BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – September 9th, 2021

Written Exam for Computer Science

1. Let us consider the subalgorithm `what(n)`, where ***n*** is a natural number ($1 \le n \le 10000$).

```
Subalgorithm what(n):
    nr ← 0
    For d ← 1, n execute
        If n MOD d = 0 then
            nr ← nr + 1
        EndIf
    EndFor
    If nr = 2 then
      return true
    else
      return false
    EndIf
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns *true* if ***n*** is odd.
B. The subalgorithm returns *true* if ***n*** is even.
C. The subalgorithm returns *true* if ***n*** is prime.
D. The subalgorithm returns *true* if ***n*** is a perfect square.

2. Given that $x < y$ (x and y are real numbers), which of the following expressions have the value *true* if and only if the value of ***t*** (***t*** is a real number) DOES NOT belong to the interval (x, y)?

A. `(t > x) OR (t < y)`
B. `(t ≤ x) OR (t ≥ y)`
C. `(t ≤ x) AND (t ≥ y)`
D. `(t > x) AND (t < y)`

3. Let us consider the subalgorithm `f(n)`, where ***n*** is a natural number ($1 \le n \le 10000$):

```
Subalgorithm f(n):
    r ← 0
    While n > 0 execute
        r ← r + (n MOD 10) * (n MOD 2)
        n ← n DIV 10
    EndWhile
    return r
EndSubalgorithm
```

Select the choices that correctly fill in the underlined section from the subalgorithm below such that both subalgorithms will always return the same value.

```
Subalgorithm fr(n):
    If n > 0 then
        return ________________
    EndIf
    return 0
EndSubalgorithm
```

A. `(n MOD 2) * (n MOD 10) + fr(n DIV 10)`
B. `(n MOD 2) * (n MOD 10) * fr(n DIV 10)`
C. `(n MOD 10) + fr(n DIV 10)`
D. `(n MOD 2) * (n MOD 10) + fr(n MOD 10)`

4. Let us consider the subalgorithm `f(n)` where ***n*** is a natural number ($1 \leq n \leq 10000$).

```
Subalgorithm f(n):
        For i ← 1, n execute
            For j ← 1, 2 * i - 1 execute
               write '*'
            EndFor
        EndFor
EndSubalgorithm
```

Which of the following statements are true?

A. For ***n*** = 3 the subalgorithm prints 3 stars
B. For ***n*** = 3 the subalgorithm prints 9 stars
C. In order for the subalgorithm to print 1154 stars, the value of ***n*** must be 34
D. In order for the subalgorithm to print 289 stars, the value of ***n*** must be 17

5. The subalgorithm below has as input parameters an array ***v*** of ***n*** natural numbers (***v***[1], ***v***[2], …, ***v***[***n***]) and the whole number ***n*** ($2 \leq n \leq 10000$). The / operator denotes real division (e.g. `3/2=1.5`). Array ***v*** contains at least one even number and at least one odd number.

```
Subalgorithm fn(v, n):
    a ← 0
    b ← 0
    For i ← 1, n execute
        If v[i] MOD 2 = 0 then
            a ← a + v[i]
            b ← b + 1
        EndIf
    EndFor
    return a / b
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns the number of even elements from the array ***v***
B. The subalgorithm returns the average of the even elements from the array ***v***
C. The subalgorithm returns the sum of the even elements from the array ***v***
D. The subalgorithm returns the average of the odd elements from the array ***v***

