BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Zulassungsprüfung - 19. Juli 2023
Schriftliche Prüfung in Informatik

WICHTIGER HINWEIS:
Falls nicht anders erklärt:

- Es wird angenommen, dass alle arithmetischen Operationen mit unbegrenzten Datentypen durchgeführt werden (kein *Über-/Unterlauf*).
- Die Indexnummerierung aller Zeichenketten beginnt bei 1.
- Alle Einschränkungen beziehen sich auf die aktuellen Parameterwerte zum Zeitpunkt des ersten Aufrufs.
- Eine Teilfolge eines Vektors besteht aus Elementen, die aufeinanderfolgende Positionen im Vektor einnehmen.

1. Gegeben sei der Algorithmus `F(x)`, wobei ***x*** eine natürliche Zahl ist ($1 \leq \boldsymbol{x} \leq 10^6$):

```
Algorithm F(x):
    If x = 0 then
        Return 0
    Else
        If x MOD 3 = 0 then
            Return F(x DIV 10) + 1
        Else
            Return F(x DIV 10)
        EndIf
    EndIf
EndAlgorithm
```

Bei welchen der folgenden Aufrufe wird 4 zurückgegeben?

A. `F(21369)`
B. `F(6933)`
C. `F(4)`
D. `F(16639)`

2. Gegeben sei der Algorithmus `ceFace(a, b)`, wobei ***a*** und ***b*** natürliche Zahlen sind ($1 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10^4$), die nicht die Ziffer 0 enthalten.

```
Algorithm ceFace(a, b):
    p ← 0
    While a ≠ 0 execute
        c ← a MOD 10
        p ← p * 10 + c
        a ← a DIV 10
    EndWhile
    If p = b then
        Return True
    Else
        Return False
    EndIf
EndAlgoritm
```

Der Algorithmus `ceFace(a, b)` gibt *True* zurück, genau dann, wenn:

A. die Zahlen ***a*** und ***b*** gleich sind
B. ***a*** und ***b*** palindromische Zahlen sind
C. die Zahl ***a*** das Spiegelbild der Zahl ***b*** ist
D. die letzte Ziffer von ***a*** gleich ist der letzten Ziffer von ***b***

3. Gegeben sei der Algorithmus `ceFace(n)`, wobei ***n*** eine natürliche Zahl ist ($1 \leq \boldsymbol{n} \leq 10^3$). Der Operator "/" steht für die reelle Division, z. B.: 3 / 2 = 1,5.

```
Algorithm ceFace(n):
    s ← 0
    For i ← 1, n execute
        p ← (i + 1) * (i + 2)
        s ← s + (i / p)
    EndFor
    Return s
EndAlgorithm
```

Geben Sie den Ausdruck an, dessen Wert vom Algorithmus zurückgegeben wird.

A. $\frac{1}{1} + \frac{1}{1+2} + \cdots + \frac{1}{1+2+\cdots+\boldsymbol{n}}$

B. $\frac{1}{2*3} + \frac{2}{3*4} + \cdots + \frac{n}{(\boldsymbol{n}+1)*(\boldsymbol{n}+2)}$

C. $\frac{1}{1} + \frac{1}{1*2} + \cdots + \frac{1}{1*2*\ldots*\boldsymbol{n}}$

D. $\frac{1}{2*3} + \frac{2}{3*4} + \cdots + \frac{\boldsymbol{n}-1}{\boldsymbol{n}*(\boldsymbol{n}+1)}$

4. Gegeben sei der Algorithmus `f(n, x)`, wobei ***n*** eine natürliche Zahl ($3 \leq \boldsymbol{n} \leq 10^4$) und ***x*** ein Vektor aus ***n*** natürlichen Zahlen ist (***x***[1], ***x***[2], ..., ***x***[***n***], $1 \leq \boldsymbol{x}[\boldsymbol{i}] \leq 10^4$, für ***i*** = 1, 2, ..., ***n***).

```
Algorithm f(n, x):
    k ← 0
    For i ← 1, n - 1 execute
        If k = 0 then
            If x[i] = x[i + 1] then
                Return False
            EndIf
            If x[i] < x[i + 1] then
                k ← 1
            EndIf
        EndIf
        If k = 1 then
            If x[i] ≥ x[i + 1] then
                Return False
            EndIf
        EndIf
    EndFor
    If x[n - 1] ≥ x[n] then
        Return False
    EndIf
    Return True
EndAlgorithm
```

Bei welchen der folgenden Aufrufe gibt der Algorithmus *True* zurück?

