BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 19th, 2021

Written Exam for Computer Science

1. Let us consider the following subalgorithm, with the input parameter the natural number n and that returns a natural number.

```
Subalgorithm compute(n):
    E ← 1
    P ← 1
    i ← 2
    While i ≤ n do
        P ← (-1) * P * i
        E ← E + P
        i ← i + 1
    EndWhile
    return E
EndSubalgorithm
```

What is the value returned by the subalgorithm, considering that $n \geq 1$?

A. $1! - 2! + 3! - 4! + ... + (-1)^{n+1} \cdot n!$
B. $1 - 1! + 2! - 3! + ... + (-1)^{n} \cdot n!$
C. $1 - 1 \cdot 2 + 1 \cdot 2 \cdot 3 - 1 \cdot 2 \cdot 3 \cdot 4 + ... + (-1)^{n+1} \cdot 1 \cdot 2 \cdot ... \cdot n$
D. $1 + 1 \cdot 2 - 1 \cdot 2 \cdot 3 + 1 \cdot 2 \cdot 3 \cdot 4 + ... + (-1)^{n} \cdot 1 \cdot 2 \cdot ... \cdot n$

2. An Excel file contains n records, numbered from 1 to n. These records are copied into a Word file, where the records have to be arranged into r rows and c columns on each page (except the first and the last page). The first page of the Word document has a header, therefore its number of rows is r_1, $r_1 < r$ (the number of rows on the first page is smaller).

The records will be arranged into the Word file on each page starting from top to bottom on each column, the columns being filled in from left to right: if the first record on some page is numbered i, then the record numbered $(i + 1)$ will be placed below it; the record numbered $(i + r)$ will be the first record on the column 2 on that page, and so on.

Considering that $n = 5000$, $r = 46$, $r_1 = 12$ and $c = 2$, on what page and column of the Word document will the record having the order number $i = 3245$ be placed?

A. Page 36, last column
B. Page 37, first column
C. Page 37, last column
D. Page 38, first column

3. Let us consider the subalgorithm `whatDoesItDo(m)`, where m is a natural number ($10 \leq m \leq 10000$).

```
Subalgorithm whatDoesItDo(m):
    If m = 0 then
        return 0
    EndIf
    If m MOD 9 = 0 then
        return 9
    EndIf
    return m MOD 9
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns the remainder of the division of number ***m*** by 9.
B. The subalgorithm returns the number of divisors of number ***m*** that are divisible by 9.
C. The subalgorithm returns the control digit of number ***m*** (the sum of its digits, then the digit sum of this sum, until the obtained sum is formed out of a single digit).
D. The subalgorithm returns the control digit of number ***m*** (the sum of its digits, then the digit sum of this sum, until the obtained sum is formed out of a single digit) if and only if number ***m*** is divisible by 9.

4. In order to generate the numbers with ***n*** digits composed only of the digits 0, 2, 9, one uses an algorithm which, for ***n*** = 2, generates in increasing order the numbers 20, 22, 29, 90, 92, 99.

If ***n*** = 4 and the same algorithm is used, which number is generated immediately after 2009?

A. 2022
B. 2090
C. 2010
D. None of the other choices

5. Let us consider the subalgorithm `search(n)`, where ***n*** is a natural number (0 ≤ ***n*** ≤ 1000000).

```
Subalgorithm search(n):
    v ← 0
    If n = 0 then
        return 1
    else
        m ← n
        While m > 0 do
            If m MOD 10 = 0 then
                v ← v + 1
            EndIf
            m ← m DIV 10
        EndWhile
        return v
   EndIf
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm computes and returns the number of digits of ***n***.
B. The subalgorithm returns 1 if ***n*** is a power of 10 and 0 otherwise.
C. The subalgorithm returns 1 if ***n***'s last digit is 0 and 0 otherwise.
D. The subalgorithm computes and returns the number of digits 0 of ***n***.