6. The subalgorithm below has as input parameters an array ***v*** of ***n*** natural numbers (***v***[1], ***v***[2], …, ***v***[***n***]) and the whole number ***n*** ($1 \leq n \leq 10000$).

```
Subalgorithm fn(v, n):
    a ← 0
    For i ← 1, n execute
        ok ← 1
        b ← v[i]
        While (b ≠ 0) AND (ok = 1) execute
            If b MOD 2 = 0 then
                ok ← 0
            EndIf
            b ← b DIV 10
        EndWhile
        If ok = 1 then
            a ← a + v[i]
        EndIf
    EndFor
    return a
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns the sum of the odd elements from the array ***v***
B. The subalgorithm returns the sum of the elements from the array ***v*** that are powers of 2
C. The subalgorithm returns the sum of the elements from the array ***v*** that are composed of only even digits
D. The subalgorithm returns the sum of the elements from the array ***v*** that are composed of only odd digits

7. Which of the following subalgorithms compute the absolute value of an integer? We will assume that a logical expression has the value 1 if it is true and 0 if it is false.

A.
```
Subalgorithm modul(n):
    return n * (-2 * (n < 0) + 1)
EndSubalgorithm
```

B.
```
Subalgorithm modul(n):
    If n < 0 then
        return n * (-1)
    EndIf
    return n
EndSubalgorithm
```

C.
```
Subalgorithm modul(n):
    If n < 0 then
        return n * (-1)
    else
        return n
    EndIf
EndSubalgorithm
```

D.
```
Subalgorithm modul(n):
    If n > 0 then
        return n * (-1)
    else
        return n
    EndIf
EndSubalgorithm
```

8. What is the value of the expression below, if ***x*** = 15 and ***y*** = 17?

```
(NOT (x MOD 10 = 0)) AND (y MOD 2 = 0) AND (x < y)
```

A. *true* B. *false* C. Error D. The expression cannot be evaluated

9. Let us consider the recursive subalgorithm `what(n, i)`, where ***n*** is a natural number ($2 \le n \le 1000$).

```
Subalgorithm what(n, i):
    If i = 1 then
        return i
    else
        If n MOD i = 0 then
            return i + what(n, i - 1)
        else
            return what(n, i - 1)
        EndIf
    EndIf
EndSubalgorithm
```

Which of the following statements are true when calling `what(n, n)`?

A. The subalgorithm returns the successor of the greatest divisor of ***n***
B. The subalgorithm returns the sum of the non-prime natural numbers up to and including ***n***
C. The subalgorithm returns the sum of the proper divisors of ***n***
D. The subalgorithm returns the sum of the proper and improper divisors of ***n***

10. The subalgorithm `magic(s, n)` has as input parameters an array ***s*** of ***n*** characters (***s***[1], ***s***[2], …, ***s***[***n***]) and the whole number ***n*** ($1 \le n \le 10000$).

```
Subalgorithm magic(s, n):
    i ← 1
    f ← 1
    While i ≤ n DIV 2 execute
        If s[i] ≠ s[n - i + 1] then
            f ← 0
        EndIf
        i ← i + 1
    EndWhile
    return f
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns 1 if ***s*** has an even number of characters.
B. The subalgorithm returns 1 if ***s*** has an odd number of characters.
C. The subalgorithm returns 1 if ***s*** is a palindrome.
D. The subalgorithm returns 1 if ***s*** contains only distinct characters.

11. Which of the following expressions have the value *true* if and only if ***x*** is an odd and negative number? We denote with |***x***| the absolute value of ***x***.

A. `(|x| MOD 2 = 1) AND (x < 0)`
B. `NOT ((|x| MOD 2 = 0) AND (x ≥ 0))`
C. `NOT ((|x| MOD 2 = 0) OR (x ≥ 0))`
D. `(|x| MOD 2 ≠ 0) OR (x < 0)`

12. The subalgorithm `what(n)` has as input parameter the natural number ***n*** ($0 \le n \le 10000$).

```
Subalgorithm what(n):
    s ← 0
    While n > 0 execute
       c ← n MOD 10
       If c MOD 2 ≠ 0 then
           s ← s + c
       EndIf
       n ← n DIV 10
    EndWhile
    return s
EndSubalgorithm
```

What will the call `what(1234)` return?

A. 4 B. 10 C. 60 D. 0

13. Let us consider an array of characters and a function ***f*** which receives a character parameter and returns 1 if it is a digit and 0 otherwise. Which of the following approaches determines if the array of characters is composed only of digits?

A. Check if the function ***f***, applied on each character of the array of characters always returns 1.
B. Check if the sum of the values returned by ***f***, applied on each character of the array of characters, is equal to the length of the array of characters.
C. Check if the function ***f***, applied on each character of the array of characters, returns 1 at least once.
D. Apply the function ***f*** on randomly chosen characters from the array of characters until the number of returned values equal to 1 is equal to the length of the array.

