BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission exam – September 8th 2023
Written Exam for Computer Science

IMPORTANT NOTE:
In the absence of other specifications:

- Assume that all arithmetic operations are performed on unlimited data types (there is no *overflow* / *underflow*).
- The indexing of all arrays begins at 1.
- All restrictions refer to the values of the actual parameters at the moment of the initial call.
- A subsequence of an array consists of elements of that array that occupy consecutive positions.

1. Let us consider the algorithm `ceFace(a, b)`, where ***a*** and ***b*** are natural numbers ($0 \le a, b \le 10^4$).

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

What is the effect of the call `ceFace(a, a)`?

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

2. Let us consider the algorithm `creareTablou(n, m, x)`, where ***n, m*** are natural numbers ($1 \le n, m \le 100$), and ***x*** is a bidimensional array with ***n*** * ***m*** integer number elements ($x[1][1], x[1][2], ..., x[n][m], 0 \le x[i][j] \le 10^4$, for $i = 1, 2, ..., n; j = 1, 2, ..., m$).

```
Algorithm creareTablou(n, m, x):
    k ← 0
    For i ← 1, n execute
        For j ← 1, m execute
            If k MOD 2 ≠ 0 then
                x[i][j] ← k * k
            EndIf
            Write x[i][j], " "
            k ← k + 1
        EndFor
        Write new line
    EndFor
EndAlgorithm
```

What does the algorithm display if the elements of the array ***x*** are initialized with 0?

A. The algorithm displays the elements of the bidimensional array ***x***, in which there are elements equal to 0 and the first (***n*** * ***m***) **DIV** 2 odd perfect squares.
B. The algorithm displays the elements of the bidimensional array ***x***, in which there are values equal to 0 and the first even perfect squares.
C. The algorithm displays the elements of the bidimensional array ***x***, in which there are the first (***n*** * ***m***) **DIV** 2 even perfect squares.
D. The algorithm displays the elements of the bidimensional array ***x***, in which – if we laid out the elements one line after the other – the odd perfect squares would be in ascending order, possibly preceded and/or succeeded by values equal to 0.

3. Let us consider the algorithm `something(n, x)`, where ***n*** is a natural number ($1 \le n \le 10^4$), and ***x*** is an array of ***n*** natural numbers ($x[1], x[2], ..., x[n], 1 \le x[i] \le 10^6$, for $i = 1, 2,.., n$).

```
Algorithm something(n, x):
    s ← 0
    For i ← 1, n execute
        nr ← 1
        While x[i] > 9 execute
            nr ← nr + 1
            x[i] ← x[i] DIV 10
        EndWhile
        s ← s + nr
    EndFor
    Return s
EndAlgorithm
```

What does the call `something(5, [222, 2043, 29, 2, 20035])` return?

A. 16
B. 10
C. 11
D. 15

4. Let us consider the algorithm `ceFace(n, v, a)`, where ***n*** and ***v*** are two natural numbers ($1 \leq n, v \leq 10^4$), and ***a*** is an array of natural numbers with ***n*** elements (***a***[1], ***a***[2], ..., ***a***[***n***]).

```
Algorithm ceFace(n, v, a):
    For i ← 1, n execute
        d ← v
        If a[i] ≠ 0 then
            găsit ← False
            While (d ≤ v * a[i]) AND (NOT găsit) execute
                If ((d DIV a[i]) * a[i] = d) AND ((d DIV v) * v = d) then
                    găsit ← True
                Else
                    d ← d + 1
                EndIf
            EndWhile
        EndIf
        v ← d
    EndFor
    Return v
EndAlgorithm
```

What is the value returned by the algorithm, if ***n*** = 4, ***v*** = 3 and ***a*** = [5, 4, 2, 10]?

A. 20 B. 120 C. 60 D. 15

5. Let us consider the algorithm `calcul(v, n)`, where ***n*** is a natural number ($1 \leq n \leq 10^4$), and ***v*** is an array with ***n*** elements which are natural numbers (***v***[1], ***v***[2], ..., ***v***[***n***], $1 \leq v[i] \leq 10^4$, for ***i*** = 1, 2, ..., ***n***):

```
Algorithm calcul(v, n):
    i ← 1
    While i ≤ n DIV 2 execute
        p ← 0
        While v[i] ≠ 0 execute
            p ← p + 1
            v[i] ← v[i] DIV 10
        EndWhile
        q ← 0
        While v[n + 1 - i] ≠ 0 execute
            q ← q + 1
            v[n + 1 - i] ← v[n + 1 - i] DIV 10
        EndWhile
        If p ≠ q then
            Return False
        EndIf
        i ← i + 1
    EndWhile
    Return True
EndAlgorithm
```

In which of the following situations the algorithm returns *True*?

