BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission exam – September 6th, 2024
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 consists of elements occupying consecutive positions in the array.
- A subsequence of an array/string consists of elements not necessarily occupying consecutive positions in the array/string, but in the order in which they appear in the given array/string.
- If on the same row there are several consecutive assignment statements, they are separated by "; ".

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

```
Algorithm decide(n, x):
    b ← True
    i ← 1
    While b AND (i < n) execute
        b ← (x[i] < x[i + 1])
        i ← i + 1
    EndWhile
    Return b
EndAlgorithm
```

For which of the following situations will the algorithm return *True*?

A. If array x consists of the values 1, 2, 3, …, 10
B. If array x is strictly ascending
C. If array x has no negative elements
D. If array x has all its negative elements before the positive elements

2. Consider the algorithm `afiseaza(n, a)`, where n is a natural number ($1 \leq n \leq 10^3$), and a is an array of n integer elements ($a[1], a[2], ..., a[n]$, where $-100 \leq a[i] \leq 100$, for $i = 1, 2, ..., n$):

```
Algorithm afiseaza(n, a):
    i ← 1; j ← n
    While i ≤ j execute
        If a[i] < a[j] then
            Write a[i], " "
            i ← i + 1
        Else
            Write a[j], " "
            j ← j - 1
        EndIf
    EndWhile
EndAlgorithm
```

Which of the following statements are true?

A. If the given array is sorted in ascending order, the values from the array will be displayed in descending order.
B. If the given array is sorted in descending order, the last element displayed is the maximum element.
C. If $n = 10$ and $a = [0, 2, 4, 6, 8, 10, 8, 6, 4, 2]$, the values from the array will be displayed in ascending order.
D. If the maximum element is on the first position, the values from the array will be displayed in reverse order.

3. What is the relationship between numbers $X = 6543_{(8)}$ in base 8 and $Y = CEF_{(16)}$ in base 16?

A. $X > Y$ B. $X \leq Y$ C. $X \geq Y$ D. $X = Y$

4. Consider the algorithm `f(n)`, where n is a non-zero natural number ($1 \leq n \leq 15$).

```
1.  Algorithm f(n):
2.      x ← 10; y ← 13
3.      While n ≠ 0 execute
4.          z ← (x + y) MOD 2
5.          n ← n DIV 2
6.          If z MOD 2 = 0 then
7.              x ← (x * 3 + y * 4) MOD z
8.              y ← (y + x) * z
9.          Else
10.             x ← x + 1
11.             y ← y - 1
12.         EndIf
13.     EndWhile
14.     Return z
15. EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns the same value for any natural number $1 \leq n \leq 15$.
B. The algorithm returns distinct values for natural numbers n with the property $1 \leq n \leq 10$.
C. If $n = 11$, the algorithm returns 0.
D. If we replace the instruction on line 10 with `x ← x - 1`, and the one on line 11 with `y ← y + 1` the algorithm returns the same value as in the original version for any natural number $1 \leq n \leq 15$.

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

```
Algorithm numere(n, x):
    i ← 1; nr ← n
    While i ≤ n execute
        If (x[i] MOD 10) MOD 2 = 0 then
            nr ← nr + 1
        Else
            nr ← nr – 1
        EndIf
        i ← i + 1
    EndWhile
    Return nr = n
EndAlgorithm
```

Which of the following statements are true?

A. The call `numere(3, [1, 2, 3])` returns *True.*
B. The call `numere(3, [1, -2, 3])` returns *False.*
C. The call `numere(4, [1, 2, 3, -4])` returns *False.*
D. The call `numere(4, [1, 2, 3, 4])` returns *True.*

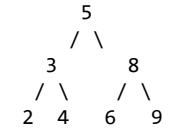
6. Consider the algorithm `ceFace(v, n)`, where ***v*** is an array of ***n*** ($1 \leq n \leq 10^4$) natural numbers (***v***[1], ***v***[2], ..., ***v***[***n***], where $1 \leq v[i] \leq 10^4$ for ***i*** = 1, 2, ..., ***n***).

