

Aufnahmeprüfung – 17. Juli 2026
Schriftliche Prüfung in Informatik

WICHTIGER HINWEIS:

Falls nicht anders angegeben:

- Alle arithmetischen Operationen erfolgen auf unbeschränkten Datentypen (es gibt keinen *overflow* / *underflow*).
- Die Nummerierung der Indizes aller Vektoren, Matrizen und Zeichenketten beginnt bei 1.
- Alle Einschränkungen beziehen sich auf die Werte der aktuellen Parameter beim ursprünglichen Aufruf.
- Eine Teilsequenz eines Vektors oder einer Zeichenkette besteht aus Elementen, die im Vektor bzw. in der Zeichenkette aufeinanderfolgende Positionen belegen.
- Mehrere aufeinanderfolgende Zuweisungen in derselben Zeile sind durch ";" getrennt.

1. Gegeben sei der Algorithmus `ceface(n)`, wobei n eine natürliche Zahl ist ($0 \leq n \leq 10^3$) ist und die Operation `&` das bitweise **AND** bezeichnet.

```
Algorithm ceface(n):  
  If (n & 1) ≠ 0 then  
    Return 0  
  EndIf  
  Return 1  
EndAlgorithm
```

Welche der folgenden Aussagen sind wahr?

- A. Ist n eine Potenz von 2, gibt der Algorithmus 1 zurück.
- B. Ist n gerade, gibt der Algorithmus 1 zurück.
- C. Der Algorithmus gibt genau dann 0 zurück, wenn n ungerade ist.
- D. Für $n = 0$ gibt der Algorithmus 0 zurück.

2. Gegeben sei der Algorithmus `gramana(n, x)`, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^3$), und x eine Zeichenkette aus n Zeichen ist ($x[1], x[2], \dots, x[n]$, für $i = 1, 2, \dots, n$), die alle Kleinbuchstaben sind.

```
Algorithm gramana(n, x):  
  For i ← n, 1, -1 execute  
    For j ← 2, i execute  
      If x[j - 1] > x[j] then  
        car ← x[j]  
        x[j] ← x[j - 1]  
        x[j - 1] ← car  
      EndIf  
    EndFor  
  EndFor  
EndAlgorithm
```

Welche der folgenden Aussagen sind wahr?

- A. Die Zeitkomplexität des Algorithmus ist $O(n^2)$.
- B. Der Algorithmus führt eine zyklische Permutation der Elemente der gegebenen Zeichenkette durch.
- C. Der Algorithmus sortiert die Zeichenkette x lexikografisch in aufsteigender Reihenfolge.
- D. Der Algorithmus sortiert die Zeichenkette x lexikografisch in absteigender Reihenfolge.

3. Für die Binärzahl $101100110000_{(2)}$, geben Sie an, welche der folgenden Aussagen wahr sind.

- A. Die Zahl ist teilbar durch $16_{(10)}$
- B. Die Zahl ist kleiner als $10000_{(10)}$
- C. Die Zahl ist gerade.
- D. Die Zahl ist teilbar durch $32_{(10)}$.

4. Gegeben sei der Algorithmus `test(n)`, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^3$).

```
Algorithm test(n):  
  c ← 1  
  While n > 0 execute  
    n ← n - 2  
    c ← c * 2  
  EndWhile  
  Return c  
EndAlgorithm
```

Welche der folgenden Aussagen sind wahr?

- A. Der Aufruf `test(n)` liefert stets einen Wert größer als n .
- B. Der Aufruf `test(n)` liefert stets einen kleineren Wert als der Aufruf `test(n + 1)`.
- C. Wenn die verschiedenen natürlichen Zahlen n und m dieselbe Parität haben, liefern die Aufrufe `test(n)` und `test(m)` verschiedene Werte.
- D. Der Algorithmus `test(n)` hat die Zeitkomplexität $O(\log n)$.

