n \le 100$).

Which of the following expressions are *True* if and only if ***n*** does **NOT** belong to the set: $\{-8\} \cup \{-4, -3, ..., 8\}$?

A. `(n ≤ -8) AND (n ≥ -8) AND (n ≤ -4) AND (n ≥ 8)`
B. `(n < -8) OR ((n > -8) AND (n < -4)) OR (n > 8)`
C. `(n < -8) OR ((n > -8) OR (n < -4)) AND (n > 8)`
D. `((n < -4) AND (n ≠ -8)) OR (n > 8)`

5. Consider the algorithm `cautBin(st, dr, y, x, n)`, where ***st*** and ***dr*** are natural numbers, ***x*** is an array sorted in ascending order with ***n*** integer elements ($1 \le st, dr, n \le 10^4$, ***x***[1], ***x***[2], ..., ***x***[***n***], $-10^3 \le x[i] \le 10^3$ for ***i*** = 1, 2, ..., ***n***) and ***y*** is an integer number ($-10^3 \le y \le 10^3$, $x[1] < y$) that is not part of the array. The algorithm is called as follows: `cautBin(1, n, y, x, n)`.

```
Algorithm cautBin(st, dr, y, x, n):
    If st < dr then
        mij ← (st + dr) DIV 2
        If y < x[mij] then
            Return cautBin(st, mij, y, x, n)
        Else
            Return cautBin(mij + 1, dr, y, x, n)
        EndIf
    Else
        .........
    EndIf
EndAlgorithm
```

What statements should be added on the dotted line such that the algorithm returns the position of the closest element in the array that is greater than ***y***. If such a number does not exist, the algorithm returns -1.

A.
```
If y > x[dr] then
    Return dr + 1
Else
    Return -1
EndIf
```

B.
```
If y < x[dr] then
    Return dr
Else
    Return -1
EndIf
```

C.
```
If y > x[st] then
    Return st + 1
Else
    Return -1
EndIf
```

D.
```
If y < x[st] then
    Return st
Else
    Return -1
EndIf
```

6. Consider the algorithm `calculeaza(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 calculeaza(x, n):
    If n MOD 2 = 1 then
        s ← x[n]
    Else
        s ← 0
    EndIf
    For i ← 1, n - 2, 2 execute
        s ← s + x[i] + x[i + 1]
    EndFor
    Return s
EndAlgorithm
```

Which of the following statements are true?

A. The call `calculeaza([3, -8, -2, 15, -1, 0, 3, 1, 3], 9)` returns 11.
B. The call `calculeaza([2, -1, 7, 5, -9, 0, 3, 1, 12], 9)` returns 4.
C. The call `calculeaza([10, 2, 5, 78, 23, 4, 11], 7)` returns 133.
D. The call `calculeaza([-3, 8, -2, 15, -1, 10], 6)` returns 27.

7. Consider the algorithm `f(n, a, p)`, where ***n*** and ***p*** are natural numbers ($1 \le n, p \le 10^5$) and ***a*** is an array containing ***n*** digits (***a***[1], ***a***[2], ..., ***a***[***n***], $0 \le a[i] \le 9$, for ***i*** = 1, 2, ..., ***n***), where at least one digit is different from 0:

```
Algorithm f(n, a, p):
    s ← 0
    For i ← 1, n execute
       s ← s + a[i]
    EndFor
    For i ← 1, p execute
       If s MOD 3 = 0 then
          s ← s DIV 3
       Else
          Return False
       EndIf
    EndFor
    Return True
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns *True* if and only if the sum of the elements of the array ***a*** is a multiple of 3^p.
B. The algorithm returns *True* if and only if the sum of the elements of the array ***a*** is a power of 3.
C. The algorithm returns *False* if and only if the sum of the elements of the array ***a*** is not divisible by 3.
D. The algorithm returns *True* for the call `f(6, [9, 1, 8, 8, 4, 6], 2)`.

