

Zulassung - 17. Juli 2025
Schriftliche Prüfung in Informatik

WICHTIGE NOTE:

Sofern nicht anders angegeben:

- Alle arithmetischen Operationen werden mit unbegrenzten Datentypen durchgeführt (kein Über-/Unterlauf).
- Die Indexnummerierung aller Vektoren, Matrizen und Strings beginnt bei 1.
- Alle Beschränkungen beziehen sich auf die aktuellen Parameterwerte zum Zeitpunkt des ersten Aufrufs.
- Eine Teilsequenz eines Vektors oder einer Zeichenkette besteht aus Elementen, die aufeinanderfolgende Positionen im Vektor bzw. in der Zeichenkette einnehmen.
- Wenn mehrere aufeinanderfolgende Zuweisungsanweisungen in derselben Zeile vorkommen, werden sie durch ";" getrennt.

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

Welche der folgenden Implementierungen des Algorithmus $\text{ceFace}(n, x)$ liefert den größten Wert im Vektor x ?

A.
Algorithm $\text{aux}(n, x, i, b)$:
 If $i > n$ **then**
 Return b
 EndIf
 If $b < x[i]$ **then**
 $b \leftarrow x[i]$
 EndIf
 Return $\text{aux}(n, x, i + 1, b)$
EndAlgorithm
Algorithm $\text{ceFace}(n, x)$:
 Return $\text{aux}(n, x, 1, x[1])$
EndAlgorithm

C.
Algorithm $\text{ceFace}(n, x)$:
 $i \leftarrow 1$
 $b \leftarrow x[i]$
 While $i < n$ **execute**
 If $b < x[i + 1]$ **then**
 $b \leftarrow x[i + 1]$
 EndIf
 $i \leftarrow i + 1$
 EndWhile
 Return b
EndAlgorithm

B.
Algorithm $\text{ceFace}(n, x)$:
 $i \leftarrow 1$
 While $i < n$ **execute**
 If $x[i] \geq x[i + 1]$ **then**
 Return *False*
 EndIf
 $i \leftarrow i + 1$
 EndWhile
 Return *True*
EndAlgorithm

D.
Algorithm $\text{ceFace}(n, x)$:
 $i \leftarrow 1$
 $b \leftarrow x[i]$
 While $i < n$ **execute**
 If $b > x[i + 1]$ **then**
 $b \leftarrow x[i + 1]$
 EndIf
 $i \leftarrow i + 1$
 EndWhile
 Return b
EndAlgorithm

2. Gegeben seien die Algorithmen $f_1(a, b)$ und $f_2(a, b)$, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10^4$).

Algorithm $f_1(a, b)$:
 While $b \neq 0$ **execute**
 $\text{temp} \leftarrow b$
 $b \leftarrow a \bmod b$
 $a \leftarrow \text{temp}$
 EndWhile
 Return a
EndAlgorithm
Algorithm $f_2(a, b)$:
 Return $f_1(a, b) = 1$
EndAlgorithm

In welchen Situationen gibt der Algorithmus $f_2(a, b)$ *True* zurück?

- A. Genau dann, wenn die Zahlen a und b *Fibonacci-Zahlen* sind.
- B. Genau dann, wenn die Summe der Ziffern der Zahl $a + b$ gleich der Summe der Ziffern der Zahl $a - b$ ist.
- C. Genau dann, wenn die Anzahl der geraden Ziffern von $a - b$ gleich der Anzahl der ungeraden Ziffern von $a + b$ ist.
- D. Genau dann, wenn die Zahlen a und b Teilerfremd sind.

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

```

Algorithm f(n):
  If n = 1 then
    Return 0
  Else
    Return (2 * n - 3) * (2 * n - 1) + f(n - 1)
  EndIf
EndAlgorithm

```

Welche der angegebenen Summen werden durch den Algorithmus $f(n)$ berechnet?