6. Let us consider the subalgorithm `abc(a, n, p)`, where ***n*** is a natural number (1 ≤ ***n*** ≤ 10000), ***p*** is a whole number (-10000 ≤ ***p*** ≤ 10000) and ***a*** is an array of ***n*** non-zero natural numbers (***a***[1], ***a***[2], …, ***a***[n]).

```
Subalgorithm abc(a, n, p):
    If n < 1 then
        return 0
    else
        If (1 ≤ p) AND (p ≤ n) then
            return a[p]
        else
            return -1
        EndIf
    EndIf
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns -1 if and only if ***p*** is negative or greater than ***n***.
B. The subalgorithm returns the element at position ***p*** if ***p*** is strictly greater than 0 and less than or equal to the array's length.
C. The subalgorithm never returns 0 for parameter values that meet the preconditions from the statement.
D. The subalgorithm returns the element at position ***p*** if ***p*** is greater than or equal to 0 and strictly smaller than the array's length. In case ***p*** is not between 1 and ***n***, the subalgorithm returns -1.

7. Which of the following sequences determines in variable ***i*** the length of an array of characters that ends with character '*' (star)? The index of the first character is 1 and the star character is part of the character array.

A.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
```
B.
```
i ← 1
While x[i] = '*' do
    i ← i + 1
EndWhile
i ← i - 1
```
C.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
i ← i + 1
```
D.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
i ← i - 1
```

8. 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 do
            i ← 2 * i
        EndWhile
        j ← j DIV 3
    EndWhile
    return j
EndSubalgorithm
```

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

A. $O(\log_2 \boldsymbol{n})$
B. $O(\log_2^2 \boldsymbol{n})$
C. $O(\log_3^2 \boldsymbol{n})$
D. $O(\log_2 \log_3 \boldsymbol{n})$

9. The subalgorithm `howMany(n, m)` has as input parameters the natural numbers ***n*** and ***m***.

```
Subalgorithm howMany(n, m):
    If n ≤ m then
        If (n MOD 2 = 0) AND (n MOD 3 ≠ 0) then
             return 1 + howMany(n + 1, m)
        else
             return howMany(n + 1, m)
        EndIf
    else
       return 0
    EndIf
EndSubalgorithm
```

Which of the following statements are true?

A. If ***n*** = 0 and ***m*** = 1, the subalgorithm returns the value 0.
B. If ***n*** = 4 and ***m*** = 21, the subalgorithm returns the value 6.
C. If ***n*** = 7 and ***m*** = 120, the subalgorithm returns the value 36.
D. If ***n*** = 1 and ***m*** = 215, the subalgorithm returns the value 72.

10. Let us consider the subalgorithm `verify(n)`, where ***n*** is a natural number (1 ≤ ***n*** ≤ 100000).

```
Subalgorithm verify(n):
  While n > 0 do
    If (n MOD 3) > 1 then
      return 0
    EndIf
    n ← n DIV 3
  EndWhile
  return 1
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns 1 if ***n*** is a power of 3 and 0, otherwise.
B. The subalgorithm returns 1 if ***n***'s representation in base 3 contains only the digits 0 and/or 1; 0, otherwise.
C. The subalgorithm returns 1 if ***n*** can be written as a sum of distinct powers of 3; 0, otherwise.
D. The subalgorithm returns 1 if ***n***'s representation in base 3 contains only the digit 2; 0, otherwise.

11. For a natural number ***nr*** (1000 ≤ ***nr*** ≤ 1000000), the decrementation operation is defined as follows: if the last digit of ***nr*** is not 0, then we subtract 1 from ***nr***, otherwise, we divide ***nr*** by 10 and keep the integer part of it. Which of the following subalgorithms, when calling `decrement(nr, k)`, returns the number obtained by applying the decrementation operation ***k*** times (0 ≤ ***k*** ≤ 100) on number ***nr***? For example, for ***nr*** = 15243 and ***k*** = 10, the result is 151.