8. The maximum number of edges in an undirected graph with ***n*** nodes and ***p*** ($0 < p \le n$) connected components is:

A. $\frac{(n-p)\times(n-p+1)}{2}$ B. $(n-p)\times(n-p+1)$ C. $\frac{(n-p)\times(n-p+1)}{4}$ D. $\frac{(n-p)\times(n+p+1)}{2}$

9. Consider a natural number $\boldsymbol{n}$ ($10 \le \boldsymbol{n} \le 10^4$).

Which of the following implementations of the algorithm `f(n)` returns the reverse of the number $\boldsymbol{n}$?

A.

```
Algorithm f(n):
    If n > 0 then
        Return n MOD 10 + 10 * f(n DIV 10)
    EndIf
    Return 0
EndAlgorithm
```

B.

```
Algorithm f1(n, ogl):
    If n > 0 then
        Return f1(n DIV 10, n MOD 10 + 10 * ogl)
    EndIf
    Return ogl
EndAlgorithm

Algorithm f(n):
    Return f1(n, 0)
EndAlgorithm
```

C.

```
Algorithm f(n):
    ogl ← 0
    While n > 0 execute
        ogl ← (n MOD 10) * 10 + ogl
        n ← n DIV 10
    EndWhile
    Return ogl
EndAlgorithm
```

D.

```
Algorithm f(n):
    ogl ← 0
    While n > 0 execute
        ogl ← ogl * 10 + n MOD 10
        n ← n DIV 10
    EndWhile
    Return ogl
EndAlgorithm
```

10. Consider the algorithm `ceFace(x1, y1, x2, y2, x3, y3)`, where $(\boldsymbol{x1}, \boldsymbol{y1})$, $(\boldsymbol{x2}, \boldsymbol{y2})$ and $(\boldsymbol{x3}, \boldsymbol{y3})$ are the coordinates of three distinct geometric points.

```
Algorithm ceFace(x1, y1, x2, y2, x3, y3):
    t ← x1 * (y2 - y3)
    v ← x2 * (y1 - y3)
    z ← x3 * (y1 - y2)
    Return (t - v + z) ≠ 0
EndAlgorithm
```

Which of the following statements are true for the call `ceFace(x1, y1, x2, y2, x3, y3)`?

A. Returns *True* if the given points form a non-degenerate triangle.
B. Returns *False* if the given points are collinear.
C. Returns *False* if the given points form a non-degenerate triangle.
D. Returns *True* if the given points are collinear.

11. Consider the algorithm `h(A, n)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 10^3$), and $\boldsymbol{A}$ is an array of $\boldsymbol{n}$ integer elements ($\boldsymbol{A}[1], \boldsymbol{A}[2], ..., \boldsymbol{A}[\boldsymbol{n}]$, where $0 \le \boldsymbol{A}[\boldsymbol{i}] \le 100$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$):

```
Algorithm h(A, n):
    If n = 0 then
        Return 0
    EndIf
    Return h(A, n - 1) + (A[n] MOD 2) * (A[n] MOD 10) * (n MOD 2)
EndAlgorithm
```

For which calls will the algorithm return the value 0?

A. `h([25, 14, 35, 26, 2, 10], 6)`
B. `h([14, 25, 26, 2, 10, 35], 6)`
C. `h([12, 5, 22, 4, 32, 8, 46, 9, 54, 3], 10)`
D. `h([3, 4, 7], 3)`

12. Consider the algorithm `ceFace(n)`, where $\boldsymbol{n}$ is a natural number ($0 \le \boldsymbol{n} \le 10$).

```
1. Algorithm ceFace(n):
2.     e ← 1
3.     For f ← 1, n execute
4.         s ← 0
5.         For j ← 1, f execute
6.             s ← s + j
7.         EndFor
8.         e ← e * s
9.     EndFor
10.    Return e
11. EndAlgorithm
```

Which of the following statements are true?

A. After the call `ceFace(5)`, the algorithm returns the value 2700.
B. Regardless of the value of $\boldsymbol{n}$, the algorithm `ceFace(n)` will never return the value 0.
C. The value returned after the call `ceFace(9)` has the same number of zeroes at the end as the value returned after the call `ceFace(10)`.
D. After the call `ceFace(10)`, line 6 is executed 45 times.

