

Zulassungsprüfung - 06. September 2024
Schriftliche Prüfung in Informatik

WICHTIGER HINWEIS:

Falls nicht anders erklärt:

- Alle arithmetischen Operationen werden mit unbegrenzten Datentypen durchgeführt (kein *Über-/Unterlauf*).
- Die Indexnummerierung aller Vektoren, Arrays und Strings beginnt bei 1.
- Alle Einschränkungen beziehen sich auf die aktuellen Parameterwerte zum Zeitpunkt des ersten Aufrufs.
- Eine Teilsequenz eines Vektors besteht aus Elementen, die aufeinanderfolgende Positionen im Vektor einnehmen.
- Eine Teilzeichenkette eines Vektors/einer Zeichenkette besteht aus Elementen, die sich nicht unbedingt an aufeinanderfolgenden Positionen im Vektor/der Zeichenkette befinden, in der Reihenfolge, in der sie in der gegebenen Zeichenkette erscheinen.
- Wenn mehrere aufeinanderfolgende Zuweisungsanweisungen in der gleichen Zeile erscheinen, werden sie durch ";" voneinander getrennt.

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

```
Algorithm decide(n, x):  
  b ← True  
  i ← 1  
  While b AND (i < n) execute  
    b ← (x[i] < x[i + 1])  
    i ← i + 1  
  EndWhile  
  Return b  
EndAlgorithm
```

Für welchen der folgenden Fälle gibt der Algorithmus *True* zurück?

- A. Falls der Vektor x aus den Werten 1, 2, 3, ..., 10 besteht
- B. Wenn der Vektor x streng aufsteigend ist
- C. Wenn der Vektor x keine negativen Elemente hat
- D. Falls im Vektor x die negativen Elemente vor den positiven Elementen vorkommen

2. Gegeben sei der Algorithmus $afiseaza(n, a)$, wobei n eine natürliche Zahl ($1 \leq n \leq 10^3$) und a ein Vektor mit n ganzzahligen Elementen ist ($a[1], a[2], \dots, a[n]$, wobei $-100 \leq a[i] \leq 100$, für $i = 1, 2, \dots, n$):

```
Algorithm afiseaza(n, a):  
  i ← 1; j ← n  
  While i ≤ j execute  
    If a[i] < a[j] then  
      Write a[i], " "  
      i ← i + 1  
    Else  
      Write a[j], " "  
      j ← j - 1  
    EndIf  
  EndWhile  
EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

- A. Wenn der Eingabevektor aufsteigend sortiert ist, werden die Werte im Ausgabevektor in absteigender Reihenfolge angezeigt.
- B. Wenn der Eingabevektor absteigend sortiert ist, ist das letzte angezeigte Element das maximale Element.
- C. Wenn $n = 10$ und $a = [0, 2, 4, 6, 8, 10, 8, 6, 4, 2]$ ist, werden die Werte im Vektor in aufsteigender Reihenfolge angezeigt.
- D. Befindet sich das maximale Element an der ersten Position, werden die Werte im Vektor in umgekehrter Reihenfolge angezeigt.

3. Welche Beziehung besteht zwischen den Zahlen $X = 6543_{(8)}$ zur Basis 8 und $Y = CEF_{(16)}$ zur Basis 16?

- A. $X > Y$
- B. $X \leq Y$
- C. $X \geq Y$
- D. $X = Y$

4. Gegeben sei der Algorithmus $f(n)$, wobei n die natürliche Zahl n ($1 \leq n \leq 15$) ist.

```
1. Algorithm f(n):  
2.   x ← 10; y ← 13  
3.   While n ≠ 0 execute  
4.     z ← (x + y) MOD 2  
5.     n ← n DIV 2  
6.     If z MOD 2 = 0 then  
7.       x ← (x * 3 + y * 4) MOD z  
8.       y ← (y + x) * z  
9.     Else  
10.      x ← x + 1  
11.      y ← y - 1  
12.     EndIf  
13.   EndWhile  
14.   Return z  
15. EndAlgorithm
```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Algorithmus gibt für jede natürliche Zahl $1 \leq n \leq 15$ denselben Wert zurück.
- B. Der Algorithmus gibt unterschiedliche Werte für die natürlichen Zahlen n mit der Eigenschaft $1 \leq n \leq 10$ zurück.
- C. Ist $n = 11$, gibt der Algorithmus 0 zurück.
- D. Wenn wir die Anweisung in der Zeile 10 in $x \leftarrow x - 1$ und die Anweisung in der Zeile 11 in $y \leftarrow y + 1$ ändern, gibt der Algorithmus für jede natürliche Zahl $1 \leq n \leq 15$ denselben Wert wie in der ursprünglichen Version zurück.

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