A. If the array ***v*** consists of the values [12, 12, 2, 5466, 3, 111, 1, 3, 44] and ***n*** = 9.
B. If the array ***v*** consists of the values [12, 345, 2, 5466, 3, 111, 10] and ***n*** = 7.
C. If the elements of the array ***v*** have the same number of digits.
D. If the array consisting of the number of digits of the elements of array ***v*** forms a palindrome; for example, from ***v*** = [8, 37, 3] the array [1, 2, 1] is formed, which is a palindrome.

6. Let us consider the algorithm `alg(n)`, where ***n*** is a natural number ($0 \leq n \leq 10^4$).

```
Algorithm alg(n):
    If n = 0 then
        Return 0
    Else
        If n MOD 2 = 0 then
            Return alg(n DIV 10) + n MOD 10
        Else
            Return alg(n DIV 10)
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. The call `alg(123)` returns 6.
B. The algorithm calculates the sum of the digits found on even positions in the given number.
C. The algorithm calculates the sum of the even digits from the given number.
D. The algorithm calculates the sum of the digits of the given number.

7. Let us consider the algorithm `f(x)`, where x is a non-zero natural number ($1 \leq x \leq 10^5$).

```
Algorithm f(x):
    If x > 0 then
        x ← x DIV 2
        f(x)
        Write x, " "
        x ← x DIV 2
        f(x)
    EndIf
EndAlgorithm
```

What will be displayed after the call `f(10)`?

A. `0 1 2 0 5 0 1`
B. `0 1 2 5 1 0`
C. `1 2 1 5 2 1`
D. `1 2 1 1 5 1 2`

8. Let us consider the square matrix M of size n that contains natural numbers, where n is a non-zero natural number ($1 \leq n \leq 10^4$, $M[1][1]$, ..., $M[1][n]$, $M[2][1]$, ..., $M[2][n]$, ..., $M[n][1]$, ..., $M[n][n]$, $1 \leq M[i][j] \leq 10^4$, for $i = 1, 2, ..., n$, $j = 1, 2, ..., n$). Let us consider the following algorithm:

```
Algorithm what(M, n):
    up ← 1
    down ← n
    left ← 1
    right ← n
    While left ≤ right AND up ≤ down execute
        For i ← left, right execute
            Write M[up][i], " "
        EndFor
        up ← up + 1
        For i ← up, down execute
            Write M[i][right], " "
        EndFor
        right ← right - 1
        For i ← right, left, -1 execute
            Write M[down][i], " "
        EndFor
        down ← down - 1
        For i ← down, up, -1 execute
            Write M[i][left], " "
        EndFor
        left ← left + 1
    EndWhile
EndAlgoritm
```

What will be displayed for the following matrix M?

1	2	3
8	9	4
7	6	5

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

9. Let us consider the algorithm `ce_face(a, b)`, where a and b are natural numbers ($1 \leq a, b \leq 10^4$).

```
Algorithm ce_face(a, b):
    If a = 1 then
        Return 1
    Else
        If a MOD b = 0 then
            Return ce_face(a DIV b, b)
        Else
            Return 0
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. For the call `ce_face(1, 2)` the algorithm returns 1
B. For the call `ce_face(24, 2)` the algorithm returns 0
C. For the call `ce_face(2024, 4)` the algorithm returns 4
D. For the call `ce_face(8, 3)` the algorithm returns 2

10. Let us consider the algorithms `decide(n)` and `compute(m)`, where n and m are non-zero natural numbers ($1 \leq n, m \leq 10^4$):

```
Algorithm decide(n):
    result ← -1
    m ← 0
    While n > 0 execute
        m ← m * 10 + n MOD 10
        n ← n DIV 10
    EndWhile
    If m MOD 3 = 0 then
        result ← 1
    EndIf
    Return result
EndAlgorithm
```

```
Algorithm compute(m):
    cnt ← 0
    For k ← 0, m – 1 execute
        cnt ← cnt + decide(k)
    EndFor
    Return cnt
EndAlgorithm
```

For what values of m the algorithm `compute(m)` will return -33?