```
Algorithm ceFace(v, n):
    a ← 0; b ← 1
    For i ← n, 2, -1 execute
        If v[i] = v[i - 1] + 1 then
            b ← b + 1
        Else
            b ← 1
        EndIf
        If b > a then
            a ← b
        EndIf
    EndFor
    Return a
EndAlgorithm
```

What will the `ceFace(v, n)` algorithm return?

A. The length of the longest subarray consisting of consecutive numbers in ascending order in array ***v***.
B. The length of the longest subarray consisting of consecutive numbers in descending order in array ***v***.
C. The number of ascending subarrays in array ***v***.
D. The length of the longest subsequence consisting of consecutive numbers in ascending order in array ***v***.

7. Consider the following binary tree:



Which of the following node sequences correspond to the postorder traversal of the tree?

A. 2, 3, 4, 5, 6, 8, 9
B. 4, 3, 2, 9, 8, 6, 5
C. 2, 4, 3, 6, 9, 8, 5
D. 9, 6, 8, 5, 3, 2, 4

8. Consider the algorithm `prelucrare(n, m, x)`, where ***n*** and ***m*** are natural numbers ($1 \leq n \leq 100$, $1 < m \leq 100$), and ***x*** is a matrix with ***n*** * ***m*** natural number elements (***x***[1][1], ***x***[1][2], ..., ***x***[***n***][***m***], where initially ***x***[***i***][***j***] = 0, for ***i*** = 1, 2, ..., ***n***; ***j*** = 1, 2, ..., ***m***):

```
Algorithm prelucrare(n, m, x):
    k ← 1; i ← k
    While i ≤ n execute
        j ← k + 1
        While j ≤ m execute
            If k MOD 2 = 0 then
                x[i][j] ← k * k
            EndIf
            Write x[i][j], " "
            k ← k + 1; j ← j + 1
        EndWhile
        i ← i + 1
    EndWhile
EndAlgorithm
```

What does this algorithm display?

A. A sequence of ***n*** values.
B. If ***m*** is even, a sequence of values in which the value 0 alternates with values that represent even square numbers, and the first and last values are 0.
C. A sequence of ***m*** – 1 values.
D. A sequence of values in which the value 0 alternates with values that represent odd square numbers.

9. Consider two bit arrays, ***x*** with ***n*** elements and ***y*** with ***m*** elements, where ***n*** and ***m*** are non-zero natural numbers ($0 < n, m \leq 64$). The elements of the arrays are either 0 or 1. Let ***b1*** and ***b2*** be two bits for which we define the operation $op(b1, b2) = \begin{cases} 0 & if\ \ b1 = b2 \\ 1 & otherwise \end{cases}$. We define the *os* operation as an application of the *op* operation on the elements of ***x*** and ***y***, but starting from the last elements of the arrays (so firstly, we apply *op* on ***x***[***n***] and ***y***[***m***]). If the two arrays have a different number of elements, the elements from the beginning of the longer array, that have no pair in the other array, remain unaffected. For example, for arrays [1, 1, 1, 0, 1, 0] and [1, 1, 1, 0], the result of the *os* operation will be [1, 1, 0, 1, 0, 0]. The algorithm creates an array ***r*** with maximum(***n***, ***m***) elements.

The purpose of the `OperatieSpeciala(x, n, y, m)` algorithm is to implement the *os* operation described above, and to return the resulting array and its length. The algorithm `Zero(k)` returns an array of ***k*** elements, all equal to zero.

```
1.Algorithm OperatieSpeciala(x, n, y, m):
2.    length ← n
3.    lenF ← m
4.    r ← Zero(m)
5.    If m < n then
6.        length ← m
7.        lenF ← n
8.        r ← Zero(n)
9.    EndIf
10.   For i ← 1, length execute
11.       If (x[i] + y[i]) MOD 2 = 0 then
12.           r[i] ← 0
13.       Else
14.           r[i] ← 1
15.       EndIf
16.   EndFor
17.   For i ← length + 1, m execute
18.       r[i] ← y[i]
19.   EndFor
20.   For i ← length + 1, n execute
21.       r[i] ← x[i]
22.   EndFor
23.   Return r, lenF
24.EndAlgorithm
```

Which of the following statements are true?

A. The algorithm `OperatieSpeciala(x, n, y, m)` correctly implements the *os* operation and returns the resulting array and its length.
B. For those input data for which the instructions from lines 18 and 21 are executed the same number of times, the returned result is correct.
C. The implementation would be correct if we replaced the instruction from line 11 with
`If (x[n - i] + y[m - i]) MOD 2 = 0 then`
D. The result of the `OperatieSpeciala(x, n, y, m)` algorithm is not correct, and the returned array contains the elements in reverse order.

10. Consider the algorithm `ceFace(x, m, y, n)`, where ***x*** is an array of characters of length ***m*** ($1 \leq m \leq 100$) and ***y*** an array of characters of length ***n*** ($1 \leq n \leq 100$), such that $m < n$.