A. `f(6, [1000, 512, 23, 22, 1, 2])`
B. `f(6, [6, 4, 1, 1, 2, 3])`
C. `f(8, [3000, 2538, 799, 424, 255, 256, 299, 1001])`
D. `f(3, [3, 2, 1])`

5. Gegeben sei der Algorithmus `calcul(a, b, c, d)`, wobei ***a***, ***b***, ***c***, ***d*** natürliche Zahlen ungleich Null sind ($1 \leq \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d} \leq 100$).

```
Algorithm calcul(a, b, c, d):
    x ← a * b
    y ← c * d
    While y ≠ 0 execute
        z ← x MOD y
        x ← y
        y ← z
    EndWhile
    Return x
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Der Algorithmus liefert den größten gemeinsamen Teiler der Zahlen ***a***, ***b***, ***c***, ***d***.
B. Der Algorithmus liefert den größten gemeinsamen Teiler der Zahlen ***a*** * ***b*** und ***c*** * ***d***.
C. Der Algorithmus liefert das kleinste gemeinsame Vielfache der Zahlen ***a***, ***b***, ***c***, ***d***.
D. Der Algorithmus liefert das kleinste gemeinsame Vielfache der Zahlen ***a*** * ***b*** und ***c*** * ***d***.

6. Gegeben sei der Algorithmus `p(na, a, nb, b)`, wobei ***na*** und ***nb*** natürliche Zahlen sind ($0 \leq \boldsymbol{na}, \boldsymbol{nb} \leq 10^4$), ***a*** und ***b*** Vektoren mit ***na***, bzw. ***nb*** natürlichen Zahlen sind (***a***[1], ***a***[2], ..., ***a***[***na***], $1 \leq \boldsymbol{a}[\boldsymbol{i}] \leq 10^4$, für ***i*** = 1, 2, ..., ***na*** und ***b***[1], ***b***[2], ..., ***b***[***nb***], $1 \leq \boldsymbol{b}[\boldsymbol{i}] \leq 10^4$, für ***i*** = 1, 2, ..., ***nb***). Die lokale Variable ***c*** ist ein Vektor.

```
Algorithm p(na, a, nb, b):
    i ← 1
    j ← 1
    nc ← 0
    While i ≤ na AND j ≤ nb execute
        nc ← nc + 1
        If a[i] < b[j] then
            c[nc] ← a[i]
            i ← i + 1
        Else
            c[nc] ← b[j]
            j ← j + 1
        EndIf
    EndWhile
    Return nc
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Wenn ***na*** = 0 und ***nb*** = 0, dann ist der von ***nc*** zurückgegebene Wert gleich 0.
B. Wenn die Elemente in ***a*** und ***b*** aufsteigend sortiert sind, dann sind auch die in ***c*** hinterlegten Elemente aufsteigend sortiert.
C. Der von ***nc*** zurückgegebene Wert ist immer gleich ***na*** + ***nb***.
D. Wenn ***na***, ***nb*** > 0 und das größte Element in ***a*** kleiner ist als alle Elemente in ***b***, dann hat ***c*** genau die gleichen Elemente wie ***a***.

7. Gegeben sei der Algorithmus `suma(n, a, m, b)`, wobei n und m natürliche Zahlen sind ($1 \leq n, m \leq 10^5$) und a und b zwei aufsteigend geordnete Zahlenfolgen mit n, bzw. m natürlichen Zahlen sind ($a[1], a[2], ..., a[n]$ und $b[1], b[2], ..., b[m]$):

```
Algorithm suma(n, a, m, b):
    s ← 0
    For i ← 1, n, 2 execute
        j ← 1
        While j ≤ a[i] AND j ≤ m execute
            s ← s + b[j]
            j ← j + 1
        EndWhile
    EndFor
    Return s
EndAlgorithm
```

Welchen Wert liefert der Algorithmus, wenn $n = 4$, $a = [1, 3, 4, 7]$, $m = 6$ und $b = [2, 4, 6, 8, 10, 12]$?