A.
```
Subalgorithm decrement(nr, k):
    If k = 0 then
        return nr
    else
        If nr MOD 10 ≠ 0 then
            return decrement(nr DIV 10, k - 1)
        else
            return decrement(nr - 1, k - 1)
        EndIf
    EndIf
EndSubalgorithm
```

B.
```
Subalgorithm decrement(nr, k):
    While k > 0 do
        If nr MOD 10 = 0 then
            nr ← nr DIV 10
        else
            nr ← nr - 1
        EndIf
    EndWhile
    return nr
EndSubalgorithm
```

C.
```
Subalgorithm decrement(nr, k):
    For i ← 1, k do
        If nr MOD 10 > 0 then
            nr ← nr - 1
        else
            nr ← nr DIV 10
        EndIf
    EndFor
    return nr
EndSubalgorithm
```

D.
```
Subalgorithm decrement(nr, k):
    If k = 0 then
        return nr
    else
        If k > nr MOD 10 then
            nr1 ← nr DIV 10
            return decrement(nr1,  k - nr MOD 10 - 1)
        else
            return decrement(nr - k, 0)
        EndIf
    EndIf
EndSubalgorithm
```

12. Let us consider the following subalgorithm, with the input parameters the array ***x*** with ***n*** natural numbers (***x***[1], ***x***[2], …, ***x***[***n***]) and the whole number ***n***.

```
Subalgorithm f(x, n):
    If n = 1 then
        return 100
    else
        If x[n] > f(x, n - 1) then
            return x[n]
        else
            return f(x, n - 1)
        EndIf
    EndIf
EndSubalgorithm
```

What is the result of executing the subalgorithm for ***x*** = [101, 7, 6, 3] and ***n*** = 4?

A. 101
B. 3
C. 100
D. 7

13. The following subalgorithm has as input parameters the array ***a*** with ***n*** natural numbers (***a***[1], ***a***[2], …, ***a***[***n***]) and the natural number ***n*** ($2 \leq n \leq 10000$).

```
Subalgorithm h(a, n):
    If n ≤ 0 then
        return 0
    EndIf
    If (n MOD 2 = 0) AND (a[n] MOD 2 = 0) then
        return h(a, n - 1) + a[n]
    EndIf
    return h(a, n - 1) - a[n]
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns the difference between the sum of elements having the same parity as their position and the sum of elements having different parity than their position in array ***a***.
B. The subalgorithm returns the difference between the sum of even elements on even positions and the sum of odd elements on odd positions of array ***a***.
C. The subalgorithm returns the difference between the sum of the even elements and the sum of the odd elements of array ***a***.
D. The subalgorithm returns the difference between the sum of the even elements from even positions and the sum of the other elements of array ***a***.

14. Let us consider the subalgorithm `whatDoesItDo(n)`, with the parameter ***n*** a non-zero natural number.

```
Subalgorithm whatDoesItDo(n):
    i ← 1
    While n > 0 do
        If n MOD 2 ≠ 0 then
            write i
        EndIf
        i ← i + 1
        n ← n DIV 2
    EndWhile
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm prints the sequence: 12345 for ***n*** = 31.
B. The subalgorithm prints the sequence: 234 for ***n*** = 14.
C. The subalgorithm prints 1 in the beginning of the sequence, when ***n*** is an odd number.
D. The subalgorithm prints a single number for $n = 2^k$, where ***k*** is a natural number.

15. Let us consider the set ***S***, consisting of ***n*** intervals described by the lower boundary l_i and the upper boundary u_i ($l_i < u_i \; \forall \; i = 1...n$). The subset $S' \subseteq S$, consisting of ***m*** elements is built such that there are no two intervals in ***S’*** that intersect one with another and ***m*** has the highest possible value.
Which of the following strategies provide a correct solution for the problem?