```

Algorithm numere(n, x):
  i ← 1; nr ← n
  While i ≤ n execute
    If (x[i] MOD 10) MOD 2 = 0 then
      nr ← nr + 1
    Else
      nr ← nr - 1
    EndIf
    i ← i + 1
  EndWhile
  Return nr = n
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Aufruf $\text{numere}(3, [1, 2, 3])$ gibt *True* zurück.
- B. Der Aufruf $\text{numere}(3, [1, -2, 3])$ gibt *False* zurück.
- C. Der Aufruf $\text{numere}(4, [1, 2, 3, -4])$ gibt *False* zurück.
- D. Der Aufruf $\text{numere}(4, [1, 2, 3, 4])$ gibt *True* zurück.

6. Gegeben sei der Algorithmus $\text{ceFace}(v, n)$, wobei v ein Vektor aus n ($1 \leq n \leq 10^4$) natürlichen Zahlen ist ($v[1], v[2], \dots, v[n]$, wobei $1 \leq v[i] \leq 10^4$ für $i = 1, 2, \dots, n$).

```

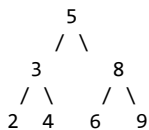
Algorithm ceFace(v, n):
  a ← 0; b ← 1
  For i ← n, 2, -1 execute
    If v[i] = v[i - 1] + 1 then
      b ← b + 1
    Else
      b ← 1
    EndIf
    If b > a then
      a ← b
    EndIf
  EndFor
  Return a
EndAlgorithm

```

Was gibt der $\text{ceFace}(v, n)$ Algorithmus zurück?

- A. Die Länge der längsten Teilfolge von aufeinanderfolgenden steigenden Zahlen im Vektor v .
- B. Die Länge der längsten Teilfolge aufeinanderfolgender absteigender Zahlen im Vektor v .
- C. Die Anzahl der aufsteigenden Teilfolgen im Vektor v .
- D. Die Länge der längsten Teilfolge von aufeinanderfolgenden aufsteigender Zahlen im Vektor v .

7. Gegeben sei der folgende Binärbaum:



Welche der folgenden Knotenfolgen entspricht der Durchquerung des Baums in Postorder?

- A. 2, 3, 4, 5, 6, 8, 9
- B. 4, 3, 2, 9, 8, 6, 5
- C. 2, 4, 3, 6, 9, 8, 5
- D. 9, 6, 8, 5, 3, 2, 4

8. Gegeben sei der Algorithmus $\text{prelucrare}(n, m, x)$, wobei n und m natürliche Zahlen sind ($1 \leq n \leq 100, 1 < m \leq 100$) und x eine Matrix mit $n * m$ Elementen aus natürlichen Zahlen ist ($x[1][1], x[1][2], \dots, x[n][m]$, wobei anfangs $x[i][j] = 0$, für $i = 1, 2, \dots, n; j = 1, 2, \dots, m$):

```

Algorithm prelucrare(n, m, x):
  k ← 1; i ← k
  While i ≤ n execute
    j ← k + 1
    While j ≤ m execute
      If k MOD 2 = 0 then
        x[i][j] ← k * k
      EndIf
      Write x[i][j], " "
      k ← k + 1; j ← j + 1
    EndWhile
    i ← i + 1
  EndWhile
EndAlgorithm

```

Was zeigt dieser Algorithmus an?

- A. Eine Zeichenkette mit n Werten.
- B. Wenn m gerade ist, eine Reihe von Werten, bei denen sich der Wert 0 mit Werten abwechselt, die gerade perfekte Quadrate darstellen, und der erste und letzte Wert 0 ist.
- C. Eine Folge von $m - 1$ Werten.
- D. Eine Reihe von Werten, bei denen sich der Wert 0 mit Werten abwechselt, die ungerade perfekte Quadrate darstellen.