5. Gegeben sei der Algorithmus $\text{mister}(x, n)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^3$), und x ein Vektor mit n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n]$, $10 \leq x[i] \leq 10^5$, für $i = 1, 2, \dots, n$).

```

Algorithm mister(x, n):
  For i ← 1, n - 1 execute
    For j ← i + 1, n execute
      a ← x[i]
      b ← x[j]
      While a > 10 execute
        a ← a DIV 10
      EndWhile
      While b > 10 execute
        b ← b DIV 10
      EndWhile
      If a < b then
        aux ← x[i]
        x[i] ← x[j]
        x[j] ← aux
      EndIf
    EndFor
  EndFor
  Return x
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Der Algorithmus $\text{mister}(x, n)$ sortiert den Vektor x absteigend nach der ersten Ziffer seiner Elemente.
- B. Der Aufruf $\text{mister}([345, 21, 98, 412, 67], 5)$ liefert den Vektor $[98, 67, 412, 345, 21]$.
- C. Der Algorithmus $\text{mister}(x, n)$ sortiert den Vektor x aufsteigend nach der letzten Ziffer der Zahlen.
- D. Der Aufruf $\text{mister}([10, 305, 47, 89], 4)$ liefert den Vektor $[89, 47, 305, 10]$.

6. An einem Sportturnier nehmen n ($2 \leq n \leq 100$) Spieler teil. Jede Partie wird zwischen 2 Spielern gespielt; der Verlierer jeder Partie scheidet aus dem Turnier aus.

Wie viele Partien müssen gespielt werden, um zu wissen, wer das Turnier gewinnt?

- A. $n - 1$
- B. n
- C. $n \text{ DIV } 2$
- D. $(n \text{ DIV } 2) + 1$

7. Ein logischer Ausdruck aus Termen, die mit dem Operator **OR** ($t_1 \text{ OR } t_2 \text{ OR } \dots \text{ OR } t_n$) verbunden sind, wird von links nach rechts ausgewertet, bis der erste Term mit Wert *True* gefunden wird, falls vorhanden (dann ist der Wert des Ausdrucks *True*, die restlichen Terme entfallen). Falls alle Terme zu *False* ausgewertet werden, hat der Ausdruck den Wert *False*.

Gegeben sei der Algorithmus $\text{test}(n)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^5$).

```

Algorithm test(n):
  Write '*'
  If n < 10 then
    Return False
  EndIf
  If n MOD 2 = 0 then
    Return True
  EndIf
  If n MOD 5 = 0 then
    Return True
  EndIf
  Return test(n DIV 2) OR test(n DIV 5) OR test(n DIV 10) // #
EndAlgorithm

```

Angenommen, bei der Ausführung der mit # markierten Zeile (ein logischer Ausdruck aus drei mit **OR** verbundenen Termen) wird wie oben vorgegangen: Wie viele Zeichen '*' gibt der Aufruf $\text{test}(159)$ aus?

- A. 10
- B. 15
- C. 3
- D. 8

8. Gegeben sei der Algorithmus $f(n)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^3$).

```

Algorithm f(n):
  If n = 1 then
    Return 0
  EndIf
  Return f(n DIV 2) + 1
EndAlgorithm

```

Was liefert der Aufruf $f(n)$?

- A. Der ganzzahlige Teil des Wertes $\log_2 n$.
- B. Die minimale Anzahl von Bits, um n in Basis 2 darzustellen.
- C. Die Anzahl der 1-Bits in der Darstellung von n in Basis 2.
- D. Ein Wert verschieden von n .

9. Gegeben sei der Algorithmus $ff(n, v)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^4$), und v ein Vektor mit n ganzzahligen Elementen ist ($v[1], v[2], \dots, v[n]$, $-100 \leq v[i] \leq 100$, für $i = 1, 2, \dots, n$).

```

Algorithm ff(n, v):
  c ← 0
  For j ← 2, n execute
    a ← v[j]
    i ← j - 1
    While i > 0 AND v[i] ≥ a execute
      v[i + 1] ← v[i]
      i ← i - 1
      c ← c + 1
    EndWhile
    v[i + 1] ← a
  EndFor
  Return c
EndAlgorithm

```

Welchen Wert liefert der Aufruf
 $ff(5, [-4, 1, 7, -8, 1])$?

- A. 4
- B. 2
- C. 5
- D. 8

10. Gegeben sei der Algorithmus $f(a, b)$, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10^5$).

```

Algorithm f(a, b):
  If b = 0 then
    Return 0
  EndIf
  If b MOD 2 = 0 then
    Return f(a + a, b DIV 2)
  Else
    Return f(a + a, b DIV 2) + a
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Der Algorithmus liefert das Produkt $a * b$.
- B. Die Zeitkomplexität des Algorithmus ist $O(\log b)$.
- C. Der Aufruf $f(2, 3)$ gibt 8 zurück.
- D. Der Algorithmus liefert den Wert a^b .