- A. $\sum_{k=1}^n (2k - 3) (2k - 1)$
- B. $\sum_{k=1}^{n-1} (2k - 1) (2k + 1)$
- C. $\sum_{k=2}^{n-1} (2k - 1) (2k + 1)$
- D. $\sum_{k=2}^n (2k - 3) (2k - 1)$

4. Gegeben sei der Algorithmus $f(x, y)$, wobei x und y natürliche Zahlen sind ($3 \leq x, y \leq 10^3$):

```

Algorithm f(x, y):
  If x = 1 then
    Return y
  EndIf
  If x MOD 2 = 0 then
    If x > 0 then
      Return f(x DIV 2, y * 2)
    Else
      Return 0
    EndIf
  EndIf
  Return y + f(x DIV 2, y * 2)
EndAlgorithm

```

Welcher der folgenden Algorithmen liefert für beliebige Werte von x und y denselben Wert wie der Algorithmus $f(x, y)$?

A.

```

Algorithm a(x, y):
  If x = 0 then
    Return 0
  EndIf
  If x MOD 2 = 0 then
    Return 2 * a(x DIV 2, y)
  EndIf
  Return y + a(x - 1, y)
EndAlgorithm

```

C.

```

Algorithm c(x, y):
  If x = 0 then
    Return 0
  EndIf
  Return y + c(x - 1, y)
EndAlgorithm

```

B.

```

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

```

D.

```

Algorithm d(x, y):
  s ← 0
  While x > 0 execute
    s ← s + y
    x ← x - 1
  EndWhile
  Return s
EndAlgorithm

```

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

```

Algorithm ceFace(x, n):
  For i ← 1, 10 execute
    For j ← 1, n - 1 execute
      If x[j] > x[j + 1] then
        tmp ← x[j]
        x[j] ← x[j + 1]
        x[j + 1] ← tmp
      EndIf
    EndFor
  EndFor
EndAlgorithm

```

Welche der folgenden Aussagen ist nach dem Aufruf von $ceFace(x, n)$ wahr?

- A. Die erste Position des Vektors x wird durch das minimale Element des Vektors belegt.
- B. Die letzte Position des Vektors x wird durch das maximale Element des Vektors belegt.
- C. Der Vektor x ist aufsteigend sortiert.
- D. Falls $n > 10$, dann werden die ersten 10 Elemente des Vektors x aufsteigend sortiert.

6. Gegeben seien die natürliche Zahl n ($1 \leq n \leq 10^4$) und der Vektor x dessen Elemente natürliche Zahlen sind ($x[1]$, $x[2]$, ..., $x[n]$, $1 \leq x[i] \leq 10^3$, für $i = 1, 2, \dots, n$). Der Algorithmus $\text{gcd}(a, b)$ gibt den größten gemeinsamen Teiler der natürlichen Zahlen a und b zurück.

Welche der folgenden Implementierungen des Algorithmus $\text{gcdVector}(x, n)$ liefert den größten gemeinsamen Teiler der n Elemente des Vektors x ?

- A.
- ```

Algorithm gcdVector(x, n):
 If n = 1 then
 Return 1
 EndIf
 Return gcd(x[n], gcdVector(x, n - 1))
EndAlgorithm

```
- B.
- ```

Algorithm gcdVector(x, n):
  result ← x[1]
  For i ← 2, n execute
    result ← gcd(x[i - 1], x[i])
  EndFor
  Return result
EndAlgorithm

```
- C.
- ```

Algorithm gcdVector(x, n):
 result ← gcd(x[1], x[n])
 i ← 2; j ← n - 1
 While i ≤ j execute
 result ← gcd(result, gcd(x[i], x[j]))
 i ← i + 1
 j ← j - 1
 EndWhile
 Return result
EndAlgorithm

```
- D.
- ```

Algorithm gcdVector2(x, n, i):
  If i = n then
    Return x[n]
  EndIf
  Return gcd(x[i], gcdVector2(x, n, i + 1))
EndAlgorithm

Algorithm gcdVector(x, n):
  Return gcdVector2(x, n, 1)
EndAlgorithm

```

7. Gegeben sei der Algorithmus $\text{afla}(n, x)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^4$) und x ein Vektor mit n ganzzahligen Elementen ist ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$):

```

Algorithm afla(n, x):
  M ← x[1]
  For i ← 1, n execute
    For j ← i, n execute
      For k ← j, n execute
        If M < x[i] + x[j] + x[k] then
          M ← x[i] + x[j] + x[k]
        EndIf
      EndFor
    EndFor
  EndFor
  Return M
EndAlgorithm

```

Wie hoch ist die Zeitkomplexität des Algorithmus?