A. 42
B. 22
C. 20
D. Es ist nicht möglich zu bestimmen, welcher Wert zurückgegeben wird

8. Gegeben sei der Algorithmus `verifica(n, p1, p2)`, wobei n, $p1$ und $p2$ natürliche Zahlen sind ($1 \leq n, p1, p2 \leq 10^6$):

```
Algorithm verifica(n, p1, p2):
    bt ← (p1 + p2) DIV 2
    If p1 > p2 then
        Return False
    EndIf
    If bt * bt = n then
        Return True
    EndIf
    If bt * bt > n then
        Return verifica(n, p1, bt - 1)
    EndIf
    Return verifica(n, bt + 1, p2)
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Wenn die Zahlen **p1**, **p2** und n teilerfremd sind, gibt der Aufruf `verifica(n, p1, p2)` *True* zurück.
B. Der Algorithmus verwendet die binäre Suchmethode, und wenn die Zahl n eine Primzahl ist, gibt der Aufruf `verifica(n, 1, n)` *True* zurück.
C. Bei dem Aufruf `verifica(n, 1, n)` gibt der Algorithmus nur dann *True* zurück, wenn die Zahl n ein perfektes Quadrat ist.
D. Wenn $\mathbf{p1} \leq n \leq \mathbf{p2}$ und in den Intervallen [**p1**, n] und [n, **p2**] mindestens ein perfektes Quadrat vorhanden ist, gibt der Aufruf `verifica(n, p1, p2)` *True* zurück.

9. Gegeben sei der Algorithmus `ceFace(n)`, wobei n eine natürliche Zahl ist ($1 \leq n \leq 3000$).

```
Algorithm ceFace(n):
    s ← 0
    i ← 1
    While s < n execute
        s ← s + i
        If s = n then
            Return True
        Else
            i ← i + 2
        EndIf
    EndWhile
    Return False
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Wenn $n = 36$, gibt der Algorithmus *True* zurück.
B. Wenn n gleich einer Summe von aufeinanderfolgenden ungeraden Zahlen ab 1 ist, gibt der Algorithmus *True* zurück.
C. Wenn n ein perfektes Quadrat ist, gibt der Algorithmus *True* zurück, andernfalls *False*.
D. Wenn $n = 64$, gibt der Algorithmus *False* zurück.

10. Gegeben sei der Algorithmus `ceFace(a)`, wobei a eine natürliche Zahl ist ($1 \leq a \leq 10^4$).

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

Geben Sie das Ergebnis des Algorithmus an.

A. Der Algorithmus liefert das kleinste Palindrom, das größer oder gleich a ist.
B. Der Algorithmus liefert das größte Palindrom, das kleiner oder gleich a ist.
C. Der Algorithmus liefert das kleinste Palindrom, das größer als a ist.
D. Der Algorithmus liefert die kleinste gerade Zahl, die größer als a ist.

11. Gegeben sei der Algorithmus `calcul(v, n)`, wobei $\boldsymbol{n}$ eine natürliche Zahl ($1 \leq \boldsymbol{n} \leq 10^4$) und $\boldsymbol{v}$ ein Vektor aus $\boldsymbol{n}$ natürlichen Zahlen ist ($\boldsymbol{v}[1], \boldsymbol{v}[2], ..., \boldsymbol{v}[\boldsymbol{n}]$, $1 \leq \boldsymbol{v}[\boldsymbol{i}] \leq 10^4$, für $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$):

```
Algorithm calcul(v, n):
    i ← 2
    x ← 0
    If v[1] MOD 2 ≠ 0 then
        Return False
    EndIf
    While i ≤ n execute
        If x = 0 AND v[i] MOD 2 = 0 then
            Return False
        Else
            If x = 1 AND v[i] MOD 2 = 1 then
                Return False
            Else
                i ← i + 1
                x ← (x + 1) MOD 2
            EndIf
        EndIf
    EndWhile
    Return True
EndAlgorithm
```

In welchen der folgenden Situationen gibt der Algorithmus *True* zurück?