11. Gegeben sei der Algorithmus $ceFace(a, n, s)$, wobei a ein Vektor von n ($1 \leq n \leq 20$) natürlichen Zahlen ist ($a[1], a[2], \dots, a[n]$, $0 \leq a[i] \leq 10^3$ für $1 \leq i \leq n$), der aufsteigend sortiert ist, und s eine natürliche Zahl ist ($0 \leq s \leq 10^3$).

```

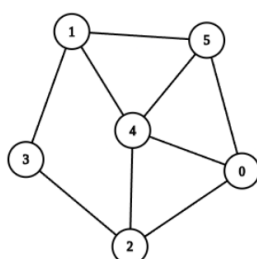
Algorithm ceFace(a, n, s):
  c ← 0
  For i ← 1, n - 2 execute
    j ← i + 1
    k ← n
    While j < k execute
      sum ← a[i] + a[j] + a[k]
      If sum = s then
        c ← c + 1
        j ← j + 1
        k ← k - 1
      Else
        If sum < s then
          j ← j + 1
        Else
          k ← k - 1
        EndIf
      EndIf
    EndWhile
  EndFor
  Return c
EndAlgorithm

```

Welche Aufrufe liefern den Wert 3?

- A. $ceFace([1, 2, 3, 4, 5, 6], 6, 10)$
- B. $ceFace([1, 2, 3, 4, 4, 5, 6], 7, 10)$
- C. $ceFace([1, 2, 3, 4, 4, 5, 6], 7, 8)$
- D. $ceFace([1, 1, 1, 2, 2, 2], 6, 4)$

12. Gegeben sei der folgende Graph:



Welche der folgenden Knotenfolgen ist ein Hamiltonkreis?

- A. 0 1 2 3 4 5 0
- B. 1 3 2 0 4 5 1
- C. 1 3 2 4 2 0 5
- D. 4 2 3 1 5 0 4

13. Präslea der Tapfere hat in seinem Garten einen Zauberbaum mit 27 goldenen und 24 silbernen Äpfeln. Jeden Tag stiehlt der Drache zwei Äpfel; zugleich wächst am Baum ein neuer Apfel: Sind die beiden genommenen Äpfel von derselben Art (beide golden oder beide silbern), wächst an ihrer Stelle ein silberner Apfel; nimmt der Drache einen goldenen und einen silbernen, wächst an ihrer Stelle ein goldener Apfel.

Welche der folgenden Aussagen sind wahr?

- A. Am 50. Tag hat der Drache einen goldenen und einen silbernen Apfel genommen.
- B. Es lässt sich nicht bestimmen, welche Äpfel der Drache am 50. Tag genommen hat.
- C. Es gibt eine Abfolge von Diebstählen, sodass irgendwann nur silberne Äpfel am Baum sind.
- D. Im beschriebenen Prozess sinkt die Gesamtzahl der Äpfel am Baum jeden Tag um genau 1.

14. Gegeben sei der Algorithmus $f(X, n, k)$, wobei n und k natürliche Zahlen sind ($1 \leq k \leq n \leq 10$), und X ein Vektor von n natürlichen Zahlen ist ($X[1], X[2], \dots, X[n]$).

```

Algorithm f(X, n, k):
  If k = n + 1 then
    aux ← 0; ok ← True
    For j ← 1, n execute
      If X[j] = 1 then
        aux ← aux + 1
      EndIf
      If j > 1 AND X[j] = 1 AND X[j - 1] = 1 then
        ok ← False
      EndIf
    EndFor
    If ok AND aux = n DIV 2 then
      For j ← 1, n execute
        Write X[j]
      EndFor
      Write New Line
    EndIf
  Else
    For i ← 0, 1 execute
      X[k] ← i
      f(X, n, k + 1)
    EndFor
  EndIf
EndAlgorithm

```

Nach dem Aufruf $f([0, 0, 0, 0], 4, 1)$ ist die erste ausgegebene Zeile 0101. Welche sind die nächsten beiden ausgegebenen Zeilen?