14. Which of the following algorithms can be implemented such that they have a linear time complexity (O(***n***))?

A. The sequential search of an element in an array of ***n*** numbers.
B. The insertion sort algorithm applied on a one-dimensional array of ***n*** numbers.
C. The algorithm that searches for the maximum number in an unsorted array of ***n*** numbers.
D. The algorithm that computes the sum of the elements from the main diagonal of a square matrix with ***n*** rows and ***n*** columns.

15. Let us consider the subalgorithm `f(a, b)`, where ***a*** and ***b*** are natural numbers ($1 \le a, b \le 10000$).

```
Subalgorithm f(a, b):
    m ← a
    While b MOD m > 0 execute
        m ← m + 1
    EndWhile
    return m
EndSubalgorithm
```

For which of the following calls will the body of the `While` loop be executed at most once?

A. `f(10, 11)` B. `f(10, 10)` C. `f(10, 9)` D. `f(10, 15)`

16. Let us consider the subalgorithm `f(a, b)`, where ***a*** and ***b*** are natural numbers ($1 \le a, b \le 10000$).

```
Subalgorithm f(a, b):
    c ← 0
    d ← 0
    p ← 1
    While a + b + c > 0 execute
        c ← a MOD 10 + b MOD 10 + c
        d ← d + (c MOD 10) * p
        p ← p * 10
        a ← a DIV 10
        b ← b DIV 10
        c ← c DIV 10
    EndWhile
    return d
EndSubalgorithm
```

What will the `f(493, 1836)` call return?

A. 2329 B. 2229 C. 2430 D. 3292

17. Let us consider the subalgorithm `print(M, n)` which receives as parameter an array ***M*** of ***n*** (***n*** ≤ 10) whole numbers (***M***[1], ***M***[2], …, ***M***[***n***]) representing a set.

```
Subalgorithm print(M, n):
    nr ← 2^n
    k ← 0
    While k < nr execute
       curent ← k
       write '{'
       For j = 1, n execute
           r ← curent MOD 2
           curent ← curent DIV 2
           If r = 1 then
              write M[j]
           EndIf
       EndFor
       write '}'
       write new line
       k ← k + 1
    EndWhile
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm prints all the permutations of the set ***M***.
B. The subalgorithm prints all the combinations of the elements of the set ***M*** taken ***i*** at a time, where ***i*** = 0, 1, ..., ***n*** (not necessarily in this order).
C. The subalgorithm prints all the permutations of the elements of the set ***M*** taken ***i*** at a time, where ***i*** = 0, 1, ..., ***n*** (not necessarily in this order).
D. The subalgorithm prints all the subsets of the set ***M***.

18. Let us consider subalgorithm `s(a, b, c)`, where ***a***, ***b***, ***c*** are positive natural numbers ($1 \le a, b, c \le 10000$).

```
Subalgorithm s(a, b, c):
    If (a = 1) OR (b = 1) OR (c = 1) then
        return 1
    else
        If a > b then
            return a * s(a - 1, b, c)
         else
             If a < b then
                return b * s(a, b - 1, c)
            else
                return c * s(a - 1, b - 1, c - 1)
            EndIf
        EndIf
    EndIf
EndSubalgorithm
```

Which of the following statements are true if ***a*** = ***b*** and ***a*** < ***c***:

A. The subalgorithm computes and returns ***c***!
B. The subalgorithm computes and returns ***c***! / (***c*** – ***a*** + 1)!
C. The subalgorithm computes and returns ***c***! / (***c*** – ***a*** – 1)!
D. The subalgorithm computes and returns the number of permutations of ***c*** taken (***a*** - 1) at a time

19. The following subalgorithm has as input parameters the array ***A*** of ***n*** natural numbers (***A***[1], ***A***[2], …, ***A***[***n***]) and the natural number ***n*** ($1 \le n \le 10000$). For the natural numbers ***x*** and ***y***, ***x***^***y*** has the meaning ***x*** raised to the power of ***y*** (x^y).

```
Subalgorithm h(A, n):
    If n = 0 then
        return 0
    else
        return A[n] * (-1)^(1 - A[n] MOD 2) + h(A, n - 1)
   EndIf
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns the difference between the sum of the elements on even positions and the sum of the elements on odd positions in array ***A***.
B. The subalgorithm returns the difference between the sum of the even elements and the sum of the odd elements of array ***A***.
C. The subalgorithm returns the difference between the sum of the odd elements and the sum of the even elements of array ***A***.
D. None of the other answers are correct.