A. Wenn der Vektor $\boldsymbol{v}$ aus den Werten [2, 3, 10, 7, 20, 5, 18] besteht und $\boldsymbol{n} = 7$

B. Wenn der Vektor $\boldsymbol{v}$ Werte nach folgendem Muster hat: ungerade, gerade, ungerade, gerade...

C. Wenn der Vektor $\boldsymbol{v}$ aus den Werten [3, 8, 17, 20, 15, 10] besteht und $\boldsymbol{n} = 6$

D. Wenn der Vektor $\boldsymbol{v}$ Werte nach folgendem Muster hat: gerade, ungerade, gerade, ungerade...

12. Gegeben sei der Algorithmus `ceFace(a, n)`, wobei $\boldsymbol{n}$ eine natürliche Zahl ungleich Null ist ($2 \leq \boldsymbol{n} \leq 10^4$) und $\boldsymbol{a}$ ein Vektor aus $\boldsymbol{n}$ ganzen Zahlen ist ($\boldsymbol{a}[1], \boldsymbol{a}[2], ..., \boldsymbol{a}[\boldsymbol{n}]$, $-100 \leq \boldsymbol{a}[\mathrm{i}] \leq 100$, $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$). In dem Vektor $\boldsymbol{a}$ gibt es mindestens eine positive Zahl.

```
Algorithm ceFace(a, n):
    b ← 0
    c ← b
    For i ← 1, n execute
        b ← b + a[i]
        If b < 0 then
            b ← 0
        EndIf
        If b > c then
            c ← b
        EndIf
    EndFor
    Return c
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Der Algorithmus liefert die Summe aller Elemente des Vektors $\boldsymbol{a}$.

B. Der Algorithmus liefert die Summe der maximal langen Teilfolge, die nur positive Elemente des Vektors $\boldsymbol{a}$ enthält.

C. Der Algorithmus liefert die Summe aller positiven Elemente des Vektors $\boldsymbol{a}$.

D. Der Algorithmus liefert die Summe einer Teilfolge mit der maximalen Summe des Vektors $\boldsymbol{a}$.

13. Betrachten wir eine ganzzahlige Matrix $\boldsymbol{A}$ mit $\boldsymbol{n}$ Zeilen und $\boldsymbol{m}$ Spalten ($1 \leq \boldsymbol{n}, \boldsymbol{m} \leq 10^4$). Unter der Voraussetzung, dass $\boldsymbol{n} * \boldsymbol{m} = \boldsymbol{p} * \boldsymbol{q}$ ist, wollen wir diese Matrix in eine Matrix $\boldsymbol{B}$ aus ganzen Zahlen mit $\boldsymbol{p}$ Zeilen und $\boldsymbol{q}$ Spalten ($1 \leq \boldsymbol{p}, \boldsymbol{q} \leq 10^4$) umwandeln, wie im folgenden Beispiel, wobei $\boldsymbol{n} = 4$, $\boldsymbol{m} = 6$, $\boldsymbol{p} = 3$ und $\boldsymbol{q} = 8$. Die Zeilen und Spalten sind von 1 an nummeriert.

A:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24

B:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24

Welcher der folgenden Algorithmen stellt einen Algorithmus dar, der für das Paar natürlicher Zahlen $\boldsymbol{i}$ und $\boldsymbol{j}$ ($1 \leq \boldsymbol{i} \leq \boldsymbol{n}$, $1 \leq \boldsymbol{j} \leq \boldsymbol{m}$), die Indizes in der Matrix $\boldsymbol{A}$ darstellen, das Paar von Indizes in der Matrix $\boldsymbol{B}$ zurückgibt, das dem Wert von $\boldsymbol{A}[\boldsymbol{i}][\boldsymbol{j}]$ entspricht?

A.
```
Algorithm reshape(i, j, n, m, p, q):
    Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm
```

B.
```
Algorithm reshape(i, j, n, m, p, q):
    i ← i - 1
    j ← j - 1
    Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm
```

C.
```
Algorithm reshape(i, j, n, m, p, q):
    i ← i - 1
    j ← j - 1
    Return (i * m + j) DIV q + 1,
                            (i * m + j) MOD q + 1
EndAlgorithm
```

D.
```
Algorithm reshape(i, j, n, m, p, q):
    Return (i * m + j - 1) DIV q + 1,
                       (i * m + j - 1) MOD q + 1
EndAlgorithm
```

14. Gegeben sei der Algorithmus `ceFace(n, m)`, wobei ***n*** eine natürliche Zahl ist ($1 \leq n \leq 10^4$), und ***m*** eine Matrix mit ***n*** Zeilen und ***n*** Spalten ist, deren Elemente natürliche Zahlen sind (***m***[1][1], ..., ***m***[1][***n***], ***m***[2][1], ..., ***m[2]***[***n***], ..., ***m***[***n]***[1], ..., ***m***[***n***][***n]***). Wir nehmen an, dass die Elemente der Matrix ***m*** anfangs gleich 0 sind.