A. The intervals from set ***S*** are ordered in ascending order based on their lower boundary. The first interval from the ordered array of intervals is added to ***S’***. The ordered array is navigated following the sorted order and if an interval that does not intersect with the last interval added to ***S’*** is found, then it will be added to ***S’***.
B. The intervals from set ***S*** are ordered in ascending order based on their upper boundary. The first interval from the ordered array of intervals is added to ***S’***. The ordered array is navigated following the sorted order and if an interval that does not intersect with the last interval added to ***S’*** is found, then it will be added to ***S’***.

C. The intervals from set ***S*** are ordered in ascending order based on their length. The first interval from the ordered array of intervals is added to ***S'***. The ordered array is navigated following the sorted order and if an interval that does not intersect with the last interval added to ***S'*** is found, then it will be added to ***S'***.
D. The intervals from set ***S*** are ordered in ascending order based on the number of intervals from ***S*** they intersect with. The first interval from the ordered array of intervals is added to ***S'***. The ordered array is navigated following the sorted order and if an interval that does not intersect with the last interval added to ***S'*** is found, then it will be added to ***S'***.

16. Let us consider the subalgorithm `f(a, b)`, with the input parameters two natural numbers ***a*** and ***b*** ($1 \leq \boldsymbol{a} < \boldsymbol{b} \leq 1000$).

```
Subalgorithm f(a, b):
    m ← 0
    For n ← a, b do
        c ← 0
        For d ← 1, n do
            If n MOD d = 0 then
                c ← c + 1
            EndIf
        EndFor
        If c > m then
            m ← c
        EndIf
    EndFor

    For n ← a, b do
        c ← 0
        For d ← 1, n do
            If n MOD d = 0 then
                c ← c + 1
            EndIf
        EndFor
        If c = m then
            write n
        EndIf
    EndFor
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm prints the maximum between the number of divisors of ***a*** and the number of divisors of ***b***.
B. The subalgorithm prints the natural numbers from interval [***a***, ***b***] that have the greatest number of divisors.
C. The subalgorithm prints the number of divisors for each natural number from interval [***a***, ***b***].
D. The subalgorithm prints the natural numbers from interval [***a***, ***b***] that have the greatest number of proper divisors.

17. Let us consider the natural numbers ***a*** and ***b***, where $\boldsymbol{b} \neq 0$. Which of the following alternatives computes:

- `a DIV b`, if `a MOD b = 0`
- `(a / b)` rounded up to the next whole number, if `a MOD b ≠ 0`

A. `(a - 1) DIV b`
B. `(a + b + 1) DIV b`
C. `(a + b - 1) DIV b`
D. `((a + 2 * b - 1) DIV b) - 1`

18. Johnny has to implement the binary search algorithm of an element $\boldsymbol{a}$ in an array $\boldsymbol{V}$ of $\boldsymbol{n}$ ($1 \leq \boldsymbol{n} \leq 1000$) whole numbers sorted in ascending order ($\boldsymbol{V}[1]$, $\boldsymbol{V}[2]$, …, $\boldsymbol{V}[\boldsymbol{n}]$). He writes the following subalgorithm:

```
Subalgorithm BinarySearch(a, n, V):
     st ← 1
     dr ← n
     While dr – st > 1 execute
          m ← (st + dr) DIV 2
          If a ≤ V[m] then
               dr ← m
          else
               st ← m
          EndIf
     EndWhile
     return dr
EndSubalgorithm
```

Which of the following statements are true?

A. If $\boldsymbol{n} = 1$ then the value returned by the subalgorithm is always 1.
B. For any $\boldsymbol{n} \geq 1$, the subalgorithm written by Johnny returns the value 1 when $\boldsymbol{a}$ is smaller than all the elements of the array.
C. When the element $\boldsymbol{a}$ exists in the array, the subalgorithm written by Johnny DOES NOT always return the position (index from array $\boldsymbol{V}$) at which the element is located.
D. For any $\boldsymbol{n} > 1$, the subalgorithm written by Johnny returns the value $\boldsymbol{n}$ when $\boldsymbol{a}$ is greater than all the elements of the array.

19. Let us consider the subalgorithm `compute(x, n)`, where the input parameters are the natural numbers $\boldsymbol{n}$ and $\boldsymbol{x}$, where $1 \leq \boldsymbol{x} \leq \boldsymbol{n} < 10$.