```
1. Algorithm ceFace(x, m, y, n):
2.     i ← 1
3.     ok ← True
4.     While ok AND i ≤ m execute
5.         If i ≤ m AND x[i] ≠ y[i] then
6.             ok ← False
7.         Else
8.             i ← i + 1
9.         EndIf
10.    EndWhile
11.    _____________
12. EndAlgorithm
```

What instruction should be placed on line 11, such that the algorithm returns *True* if the array ***x*** is a prefix for array ***y***? *Example*: if ***x*** = `"abc"` and ***y*** = `"abcd"`, ***x*** is a prefix of ***y*** and the algorithm returns *True*.

A. `Return (i = m) OR (i = n)`
B. `Return i = m`
C. `Return i > m`
D. `Return ok`

11. Consider a matrix ***A*** with ***m*** rows and ***n*** columns (***A***[1][1], ***A***[1][2], ..., ***A***[***m***][***n***]), where ***m*** and ***n*** are natural numbers ($1 < m \leq 25$, $1 < n \leq 25$), and $1 \leq A[i][j] \leq 10^3$, for $i = 1, 2, ..., m$; $j = 1, 2, ..., n$.

Which of the following algorithms return the sum of the elements from column ***k*** ($1 < k \leq n$)?

A.
```
Algorithm suma(A, m, n, k):
    s ← 0
    For i ← n, 1, -1 execute
        s ← s + A[i][k]
    EndFor
    Return s
EndAlgorithm
```

B.
```
Algorithm suma(A, m, n, k):
    s ← 0; i ← 1
    While i ≤ m execute
        s ← s + A[i][k]
        i ← i + 1
    EndWhile
    Return s
EndAlgorithm
```

C.
```
Algorithm suma(A, m, n, k):
    s ← 0
    For i ← 1, m execute
        s ← s + A[k][i]
    EndFor
    Return s
EndAlgorithm
```

D.
```
Algorithm suma(A, m, n, k):
    s ← 0; k ← 1
    While k ≤ n execute
        s ← s + A[k][k]
        k ← k + 1
    EndWhile
    Return s
EndAlgorithm
```

12. Consider algorithm `F(n)`, where ***n*** is a non-zero natural number ($1 \leq n \leq 10^6$). The algorithm `sqrt(n)` returns the square root of ***n*** and has a complexity of $O(1)$. The notation [a] represents the integer part of value a. The „/" operator represents real division, for example: 3 / 2 = 1.5.

```
Algorithm F(n):
    If n = 1 then
        Return 1
    EndIf
    i ← [n / sqrt(n)]
    Return 1 + F(i)
EndAlgorithm
```

Which of the following statements are true?

A. Algorithm `F(n)` has a time complexity of $O(\log \log n)$.
B. After the call `F(200)` the value 4 is obtained.
C. After the call `F(250)` the value 5 is obtained.
D. Algorithm `F(n)` has a time complexity of $O(1)$.

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

```
Algorithm check(n, x):
    If n < 3 then
        Return False
    EndIf
    p ← select(n, x)
    If p = 1 OR p = n then
        Return False
    EndIf
    For i ← 2, p execute
        If x[i] ≥ x[i - 1] then
            Return False
        EndIf
    EndFor
    For i ← p + 1, n - 1 execute
        If x[i] ≥ x[i + 1] then
            Return False
        EndIf
    EndFor
    Return True
EndAlgorithm
```