13. Consider the algorithm `ceva(n)` where ***n*** is a natural number ($1 \le n \le 10^9$).

```
Algorithm aux(n):
    v1 ← 1
    v2 ← 1
    While v1 < n execute
        v3 ← v1 + v2
        v1 ← v2
        v2 ← v3
    EndWhile
    Return v1 = n
EndAlgorithm
```

```
Algorithm ceva(n):
    If aux(n) then
        Return True
    EndIf
    p ← 10
    gata ← False
    While (n DIV p ≠ 0) AND (NOT gata) execute
        nr1 ← n MOD p
        nr2 ← (n – nr1) DIV p
        If aux(nr1) then
            gata ← ceva(nr2)
        EndIf
        p ← p * 10
    EndWhile
    Return gata
EndAlgorithm
```

Considering that the first 6 numbers of the *Fibonacci* sequence are 1, 1, 2, 3, 5, 8, which of the following statements are true?

A. The algorithm `ceva(n)` returns *True* if and only if ***n*** is a *Fibonacci* number.
B. The algorithm `ceva(n)` checks whether ***n*** can be written as a sum of *Fibonacci* numbers.
C. The algorithm `ceva(n)` checks whether ***n*** can be written as a product of *Fibonacci* numbers.
D. If ***n*** = 1234589, then the algorithm `ceva(n)` returns *True*.

14. Consider the algorithm `ceFace(n, f, p)`, where ***n*** is a natural number ($0 \le n \le 10^{10}$), ***p*** is a natural number ($0 \le p \le 100$) and ***f*** is an integer number ($-1 \le f \le 1$).

```
Algorithm ceFace(n, f, p):
    If n = 0 then
        Return f = 1
    EndIf
    c ← n MOD 10
    n ← n DIV 10
    If f = -1 then
        If c < p then
            Return ceFace(n, 0, c)
        Else
            Return False
        EndIf
    EndIf
    If f = 0 then
        If c < p then
            Return ceFace(n, 0, c)
        Else
            If c > p then
                Return ceFace(n, 1, c)
            Else
                Return False
            EndIf
        EndIf
    EndIf
    If f = 1 then
        If c > p then
            Return ceFace(n, 1, c)
        Else
            Return False
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements about the result of the call `ceFace(n DIV 10, -1, n MOD 10)` are true?

A. For any value of ***n*** < 101 it will be *False*.
B. For ***n*** = 8976532014 it will be *True*.
C. If ***n*** contains at least two equal digits, it will be *False*.
D. If ***n*** does not contain the digit 0 and the call returns *True*, then the call will return *True* for the reversed value of ***n*** as well.

15. To determine a digit that appears most frequently in a number, we implement three algorithms: `cifreA(n)`, `cifreB(n)` and `cifreC(n)`, where ***n*** is a natural number ($1 \le n \le 10^{12}$).

```
Algorithm cifreA(n):
    c ← n
    maxf ← -1; maxd ← -1
    While c > 0 execute    // (*)
        d ← c MOD 10
        copie ← n; cnt ← 0
        While copie > 0 execute
            If copie MOD 10 = d then
                cnt ← cnt + 1
            EndIf
            copie ← copie DIV 10
        EndWhile
        If cnt > maxf then
            maxf ← cnt
            maxd ← d
        EndIf
        c ← c DIV 10
    EndWhile
    Return maxd
EndAlgorithm
```

```
Algorithm cifreB(n):
    maxf ← -1
    maxd ← -1
    For i ← 0, 9 execute    // (*)
        c ← n
        cnt ← 0
        While c > 0 execute
            If c MOD 10 = i then
                cnt ← cnt + 1
            EndIf
            c ← c DIV 10
        EndWhile
        If cnt > maxf then
            maxf ← cnt
            maxd ← i
        EndIf
    EndFor
    Return maxd
EndAlgorithm
```