```
Algorithm ceFace(n, m):
    a ← 0
    b ← 1
    For j ← 1, n execute
        i ← 1
        While i + j ≤ n – 1 execute
            If (i MOD 2 = 1) AND (j MOD 2 = 1) then
                m[i][j] ← b
                c ← a + b
                a ← b
                b ← c
            EndIf
            i ← i + 1
        EndWhile
    EndFor
EndAlgorithm
```

Welche der folgenden Aussagen sind **FALSCH**?

A. Wenn ***n*** = 11, ist der Wert von ***m***[6][4] gleich mit 21
B. Wenn ***n*** = 7 ist, ist der Wert von ***m***[3][5] gleich mit 4
C. Wenn ***n*** = 10 ist, ist der Wert von ***m***[6][4] gleich mit 21
D. Wenn ***n*** = 7 ist, beträgt der Maximalwert in der Matrix 8

15. Die folgenden Algorithmen verarbeiten einen geordneten aufsteigenden Vektor ***x*** mit ***n*** Elementen natürlicher Zahlen ($1 \leq n \leq 10^4$, ***x***[1], ***x***[2], ..., ***x***[***n***]). Die Parameter ***first*** und ***last*** sind natürliche Zahlen (1 ≤ ***first*** ≤ ***last*** ≤ ***n***).

Wählen Sie die Algorithmen, die die geringste Zeitkomplexität haben, wenn sie in der Form `A(x, 1, n, n)` aufgerufen werden.

A.

```
Algorithm A(x, first, last, n):
    If first > last then
        Return 0
    EndIf
    m ← (first + last) DIV 2
    If x[m] = n then
        Return m
    Else
        If x[m] > n then
            Return A(x, first, m - 1, n)
        Else
            If x[m] < n then
                Return A(x, m + 1, last, n)
            EndIf
        EndIf
    EndIf
EndAlgorithm
```

B.

```
Algorithm A(x, first, last, n):
    While first < last execute
        m ← (first + last) DIV 2
        If x[m] = n then
            Return m
        Else
            If x[m] > n then
                last ← m - 1
            Else
                If x[m] < n then
                    first ← m + 1
                EndIf
            EndIf
        EndIf
    EndWhile
    Return 0
EndAlgorithm
```

C.

```
Algorithm A(x, first, last, n):
    For i ← first, last execute
        If x[i] = n then
            Return i
        EndIf
    EndFor
    Return 0
EndAlgorithm
```

D.

```
Algorithm A(x, first, last, n):
    For i ← first, last execute
        If x[i] = n then
            x[i] ← 3 * n
        EndIf
    EndFor
EndAlgorithm
```

16. Andrei spielt mit dem folgenden Algorithmus, wobei ***n*** und ***m*** natürliche, von Null verschiedene Zahlen sind ($1 \leq n, m \leq 10^4$). Der Algorithmus `abs(x)` gibt den absoluten Wert von ***x*** zurück.