20. An Excel file contains $\boldsymbol{n}$ records numbered from 1 to $\boldsymbol{n}$. These records are copied into a Word file, where the records will be arranged in maximum $\boldsymbol{r}$ rows and exactly $\boldsymbol{c}$ columns on each page. It is guaranteed that the value of $\boldsymbol{n}$ will always allow arranging the records in exactly **c** columns.

Let us denote by $\boldsymbol{x_1}, ..., \boldsymbol{x_c}$ the number of records that are copied on each column for a specific page.

The first page of the Word document includes a header, so the number of rows is $\boldsymbol{r_1}$, $\boldsymbol{r_1} < \boldsymbol{r}$ (the number of rows on the first page is smaller), namely $\boldsymbol{x_p} = \boldsymbol{r_1}$, $\forall\ 1 \leq \boldsymbol{p} \leq \boldsymbol{c}$.

The records will be arranged in the Word file on each page starting from top to bottom on each column, the columns being filled in from the left hand side to the right hand side: if the first record on some page is numbered $\boldsymbol{i}$, the record numbered as ($\boldsymbol{i}$ + 1) will be placed below it, while the record numbered as ($\boldsymbol{i}$ + $\boldsymbol{x_1}$) will be the first record on the second column of the same page, and so on.

On the last page of the Word document, it is required that the number of records is balanced across the columns, meaning that the difference between the number of records on any two columns is at most 1 ($|\boldsymbol{x_j} - \boldsymbol{x_k}| \leq 1$, $\forall\ 1 \leq \boldsymbol{j}, \boldsymbol{k} \leq \boldsymbol{c}, \boldsymbol{j} \neq \boldsymbol{k}$).

For all other pages (except for the first and the last page) $\boldsymbol{x_p} = \boldsymbol{r}$, $\forall\ 1 \leq \boldsymbol{p} \leq \boldsymbol{c}$.

For $\boldsymbol{n}$ = 5883, $\boldsymbol{r}$ = 46, $\boldsymbol{r_1}$ = 12 and $\boldsymbol{c}$ = 2, on which row of the page will the last record of the document be placed (the record numbered with $\boldsymbol{i}$ = 5883)?

A. 29 B. 30 C. 31 D. 32

21. Let us consider the subalgorithm `compute(a, b, c, d, e)`, which receives as parameters five whole numbers $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$ and $\boldsymbol{e}$ ($1 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10000$, $2 \leq \boldsymbol{c} \leq 16$, $1 \leq \boldsymbol{d} < \boldsymbol{c}$).

```
Subalgorithm compute(a, b, c, d, e):
    If a = 0 AND b = 0 then
        If e = 0 then
            return 1
        else
            return 0
        EndIf
    EndIf
    If (a MOD c = d) AND (b MOD c = d) then
        return compute(a DIV c, b DIV c, c, d, e)
    EndIf
    If a MOD c = d then
        return compute(a DIV c, b DIV c, c, d, e + 1)
    EndIf
    If b MOD c = d then
        return compute(a DIV c, b DIV c, c, d, e - 1)
    else
        return compute(a DIV c, b DIV c, c, d, e)
    EndIf
EndSubalgorithm
```

Which of the following statements are true for the call `compute(a, b, c, d, 0)`?

A. Returns 1 if the base $\boldsymbol{c}$ representations of the numbers $\boldsymbol{a}$ and $\boldsymbol{b}$ contain the digit $\boldsymbol{d}$ the same number of times, 0 otherwise
B. Returns 1 if the digit $\boldsymbol{d}$ occurs in the base $\boldsymbol{c}$ representation of the number $\boldsymbol{a}$ and in the base $\boldsymbol{c}$ representation of the number $\boldsymbol{b}$, 0 otherwise
C. Returns 1 if the digit $\boldsymbol{d}$ occurs in the base $\boldsymbol{c}$ representation of the number $\boldsymbol{a}$ or in the base $\boldsymbol{c}$ representation of the number $\boldsymbol{b}$, 0 otherwise
D. Returns 1 if the digit $\boldsymbol{d}$ does not occur in the base **c** representations of $\boldsymbol{a}$ and $\boldsymbol{b}$, 0 otherwise