```
Algorithm cifreC(n):
    maxf ← -1; maxd ← -1
    For i ← 9, 0, -1 execute
        c ← n; cnt ← 0
        While c > 0 execute
            If c MOD 10 = i then
                cnt ← cnt + 1
            EndIf
            c ← c DIV 10
        EndWhile
        If cnt > maxf then
            maxf ← cnt
            maxd ← i
        EndIf
    EndFor
    Return maxd
EndAlgorithm
```

Which of the following statements are true?

A. `cifreA(123453) = cifreB(123453) = cifreC(123453)`
B. `cifreA(123456) = cifreB(123456) = cifreC(123456)`
C. There is at least one number ***n*** for which the three algorithms return three different values.
D. For any number ***n***, the **While** loop marked with (*) in the algorithm `cifreA(n)` is executed fewer times than the **For** loop marked with (*) in the algorithm `cifreB(n)`.

16. Consider the algorithm `getSomeMax(n, x)`, where ***n*** is a natural number ($1 \le n \le 10^3$), and ***x*** is an array with ***n*** integer elements (***x***[1], ***x***[2], ..., ***x***[***n***], where $-10^3 \le x[i] \le 10^3$, for ***i*** = 1, 2, ..., ***n***). The algorithm `zero(k)` returns an array with ***k*** elements, all equal to zero.

```
Algorithm getSomeMax(n, x):
    y ← zero(n + 1)
    For i ← 1, n execute
        y[i + 1] ← y[i] + x[i]
    EndFor
    sm ← y[2]
    For i ← 2, n execute
        For j ← i, n execute
            s ← y[j] - y[i - 1]
            If s > sm then
                sm ← s
            EndIf
        EndFor
    EndFor
    Return sm
EndAlgorithm
```

Which of the following statements are true?

A. If ***n*** = 1, the value returned by the algorithm `getSomeMax(n, x)` is the value of ***x***[1].
B. The value returned by the algorithm in the call `getSomeMax(8, [5, 7, -4, 6, -3, -2, 6, -7])` is 10.
C. If ***n*** = 100 and ***x*** = [1, 2, 3, … , 99, 100], the value returned by the `getSomeMax(n, x)` algorithm is 4950.
D. If all values in the array ***x*** are strictly negative, the `getSomeMax(n, x)` algorithm returns the largest element in the array.

17. Consider the algorithm `afla(n, x)`, where $\boldsymbol{n}$ is a natural number ($3 \le \boldsymbol{n} \le 10^4$), and $\boldsymbol{x}$ is an array with $\boldsymbol{n}$ integer elements ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$, $-100 \le \boldsymbol{x}[\boldsymbol{i}] \le 100$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$):

```
1. Algorithm afla(n, x):
2.     M1 ← x[1]; M2 ← x[2]; M3 ← x[3]
3.     For i ← 1, n execute
4.         If x[i] > M1 then
5.             M3 ← M2
6.             M2 ← M1
7.             M1 ← x[i]
8.         Else
9.             If x[i] > M2 then
10.                M3 ← M2
11.                M2 ← x[i]
12.            Else
13.                If x[i] > M3 then
14.                    M3 ← x[i]
15.                EndIf
16.            EndIf
17.        EndIf
18.    EndFor
19.    Return M1, M2, M3
20.EndAlgorithm
```

Which of the following statements are true?

A. After the call `afla(6, [1, 2, 3, 4, 5, 6])` the algorithm returns 6, 5, 4.

B. If the statements on lines 8 and 12 were replaced with **`EndIf`**, and the instructions on lines 16 and 17 were deleted, the algorithm would return the same result as the initial algorithm.

C. If at the beginning ***M1***, ***M2*** and ***M3*** would take the values $\boldsymbol{x}[3]$, $\boldsymbol{x}[2]$ and $\boldsymbol{x}[1]$ respectively, the algorithm would return the same result as the initial algorithm.

D. If on line 3 instead of `For i ← 1, n execute` we would have `For i ← 4, n execute`, the algorithm would return the same result as the initial algorithm.

18. Consider the algorithm `ceFace(A, n)`, where $\boldsymbol{n}$ is a natural number ($1 \le \boldsymbol{n} \le 20$), and $\boldsymbol{A}$ is a square matrix with $\boldsymbol{n}$ rows and $\boldsymbol{n}$ columns, which contains natural numbers: ($\boldsymbol{A}[1][1], \boldsymbol{A}[1][2], ..., \boldsymbol{A}[\boldsymbol{n}][\boldsymbol{n}]$, where $0 \le \boldsymbol{A}[\boldsymbol{i}][\boldsymbol{j}] \le 200$, for $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$ and $\boldsymbol{j} = 1, 2, ..., \boldsymbol{n}$).