- A. 1001
1100
- B. 1010
0110
- C. 0110
1010
- D. 1001
1010

15. Gegeben sei der Algorithmus $LU(A, n)$, wobei A eine quadratische Matrix der Dimension n ($A[1][1], A[1][2], \dots, A[1][n], \dots, A[n][n]$, $0 < A[i][j] < 10^3$, für $i, j = 1, \dots, n$) ist, die natürliche Zahlen enthält, und n eine natürliche Zahl ist ($2 \leq n \leq 10$). Der Algorithmus $zerosMat(n)$ liefert eine quadratische Matrix der Dimension n , deren sämtliche Elemente gleich 0 sind.

```

Algorithm LU(A, n):
  L ← zerosMat(n); U ← zerosMat(n)
  For i ← 1, n execute
    L[i][i] ← 1
    For j ← i, n execute
      sum ← 0
      For k ← 1, i - 1 execute
        sum ← sum + L[i][k] * U[k][j]
      EndFor
      U[i][j] ← A[i][j] - sum
    EndFor
    For j ← i + 1, n execute
      sum ← 0
      For k ← 1, i - 1 execute
        sum ← sum + L[j][k] * U[k][i]
      EndFor
      L[j][i] ← (A[j][i] - sum) DIV U[i][i]
    EndFor
  EndFor
  Return L, U
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Nach dem Aufruf $LU(A, n)$ sind bei der zurückgegebenen Matrix L alle Elemente oberhalb der Hauptdiagonale gleich 0.
- B. Die Zeitkomplexität des Algorithmus ist $O(n^2)$.
- C. Wenn $A = \begin{pmatrix} 2 & 1 & 2 \\ 4 & 6 & 3 \\ 4 & 2 & 8 \end{pmatrix}$ und $n = 3$ ist, ist der Wert der nach dem Aufruf $LU(A, n)$ zurückgegebenen Matrizen L und U gleich $L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$ und $U = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 4 \end{pmatrix}$.
- D. Nach dem Aufruf $LU(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, 2)$, ist das Produkt der Hauptdiagonalelemente der gelieferten Matrix U gleich der Determinante der Matrix A .

16. Gegeben sei der Algorithmus $f(x, y, v)$, wobei x, y und v natürliche Zahlen sind ($1 \leq x, y, v \leq 100, x \leq v$).

```

Algorithm  $f(x, y, v)$ :
  If  $x \geq v$  then
    Write '1'
  Else
    If  $x \bmod y \neq 0$  then
       $f(x + 1, y, v - 1)$ 
    Else
      If  $y \bmod v \neq 0$  then
         $f(x, y + 1, v - 1)$ 
        Write '*'
      Else
         $f(x + 1, y, v - 2)$ 
        Write '2'
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Die Aufrufe $f(3, 5, 10)$, $f(5, 6, 10)$ und $f(2, 5, 9)$ erzeugen dieselbe Ausgabe.
- B. Der Aufruf $f(2, 2, 15)$ gibt der Algorithmus "1*****" aus
- C. Der Aufruf $f(2, 3, 14)$ erzeugt eine Ausgabe, die die Ziffer 2 enthält.
- D. Die Aufrufe $f(2, 2, 11)$, $f(1, 2, 9)$ und $f(3, 7, 9)$ erzeugen dieselbe Ausgabe.

17. Gegeben sei der Algorithmus $f(n, x)$, wobei n eine natürliche Zahl ist ($3 \leq n \leq 10^4$), und x ein Vektor von n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n]$, $1 \leq x[i] \leq 10^4$, für $i = 1, 2, \dots, n$).

```

Algorithm  $f(n, x)$ :
  If  $x[n] - x[n - 1] \neq x[n - 2]$  then
    Return 0
  EndIf
   $k \leftarrow 3$ 
  For  $i \leftarrow n - 1, 3, -1$  execute
    If  $x[i] - x[i - 1] \neq x[i - 2]$  then
      Return  $k$ 
    EndIf
     $k \leftarrow k + 1$ 
  EndFor
  Return  $k$ 
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Der Aufruf $f(4, [1, 1, 2, 3])$ gibt 4 zurück.
- B. Der Algorithmus liefert genau dann den Wert n , wenn die Werte im Vektor x Teil einer Folge von Fibonacci-Zahlen sind.
- C. Es gibt keinen Vektor mit 3 Elementen, für den der Algorithmus 1 liefert.
- D. Liefert der Algorithmus den Wert n , sind die Werte im Vektor x aufsteigend geordnet.

18. Gegeben sei der Algorithmus $verifica(s1, n1, s2, n2)$, wobei $s1$ und $s2$ Zeichenketten nur aus Kleinbuchstaben des englischen Alphabets sind, mit den Längen $n1$ bzw. $n2$ ($1 \leq n1, n2 \leq 100$). Der Algorithmus $zerosVec(n)$ liefert einen Vektor natürlicher Zahlen mit n Elementen, alle mit 0 initialisiert.