- A. $O(3 * n)$
 B. $O(n^3)$
 C. $O(n / 3)$
 D. $O(\log_3 n)$

8. Gegeben sei der Algorithmus $\text{ceFace}(n)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 100$). Der Algorithmus $\text{min}(a, b)$ liefert den Minimalwert der Zahlen a und b .

```

Algorithm ceFace(n):
  c1 ← 0
  c2 ← 0
  For i ← 1, n execute
    v ← i
    While v MOD 2 = 0 execute
      c1 ← c1 + 1
      v ← v DIV 2
    EndWhile
    While v MOD 5 = 0 execute
      c2 ← c2 + 1
      v ← v DIV 5
    EndWhile
  EndFor
  Return min(c1, c2)
EndAlgorithm

```

Welche der folgenden Aussagen ist wahr?

- A. Der Algorithmus gibt die Anzahl der geraden Zahlen im Intervall $[1, n]$ zurück.
 B. Der Algorithmus gibt die Anzahl der durch 5 teilbaren Zahlen im Intervall $[1, n]$ zurück.
 C. Der Algorithmus gibt die Anzahl der aufeinanderfolgenden 0-Ziffern am Ende der Summe der ersten n natürlichen Zahlen zurück.
 D. Keine der obigen Antworten ist richtig.

9. Gegeben sei der Vektor x mit n Elementen ($2 \leq n \leq 10^5$) ganze Zahlen ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$). Der Algorithmus $\text{max}(a, b)$ gibt den Maximalwert der Zahlen a und b zurück.

Welche der folgenden Implementierungen des Algorithmus $\text{maxPePozitiePara}(x, n)$ gibt das größte Element an einer geraden Position im Vektor zurück?

- A.
- ```

Algorithm maxPePozitiePara(x, n):
 maxVal ← x[2]
 For i ← 4, n, 2 execute
 If x[i] > maxVal then
 maxVal ← x[i]
 EndIf
 EndFor
 Return maxVal
EndAlgorithm

```
- B.
- ```

Algorithm maxPePozitiePara(x, n):
    maxVal ← -100
    For i ← 1, n, 2 execute
        If x[i] > maxVal then
            maxVal ← x[i]
        EndIf
    EndFor
    Return maxVal
EndAlgorithm

```
- C.
- ```

Algorithm maxPePozitiePara2(x, n, i):
 If i > n then
 Return -100
 EndIf
 maxRest ← maxPePozitiePara2(x, n, i + 2)
 Return max(x[i], maxRest)
EndAlgorithm

```
- D.
- ```

Algorithm maxPePozitiePara2(x, n, i):
    If i > n then
        Return -100
    EndIf
    maxRest ← maxPePozitiePara2(x, n, i + 2)
    Return max(x[i], maxRest)
EndAlgorithm

```
- Algorithm maxPePozitiePara(x, n):
- ```

Return maxPePozitiePara2(x, n, 1)
EndAlgorithm

```
- Algorithm maxPePozitiePara(x, n):
- ```

Return maxPePozitiePara2(x, n, 2)
EndAlgorithm

```

10. Angenommen die folgenden Annahmen sind wahr:

1. Ana geht nur spazieren, wenn es sonnig ist.
2. Maria geht nur spazieren, wenn Ana spazieren geht.
3. Wenn es sonnig ist, geht Tudor spazieren.

Welche der folgenden Schlussfolgerungen ergeben sich aus den Hypothesen?