```
Algorithm ceFace(A, n):
    // Început partea 1
    For i ← 1, n execute
        For j ← i + 1, n execute
            temp ← A[i][j]
            A[i][j] ← A[j][i]
            A[j][i] ← temp
        EndFor
    EndFor
    // Sfârșit partea 1

    // Început partea 2
    For i ← 1, n execute
        For j ← 1, n DIV 2 execute
            temp ← A[i][j]
            A[i][j] ← A[i][n - j + 1]
            A[i][n - j + 1] ← temp
        EndFor
    EndFor
    // Sfârșit partea 2
EndAlgorithm
```

Which of the following statements are true?

A. Upon executing the algorithm `ceFace(A, 3)`, the matrix $\boldsymbol{A} = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6, \\ 7 & 8 & 9 \end{matrix}$ will become $\begin{matrix} 7 & 4 & 1 \\ 8 & 5 & 2. \\ 9 & 6 & 3 \end{matrix}$

B. If the input matrix $\boldsymbol{A}$ is the identity matrix of order 3, then it does not change as a result of the execution of the algorithm `ceFace(A, 3)`

C. The algorithm `ceFace(A, n)` applies a 90° rotation to the right on the given matrix, modifying it accordingly.

D. If we swap the part of the algorithm between `Început partea 1` and `Sfârșit partea 1` with the one between `Început partea 2` and `Sfârșit partea 2`, the algorithm `ceFace(A, n)` would return the same result as the initial algorithm.

19. Consider the algorithm `ceFace(n)`, where $\boldsymbol{n}$ is a natural number ($0 \le \boldsymbol{n} \le 200$).

```
1. Algorithm ceFace(n):
2.    e ← 0
3.    For i ← 1, n execute
4.        If i MOD 2 = 0 then
5.            e ← e - 2 * i * i
6.        Else
7.            e ← e + 2 * i * i
8.        EndIf
9.    EndFor
10.   Return e
11. EndAlgorithm
```

Which of the following statements are true?

A. For any even number $\boldsymbol{n}$, the algorithm will return a negative value.

B. The algorithm computes the value of the expression $0 + 1 * 2 - 2 * 4 + 3 * 6 - 4 * 8 + \ldots + (-1)^{\boldsymbol{n}-1} * \boldsymbol{n} * 2 * \boldsymbol{n}$

C. If the algorithm `ceFace(n)` returns a negative value, then $\boldsymbol{n}$ is an even number.

D. There exists a single value of $\boldsymbol{n}$, for which the statement on line 7 is executed exactly 7 times.

20. Consider the algorithm `ceFace(n, x)`, where ***n*** is a natural number ($2 \leq n \leq 10^3$), 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***). The algorithm `zero(k)` returns an array with ***k*** elements, all equal to zero. The algorithms `minim(n, x)`, `maxim(n, x)` return the minimum and maximum value of the array ***x*** with ***n*** elements.