```
Subalgorithm compute(x, n):
    b ← 1
    For i ← 1, n - x do
        b ← b * i
    EndFor
    a ← b
    For i ← n – x + 1, n do
        a ← a * i
    EndFor
    return a DIV b
EndSubalgorithm
```

Which of the following statements are true?

A. If $\boldsymbol{n} = 5$ and $\boldsymbol{x} = 2$, then the subalgorithm returns 20.
B. If $\boldsymbol{n} = 3$ and $\boldsymbol{x} = 2$, then the subalgorithm returns 6.
C. The subalgorithm returns the cardinal number of the set $\{\overline{c_1 c_2 \ldots c_x} : \boldsymbol{c_i} \neq \boldsymbol{c_j}\ \forall\ 1 \leq \boldsymbol{i}, \boldsymbol{j} \leq \boldsymbol{x}, \boldsymbol{i} \neq \boldsymbol{j}, 1 \leq \boldsymbol{c_i} \leq \boldsymbol{n}\}$
D. The subalgorithm executes $\boldsymbol{n}$ multiplication operations.

20. Let us consider subalgorithm `what(n,k)`, which receives as parameters two non-zero natural numbers $\boldsymbol{n}$ and $\boldsymbol{k}$ ($1 \leq \boldsymbol{n}, \boldsymbol{k} \leq 1000000$).

```
Subalgorithm what(n, k):
    While n ≥ 1 execute
        If k ≤ n then
            i ← k
        else
            i ← n
        EndIf
        n ← n - i
        x ← 1
        While i ≥ 1 execute
            Write x,' '
            x ← x + 1
            i ← i - 1
        EndWhile
    EndWhile
EndSubalgorithm
```

Which of the following statements are true?

A. For ***n*** = 8 and ***k*** = 3 the subalgorithm prints the array 1 2 3 1 2 3 1 2
B. For ***k*** = 2, the smallest value of ***n*** for which the value 1 is printed 3 times is ***n*** = 3.
C. For ***k*** = 5, the smallest value of ***n*** for which the value 2 is printed 37 times is ***n*** = 182.
D. For ***n*** = 7 and ***k*** = 3 the subalgorithm prints 1 2 3 1 2 3

21. Let us consider the subalgorithm `compute(a, b, c)`, with input parameters non-zero natural numbers, that computes the greatest common divisor of the three given numbers.
Which of the following implementations for the subalgorithm are correct?

A.

```
Subalgorithm compute(a, b, c):
    While (a ≠ b) OR (a ≠ c) OR (b ≠ c) do
        x ← a
        If a ≠ x then
            a ← a - x
        EndIf
        If b ≠ x then
            b ← b - x
        EndIf
        If c ≠ x then
            c ← c - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

B.

```
Subalgorithm compute(a, b, c):
    x ← a
    y ← b
    While x ≠ y do
        If x > y then
            x ← x - y
        else
            y ← y - x
        EndIf
    EndWhile
    z ← c
    While x ≠ z do
        If x > z then
            x ← x - z
        else
            z ← z - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

C.

```
Subalgorithm compute(a, b, c):
    While (a ≠ b) OR (a ≠ c) OR (b ≠ c) do
        x ← a
        If b < x then
            x ← b
        EndIf
        If c < x then
            x ← c
        EndIf
        If a ≠ x then
            a ← a - x
        EndIf
        If b ≠ x then
            b ← b - x
        EndIf
        If c ≠ x then
            c ← c - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

D.

```
Subalgorithm compute(a, b, c):
    x ← a
    y ← b
    r ← x MOD y
    While r ≠ 0 do
        x ← y
        y ← r
        r ← x MOD y
    EndWhile
    z ← c
    r ← y MOD z
    While r ≠ 0 do
        y ← z
        z ← r
        r ← y MOD z
   EndWhile
   return z
EndSubalgorithm
```