- A. Wenn Maria spazieren geht, geht auch Tudor spazieren.
- B. Wenn es nicht sonnig ist, dann geht Ana nicht spazieren.
- C. Wenn es sonnig ist, dann geht Maria spazieren.
- D. Wenn es nicht sonnig ist, dann geht Tudor nicht spazieren.

11. Gegeben sei die Zahl $x = 10000110111_{(2)}$ zur Basis 2 und die Zahl $y = 11011_{(4)}$ zur Basis 4.

Welcher ist der Wert der Summe $x + y$ zur Basis 10?

- A. 1079
- B. 1404
- C. 2285
- D. Keines von A, B und C

12. Gegeben sei der Algorithmus $\text{ceFace}(n, a)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^4$) und a ein Vektor dessen n Elemente natürliche Zahlen sind ($a[1], a[2], \dots, a[n]$, $1 \leq a[i] \leq 10^9$, für $i = 1, 2, \dots, n$).

```

1. Algorithm ceFace(n, a):
2.     m ← 0
3.     For i ← 1, n execute
4.         nr ← 1
5.         While a[i] > 9 execute
6.             nr ← nr * 10
7.             a[i] ← a[i] DIV 10
8.         EndWhile
9.         If m < a[i] * nr then
10.            m ← a[i] * nr
11.         EndIf
12.     EndFor
13.     Return m
14. EndAlgorithm

```

Welche der folgenden Aussagen ist richtig?

- A. Nach dem Aufruf $\text{ceFace}(5, [222, 2043, 29, 2, 20035])$ gibt der Algorithmus den Wert 2000 zurück.
- B. Unabhängig von den Werten von n und a liefert der Algorithmus $\text{ceFace}(n, a)$ eine Zahl, die im Vektor a nicht vorkommt.
- C. Nach dem Aufruf $\text{ceFace}(5, [34, 254, 21, 543, 123])$ gibt der Algorithmus den Wert 500 zurück.
- D. Wenn die Anweisung in Zeile 10 n -mal ausgeführt wird, folgt daraus, dass der anfangs gegebene Vektor a streng aufsteigend sortiert wurde.

13. Wir konstruieren einen ungerichteten Graphen wie folgt: Für jede natürliche Zahl n , $2 \leq n \leq 20$ ist, fügen wir einen mit dieser Zahl beschrifteten Knoten hinzu, und zwischen zwei beschrifteten Knoten x und y fügen wir eine Kante hinzu, wenn y ein echter Teiler von x ist. Welche der folgenden Aussagen ist wahr?

- A. Der konstruierte Graph hat 5 zusammenhängende Komponenten.
- B. Nur die Knoten 12, 18 und 20 haben einen maximalen Grad.
- C. Für zwei beliebige Zahlen x, y ($2 \leq x \leq y \leq n$), die nicht prim sind, ist der Grad von x größer oder gleich dem Grad von y .
- D. Es gibt nur einen Knoten mit dem Grad 1.

14. Gegeben sei der Algorithmus `ceva(n, v)`, wobei v ein Vektor von n ($10 \leq n \leq 10^5$) ganzen Zahlen ist ($v[1], v[2], \dots, v[n]$, $-10 \leq v[i] \leq 10$, für $i = 1, 2, \dots, n$). Der Algorithmus `zero(k)` gibt einen Vektor mit k Elementen zurück, die alle gleich Null sind.

```

1. Algorithm ceva(n, v):
2.   fr ← zero(21)
3.   s ← 0
4.   For i ← 1, n execute
5.     If fr[v[i] + 11] = 0 then
6.       s ← s + 1
7.     EndIf
8.     fr[v[i] + 11] ← 1
9.   EndFor
10.  Write s
11.  p ← 1
12.  For i ← 1, n execute
13.    If fr[v[i] + 11] = 1 then
14.      p ← p * v[i]
15.      fr[v[i] + 11] ← 0
16.    EndIf
17.  EndFor
18.  Return p
19. EndAlgorithm

```

Welche der folgenden Aussagen ist richtig?