```
Algorithm select(n, x):
    r ← 0
    v ← x[1]
    For i ← 2, n execute
        If x[i] < v then
            r ← i
            v ← x[i]
        EndIf
    EndFor
    Return r
EndAlgorithm
```

Which of the following statements are true?

A. If array ***x*** is sorted in descending order and has at least 3 elements, algorithm `check(n, x)` returns *True*.
B. If array $x = [12, 10, 8, 5, 9, 11, 15, 18]$ and $n = 8$ algorithm `check(n, x)` returns *True*.
C. If array $x = [20, 10, 5, 1, 2, 4, 6, 10, 8]$ and $n = 9$ algorithm `check(n, x)` returns *False*.
D. If array ***x*** is sorted in strictly ascending order and has at least 3 elements, algorithm `check(n, x)` returns *True*.

14. Consider algorithm `f(x, n)`, 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 $i = 1, 2, ..., n$). The notation [ ] represents an empty array, and [***a***, ***b***] represents an array with 2 elements, ***a*** and ***b***.

```
Algorithm f(x, n):
    If n < 2 then
        Return []
    EndIf
    If n = 2 then
        If x[1] > x[2] then
            Return [x[1], x[2]]
        Else
            Return [x[2], x[1]]
        EndIf
    EndIf
    y ← f(x, n - 1)
    If x[n] > y[1] then
        Return [x[n], y[1]]
    Else
        If x[n] > y[2] then
            Return [y[1], x[n]]
        Else
            Return y
        EndIf
    EndIf
EndAlgorithm
```

What will be returned for the call `f([4, 15, 5, 8, 10, 18, 16, 19, 1, 12], 10)`?

A. `[19, 18]`
B. `[18, 19]`
C. `[16, 19]`
D. `[19, 16]`

15. Consider the algorithm `numere(x, n, e)`, where ***n*** is a natural number ($1 \leq n \leq 10^4$), ***x*** is an array of ***n*** integer elements (***x***[1], ***x***[2], ..., ***x***[***n***], where $-100 \leq x[i] \leq 100$, for $i = 1, 2, ..., n$), and ***e*** is the value of an element in the array:

```
Algorithm numere(x, n, e):
    i ← 1
    c ← 0
    b ← True
    If n MOD 2 = 0 then
        Return False
    EndIf
    While (i ≤ n) AND b execute
        If x[i] < e then
            c ← c + 1
        Else
            b ← False
        EndIf
        i ← i + 1
    EndWhile
    Return c = (n – i + 1)
EndAlgorithm
```

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

A. If the array has an even number of elements and is sorted in descending order until the element that has the value ***e*** inclusively, which is found on position ***n*** **DIV** 2.
B. If the array has an odd number of elements and is sorted in strictly ascending order until the element that has the value ***e*** inclusively, which is found on position ***n*** **DIV** 2 + 1.
C. If the array has an odd number of elements and is sorted in descending order until the element that has the value ***e*** inclusively, which is found on position ***n*** **DIV** 2 + 1.
D. If the array has an odd number of elements, the value ***e*** is found on position ***n*** **DIV** 2 + 1 and before ***e*** there are only smaller values and after ***e*** there are only higher values.

16. Which of the following algorithms display the representation in base ***b*** of the number ***a*** where ***a***, ***b*** are natural numbers given in base 10 ($1 \leq a \leq 10^4$, $2 \leq b \leq 9$, $a > b$)?

A.

```
Algorithm afiseaza(a, b):
    If a ≠ 0 then
        Write a MOD b
        afiseaza(a DIV b, b)
    EndIf
EndAlgorithm
```

B.

```
Algorithm afiseaza(a, b):
    If a ≠ 0 then
        afiseaza(a DIV b, b)
        Write a MOD b
    EndIf
EndAlgorithm
```

C.

```
Algorithm afiseaza(a, b):
    While a > 0 execute
        Write a MOD b
        a ← a DIV b
    EndWhile
EndAlgorithm
```

D.

```
Algorithm afiseaza(a, b):
    nrNou ← 0
    putere ← 1
    While a > 0 execute
        nrNou ← nrNou + (a MOD b) * putere
        a ← a DIV b
        putere ← putere * b
    EndWhile
    Write nrNou
EndAlgorithm
```

17. Consider the algorithm `f(x, y)`, where ***x*** and ***y*** are two natural numbers ($1 \leq x \leq 100$, $1 \leq y \leq 100$).