22. Let us consider the subalgorithms `val(p, s, i, n, x)` and `val_exp(p, n, x)` with the following parameter specification: an array ***p*** of ***n*** whole numbers (***p***[1], ***p***[2], …, ***p***[***n***]), the natural numbers ***s***, ***i*** and ***n*** (***n*** ≤ 1000, ***n*** = 2^k, ***k*** < 10), and the real number ***x***. The values of the array ***p*** represent the coefficients of the expression in increasing order of the exponents, the maximum exponent being equal to ***n*** - 1, for an expression of the form $p[1] + p[2] \cdot x + p[3] \cdot x^2 + ... + p[n] \cdot x^{n-1}$

E.g: ***p*** = [1, 2, 3, 4] corresponds to the expression $E(x)=1 + 2x + 3x^2 + 4x^3$.

```
Subalgorithm val(p, s, i, n, x):
    If s + i ≤ n then
        ______________________
    else
            return p[s]
    EndIf
EndSubalgorithm

Subalgorithm val_exp (p, n, x):
    return val(p, 1, 1, n, x)
EndSubalgorithm
```

Which of the following choices correctly fill in the underlined section such that the subalgorithm `val_exp(p, n, x)` returns the value of the expression ***E***(***x***)?

```
A. return p[s] + x * val(p, s + i, i * 2, n , x * x)
B. return val(p, s, i * 2, n - i, x * x) + x * val(p, s + i, i * 2, n, x * x)
C. return val(p, s + i, i * 2, n, x * x) + x * val(p, s, i * 2, n - i, x * x)
D. return p[s] + x * val(p, s + i, i, n, x)
```

23. Let us consider the subalgorithm `f(a)`, which receives as parameter a natural number ***a*** (2 ≤ ***a*** < 1000000) and returns *true* if a natural number ***d***, 1 < ***d*** < ***a*** exists such that ***d*** divides ***a***, and *false* otherwise. We use [x] to represent the integer part of the number x.

Which of the following versions of the subalgorithm `f(a)` are correct?

A.

```
Subalgorithm f(a):
    If a = 2 then
        return false
    EndIf
    If a MOD 2 = 0 then
        return true
    EndIf
    For d ← 3, [√a] - 1, 2 execute
        If a MOD d = 0 then
            return true
        EndIf
    EndFor
    return false
EndSubalgorithm
```

B.

```
Subalgorithm f(a):
    For d ← 2,[√a] execute
        If a MOD d = 0 then
            return true
        EndIf
    EndFor
    return false
EndSubalgorithm
```

C.

```
Subalgorithm f(a):
    If a ≤ 2 then
        return false
    EndIf
    If a MOD 2 = 0 then
        return true
    EndIf
    For d ← 3, [√a], 2 execute
        If a MOD d = 0 then
            return true
        EndIf
    EndFor
    return false
EndSubalgorithm
```

D.

```
Subalgorithm f(a):
     d ← a - 1
     While true execute
         If a MOD d = 0 then
             return true
         EndIf
         d ← d - 1
     EndWhile
     return false
EndSubalgorithm
```

24. Consider the expression below, where $1 < \boldsymbol{A} < 2021$ and $1 < \boldsymbol{n} < 10202110$.

$$\boldsymbol{E}(\boldsymbol{A}, \boldsymbol{n}) = (\boldsymbol{A} + \boldsymbol{A}^2 + A^3 + \ldots + \boldsymbol{A}^{\boldsymbol{n}}) \textbf{ MOD } 2021$$

Which of the following subalgorithms correctly compute the value of `E(A, n)` and have the specified time complexity?

All computations are carried out on 32 bit data types. The computation of $\boldsymbol{x}^{\boldsymbol{k}}$ is assumed to be done in O(log $\boldsymbol{k}$).