- A. Wenn der in Zeile 10 angezeigte Wert größer als 10 ist, ist das zurückgegebene Ergebnis eine gerade Zahl.
- B. Wenn der in Zeile 10 angezeigte Wert größer als 11 ist, ist das zurückgegebene Ergebnis eine negative Zahl.
- C. Wenn der in Zeile 10 angezeigte Wert 21 ist, lautet das zurückgegebene Ergebnis $(10!)^2$.
- D. Der größtmögliche in Zeile 10 angezeigte Wert, bei dem das zurückgegebene Produkt nicht mit der Ziffer 0 endet, ist 16.

15. Gegeben sei der Algorithmus `f(x)`, wobei x eine natürliche Zahl ist ($0 \leq x \leq 10^5$).

```

Algorithm f(x):
  If x ≤ 1 then
    Return 1
  EndIf
  Return f(x - 1) + f(x - 2)
EndAlgorithm

```

Welche der folgenden Aussagen ist richtig?

- A. Nach dem Aufruf `f(10)` werden insgesamt 177 Aufrufe des Algorithmus `f(x)` durchgeführt, einschließlich des ersten Aufrufs.
- B. Der durch den Aufruf `f(x)` zurückgegebene Wert ist Teil der Fibonacci-Kette (1, 1, 1, 2, 3, 5, 8, 13, ...).
- C. Der Rückgabewert des Aufrufs `f(10)` ist 89.
- D. Der Aufruf `f(5)` führt zu insgesamt 4 Additionen.

16. Gegeben sei der Algorithmus `ceva(A, n, r, c, nr, x)`, wobei n eine natürliche Zahl ist ($1 < n \leq 10$), A ist eine $n \times n$ -Matrix deren Elemente natürliche Zahlen sind ($A[1][1], A[1][2], \dots, A[n][n]$), r, c, x sind natürliche Zahlen ($1 \leq r, c, x \leq 10$), und nr ist eine ganze Zahl ($-10^3 \leq nr \leq 10^3$). Wenn $n = 4$ und anfangs $A[3][2] = 5$ und $A[1][4] = 8$ ist, welche der folgenden Aufrufsequenzen wird die Elemente der Matrix A so verändern, dass $A[3][2]$ gleich 50 und $A[1][4]$ gleich 16 ist?

```

Algorithm ceva(A, n, r, c, nr, x):
  If (r + x - 1 ≤ n) AND (c + x - 1 ≤ n) then
    For i ← r, r + x - 1 execute
      For j ← c, c + x - 1 execute
        A[i][j] ← A[i][j] * nr
      EndFor
    EndFor
  EndIf
EndAlgorithm

```

- | | |
|---|--|
| <p>A.</p> <pre> ceva(A, n, 3, 2, 5, 1) ceva(A, n, 2, 1, 4, 3) ceva(A, n, 3, 3, 16, 2) </pre> <p>C.</p> <pre> ceva(A, n, 2, 3, 4, 2) ceva(A, n, 1, 1, 2, 4) ceva(A, n, 3, 1, 2, 1) ceva(A, n, 2, 1, 5, 3) </pre> | <p>B.</p> <pre> ceva(A, n, 1, 3, 2, 2) ceva(A, n, 3, 2, 2, 3) ceva(A, n, 2, 1, 10, 3) </pre> <p>D.</p> <p>Keine der Aufrufsequenzen führt zu der in der Anweisung angegebenen Veränderung.</p> |
|---|--|

17. Gegeben sei der Algorithmus $\text{cauta}(x, n, e)$, wobei x ein Vektor bestehend aus n ($1 \leq n \leq 10^4$) natürliche Zahlen ist ($x[1], x[2], \dots, x[n]$, für $i = 1, 2, \dots, n, 0 \leq x[i] \leq 10^4$) und e eine natürliche Zahl ist ($1 \leq e \leq 10^4$).