```

Algorithm  $verifica(s1, n1, s2, n2)$ :
  If  $n1 \neq n2$  then
    Return False
  EndIf
   $f \leftarrow zerosVec(26)$ 
  For  $i \leftarrow 1, n1$  execute
     $p \leftarrow s1[i] - 'a' + 1$ 
     $f[p] \leftarrow f[p] + 1$ 
  EndFor
  For  $i \leftarrow 1, n2$  execute
     $p \leftarrow s2[i] - 'a' + 1$ 
    ...
    If  $f[p] < 0$  then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

Durch welche der folgenden Anweisungen können die Auslassungspunkte ersetzt werden, damit der Algorithmus $verifica(s1, n1, s2, n2)$ genau dann True liefert, wenn die Zeichenkette $s2$ durch Umordnen der Zeichen von $s1$ entsteht?

- A. $f[p] \leftarrow f[p] - 1$
- B. $f[p] \leftarrow f[p] + 1$
- C. $f[p] \leftarrow f[p] - i$
- D. $f[p] \leftarrow (f[p] + 1) \bmod 2$

19. Gegeben sei der Algorithmus `produs_str(n1, n2)`, wobei ***n1*** und ***n2*** Zeichenketten sind, die natürliche Zahlen ungleich null darstellen (z. B. ***n1*** = "123" stellt 123 dar). Der Algorithmus liefert das Produkt der beiden Zahlen als Zeichenkette. So liefert `produs_str("2", "3")` den Wert "6" und `produs_str("123", "456")` den Wert "56088". Der Algorithmus `len(sir)` liefert die Länge der Zeichenkette *sir*. Der Algorithmus `int(car)` liefert die dem Zeichen *car* entsprechende Ziffer. Der Algorithmus `str(nr)` liefert die der Zahl *nr* entsprechende Zeichenkette (z. B. `str(123)` liefert "123"). Der Algorithmus `add(n, m)` liefert die Summe der Zahlen *n* und *m*, dargestellt als Zeichenketten (z. B. `add("123", "457")` liefert die Zahl 580). Der Algorithmus `zero(k)` liefert einen Vektor von *k* Nullen. Für Zeichenketten bezeichnet der Operator + die Konkatenation.

Welche der folgenden Varianten implementieren den Algorithmus `produs_str(n1, n2)` korrekt?

- A.
- ```

Algorithm produs_str(n1, n2):
 rez ← ""
 For i ← 1, len(n1) execute
 For j ← 1, len(n2) execute
 produs ← int(n1[i]) * int(n2[j])
 rez ← rez + str(produs)
 EndFor
 EndFor
 Return rez
EndAlgorithm

```
- B.
- ```

Algorithm produs_str(n1, n2):
    rez ← "0"
    For i ← len(n1), 1, -1 execute
        cif ← int(n1[i])
        For j ← 1, cif execute
            rez ← str(add(rez, n2))
        EndFor
    EndFor
    Return rez
EndAlgorithm

```
- C.
- ```

Algorithm produs_str(n1, n2):
 rez ← zero(len(n1) + len(n2))
 For i ← len(n1), 1, -1 execute
 For j ← len(n2), 1, -1 execute
 produs ← int(n1[i]) * int(n2[j])
 poz1 ← i + j - 1
 poz2 ← i + j
 sum ← produs + rez[poz2]
 rez[poz2] ← sum MOD 10
 rez[poz1] ← rez[poz1] + (sum DIV 10)
 EndFor
 EndFor
 rez_str ← ""
 For i ← 1, len(n1) + len(n2) execute
 digit ← rez[i]
 If NOT ((len(rez_str) = 0) AND (digit = 0)) then
 rez_str ← rez_str + str(digit)
 EndIf
 EndFor
 Return rez_str
EndAlgorithm