A. 100
B. 99
C. 98
D. 101

11. Let us consider the algorithm `f(n, x)`, where ***n*** and ***x*** are natural numbers ($1 \le n \le 10^5$, $2 \le x \le 10$):

```
Algorithm f(n, x):
    If n > 0 then
        f(n DIV x, x)
        Write n MOD x
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm displays the representation of the number ***n*** in base ***x***.
B. The algorithm displays the remainder of the integer division of the number ***x*** to number ***n***.
C. The algorithm displays the number of digits from the representation in base ***x*** of number ***n***.
D. The algorithm checks if the number ***n*** is divisible by ***x***.

12. Let us consider the algorithm `ceFace(n)`, where ***n*** is a natural number ($1 \le n \le 10^9$).

```
Algorithm ceFace(n):
    If n ≤ 9 then
        If n MOD 2 = 0 then
            Return n
        Else
            Return -1
        EndIf
    EndIf
    x ← n MOD 10
    y ← ceFace(n DIV 10)
    If x MOD 2 ≠ 0 then
        Return y
    EndIf
    If x > y then
        Return x
    EndIf
    Return y
EndAlgorithm
```

Which of the following statements are true?

A. The algorithm returns a number containing a single digit or -1.
B. The algorithm returns an odd number.
C. The algorithm returns the maximum odd digit from the number ***n***, or -1.
D. The algorithm returns the maximum even digit from the number ***n***, or -1.

13. Let us consider the algorithm `decide(n, x)`, where ***n*** is a natural number ($1 \le n \le 10^4$) , and ***x*** is an array with ***n*** integer numbers as elements (***x***[1], ***x***[2], ..., ***x***[***n***], $-100 \le x[i] \le 100$, for ***i*** = 1, 2, ..., ***n***):

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

In which of the following cases the algorithm returns *True*?

A. If the array ***x*** = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and ***n*** = 10
B. If ***n*** > 1 and the elements of the array ***x*** are in strictly ascending order
C. If the array ***x*** does not contain negative numbers
D. If the array ***x*** has positive elements situated before the negative ones

14. Let ***x*** and ***y*** be two positive natural numbers with the following properties: ***x*** is a power of 2 and ***y*** is a multiple of 3. Let us consider the following logical expression:

```
((x * y + 3) DIV 6 = 10) OR (((x * y) MOD 6 = 0) AND ((x + y) MOD 4 = 0))
```

Which of the following statements are true for pairs of numbers that follow the previously mentioned properties?

A. There exists a pair (***x***, ***y***) for which the expression is true.
B. There exists a pair (***x***, ***y***) for which the expression is false.
C. There exist the pairs (x_1, y_1) and (x_2, y_2), with $x_1 \ne x_2$ and $y_1 \ne y_2$ such that the expression is true for both pairs.
D. The expression is false for any pair (***x***, ***y***).

15. Let us consider two natural numbers ***n*** and ***m*** ($1 \le n, m \le 256$) and the arrays of characters ***a***, with ***n*** characters (***a***[1], ***a***[2], ..., ***a***[***n***]) and ***b*** having ***m*** characters (***b***[1], ***b***[2], ..., ***b***[***m***]).
Which of the following algorithms return *True* if the array ***a*** can be obtained starting from array ***b*** and eliminating some characters without modifying the relative positions of the remaining characters, and *False* otherwise. For example, the array `"ace"` can be formed by eliminating characters from the array `"abcde"`, but the array `"aec"` cannot be obtained in the same manner.

A.

```
Algorithm hasProperty(a, b, n, m):
    If n = 0 then
        Return True
    EndIf
    If m = 0 then
        Return False
    EndIf
    If a[n] = b[m] then
        Return hasProperty(a, b, n - 1, m - 1)
    EndIf
    Return hasProperty(a, b, n, m - 1)
EndAlgorithm
```

B.

```
Algorithm hasProperty(a, b, n, m):
    i ← 1
    j ← 1
    While i ≤ n AND j ≤ m execute
        If a[i] = b[j] then
            i ← i + 1
        EndIf
        j ← j + 1
    EndWhile
    If i > n then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

C.

```
Algorithm hasProperty(a, b, n, m):
    i ← n
    j ← m
    While i * j > 0 execute
        If a[i] = b[j] then
            i ← i - 1
        EndIf
        j ← j - 1
    EndWhile
    If i = 0 then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

D.