22. The subalgorithm `whatDoesItDo(n)` has as input parameter the natural number $\boldsymbol{n}$ ($1 \leq \boldsymbol{n} \leq 100$).

```
Subalgorithm whatDoesItDo(n):
    s ← 0
    If n MOD 2 = 0 then
        a ← 1
        While a < n do
            s ← s + a
            a ← a + 2
        EndWhile
    else
        b ← 2
        While b < n do
            s ← s + b
            b ← b + 2
        EndWhile
    EndIf
    return s
EndSubalgorithm
```

Which of the following statements are true?

A. If ***n*** is even, the subalgorithm returns the sum of the natural numbers strictly smaller than ***n***; if ***n*** is odd, it returns the sum of the even natural numbers smaller than ***n***.
B. If ***n*** is even, the subalgorithm returns the sum of the even natural numbers strictly smaller than ***n***; if ***n*** is odd, it returns the sum of the odd natural numbers smaller than ***n***.
C. If ***n*** is even, the subalgorithm returns the sum of the odd natural numbers smaller than ***n***; if ***n*** is odd, it returns the sum of the even natural numbers smaller than ***n***.
D. If ***n*** is even, the subalgorithm returns the sum of the even natural numbers strictly smaller than ***n***; if ***n*** is odd, it returns the sum of the natural numbers strictly smaller than ***n***.

23. The subalgorithm `whatDoesItDo(a)` has the input parameter the natural number ***a*** ($1 \leq a \leq 100000$).

```
Subalgorithm whatDoesItDo(a):
    b ← 0
    c ← 0
    d ← 0
    e ← 1
    While a > 0 do
        d ← a MOD 10
        If (d ≠ 4) AND (d < 7) then
            b ← b + e * (d DIV 2)
            c ← c + e * (d - d DIV 2)
        else
            b ← b + e
            c ← c + e * (d - 1)
        EndIf
        a ← a DIV 10
        e ← e * 10
    EndWhile
    write b
    write c
EndSubalgorithm
```

Which of the following pairs of values will never be printed for any valid input values?

A. 1112 and 11233
B. 1111 and 88888
C. 21001 and 33011
D. 3141 and 3258

24. Let us consider the subalgorithms `f(n, c)` and `g(n, c)`, having the input parameters the natural numbers ***n*** and ***c***.

```
Subalgorithm f(n, c):
    If n ≤ 9 then
        If n = c then
            return 1
        else
            return 0
        EndIf
    else
        If n MOD 10 = c then
            return f(n DIV 10, c) + 1
        else
            return f(n DIV 10, c)
        EndIf
    EndIf
EndSubalgorithm
```

```
Subalgorithm g(n, c):
    If c = 0 then
        return 0
    else
        If f(n, c) > 0 then
            return g(n, c - 1) + 1
        else
            return g(n, c - 1)
        EndIf
    EndIf
EndSubalgorithm
```

What is the result of the call `g(n, 9)`?

A. It returns the number of digits of number ***n***.
B. It returns the number of distinct digits of number ***n***.
C. It returns the number of digits greater than 1 of number ***n***.
D. None of the other answers are correct.

25. On a site each registered user has a secret code of ***n*** digits instead of a password. For logging in, the user does not have to introduce the entire code, instead the site randomly generates 3 distinct positions ***p***1, ***p***2 and ***p***3, such that 1 ≤ ***p***1 < ***p***2 < ***p***3 ≤ n and the user only has to introduce the digits from those positions. For example, if the user's code is 987654321 and the site randomly generates the positions 2, 5 and 7, the user has to introduce the digits 8, 5, 3.
Below, the values introduced by a user for 9 logins are given.
1, 2, 3
2, 9, 0
6, 3, 2
2, 0, 2
1, 4, 7
9, 3, 2
4, 4, 3
4, 3, 1
5, 6, 0
Assuming that all 9 logins are valid and the user code was not changed in the meantime, which of the below statements are correct.