```
01. Algorithm ceFace(n, x):
02.     min ← minim(n, x)
03.     max ← maxim(n, x)
04.     r ← max – min + 1
05.     y ← zero(r)
06.     For i ← 1, n execute
07.         y[x[i] – min + 1]  ← y[x[i] – min + 1] + 1
08.     EndFor
09.     idx ← 1
10.     For i ← 1, r execute
11.         While y[i] > 0 execute
12.             x[idx] ← i + min – 1
13.             idx ← idx + 1
14.             y[i] ← y[i] - 1
15.         EndWhile
16.     EndFor
17. EndAlgorithm
```

Which of the following statements are true?

A. If array ***x*** contains negative numbers as well, the algorithm will try to access non-existent positions in array ***y***.

B. If we replaced the instructions on lines 9 and 10 with the instruction sequence below, the algorithm `ceFace(n, x)` would return the same result as the initial algorithm.

```
x[1] ← min
idx ← 2
y[1] ← y[1] – 1
For i ← 2, r execute
```

C. After the call `ceFace(2, [5, 8])`, array ***x*** becomes: ***x*** = `[6, 9]`.

D. After the execution of the algorithm `ceFace(n, x)` the elements of array ***x*** will represent a permutation of the array's initial elements.

21. Consider the algorithm `p(x, n, a, b, c, d)`, where ***x*** is an array with ***n*** ($0 \leq n \leq 100$) integer elements (***x***[1], ***x***[2], ..., ***x***[***n***], where $-100 \leq x[i] \leq 100$, for ***i*** = 1, 2, ..., ***n***), and ***a***, ***b***, ***c***, and ***d***, are integers ($0 \leq a, b, c, d \leq 100$).

```
Algorithm p(x, n, a, b, c, d):
    If n = 0 then
        Return a = b AND c = d
    EndIf
    p1 ← p(x, n - 1, a + x[n], b, c * x[n], d)
    p2 ← p(x, n - 1, a,  b + x[n], c, d * x[n])
    Return p1 OR p2
EndAlgorithm
```

Knowing that ***x*** = [2, 9, 5, 6, 8, 4, 1, 2, 5, 3, 4, 1, 9, 6, 8, 3], which of the following statements are true?

A. After the call `p(x, 16, 0, 0, 1, 1)`, the algorithm returns *True.*

B. After the call `p(x, 16, 0, 0, 1, 1)`, the algorithm returns *False.*

C. Corresponding to the call `p(x, 16, 0, 0, 1, 1)`, the algorithm enters an infinite loop.

D. After the call `p(x, 16, 0, 0, 1, 1)`, the algorithm returns the same result for any permutation of the array ***x***.

22. Consider the algorithms `rec(n, x, i, j)` and `ceFace(n, x)`, where ***n*** is a natural number ($1 \leq n \leq 10^3$), and ***x*** is an array with ***n*** integer elements (***x***[1], ***x***[2], ..., ***x***[***n***], $-100 \leq x[k] \leq 100$, for ***k*** = 1, 2, ..., ***n***), and ***i*** and ***j*** are integers in the range [0, ***n***]. The algorithm `maxim(a, b)` returns the greater value of ***a*** and ***b***.

```
Algorithm rec(n, x, i, j):
    If i = n then
        Return 0
    EndIf
    a ← rec(n, x, i + 1, j)
    b ← 0
    If j = 0 then
        b ← 1 + rec(n, x, i + 1, i)
    Else
        If x[i] > x[j] then
            b ← 1 + rec(n, x, i + 1, i)
        EndIf
    EndIf
    Return maxim(a, b)
EndAlgorithm
```

```
Algorithm ceFace(n, x):
    Return rec(n, x, 1, 0)
EndAlgorithm
```

Which of the following statements are true?

A. For an array ***x*** ordered in strictly ascending order, the algorithm `ceFace(n, x)` will return the value ***n***.

B. The time complexity of the algorithm in the worst case is $O(n^2)$.

C. After the call `ceFace(8, [10, 15, 9, 30, 21, 50, 42, 60])` the algorithm returns the value 5.

D. After the call `ceFace(2, [3, 2])` the algorithm returns the value 1.

23. A number ***n*** is called *special* if its prime divisors are only the numbers 2, 3 and 5. For example, special numbers are 1 ($1 = 2^0 * 3^0 * 5^0$), 12 ($12 = 2^2 * 3$) or 30 ($30 = 2 * 3 * 5$). The algorithm `zero(k)` returns an array with ***k*** elements equal to 0.

Which of the instruction sequences from the answers A, B, C, D should be inserted into the algorithm `special(n)` in place of the dotted line, so that the algorithm returns the ***n***th special number, where ***n*** is a natural number ($1 \le n \le 10^5$)?