```
Algorithm f(x, y):
    If x = y then
        Write "start: "
    Else
        If x MOD y = 0 then
            f(x + 1, y + 2)
        Else
            If (x DIV y) MOD 2 = 0 then
                f(x + 2, y + 1)
                Write "*"
            Else
                f(x - 1, y + 1)
                Write "#"
            EndIf
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. The calls `f(12, 15)` and `f(8, 12)` do not display the same array of characters.
B. The calls `f(15, 12)` and `f(12, 8)` display the same array of characters.
C. The call `f(17, 23)` does not display any occurrence of the character `"#"`
D. The call `f(23, 17)` displays at least one occurrence of the character `"#"`

18. Consider the algorithm `decide(n, x, t)`, where ***x*** is an array of ***n*** natural numbers ($2 \leq n \leq 10^4$, $1 \leq x[i] \leq 10^4$, $i = 1, 2, ..., n$) and ***t*** is a natural number ($1 \leq t \leq 10^4$).

```
Algorithm decide(n, x, t):
    left ← 1; right ← n
    While x[left] + x[right] ≠ t execute
        If x[left] + x[right] < t then
            left ← left + 1
        Else
            right ← right - 1
        EndIf
    EndWhile
    Return left, right
EndAlgorithm
```

In which of the following situations the algorithm `decide(n, x, t)` determines the indices ***left***, ***right*** ($1 \leq left < right \leq n$) such that $x[left] + x[right] = t$?

A. If and only if array ***x*** contains distinct numbers.
B. If and only if array ***x*** is sorted in ascending order.
C. If array ***x*** is sorted in descending order and contains distinct numbers.
D. If array ***x*** is sorted in ascending order and in the array there is at least one pair of elements with the sum equal to ***t***.

19. Consider the algorithm `perechi(x, y)`, where ***x*** and ***y*** are non-zero natural numbers ($1 \leq x, y \leq 100$):

```
Algorithm perechi(x, y):
    nr ← 0; d ← 2
    While d ≤ x AND d ≤ y execute:
        If (x MOD d = 0) AND (y MOD d = 0) then
            nr ← nr + 1
            x ← x DIV d
            y ← y DIV d
        Else
            d ← d + 1
        EndIf
    EndWhile
    Write nr, " ", x, " ", y
EndAlgorithm
```

In which of the following variants do we only have pairs of numbers (x, y) for which the algorithm `perechi(x, y)` displays the values `1 7 11`?

A. `(14, 22), (21, 33), (35, 55), (49, 77)`
B. `(7, 11), (14, 22), (21, 33), (28, 44)`
C. `(1, 7), (1, 11)`
D. `(2, 2), (3, 3), (4, 4), (5, 5)`

20. Consider the algorithm `first(x, n)`, where ***n*** is a non-zero natural number ($2 \leq n \leq 10^4$), and ***x*** is an array of ***n*** natural numbers ($x[1], x[2], ..., x[n]$, where $1 \leq x[i] \leq 10^4$, for $i = 1, 2, ..., n$).

```
01. Algorithm first(x, n):
02.     f1 ← False
03.     f2 ← False
04.     For i ← 1, n execute
05.         If x[i] = 1 then
06.             f1 ← True
07.         EndIf
08.         If x[i] = n then
09.             f2 ← True
10.         EndIf
11.         If x[i] ≥ n then
12.             x[i] ← 1
13.         EndIf
14.     EndFor
15.     If NOT f1 then
16.         Return 1
17.     EndIf
18.     For i ← 1, n execute
19.         ...
20.     EndFor
21.     If f2 then
22.         x[n] ← n
23.     EndIf
24.     For i ← 1, n execute
25.         ...
26.             Return i
27.         EndIf
28.     EndFor
29.     Return n + 1
30. EndAlgorithm
```

What should lines 19 and 25 be replaced with, so that the algorithm returns the smallest non-zero natural number that is not found in array ***x***?