A. The user code surely does not contain the digit 8.
B. The shortest possible code has 12 digits.
C. The shortest possible code contains the digit 2 at least 3 times.
D. The sum of the digits in the shortest possible code may be 44.

26. Let us consider the subalgorithm `f(x, n)` where ***x***, ***n*** are natural numbers and ***x*** > 0.

```
Subalgorithm f(x, n):
    If n = 0 then
        return 1
    EndIf
    m ← n DIV 2
    p ← f(x, m)
    If n MOD 2 = 0 then
        return p * p
    EndIf
    return x * p * p
EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm returns x^n.
B. If we replace "`n MOD 2`" with "`m MOD 2`" then the subalgorithm will return x^n.
C. The statements after the recursive call will never be executed.
D. The subalgorithm returns x^n if and only if ***n*** is even.

27. Let us consider the subalgorithm `f2(a,b)` with the input parameters ***a*** and ***b*** natural numbers and the subalgorithm `f(arr, i, n, p)` with the input parameters the array ***arr*** with ***n*** whole numbers (***arr***[1], ***arr***[2], …, ***arr***[***n***]), and the whole numbers ***i*** and ***p***.

```
Subalgorithm f2(a, b):
    If a > b then
        return a
    else
        return b
    EndIf
EndSubalgorithm

Subalgorithm f(arr, i, n, p):
    If i = n then
        return 0
    EndIf
    n1 ← f(arr, i + 1, n, p)
    n2 ← 0
    If p + 1 ≠ i then
        n2 ← f(arr, i + 1, n, i) + arr[i]
    EndIf
    return f2(n1, n2)
EndSubalgorithm
```

State the result of the call `f(arr, 1, 9, -10)`, if array ***arr*** consists of the values (10, 1, 3, 4, 8, 12, 1, 11, 6).

A. 24
B. 37
C. 26
D. 56

28. Let us consider the subalgorithm `verify(n)`, with the input parameter a whole number ***n*** ($1 \leq n \leq 100000$) that returns *true* if ***n*** contains a digit that is equal to the sum of the other digits. For example, `verify(1517)` returns *true* because 7 = 1 + 5 + 1.
Which of the following alternatives represent correct implementations of the `verify(n)` subalgorithm?

A.
```
Subalgorithm verify(n):
    s ← 0
    c ← n
    r ← false
    While c > 0 do
        s ← s + c MOD 10
        c ← c DIV 10
    EndWhile
    c ← n
    While c > 0 do
        d ← c MOD 10
        If d = s - d then
            r ← true
        else
            r ← false
        EndIf
        c ← c DIV 10
    EndWhile
    return r
EndSubalgorithm
```

B.

```
Subalgorithm verify(n):
    m ← -1
    c ← n
    r ← false
    While c > 0 do
        d ← c MOD 10
        c ← c DIV 10
        If d > m then
            m ← d
        EndIf
    EndWhile
    c ← n
    s ← 0
    While c > 0 do
        d ← c MOD 10
        If d ≠ m then
            s ← s + d
        EndIf
        c ← c DIV 10
    EndWhile
    If s = m then
        r ← true
    EndIf
    return r
EndSubalgorithm
```

C.