```
Algorithm special(n):
    v ← zero(n)
    v[1] ← 1; c2 ← 1; c3 ← 1; c5 ← 1
    nr ← 1
    While nr < n execute
        val1 ← v[c2] * 2
        val2 ← v[c3] * 3
        val3 ← v[c5] * 5
        If val1 ≤ val2 AND val1 ≤ val3 then
            elem ← val1
            c2 ← c2 + 1
        Else
            If val2 ≤ val1 AND val2 ≤ val3 then
                elem ← val2; c3 ← c3 + 1
            Else
                elem ← val3
                c5 ← c5 + 1
            EndIf
        EndIf
        ..........
    EndWhile
    Return v[n]
EndAlgorithm
```

A.
```
v[nr] ← elem
nr ← nr + 1
```

B.
```
If v[nr] < elem then
    v[nr + 1] ← elem
    nr ← nr + 1
EndIf
```

C.
```
nr ← nr + 1
v[nr] ← elem
```

D.
```
tmp ← nr
While elem < v[tmp] AND tmp ≥ 1 execute
    v[tmp + 1] ← v[tmp]
    tmp ← tmp - 1
EndWhile
v[tmp + 1] ← elem
nr ← nr + 1
```

24. A natural number ***n*** is given ($0 \le n \le 2^{31}$) and we want to determine the number of bits having the value $k \in \{0, 1\}$ of the base-2 representation of the number ***n*** which is represented using exactly 32 bits. The algorithms use bitwise operations: `&` (**AND**), `<<` (left shift) and `>>` (right shift) having the following meanings:

- If ***x*** and ***y*** are two natural numbers, then ***x*** & ***y*** applies the bitwise **AND** operation on their binary representation: each bit in the result is 1 only if both the corresponding bits of ***x*** and ***y*** are 1; otherwise, it is 0.
- If ***x*** is a natural number, the operation ***x*** << ***i*** is equivalent to multiplying ***x*** by 2, ***i*** times; and the operation ***x*** >> ***i*** is equivalent to dividing ***x*** by 2, ***i*** times.

Which of the algorithm variants below return the required value?

A.
```
Algorithm countBits_A(n, k):
    count ← 0
    For i ← 0, 31 execute
        If ((n & (1 << i)) >> i) = k then
            count ← count + 1
        EndIf
    EndFor
    Return count
EndAlgorithm
```

B.
```
Algorithm countBits_B(n, k):
    count ← 0
    While n > 0 execute
        If (n & 1) = 1 then
            count ← count + 1
        EndIf
        n ← n >> 1
    EndWhile
    If k = 0 then
        count ← 32 – count
    EndIf
    Return count
EndAlgorithm
```

C.
```
Algorithm countBits_C(n, k):
    If n = 0 then
        If k = 0 then
            Return 32
        Else
            Return 0
        EndIf
    Else
        If (n & 1) = k then
            Return 1 + countBits_C(n >> 1, k)
        Else
            Return countBits_C(n >> 1, k)
        EndIf
    EndIf
EndAlgorithm
```

D.
```
Algorithm countBits(n, k, poz):
    If poz < 0 then
        Return 0
    Else
        If ((n & (1 << poz)) >> poz) = k then
            Return 1 + countBits(n, k, poz - 1)
        Else
            Return countBits(n, k, poz - 1)
        EndIf
    EndIf
EndAlgorithm

Algorithm countBits_D(n, k):
    Return countBits(n, k, 31)
EndAlgorithm
```

BABEŞ-BOLYAI UNIVERSITY

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Mate-Info Contest – April 12th, 2025

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

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