```
Algorithm hasProperty(a, b, n, m):
    If n > m then
        Return False
    EndIf
    i ← 1
    j ← 1
    While i < n execute
        If a[i] = b[j] then
            i ← i + 1
        EndIf
        j ← j + 1
    EndWhile
    If i > m then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

16. Let us consider the algorithm `ceva(x, n, e)`, where $\boldsymbol{x}$ is an array with $\boldsymbol{n}$ distinct integer elements ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$, $1 \leq \boldsymbol{n} \leq 10^3$ and $\boldsymbol{x}[\boldsymbol{i}] \neq \boldsymbol{x}[\boldsymbol{j}]$, for $1 \leq \boldsymbol{i} < \boldsymbol{j} \leq \boldsymbol{n}$) and $\boldsymbol{e}$ is an integer number. The algorithm searches for element $\boldsymbol{e}$ in array $\boldsymbol{x}$, and if it finds it, moves the element to the first position in the array and returns *True*, without modifying the relative order of the other elements. If $\boldsymbol{e}$ is not found in array $\boldsymbol{x}$, the algorithm returns *False* and does not modify the array. For example, for array $\boldsymbol{x}$ with elements [-100, 2, 71, 31, -62, 51] and $\boldsymbol{e} = 31$, the algorithm will return *True* and the array $\boldsymbol{x}$ will become [31, -100, 2, 71, -62, 51]. Which of the following variants represent a correct implementation for the `ceva(x, n, e)` algorithm that also has time complexity $O(\boldsymbol{n})$?

A.

```
Algorithm ceva(x, n, e):
    index ← 1
    While index ≤ n execute
        If x[index] = e then
            tmp ← x[index]
            x[index] ← x[1]
            x[1] ← tmp
            Return True
        EndIf
        index ← index + 1
    EndWhile
    Return False
EndAlgorithm
```

B.

```
Algorithm ceva(x, n, e):
    index ← 2
    tmp ← x[1]
    While index ≤ n execute
        If x[index] = e then
            x[1] ← e
            x[index] ← tmp
            Return True
        EndIf
        tmp2 ← x[index]
        x[index] ← tmp
        tmp ← tmp2
        index ← index + 1
    EndWhile
    Return False
EndAlgorithm
```

C.

```
Algorithm ceva(x, n, e):
    index ← n
    While index > 1 execute
        If x[index] = e then
            index2 ← index
            While index2 > 1 execute
                x[index2] ← x[index2 - 1]
                index2 ← index2 – 1
            EndWhile
            x[index2] ← e
        EndIf
        index ← index – 1
    EndWhile
    If x[1] = e then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

D.

None of the variants A, B, C

17. Let us consider the algorithm `expresie(x, y, z)`, where x, y, z are natural numbers ($0 \leq x, y, z \leq 10^4$):

```
Algorithm expresie(x, y, z):
    If x = 0 then
        Return z
    Else
        Return expresie(x - 1, y, x * x + y * y + z)
    EndIf
EndAlgorithm
```

Specify which expression value is calculated and returned by the algorithm:

A. $\sum_{i=1}^{x} i^2 + \sum_{i=1}^{y} x * y + \sum_{k=1}^{z} 1$

B. $\sum_{i=1}^{x} i^2 + \sum_{j=1}^{y} j^2 + z$

C. $\sum_{i=1}^{x} i^2 + x * y^2 + z$

D. $\sum_{i=1}^{x} i^2 + \sum_{j=1}^{y} j^2 + \sum_{k=1}^{z} k$

18. Let us consider the algorithm `ceFace(v, a, b)`, where v is an array of n elements with values from the set {0, 1}, ($1 \leq n \leq 10^4$, $v[1], ..., v[n]$), and a and b are natural non-zero numbers. The array v is sorted in ascending order.

```
Algorithm ceFace(v, a, b):
    If b – a + 1 = 0 then
        Return 0
    EndIf
    If v[a] = 1 then
        Return b – a + 1
    EndIf
    If v[b] = 0 then
        Return 0
    EndIf
    c ← (a + b) DIV 2
    Return ceFace(v, a, c) + ceFace(v, c + 1, b)
EndAlgorithm
```

Which of the following statements are true, considering that the initial call is `ceFace(v, 1, n)`?

A. If the array v contains at least one element with value 1, then the algorithm returns the length of the array.

B. If the array v contains only elements with value 1, then the algorithm returns the value of n.

C. If the array v contains only elements with value 0, then the algorithm returns 0.

D. The algorithm returns the number of elements with value 1 contained by array v.

19. It is known that the total number of binary arrays (that contain only the characters `0` and `1`) of length n is 2^n. For example, for $n = 2$ those arrays are `00`, `01`, `10` and `11`, their number being $2^2 = 4$. The array `100011` has length 6 and contains as subsequences all of the 4 possible arrays of length $n = 2$, since starting from the first position we have `10`, starting with the second position we have `00`, starting from the fourth position we have `01` and starting with the fifth position we have `11`.

What is the minimal length of an array that contains as subsequences all the 2^n possible binary arrays for $n = 4$?

A. 18 B. 19 C. 20 D. 21

20. Let us consider the algorithm `t(q, x, y)`, where $\boldsymbol{q}$ is a character, and $\boldsymbol{x}$ and $\boldsymbol{y}$ are non-zero natural numbers ($1 \leq \boldsymbol{x}, \boldsymbol{y} \leq 100$).