```
Algorithm problema(n, m):
    b ← abs(m - n)
    c ← n - m
    If b - c = 0 then
        a ← n MOD m
    Else
        a ← (m + 2) MOD n
    EndIf
    Return a
EndAlgorithm
```

Er stellt fest, dass es unabhängig vom Wert der Variablen ***n***, die der Spezifikation entspricht, mindestens zwei Werte von ***m*** gibt, bei denen der Algorithmus `problema(n, m)` den Wert 0 ergibt. Welche sind diese Werte von ***m***?

A. 1 und ***n***
B. 1 und ***n*** + 2
C. ***n*** und ***n*** + 2
D. 1 und ***n*** – 2

17. Ein Student möchte mit Hilfe der Backtracking-Methode alle dreistelligen ungeraden Zahlen generieren mit Ziffern, die Werte aus dem Vektor [4, 3, 8, 5, 7, 6] in der angegebenen Reihenfolge annehmen. Wenn man weiß, dass die ersten 5 erzeugten Zahlen in dieser Reihenfolge 443, 445, 447, 433, 435 sind, was ist dann die zehnte erzeugte Zahl?

A. 487 B. 453 C. 457 D. 455

18. Gegeben sei der Algorithmus `f(k, n, x)`, wobei ***k***, ***n*** natürliche Zahlen sind ($1 \leq \boldsymbol{k}, \boldsymbol{n} \leq 10^3$) und ***x*** ein Vektor aus ***n*** natürlichen Zahlen ist (***x***[1], ***x***[2], ..., ***x***[***n***], $1 \leq \boldsymbol{x}[\boldsymbol{i}] \leq 10^4$, für ***i*** = 1, 2, ..., ***n***).

```
Algorithm f(k, n, x):
    If n = 0 then
        Return 0
    Else
        d ← 0
        For i ← 2, x[n] DIV 2 execute
            If (x[n] MOD i) = 0 then
                d ← d + 1
            EndIf
        EndFor
        If d = k then
            Return 1 + f(k, n - 1, x)
        Else
            Return f(k, n - 1, x)
        EndIf
    EndIf
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Für ***x*** = [4, 9, 26, 121] ist das Ergebnis des Aufrufs von `f(1, 4, x)` gleich mit 3.
B. Für ***x*** = [4, 8, 6, 144] ist das Ergebnis des Aufrufs von `f(2, 4, x)` gleich mit 3.
C. Für ***x*** = [4, 9, 25, 144] ist das Ergebnis des Aufrufs von `f(1, 4, x)` gleich mit 3.
D. Für ***x*** = [8, 27, 25, 121] ist das Ergebnis des Aufrufs von `f(2, 4, x)` gleich mit 3.

19. Gegeben sei der Algorithmus `check(n)`, wobei ***n*** eine natürliche Zahl ist ($1 \leq \boldsymbol{n} \leq 10^5$).

```
Algorithm check(n):
    While n > 0 execute
        If n MOD 3 > 1 then
            Return False
        EndIf
        n ← n DIV 3
    EndWhile
    Return True
EndAlgorithm
```

Geben Sie die Wirkung des Algorithmus an.

A. Der Algorithmus gibt *True* zurück, wenn ***n*** eine Potenz von 3 ist, und sonst *False*.
B. Der Algorithmus gibt *True* zurück, wenn die Basis-3-Schreibung von ***n*** nur die Ziffern 0 und 1 enthält, und andernfalls *False*.
C. Der Algorithmus gibt *True* zurück, wenn ***n*** als eine Potenz von 3 oder als die Summe verschiedener Potenzen von 3 geschrieben werden kann, und andernfalls *False.*
D. Der Algorithmus gibt *True* zurück, wenn die Basis-3-Schreibweise von ***n*** nur die Ziffer 2 enthält, und andernfalls *False*.

20. Eine Veranstaltung sollte in einem bestimmten Saal I stattfinden, muss aber in den Saal II verlegt werden, in dem die Sitzplatznummerierung anders ist. In beiden Sälen gibt es ***L*** Sitzreihen ($2 \leq \boldsymbol{L} \leq 50$), wobei jede Reihe in der Mitte durch einen Gang geteilt ist und ***K*** Sitze ($2 \leq \boldsymbol{K} \leq 50$) auf jeder Seite des Ganges hat (der Saal enthält also insgesamt 2 * ***K*** * ***L*** Sitze).

Im Saal I ist jeder Sitzplatz durch eine einzige Nummer gekennzeichnet. Die Sitze links vom Gang haben gerade Nummern, und die Sitzplatznummerierung beginnt in der Reihe vor der Bühne. Die Plätze in der ersten Reihe haben also (vom Gang aus zum Rand des Saals hin) die Nummern 2, 4, 6 usw. Nachdem alle Plätze einer Reihe nummeriert sind, wird die Nummerierung in der nächsten Reihe fortgesetzt, beginnend mit dem Platz neben dem Gang mit der nächsten geraden Nummer. Die Sitze auf der rechten Seite des Ganges werden auf die gleiche Weise nummeriert, allerdings mit ungeraden Zahlen. Die Sitze in der ersten Reihe sind also (vom Gang aus zum Rand des Saals hin) mit 1, 3, 5 usw. nummeriert.