Welcher der folgenden Algorithmen liefert *True*, genau dann, wenn die Zahl e im Vektor x vorkommt?

- A.
- ```

Algorithm cauta(x, n, e):
 g ← False; i ← 1
 While i ≤ n execute
 g ← (x[i] = e)
 i ← i + 1
 EndWhile
 Return g
EndAlgorithm

```
- B.
- ```

Algorithm cauta(x, n, e):
  g ← False; i ← 1
  While NOT g AND i ≤ n execute
    g ← (x[i] = e)
    i ← i + 1
  EndWhile
  Return g
EndAlgorithm

```
- C.
- ```

Algorithm cauta(x, n, e):
 c ← 0
 For i ← 1, n execute
 If x[i] = e then
 c ← c + 1
 Else
 c ← c - 1
 EndIf
 EndFor
 Return n ≠ -c
EndAlgorithm

```
- D.
- ```

Algorithm cauta(x, n, e):
  g ← False; i ← 1
  While i ≤ n execute
    If x[i] < e + 1 AND x[i] MOD e = 0 then
      g ← True
    EndIf
    i ← i + 1
  EndWhile
  Return g
EndAlgorithm

```

18. Gegeben seien die natürlichen Zahlen m und n ($0 \leq m, n \leq 10$) und der Algorithmus $\text{Ack}(m, n)$, der den Wert der Ackerman Funktion für die Parameter m und n berechnet.

```

Algorithm Ack(m, n):
  If m = 0 then
    Return n + 1
  EndIf
  If m > 0 AND n = 0 then
    Return Ack(m - 1, 1)
  EndIf
  If m > 0 AND n > 0 then
    Return Ack(m - 1, Ack(m, n - 1))
  EndIf
EndAlgorithm

```

Wie viele Autoaufrufe werden durch den Aufruf $\text{Ack}(1, 4)$ ausgelöst?

- A. Es werden 9 Autoaufrufe erfolgen.
- B. Die gleiche Anzahl von Autoaufrufen wie $\text{Ack}(1, 2)$
- C. 4 Autoaufrufe mehr als beim Aufruf $\text{Ack}(1, 2)$
- D. Es werden 11 Autoaufrufe gemacht.

19. Gegeben sei der Algorithmus $P(s, n)$, wobei s eine Zeichenkette mit n Zeichen ($3 < n \leq 100$) ist. Der Algorithmus $\text{copy}(s, \text{poz}, \text{cate})$ gibt eine Teilfolge der Zeichenkette s zurück, die aus *cate*-Zeichen besteht und an der Position *poz* beginnt, wobei $0 \leq \text{cate} \leq n$ bzw. $1 \leq \text{poz} \leq n - \text{cate} + 1$. Bei Zeichenketten steht der Operator "+" für die Verkettungsoperation.

```

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

```

Welche der folgenden Aussagen ist wahr?

- A. Falls s_1 und s_2 zwei Strings der Länge n bzw. m sind, $n \neq m$, gibt der Algorithmus nach den Aufrufen $P(s_1, n)$ und $P(s_2, m)$ zwei Strings unterschiedlicher Länge zurück.
- B. Falls die Zeichenkette s der Länge n nur die Zeichen 'a', 'b' und 'c' enthält, gibt es 252 verschiedene Werte für s , für die der Algorithmus nach dem Aufruf $P(s, n)$ die Zeichenkette "ac" zurückgibt.
- C. Nach dem Aufruf $P(\text{"conkurs de admitere"}, 19)$ gibt der Algorithmus den Wert "de admic" zurück.
- D. Wenn die Zeichenkette s der Länge n nur die Zeichen '0' und '1' enthält, gibt es 72 verschiedene Werte für s , für die der Algorithmus nach dem Aufruf $P(s, n)$ die Zeichenkette "101" zurückgibt.