```
Algorithm t(q, x, y):
    If x ≤ y then
        Write q
    Else
        If x MOD y = 0 then
            t(q, x + 1, y - 2)
        Else
            If (x DIV y) MOD 2 ≠ 0 then
                t(q, x - 1, y + 2)
                Write 'c'
            Else
                t(q, x - 1, y - 1)
                Write "cc"
            EndIf
        EndIf
    EndIf
EndAlgorithm
```

Which of the following statements are true?

A. Calling `t('c', 33, 28)`, `t('c', 10, 6)` and `t('c', 22, 16)` will result in the same characters being displayed.
B. Calling `t('c', 33, 28)` and `t('c', 45, 40)` will not display the same characters.
C. After the call `t('c', 11, 8)` `"cc"` will be displayed.
D. After the call `t('c', 25, 16)` `"ccccc"` will not be displayed.

21. Let us consider the algorithm `hIndex(x, n)`, where $\boldsymbol{x}$ is an array with $\boldsymbol{n}$ ($1 \leq \boldsymbol{n} \leq 10^5$) non-zero natural numbers as elements ($\boldsymbol{x}[1], \boldsymbol{x}[2], ..., \boldsymbol{x}[\boldsymbol{n}]$). We define the ***h-index*** of array $\boldsymbol{x}$, as being the greatest value $\boldsymbol{v}$ for which there are at least $\boldsymbol{v}$ values in $\boldsymbol{x}$ that are greater or equal to $\boldsymbol{v}$. For example, for $\boldsymbol{x} = [3, 10, 2, 7, 10, 8, 50, 1, 1, 5]$ the ***h-index*** is 5.

```
1.  Algorithm hIndex(x, n):
2.      h ← 1
3.      cont ← True
4.      While cont = True AND h ≤ n execute
5.          pos ← h
6.          For i ← h + 1, n execute
7.              If x[i] > x[pos] then
8.                  pos ← i
9.              EndIf
10.         EndFor
11.         If pos ≠ h then
12.             tmp ← x[pos]
13.             x[pos] ← x[h]
14.             x[h] ← tmp
15.         EndIf
16.         If x[h] ≥ h then
17.             h ← h + 1
18.         Else
19.             cont ← False
20.         EndIf
21.     EndWhile
22.     ...
23. EndAlgorithm
```

Which of the following statements are true?

A. At the point when line 22 would be executed, the array $\boldsymbol{x}$ is sorted in descending order.
B. The algorithm `hIndex(x, n)` returns the ***h-index*** of array $\boldsymbol{x}$ if on line 22 we add the instruction **`Return`** `h`.
C. The algorithm `hIndex(x, n)` returns the ***h-index*** of array $\boldsymbol{x}$ if on line 22 we add the instruction **`Return`** `h - 1`.
D. If the algorithm `hIndex(x, n)` is called for an array $\boldsymbol{x}$ that is sorted in strictly descending order, then the algorithm does not return the ***h-index*** of array $\boldsymbol{x}$, regardless of what instruction we add on line 22.

22. Let us consider the algorithm `ceFace(n, k, x, p)`, where $\boldsymbol{n}$, $\boldsymbol{k}$ and $\boldsymbol{p}$ are non-zero natural numbers ($1 \leq \boldsymbol{n}, \boldsymbol{k}, \boldsymbol{p} \leq 10, \boldsymbol{p} \leq \boldsymbol{n}$), and $\boldsymbol{x}$ is an array of $\boldsymbol{p} + 1$ elements that are natural numbers ($\boldsymbol{x}[0], \boldsymbol{x}[1], ..., \boldsymbol{x}[\boldsymbol{p}]$). We assume that $\boldsymbol{x}[0]$ is initialized with the value 0.