A.
```
19: x[x[i] MOD (n + 1)] ← x[x[i] MOD (n + 1)] + n
25: If x[i] DIV (n + 1) = 0 then
```
B.
```
19: x[x[i] MOD n] ← x[x[i] MOD n] + n
25: If x[i] DIV n = 0 then
```
C.
```
19: x[x[i] MOD n] ← 1
25: If x[i] = 1 then
```
D.
```
19: x[x[i] MOD n] ← x[x[i] MOD n] + n
25: If x[i] MOD n = 0 then
```

21. Consider the algorithm `ceFace(arr, n)` where ***arr*** is an array of ***n*** ($1 \leq n \leq 100$) integer elements (***arr***[1], ***arr***[2], ..., ***arr***[***n***], where $-10^5 \leq arr[i] \leq 10^5$ for $i = 1, 2, ..., n$).

```
Algorithm ceFace(arr, n):
    sum ← 0
    For i ← 1, n execute
        sum ← sum + arr[i]
    EndFor
    If sum MOD 2 ≠ 0 then
        Return False
    EndIf
    Return auxiliar(arr, n, sum DIV 2)
EndAlgorithm
```

```
Algorithm auxiliar(arr, n, sum):
    If sum = 0 then
        Return True
    EndIf
    If n = 1 AND sum ≠ 0 then
        Return False
    EndIf
    If arr[n - 1] > sum then
        Return auxiliar(arr, n - 1, sum)
    EndIf
    Return auxiliar(arr, n - 1, sum) OR
        auxiliar(arr, n - 1, sum - arr[n - 1])
EndAlgorithm
```

Which of the following statements are true?

A. The call `ceFace([11, 5, 6, 22, 0, 7, 6, 13], 8)` returns *True*.
B. The call `ceFace([-5, -6, -22, -7, -6, -13], 6)` **DOES NOT** return *True*.
C. If array ***arr*** contains only negative values, the algorithm `auxiliar(arr, n, sum)` will enter an infinite loop.
D. If and only if the elements of array ***arr*** can be partitioned in two sets such that the averages of the elements from each set are equal, the algorithm `ceFace(arr, n)` returns *True*.

22. Consider 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***], where $1 \leq x[i] \leq 10^4$, for $i = 1, 2, ..., n$):

```
1.  Algorithm f(n, x):
2.      s1 ← h(n, x)
3.      For i ← 1, 2 * n execute
4.          x[((i + 1) MOD n) + 1] ← g(x[(i MOD n) + 1], x[((i + 1) MOD n) + 1])
5.      EndFor
6.      s2 ← h(n, x)
7.      Return x[n]
8.  EndAlgorithm
```

```
Algorithm g(a, b):
    If a * b = 0 then
        Return a + b
    EndIf
    If a = b then
        Return a
    EndIf
    If a > b then
        Return g(a - b, b)
    EndIf
    Return g(a, b - a)
EndAlgorithm
```

```
Algorithm h(n ,x):
    s ← 0
    For i ← 1, n execute
        s ← s + x[i]
    EndFor
    Return s
EndAlgorithm
```

Which of the following statements are true?

A. For the call `f(6, [12, 16, 80, 40, 28, 144])` the algorithm returns the value 4.
B. For any input array, the value ***s*1** (calculated on line 2 from algorithm `f(n, x)`) will be strictly greater than the value ***s*2** (computed on line 6 from algorithm `f(n, x)`).
C. If, in algorithm `f(n, x)` we replace the instructions on lines 3, 4 and 5 with the following sequence, at the end of algorithm `f(n, x)` array ***x*** will have the same contents as in the original algorithm.

```
For j ← 1, 2 execute
    For i ← 1, n - 1 execute
        x[i + 1] ← g(x[i], x[i + 1])
    EndFor
EndFor
```