A.

```
Subalgorithm E(A, n):
    return (A * (Aⁿ - 1) DIV (A - 1)) MOD 2021
EndSubalgorithm
```

Time complexity: O(log $\boldsymbol{n}$)

B.

```
Subalgorithm E(A, n):
    return ((A * (Aⁿ - 1)) MOD 2021) DIV ((A - 1) MOD 2021)
EndSubalgorithm
```

Time complexity: O(log $\boldsymbol{n}$)

C.

```
Subalgorithm E1(A, n):
    If n = 1 then
        return (A, A) //returns a pair of values
    EndIf
    If n MOD 2 = 1 then
        (t1, t2) ← E1(A, n - 1)
        p ← (t1 * A) MOD 2021
        return (p, (p + t2) MOD 2021)
    else
        (t1, t2) ← E1(A, n DIV 2)
        p ← (t1 * t1) MOD 2021
        return (p, ((1 + t1) * t2) MOD 2021)
    EndIf
EndSubalgorithm

Subalgorithm E(A, n):
    (aux1, aux2) ← E1(A, n)
    return aux2
EndSubalgorithm
```

Time complexity: O(log $\boldsymbol{n}$)

D.

```
Subalgorithm E(A, n):
    raspuns ← A
    For i = 2, n execute
        raspuns ← raspuns + Aⁱ
    EndFor

    return raspuns MOD 2021
EndSubalgorithm
```

Time complexity: O($\boldsymbol{n} \cdot \log \boldsymbol{n}$)

25. All numbers from 1 to 1000 are written on a circle in ascending order, clockwise. Starting from 1 and moving in clockwise order, we color each ***k***-th number (1, ***k*** + 1, 2 · ***k*** + 1, ...). The process continues until we reach an already colored number, at the end having ***x*** colored numbers. Which of the following statements are true?

A. if ***k*** = 15 then ***x*** = 300
B. if ***k*** = 45 then ***x*** = 200
C. if ***k*** = 25 then ***x*** = 40
D. if ***k*** = 30 then ***x*** = 150

26. Consider the subalgorithm `what(n, k)`, where ***n*** and ***k*** are natural numbers (1 ≤ ***n***, ***k*** ≤ 1000000).

```
Subalgorithm what(n, k):
    nr ← 0
    p ← 1
    While (n ≠ 0) AND (k ≠ 0) execute
        if n MOD 2 ≠ 0 then
            nr ←  nr + ((n DIV 10) MOD 10) * p
            p ← p * 10
        else
            k ← k - 1
       EndIf
       n ← n DIV 10
    EndWhile
    return nr
EndSubalgorithm
```

Which of the following pairs of calls return identical values?

A. `what(32345, 3)` and `what(321458, 7)`
B. `what(321458, 4)` and `what(2314587, 4)`
C. `what(2314, 3)` and `what(23145, 4)`
D. `what(23145, 3)` and `what(231458, 4)`

27. Consider the subalgorithms:

- `putere(b, p)` – computes b^p (***b*** to the power ***p***), ***b***, ***p*** – natural numbers (1 ≤ ***b*** ≤ 20, 1 ≤ ***p*** ≤ 20);
- `nrCifre(nr)` – returns the number of digits of the non-zero natural number ***nr*** (0 < ***nr*** ≤ 1000000), or the value 0 when ***nr*** = 0;
- `produs(st, dr)` – the subalgorithm from below, where ***st***, ***dr*** – natural numbers (100 < ***st*** < 1000000, 0 ≤ ***dr*** < 1000000, ***st*** – a number which, in its base 10 representation, has at least two digits different from 0).

```
Subalgorithm produs(st, dr):
    If st > 0 then
        drCrt ← ____________________________
        stCrt ← st DIV 10
        If st * dr < stCrt * drCrt then
            return produs(stCrt, drCrt)
        else
            return st * dr
        EndIf
    else
        return st * dr
    EndIf
EndSubalgorithm
```

Which of the following versions correctly fill in the underlined section such that the subalgorithm `produs(st, dr)`, by executing the following sequence of instructions

```
write produs(1092, 0)
write produs(75981, 0)
```

prints 920 and 73575?

A. `(st MOD 10) * putere(10, nrCifre(dr)) + dr`
B. `(st MOD 10) * putere(10, dr) + dr`
C. `(st MOD 10) * putere(10, nrCifre(dr))`
D. `(st MOD 10) * nrCifre(dr)`

28. Let us consider the subalgorithm `what(a, n, i, f)`, which receives as parameters an array ***a*** of ***n*** whole numbers (***a***[1], ***a***[2], …, ***a***[***n***]) and the whole numbers ***i, f*** and ***n*** ($2 \le n \le 10000$).

```
Subalgorithm what(a, n, i, f):
    If (i = n) AND (f = 2) then
        return true
    else
        If (i = n) then
            return false
        else
            If (f ≤ 1) AND (a[i] < a[i + 1]) then
                return what(a, n, i + 1, 1)
            EndIf
            If (1 ≤ f) AND (a[i] > a[i + 1]) then
                return what(a, n, i + 1, 2)
            EndIf
            return false
        EndIf
   EndIf
EndSubalgorithm
```

Which of the following statements are true, given the initial call `what(a,n,1,0)`?