```
- D.
- ```

Algorithm produs_str(n1, n2):
    rez ← zero(len(n1) + len(n2))
    For i ← len(n1), 1, -1 execute
        cifTransp ← 0
        For j ← len(n2), 1, -1 execute
            produs ← int(n1[i]) * int(n2[j]) +
                cifTransp + rez[i + j]
            rez[i + j] ← produs MOD 10
            cifTransp ← produs DIV 10
        EndFor
        rez[i] ← rez[i] + cifTransp
    EndFor
    rez_str ← ""
    For i ← 1, len(n1) + len(n2) execute
        digit ← rez[i]
        If NOT ((len(rez_str) = 0) AND (digit = 0)) then
            rez_str ← rez_str + str(digit)
        EndIf
    EndFor
    Return rez_str
EndAlgorithm

```

20. Gegeben sei der Algorithmus `P(s, n)`, wobei *s* eine Zeichenkette der Länge *n* ist ($3 \leq n \leq 255$). Der Algorithmus `copiaza(s, p, q)` liefert die Teilsequenz der Zeichenkette *s*, die auf Position *p* beginnt und auf Position *q* endet ($1 \leq p \leq q \leq n$). Der Operator + bezeichnet die Konkatenation von Zeichenketten.

```

Algorithm P(s, n):
    i ← n DIV 3
    j ← 2 * (n DIV 3)
    k ← n
    If n MOD 2 = 0 then
        Return copiaza(s, i, i + 1) + copiaza(s, j - 1, j + 1) + copiaza(s, k, k)
    Else
        Return copiaza(s, 1, 1) + copiaza(s, i, i + 1) + copiaza(s, j - 1, j + 1)
    EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Der Aufruf $P(\text{"calculator"}, 10)$ gibt "lcular" zurück.
- B. Die vom Algorithmus $P(s, n)$ gelieferte Zeichenkette hat stets weniger als n Zeichen.
- C. Der Aufruf $P(s_1, n_1)$ kann genau dann die Zeichenkette "ubbubb" liefern, wenn n_1 größer oder gleich 8 ist, und der Aufruf $P(s_2, n_2)$ kann genau dann die Zeichenkette "ubbfmi" liefern, wenn n_2 größer oder gleich 9 ist.
- D. Der Aufruf $P(s, n)$ liefert genau dann eine Zeichenkette mit paarweise verschiedenen Zeichen, wenn n größer oder gleich 9 ist und die Zeichenkette s aus paarweise verschiedenen Zeichen besteht.

21. Gegeben sei der Algorithmus $f(n, x)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^3$), x ein Vektor mit n ganzen Zahlen ist ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$), und $\text{abs}(k)$ den Betrag von k liefert.

```

1.  Algorithm f(n, x):
2.      m ← n
3.      a ← 1
4.      b ← -1
5.      For i ← 1, n execute
6.          p ← -1
7.          j ← n
8.          While p = -1 AND j > i execute
9.              If abs(x[i] - x[j]) = 1 then
10.                  p ← j
11.              EndIf
12.              j ← j - 1
13.          EndWhile
14.          t ← i - 1 + n - p
15.          If t < m then
16.              m ← t
17.              a ← i
18.              b ← p
19.          EndIf
20.      EndFor
21.      Return m
22.  EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Nach Ausführung der Anweisung in Zeile 20 gilt $m = n$ oder, falls $b \neq -1$, $\text{abs}(x[a] - x[b]) = 1$.
- B. Enthält der Vektor x die ersten n ($n \geq 3$) Elemente der Fibonacci-Folge, wird unabhängig von deren Anordnung ein Wert größer als 0 geliefert.
- C. Für einen Vektor x mit n Elementen und y einen Vektor mit denselben Elementen, zyklisch um k Positionen nach rechts verschoben ($1 \leq k < n$) ausgehend vom Ausgangsvektor x , $f(n, x)$ und $f(n, y)$ liefern denselben Wert, unabhängig von k .
- D. Der Aufruf $f(9, [8, 2, 4, 5, 2, 5, 3, 4, 6])$ gibt 6 zurück.