In Saal II wird jeder Platz durch drei Werte identifiziert. Die Reihennummer (ein Wert zwischen 1 und ***L***, wobei Reihe 1 die Reihe vor der Bühne ist), die Richtung des Platzes in Bezug auf den Gang (der Wert "stanga" oder "dreapta") und die Sitzplatznummer innerhalb der Reihe (ein Wert zwischen 1 und ***K***, wobei Platz 1 der Sitz neben dem Gang ist).

Aufgrund der Verlegung der Veranstaltung müssen die Sitzplätze auf den Eintrittskarten in Saal I (durch eine einzige Nummer dargestellt) in gültige Sitzplätze in Saal II umgewandelt werden (durch ***rând***, ***loc***, ***direcție*** dargestellt).

Welcher der folgenden Algorithmen führt bei den Eingaben ***L, K, nrLoc*** gemäß der Aussage die Transformation korrekt durch? Eine Transformation ist korrekt, wenn jeder Zuschauer einen eindeutigen Sitzplatz in Saal II hat.

A.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ←  nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

B.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ← nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

C.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    rand ← nrLoc DIV (2 * K) + 1
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
    Return rand, loc, directie
EndAlgorithm
```

D.

```
Algorithm transforma(L, K, nrLoc):
    If nrLoc MOD 2 = 1 then
        directie ← "dreapta"
        nrLoc ← nrLoc + 1
    Else
        directie ← "stanga"
    EndIf
    If nrLoc MOD (2 * K) = 0 then
        rand ← nrLoc DIV (2 * K)
    Else
        rand ← nrLoc DIV (2 * K) + 1
    EndIf
    loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2 + 1
    Return d, loc, directie
EndAlgorithm
```

21. Gegeben sei der Algorithmus `p(x, n, k, final)`, wobei ***x*** ein Vektor aus ***n*** + 1 natürlichen Zahlen ist (***x***[0], ***x***[1], ***x***[2], ..., ***x***[***n***]). Am Anfang ist ***x***[***i***] = 0, für ***i*** = 0, 1, 2, ..., ***n***. Die Variablen ***n*** und ***k*** sind natürliche Zahlen ungleich Null ($1 \leq n, k \leq 20$), und ***final*** ist vom booleschen Typ. Der Algorithmus `Afis(x, 1, n)` zeigt die Elemente ***x***[1], ***x***[2], ..., ***x***[***n***] an.

```
Algorithm p(x, n, k, final):
    While final = False execute
        While x[k] < n execute
            x[k] ← x[k] + 1
            If OK(x, k) = True then
                If k = n then
                    Afis(x, 1, n)
                Else
                    k ← k + 1
                    x[k] ← 0
                EndIf
            EndIf
        EndWhile
        __________ // hier muss der Algorithmus vervollständigt sein
    EndWhile
EndAlgorithm
```

```
Algorithm OK(x, k):
    For i ← 1, k – 1 execute
        If x[k] = x[i] then
            Return False
        EndIf
    EndFor
    Return True
EndAlgorithm
```

Mit welcher Codesequenz soll der Algorithmus vervollständigt werden, so dass nach dem Aufruf `p(x, n, 1, False)` alle Permutationen der Ordnung ***n*** jeweils nur einmal angezeigt werden.

A.

```
If k > 1 then
    k ← k – 1
Else
    final ← True
EndIf
```

B.

```
If k > 0 then
    k ← k - 1
Else
    final ← True
EndIf
```

C.

```
final ← True
```

D.

```
If k > 1 then
    k ← k - 1
    final ← True
EndIf
```

22. Gegeben seien die Algorithmen `problema(n)` und `calcul(a, b)` an, wobei ***n***, ***a***, ***b*** natürliche Zahlen sind ($0 \le n, a, b \le 9$).