D. There exists an input array with ***n*** elements for which the time complexity of the algorithm `f(n, x)` is $O(n)$.

23. Consider a non-zero even natural number ***n*** ($2 \leq n \leq 12$). We want to generate in the array of characters ***x*** all possible arrays consisting of ***n*** round brackets that open and close correctly. The algorithm `paranteze(i, desc, inc, x, n)` is called as `paranteze(2, 1, 0, x, n)`, knowing that the following initialization instructions were executed `x[1] ← '('` and `x[n] ← ')'`. The algorithm `afisare(n, x)` displays the array of characters ***x*** of length ***n***.

```
1.  Algorithm paranteze(i, desc, inc, x, n):
2.      If i = n then
3.          afisare(n, x)
4.      Else
5.          If ____ then
6.              x[i] ← '('
7.              paranteze(i + 1, desc + 1, inc, x, n)
8.          EndIf
9.          If ____ then
10.             x[i] ← ')'
11.             paranteze(i + 1, desc, inc + 1, x, n)
12.         EndIf
13.     EndIf
14. EndAlgorithm
```

What should be filled in the lines specified below such that the algorithm displays only the correct arrays of characters according to the problem statement?

A. Line 5 must be completed with `desc < n`, and line 9 must be completed with `inc < desc`
B. Line 5 must be completed with `desc < n DIV 2`, and line 9 must be completed with `inc < desc`
C. Line 5 must be completed with `desc < n`, and line 9 must be completed with `inc < n DIV 2`
D. Regardless of what comparisons are used to complete lines 5 and 9, the algorithm will not display all the arrays of characters according to the problem statement.

24. During the physical education class ***n*** students are placed one next to the other, facing the teacher who asks them to turn left. Some students turn left, others, by mistake, turn right. In a unit of time, all students that face each other turn by 180°, each student doing at most one turn. The turning continues until there are no students facing one another. State which of the following implementations for the `intoarceri(n, c)` algorithm determine the number of units of time ***t*** that pass until there are no students facing each other. Variable ***n*** is a non zero natural number ($1 \leq n \leq 100$), and the array of characters ***c*** contains ***n*** elements, where ***c*[*i*]** is either `'s'` (representing „left") either `'d'` (representing „right") depending on the direction in which the ***i***-th student turned after the teacher's request. *Examples:* if ***n*** = 6 and ***c*** = `"sdsssd"`, then ***t*** = 3; if ***n*** = 3 and ***c*** = `"sdd"`, then ***t*** = 0. The algorithm `copiaza(a, n)` returns a copy of array ***a*** with ***n*** elements.

A.
```
Algorithm intoarceri(n, c):
    t ← 0; aux ← copiaza(c, n); ok ← False
    While NOT ok execute
        ok ← True
        For i ← 1, n – 1 execute
            If (aux[i] = 'd') AND (aux[i + 1] = 's') then
                c[i] ← 's'; c[i + 1] ← 'd'
                ok ← False
            EndIf
        EndFor
        aux ← copiaza(c, n)
        If NOT ok then
            t ← t + 1
        EndIf
    EndWhile
    Return t
EndAlgorithm
```

B.
```
Algorithm intoarceri(n, c):
    t ← 0; dr ← 0; st ← 0
    For i ← 1, n execute
        If c[i] = 'd' then
            dr ← dr + 1
            If st > 0 then
                st ← st – 1
            EndIf
        Else
            If dr > 0 then
                t ← dr + st
                st ← st + 1
            EndIf
        EndIf
    EndFor
    Return t
EndAlgorithm
```

C.
```
Algorithm intoarceri(n, c):
    stop ← False; t ← 0
    While NOT stop execute
        i ← 1
        stop ← True
        While i < n execute
            If (c[i] = 'd') AND (c[i + 1] = 's') then
                c[i] ← 's'; c[i + 1] ← 'd'
                i ← i + 2
                stop ← False
            Else
                i ← i + 1
            EndIf
        EndWhile
        If NOT stop then
            t ← t + 1
        EndIf
    EndWhile
    Return t
EndAlgorithm
```

D.
```
Algorithm intoarceri(n, c):
    t ← 0; dr ← 0; st ← 0
    For i ← 1, n execute
        If c[i] = 'd' then
            dr ← dr + 1
            If st > 0 then
                t ← dr + st
                st ← st – 1
            EndIf
        Else
            If dr > 0 then
                t ← dr + st
                dr ← dr + 1
            EndIf
        EndIf
    EndFor
    Return t
EndAlgorithm
```

BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – September 6th, 2024

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

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