20. Wir schraffieren die Flächen, die von zwei Rechtecken A und B überdeckt werden. Jedes Rechteck wird durch seine linke untere Ecke und seine rechte obere Ecke definiert. Für das Rechteck A sind die Koordinaten $(ax1, ay1)$ und $(ax2, ay2)$, für das Rechteck B sind die Koordinaten $(bx1, by1)$ und $(bx2, by2)$, $0 \leq ax1, ay1, ax2, ay2, bx1, by1, bx2, by2 \leq 10^4$, $ax1 < ax2, ay1 < ay2, bx1 < bx2, by1 < by2$. Die Algorithmen $\min(a, b)$ und $\max(a, b)$ geben den Minimal- bzw. Maximalwert zwischen zwei Zahlen a und b zurück.

Welcher der folgenden Algorithmen berechnet den Flächeninhalt der schraffierten Fläche?

A.

```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                    bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    xSup ← max(0, min(ax2, bx2) - max(ax1, bx1))
    ySup ← max(0, min(ay2, by2) - max(ay1, by1))
    aSup ← xSup * ySup
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

B.

```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                    bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    If ax2 ≤ bx1 OR bx2 ≤ ax1 OR ay2 ≤ by1 OR
                                   by2 ≤ ay1 then
        aSup ← 0
    Else
        xSup ← min(ax2, bx2) - max(ax1, bx1)
        ySup ← min(ay2, by2) - max(ay1, by1)
        aSup ← xSup * ySup
    EndIf
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

C.

```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                    bx2, by2):
    arie ← (ax2 - ax1) * (ay2 - ay1)
    If ax2 ≤ bx1 OR bx2 ≤ ax1 OR ay2 ≤ by1
                                   OR by2 ≤ ay1 then
        arie ← arie + (bx2 - bx1) * (by2 - by1)
    EndIf
    Return arie
EndAlgorithm
```

D.

```
Algorithm calculArie(ax1, ay1, ax2, ay2, bx1, by1,
                    bx2, by2):
    a1 ← (ax2 - ax1) * (ay2 - ay1)
    a2 ← (bx2 - bx1) * (by2 - by1)
    aSup ← (min(ax2, bx2) - max(ax1, bx1)) +
            (min(ay2, by2) - max(ay1, by1))
    arie ← a1 + a2 - aSup
    Return arie
EndAlgorithm
```

21. Gegeben sei der Algorithmus $f(A, B, m, n, k)$, wobei m, n und k natürliche Zahlen sind ($1 \leq m, n, k \leq 100$), und A ein Vektor mit n ganzzahligen Elementen ist ($A[1], A[2], \dots, A[n]$, wobei $-100 \leq A[i] \leq 100$, für $i = 1, 2, \dots, n$), und B ein Vektor mit n Elementen ist, die alle gleich Null sind.

```
Algorithm f(A, B, m, n, k):
    aux ← 0
    For j ← 1, k - 1 execute
        aux ← aux + B[j] * A[j]
    EndFor
    If aux = m then
        Write "Solutie: "
        For j ← 1, k - 1 execute
            If B[j] = 1 then
                Write A[j], " "
            EndIf
        EndFor
        Write new line
    Else
        If k ≠ n + 1 then
            For i ← 0, 1 execute
                B[k] ← i
                f(A, B, m, n, k + 1)
            EndFor
        EndIf
    EndIf
EndAlgorithm
```

Wenn $A = [4, 5, 8, 9, 3, 12, 15, 7]$ ist, wird nach dem Aufruf von $f(A, B, 12, 8, 1)$ die erste angezeigte Lösung Solutie: 12 sein.

Wie lauten die zweite und dritte angezeigte Lösung?

A.

Solutie: 4 8
Solutie: 4 5 3

B.

Solutie: 4 8
Solutie: 5 7

C.

Solutie: 9 3
Solutie: 5 7

D.