```
Algorithm ceFace(n, k, x, p):
    If k > p then
        For i ← 1, p execute
            Write x[i]
        EndFor
        Write " " //one space
    Else
        For i ← x[k - 1] + 1, n execute
            x[k] ← i
            ceFace(n, k + 1, x, p)
        EndFor
    EndIf
EndAlgorithm
```

Specify which of the following statements are correct.

A. After the algorithm is called with `ceFace(3, 1, x, 3)` it will call itself 6 more times.
B. If $\boldsymbol{x}[0]$ is initialized with a value different than 0, after the call `ceFace(5, 1, x, 3)` the number of spaces displayed is different than 10.
C. If the algorithm is called with `ceFace(5, 1, x, 4)` the following numbers are displayed `1245 1234 1345 1235 2345`, but in a different order.
D. If the algorithm is called with `ceFace(5, 1, x, 3)` the displayed result is `123 124 125 134 135 145 234 235` in this order.

23. Let us consider the algorithm `f(sir, s, d, p)`, where ***sir*** is an array of characters, and ***s***, ***d***, ***p*** are non-zero natural numbers ($0 < s, d, p < 10^9$). The operator "+" represents the operator for concatenating two arrays of characters. The algorithm `print(a)` displays the array of characters ***a***, then moves to a new line.

```
1.  Algorithm f(sir, s, d, p):
2.      If s = p AND d = p then
3.          print(sir)
4.      EndIf
5.      If s < p then
6.          f(sir + "-1 ", s + 1, d, p)
7.      EndIf
8.      If s > d then
9.          f(sir + " 1 ", s, d + 1, p)
10.     EndIf
11. EndAlgorithm
```

Specify which of the following statements are true after the call `f("", 0, 0, 2)`:

A. Two arrays of characters are displayed on separate lines, each array containing 4 numbers whose sum is 0 (for example, the sum of the numbers from the string `"-1 1 -1 1"` is 0).
B. Only `"-1 -1 1 1"` is displayed.
C. Only `"-1 -1 1 1"` is displayed, but the algorithm does not finish its execution due to an error.
D. If on line 2 the **AND** operator were replaced with the **OR** operator, then only `"-1 -1"` would be displayed.

24. Let us consider the algorithm `ceFace(a, i, n)`, where ***i*** and ***n*** are natural numbers ($1 \le i, n \le 100$), and ***a*** is an array of ***n*** integer numbers ($a[1], a[2], ..., a[n]$, $-100 \le a[i] \le 100$). In array ***a*** there is at least one positive number. The algorithm `max(x, y, z)` returns the maximum between three integer numbers ***x***, ***y*** and ***z*** ($-10^4 \le x, y, z \le 10^4$). The algorithm `ceFace(a, 1, n)` calls the `intermediar(a, i, m, n)` algorithm, where the parameters ***a***, ***i*** and ***n*** have the meaning described above, and ***m*** is a natural number ($1 \le m \le n$).

```
Algorithm intermediar(a, i, m, n):
    s ← 0
    left ← a[m]
    For k ← m, i, -1 execute
        s ← s + a[k]
        If s > left then
            left ← s
        EndIf
    EndFor
    s ← 0
    right ← a[m]
    For i ← m, n execute
        s ← s + a[i]
        If s > right then
            right ← s
        EndIf
    EndFor
    Return max(left, right, left + right – a[m])
EndAlgorithm
```

```
Algorithm ceFace(a, i, n):
    If i ≥ n then
        Return a[i]
    EndIf
    m ← (i + n) DIV 2
    v1 ← ceFace(a, i, m - 1)
    v2 ← ceFace(a, m + 1, n)
    v3 ← intermediar(a, i, m, n)
    Return max(v1, v2, v3)
EndAlgorithm
```

Specify which of the following statements are true if the algorithm is called with `ceFace(a, i, n)`:

A. The algorithm identifies a position ***m*** of array ***a*** such that either the sum of all the elements on positions 1, 2, ..., ***m***, either the sum of all the elements on positions ***m***, ***m*** + 1, ..., ***n*** be the maximum that can be obtained for any $1 \le m \le n$, and returns the maximum sum that is obtained like this.
B. The algorithm returns the maximum sum that can be obtained by summing the elements of a subset of the values of array ***a***.
C. The algorithm returns the maximum sum that can be obtained for a subsequence of array ***a***.
D. In case that all the elements of array ***a*** are positive, the algorithm returns the sum of all the elements of array ***a***.

BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – September 8th, 2023

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

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