A. The subalgorithm returns *true* if and only if the maximum of array ***a*** is located on a position ***i***, $1 < i < n$.
B. The subalgorithm returns *true* if and only if $\exists\, k$, $(1 < k < n)$, such that $a[1] < a[2] < \ldots < a[k] > a[k+1] > \ldots > a[n]$.
C. The subalgorithm returns *false* if ***a*** is strictly increasing.
D. The subalgorithm returns *true* if and only if $\exists\, k$, $(1 < k < n)$, such that $a[k] > a[k+1] > \ldots >$ a[n].

29. Let us consider the following subalgorithm, with the input parameter the non-zero natural number ***n*** and which returns a natural number.

```
Subalgorithm f(n):
    j ← n
    While j > 1 do
       i ← 1
       While i ≤ n^4 do
           i ← 4 * i
       EndWhile
       j ← j DIV 2
    EndWhile
    return j
EndSubalgorithm
```

To which of the following complexity classes does the time complexity of the subalgorithm belong?

A. $O(\log_2 n^2)$
B. $O(\log_2^2 n^2)$
C. $O(\log_4^2 n)$
D. $O(\log_2 \log_4 n)$

30. Consider an array ***s*** of ***n*** characters from the English alphabet, (***s***[1], ***s***[2], …, ***s***[***n***]). We would like to find its longest suffix that is a palindrome. A suffix of an array of characters is a subarray of the given array that contains the last character. For example, for the array *abab*, the longest palindrome suffix is *bab*.

We assume that we have the following subalgorithm:

- `ascii(c)` - returns the ASCII code of character ***c***.

We assume that the arithmetic operations on integers do not overflow.
Which of the following implementations return the length of this suffix when calling `sufix(s, n)`?

A.

```
Subalgorithm sufix(s, n):
    hf ← 0
    hb ← 0
    raspuns ← 1
    For i ← n, 1, -1 execute
        hf ← ascii(s[i]) + 2021 * hf
        hb ← hb + ascii(s[i]) * 2021^(n - i)

        If hf = hb then
            raspuns ← n – i + 1
        EndIf
    EndFor
    return raspuns
EndSubalgorithm
```

B.

```
Subalgorithm sufix(s, n):
    hf ← 0
    hb ← 0
    raspuns ← 1
    For i ← n, 1, -1 execute
        hf ← ascii(s[i]) + 3 * hf
        hb ← hb + ascii(s[i]) * 3^(n - i)

        If hf = hb then
            raspuns ← n – i + 1
        EndIf
    EndFor
    return raspuns
EndSubalgorithm
```

C.

```
Subalgorithm sufix(s, n):
    hf ← 0
    hb ← 0
    raspuns ← 1
    For i ← n, 1, -1 execute
        hf ← ascii(s[i]) + 2021 * hb
        hb ← hf + ascii(s[i]) * 2021^(n - i)

        If hf = hb then
            raspuns ← n – i + 1
        EndIf
    EndFor
    return raspuns
EndSubalgorithm
```

D. None of the other versions are correct.

BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – September 9th, 2021
Written Exam for Computer Science
GRADING AND SOLUTIONS

DEFAULT: 10 points

1	C	3 points
2	B	3 points
3	A	3 points
4	B, D	3 points
5	B	3 points
6	D	3 points
7	A, B, C	3 points
8	B	3 points
9	D	3 points
10	C	3 points
11	A, C	3 points
12	A	3 points
13	A, B	3 points
14	A, C, D	3 points
15	A, B	3 points
16	A	3 points
17	B, D	3 points
18	B, D	3 points
19	C	3 points
20	C, D	3 points
21	A	3 points
22	B, D	3 points
23	B, C	3 points
24	C	3 points
25	B, C	3 points
26	A, D	3 points
27	A	3 points
28	B, C	3 points
29	B, C	3 points
30	A	3 points