22. Gegeben sei der Algorithmus $f(v, n, i, a, j)$, wobei i, j, n und a natürliche Zahlen sind ($0 \leq j, 1 \leq i, n \leq 20, 1 \leq a \leq 10^4$), v ein Vektor mit n von null verschiedenen natürlichen Zahlen ist ($v[1], v[2], \dots, v[n]$, $1 \leq v[k] \leq 10^4$, für $k = 1, 2, \dots, n$).

```

Algorithm f(v, n, i, a, j):
  If a = 0 AND j MOD 2 = 1 then
    Return 1
  EndIf
  If a = 0 then
    Return 0
  EndIf
  If a < 0 OR i = n + 1 then
    Return 0
  EndIf
  If v[i] MOD 2 = 0 then
    Return f(v, n, i + 1, a, j)
  EndIf
  Return f(v, n, i + 1, a - v[i], j + 1) +
    f(v, n, i + 1, a, j)
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Der Aufruf $f([2, 4, 6], 3, 1, 11, 0)$ gibt 4 zurück.
- B. Der Aufruf $f([1, 7, 11, 2, 3], 5, 2, 11, 0)$ gibt 2 zurück.
- C. Der Aufruf $f(v, n, 1, 1000, 0)$ liefert 0, unabhängig vom Wert von n und den Werten im Vektor v .
- D. Der Aufruf $f([1, 2, 3, 4, 5, 6, 3, 7], 8, 1, 7, 0)$ gibt 2 zurück.

23. Gegeben sei der Algorithmus $f(x)$, wobei x eine natürliche Zahl ist ($0 \leq x \leq 10^9$).

```

Algorithm f(x):
  If x = 0 then
    Return 0
  EndIf
  If x MOD 2 = 0 then
    Return f(x DIV 2) + ((x + 1) MOD 2)
  EndIf
  Return f(x DIV 2) - (x MOD 2)
EndAlgorithm

```

```

Algorithm f2(x):
  If x = 0 then
    Return 0
  EndIf
  v ← f2(x DIV 2)
  If x MOD 2 = 0 then
    Return v + ((x + 1) MOD 2)
  EndIf
  Return v - (x MOD 2)
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Die Aufrufe $f(100)$ und $f(24)$ liefern denselben Wert.
- B. Der Aufruf $f(1024)$ liefert den Wert 10.
- C. Im Intervall $[128, 255]$ gibt es 36 Werte, für die der Algorithmus 0 liefert.
- D. Für jede Zahl x gilt: $f_2(x) = f(x)$; die beiden Algorithmen haben dieselbe Zeitkomplexität.

24. Gegeben sei der Algorithmus $\text{nrd}(M, r, c, n)$, wobei n ($2 \leq n \leq 10$), r und c ($1 \leq r, c \leq n$) natürliche Zahlen sind, und M eine quadratische Matrix mit $n \times n$ Elementen ist ($M[1][1], \dots, M[1][n], \dots, M[n][1], \dots, M[n][n]$, wobei $M[i][j] \in \{0, 1\}$, für $i, j = 1, \dots, n$). Ein Weg in der Matrix M ist eine Folge von zeilen- oder spaltenweise benachbarten Positionen mit Wert 1, in der sich keine Position wiederholt. Zwei Wege sind verschieden, wenn sie sich in mindestens einer Position unterscheiden. Die Werte 0 sind Wände, die nicht durchquert werden können.

```

Algorithm nrd(M, r, c, n):
  If r = n AND c = n then
    Return 1
  EndIf
  If r < 1 OR r > n OR c < 1 OR c > n OR M[r][c] = 0 then
    Return 0
  EndIf
  total ← 0
  ...
  Return total
EndAlgorithm

```

Wenn man die Auslassungspunkte durch die Anweisungsfolge der entsprechenden Antwortoption ersetzt, welche der folgenden Aussagen sind wahr?

- A.


```

M[r][c] ← 0
total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)
M[r][c] ← 1

```

 Der Algorithmus liefert die Anzahl verschiedener Wege von Position $M[r][c]$ bis Position $M[n][n]$ (beim ursprünglichen Aufruf gilt $M[r][c] = M[n][n] = 1$).
- B.


```

M[r][c] ← 0
total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)
M[r][c] ← 1

```

 Die Zeitkomplexität des Algorithmus ist $O(n^4)$.
- C.


```

total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)

```

 Der Algorithmus liefert die Anzahl verschiedener Wege von Position $M[r][c]$ bis Position $M[n][n]$ (beim ursprünglichen Aufruf gilt $M[r][c] = M[n][n] = 1$).
- D.


```

total ← nrd(M, r + 1, c, n) + nrd(M, r, c + 1, n)

```

 Der Aufruf $\text{nrd}(M, 1, 1, 3)$, wobei alle Elemente der Matrix M gleich 1 sind, liefert 6.

BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Zulassung - 17. Juli 2026

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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