```
Algorithm problema(n):
    rezultat ← 0
    For k ← 0, n execute
        For p ← 0, n execute
            For j ← 0, n execute
                If p MOD 2 = 0 then
                    rezultat ← rezultat + 1
                EndIf
            EndFor
        EndFor
    EndFor
    Return rezultat
EndAlgorithm
```

```
Algorithm calcul(a, b):
    t ← 0
    For cifra ← a, b execute
        t ← t + problema(cifra)
    EndFor
    Write t
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

A. Der Aufruf `calcul(1, 8)` zeigt 1095 an.
B. Der Aufruf `calcul(1, 8)` zeigt 1094 an.
C. Der Aufruf `calcul(0, 9)` zeigt 1095 an.
D. Der Aufruf `calcul(0, 9)` zeigt 1595 an.

23. Gegeben sei der Algorithmus `checkAcc(n, f, w, lw)`, wobei ***n*** eine natürliche Zahl ungleich Null ist ($1 \le n \le 10^4$), ***f*** eine natürliche Zahl, und ***w*** eine Folge von ***lw*** ($1 \le lw \le 10^4$) natürlichen Zahlen (***w***[1], ***w***[2], ..., ***w***[***lw***], wobei $0 \le w[p] \le 10^4$, für ***p*** = 1, 2, ..., ***lw***) ist. Der Algorithmus `checkAcc(n, f, w, lw)` ruft den Algorithmus `t(i, j, k)` auf, wobei ***i***, ***j*** und ***k*** natürliche Zahlen sind. Der Algorithmus `t(i, j, k)` liefert ein boolesches Ergebnis.

```
Algorithm checkAcc(n, f, w, lw):
    acc ← True
    If lw = 0 AND f ≠ 1 then
        acc ← False
    Else
        index ← 1
        q ← 1
        While (acc = True) AND (index ≤ lw) execute
            crt ← 1
            changed ← False
            While (changed = False) AND (crt ≤ n) execute
                If t(q, w[index], crt) then
                    q ← crt
                    changed ← True
                Else
                    crt ← crt + 1
                EndIf
            EndWhile
            If changed = False then
                acc ← False
            Else
                index ← index + 1
            EndIf
        EndWhile
        If (index > lw) AND (acc = True) AND (q ≠ f) then
            acc ← False
        EndIf
    EndIf
    Return acc
EndAlgorithm
```

In welchen der folgenden Situationen wird der Algorithmus `checkAcc(2, f, w, lw)` *True* zurückgeben, wenn bekannt ist, dass der Algorithmus `t(i, j, k)` in den Fällen in der Tabelle *True* zurückgibt, ansonsten *False*?

i	*j*	*k*
1	0	1
1	1	2
2	1	2

A. ***w*** = [0, 0, 1, 1], ***lw*** = 4 und ***f*** = 1
B. ***w*** = [1, 1, 1, 0], ***lw*** = 4 und ***f*** = 2
C. ***w*** = [0, 0, 1, 1], ***lw*** = 4 und ***f*** = 2
D. ***w*** = [0, 0, 0, 0], ***lw*** = 4 und ***f*** = 1

24. Gegeben sei der Zahlenvektor ***a*** = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. Um die Elemente des Vektors ***a*** in einer anderen Reihenfolge darzustellen, konstruiert man den (zunächst leeren) Vektor ***b***. Bei jedem Schritt kann eine der beiden folgenden Operationen gewählt werden:

- *Hinzufügen* - fügt das erste Element vom Vektor ***a*** am Ende von Vektor ***b*** ein und entfernt es aus dem Vektor ***a***.
- *Löschen* - zeigt das letzte Element im Vektor ***b*** an und löscht es.

Bemerkungen:

- Die Elemente des Vektors ***a*** werden in der angegebenen Reihenfolge verarbeitet.
- Sie können die Operation *Hinzufügen* nicht verwenden, wenn der Vektor ***a*** leer ist, und Sie können die Operation *Löschen* nicht verwenden, wenn der Vektor ***b*** leer ist.
- Die Verarbeitung endet, wenn die Vektoren ***a*** und ***b*** leer sind.

Nach den oben genannten Regeln, in welcher Reihenfolge können die Zahlen **NICHT** angezeigt werden?

A. `0 1 2 3 4 5 6 7 8 9`
B. `9 8 7 6 5 4 3 2 1 0`
C. `2 4 6 5 3 7 0 1 9 8`
D. `2 3 1 4 5 0 8 9 7 6`

BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Aufnahmeprüfung – 19. Juli 2023

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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