```
Subalgorithm verify(n):
    v ← [0,0,0,0,0,0,0,0,0]
    r ← false
    While n > 0 do
        d ← n MOD 10
        If d > 0 then
            v[d] ← v[d] + 1
        EndIf
        n ← n DIV 10
    EndWhile
    m ← 9
    While v[m] = 0 do
        m ← m - 1
    EndWhile
    If v[m] = 1 then
        d ← m
        s ← 0
        m ← m - 1
        While m > 0 do
            s ← s + v[m] * m
            m ← m - 1
        EndWhile
        If d = s then
            r ← true
        EndIf
    EndIf
    return r
EndSubalgorithm
```

D. None of the other answers are correct.

29. Let us consider the subalgorithm `f(x, n, e, y, m)`, with the input parameters an array ***x*** with ***n*** elements that are whole numbers (***x***[1], ***x***[2], …, ***x***[***n***]), an array ***y*** with ***m*** elements that are whole numbers (***y***[1], ***y***[2], …, ***y***[***m***]), and a whole number ***e*** that does not appear in the array ***y*.** The subalgorithm returns an array and a natural number. Consider the following subalgorithms as well:

- `(c, p) ← concatenate(a, n, b, m)` that has the input parameters an array ***a*** with ***n*** elements that are whole numbers and an array ***b*** with ***m*** elements that are whole numbers, and returns the array ***c*** with ***p*** elements that are whole numbers, representing the concatenation of the arrays ***a*** and ***b***, namely: ***a***[1], ***a***[2], …, ***a***[***n***], ***b***[1], ***b***[2], …, ***b***[***m***]
- `(c, p) ← difference(a, n, b, m)` that has the input parameters an array ***a*** with ***n*** elements that are whole numbers and an array ***b*** with ***m*** elements that are whole numbers, and returns the array ***c*** with ***p*** elements that are whole numbers, consisting of all elements of array ***a*** (the remaining elements keeping their initial ordering) that do not appear in array ***b***

```
1.  Subalgorithm f(x, n, e, y, m):
2.      If n = 0 then
3.          return [], 0
4.      EndIf
5.      If x[1] ≠ e then
6.          s ← []
7.          s[1] ← x[1]
8.          (r1, l1) ← difference(x, n, s, 1)
9.          (r2, l2) ← f(r1, l1, e, y, m)
10.         (r3, l3) ← concatenate(s, 1, r2, l2)
11.         return r3, l3
12.     else
13.         (r1, l1) ← f(y, m, e, x, n)
14.         s ← []
15.         s[1] ← x[1]
16.         (r2, l2) ← difference(x, n, s, 1)
17.         (r3, l3) ← f(r2, l2, e, y, m)
18.         (r4, l4) ← concatenate(r1, l1, r3, l3)
19.         return r4, l4
20.     EndIf
21. EndSubalgorithm
```

Which of the following statements are true?

A. The subalgorithm `f(x, n, e, y, m)` builds a one dimensional array starting from the array ***x*** in which the occurences of element **e** are removed and replaced with the elements of array ***y*** at each occurence of element ***e***. The subalgorithm returns the built array and its dimension.

B. If arrays ***x*** and ***y*** do not have common elements, then the array returned by the subalgorithm `f(x, n, e, y, m)` will consist of distinct elements only.

C. The length of the array returned by the subalgorithm `f(x, n, e, y, m)`, having input parameters the non-empty arrays ***x*** and ***y***, can be smaller than ***n***.

D. If on line 18, instead of ***r1*** and ***l1*** we had ***y*** and ***m*** then the function would return a one dimensional array (together with its dimension) that would start with the elements of array ***y***, followed by the elements of array ***x***, the occurences of ***e*** being replaced by the elements of array ***y***.

30. Let us consider the subalgorithm `s(a, b, c)`, where ***a***, ***b***, ***c*** are non-zero natural numbers and ***b*** ≥ ***a***.

```
Subalgorithm s(a, b, c):
    If c = 0 then
        return 1
    else
        If a > b then
            return (1 / a) * s(a - 1, b, c)
        else
            If a < b then
                return (1 / b) * s(a, b - 1, c)
            else
                return c * s(a - 1, b - 1, c - 1)
            EndIf
        EndIf
    EndIf
EndSubalgorithm
```

What should be the relation between ***a***, ***b*** and ***c*** in order to obtain $1/C_b^a$ (where C_b^a represents combinations of ***b*** taken ***a*** at a time).

A. a + b = c
B. a + c = b
C. b - c = a
D. b + c = a - b

BABEŞ-BOLYAI UNIVERSITY

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission Exam – July 19th, 2021

Written Exam for Computer Science

GRADING AND SOLUTIONS

DEFAULT: 10 points

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