Solutie: 9 3
Solutie: 4 8

22. Gegeben sei der Algorithmus oareCe($n1$), wobei $n1$ eine natürliche Zahl ist ($0 \leq n1 \leq 10^6$).

```

Algorithm oareCe(n1):
  n2 ← 0
  While n1 > n2 execute
    n3 ← n1 MOD 10
    n2 ← n2 * 10 + n3
    n1 ← n1 DIV 10
  EndWhile
  If n1 = n2 then
    Return True // (*)
  EndIf
  Return n1 = (n2 DIV 10)
EndAlgorithm

```

Welche der folgenden Aussagen ist wahr?

- A. Nach dem Aufruf von oareCe(1210) gibt der Algorithmus *False* zurück.
- B. Der Aufruf oareCe(282) führt zur Ausführung der mit (*) markierten Zeile
- C. Es gibt 4 verschiedene Werte von n , eine natürliche Zahl im Intervall [101, 500], für die der Aufruf oareCe(n) zur Ausführung der mit (*) gekennzeichneten Zeile führt.
- D. Wenn der Algorithmus in der Form oareCe((($i * 6$) DIV 7) * (($j * 7$) DIV ($i + j$))) aufgerufen wird, wobei $i = 1$ und $j = 2$ ist, gibt der Algorithmus *True* zurück.

23. Gegeben sei der Algorithmus interessant(a , b , c , n), wobei c und n natürliche Zahlen sind ($1 \leq n \leq 100$, $1 \leq c \leq 10^4$), und a und b zwei Vektoren der Länge n mit ganzzahligen Elementen sind ($a[1]$, $a[2]$, ..., $a[n]$ und $b[1]$, $b[2]$, ..., $b[n]$, $1 \leq a[i]$, $b[i] \leq 100$, für $i = 1, 2, \dots, n$). Der Algorithmus max(x , y) gibt den maximalen Wert der Zahlen x und y zurück.

```

Algorithm interessant(a, b, c, n):
  If n = 0 OR c = 0 then
    Return 0
  EndIf
  If a[n] > c then
    Return interessant(a, b, c, n - 1)
  EndIf
  x ← b[n] + interessant(a, b, c - a[n], n - 1)
  y ← interessant(a, b, c, n - 1)
  Return max(x, y)
EndAlgorithm

```

Welcher Wert wird vom Aufruf interessant([2, 3, 1], [6, 10, 3], 5, 3) zurückgegeben?

- A. 13
- B. 10
- C. 16
- D. 19

24. Gegeben sei der Algorithmus divide(x , y), der den ganzzahligen Quotient der Division von x durch y berechnet, wobei x und y ganze Zahlen sind ($-10^5 \leq x, y \leq 10^5$, $y \neq 0$). Die Operation $a \ll n$ verschiebt die Bits der Zahl a um n Stellen nach links, was einer Multiplikation der Zahl mit 2^n entspricht.

```

Algorithm divide(x, y):
  negativ ← 1
  If x < 0 then
    negativ ← -negativ
    x ← -x
  EndIf
  If y < 0 then
    negativ ← -negativ
    y ← -y
  EndIf
  cat ← 0
  While x ≥ y execute
    temp ← y
    multiplu ← 1
    While x ≥ (temp << 1) execute
      temp ← temp << 1
      ... // (1)
    EndWhile
    x ← x - temp
    ... // (2)
  EndWhile
  ... // (3)
  Return cat
EndAlgorithm

```

Welche Anweisungen müssen in den mit (1), (2) und (3) gekennzeichneten Zeilen eingefügt werden, damit der Algorithmus divide(x , y) das richtige Ergebnis liefert?

- A.
 - (1): multiplu ← multiplu << 1
 - (2): cat ← cat + multiplu
 - (3): If negativ = -1 then
 - cat ← -cat
- B.
 - (1): multiplu ← multiplu << 1
 - (2): cat ← cat + 1
 - (3): cat ← -negativ * cat
- C.
 - (1): multiplu ← multiplu * 2
 - (2): cat ← cat + 1
 - (3): cat ← -negativ * cat
- D.
 - (1): multiplu ← multiplu * 2
 - (2): cat ← cat + multiplu
 - (3): cat ← negativ * cat

BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Zulassung - 17. Juli 2025

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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