9. Gegeben seien zwei Bit-Vektoren, x mit n Elementen und y mit m Elementen, wobei n und m natürliche Zahlen sind, die nicht Null sind ($0 < n, m \leq 64$). Die Elemente der Vektoren sind 0 oder 1. Seien $b1$ und $b2$ zwei Bits, für die wir die Operation $op(b1, b2) = \begin{cases} 0 & \text{falls } b1 = b2 \\ 1 & \text{sonst} \end{cases}$ definieren. Wir definieren die Operation os als Anwendung von op auf die Elemente von x und y , aber ausgehend vom Ende der Vektoren aus (wir wenden op also zunächst auf $x[n]$ und $y[m]$ an). Wenn die beiden Vektoren eine unterschiedliche Anzahl von Elementen haben, bleiben die Elemente am Anfang des längeren Vektors, die im anderen Vektor kein Paar haben, unverändert. Für die Vektoren $[1, 1, 1, 1, 0, 1, 0]$ und $[1, 1, 1, 1, 1, 0]$ lautet das Ergebnis der Operation os beispielsweise $[1, 1, 1, 0, 1, 0, 0, 0]$. Der Algorithmus erzeugt einen Vektor r mit $\max(n, m)$ Elementen.

Das Ziel des Algorithmus OperatieSpeciala (x, n, y, m) ist es, die oben beschriebene Operation os zu implementieren und den resultierenden Vektor und seine Länge zurückzugeben. Der Algorithmus zero(k) gibt einen Vektor mit k Elementen zurück, die alle gleich Null sind.

```

1. Algorithm OperatieSpeciala(x, n, y, m):
2.   length ← n
3.   lenF ← m
4.   r ← Zero(m)
5.   If m < n then
6.     length ← m
7.     lenF ← n
8.     r ← Zero(n)
9.   EndIf
10.  For i ← 1, length execute
11.    If (x[i] + y[i]) MOD 2 = 0 then
12.      r[i] ← 0
13.    Else
14.      r[i] ← 1
15.    EndIf
16.  EndFor
17.  For i ← length + 1, m execute
18.    r[i] ← y[i]
19.  EndFor
20.  For i ← length + 1, n execute
21.    r[i] ← x[i]
22.  EndFor
23.  Return r, lenF
24. EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Algorithmus OperatieSpeciala(x, n, y, m) implementiert die Operation os korrekt und gibt den resultierenden Vektor und dessen Länge zurück.
- B. Für die Eingabedaten, für die die Anweisungen in den Zeilen 18 und 21 gleich oft ausgeführt werden, ist das zurückgegebene Ergebnis korrekt.
- C. Die Implementierung würde korrekt werden, wenn wir die Anweisung in Zeile 11 durch **If (x[n - i] + y[m - i]) MOD 2 = 0 then** ersetzen.
- D. Das Ergebnis des Algorithmus OperatieSpeciala(x, n, y, m) ist nicht korrekt, und der zurückgegebene Vektor enthält die Elemente in umgekehrter Reihenfolge.

10. Gegeben sei der Algorithmus ceFace(x, m, y, n), wobei x eine Zeichenfolge der Länge m ($1 \leq m \leq 100$) und y eine Zeichenfolge der Länge n ($1 \leq n \leq 100$) ist, so dass $m < n$.

```

1. Algorithm ceFace(x, m, y, n):
2.   i ← 1
3.   ok ← True
4.   While ok AND i ≤ m execute
5.     If i ≤ m AND x[i] ≠ y[i] then
6.       ok ← False
7.     Else
8.       i ← i + 1
9.     EndIf
10.  EndWhile
11.  _____
12. EndAlgorithm

```

Welche Anweisung sollte in der Zeile 11 stehen, damit der Algorithmus *True* zurückgibt, wenn der String x das Präfix des Strings y ist? Beispiel: Wenn $x = "abc"$ und $y = "abcd"$, ist x das Präfix von y und der Algorithmus gibt *True* zurück.

- A. Return (i = m) OR (i = n)
- B. Return i = m
- C. Return i > m
- D. Return ok

11. Gegeben sei eine Matrix A mit m Zeilen und n Spalten ($A[1][1], A[1][2], \dots, A[m][n]$), wobei m und n natürliche Zahlen sind ($1 < m \leq 25, 1 < n \leq 25$) und $1 \leq A[i][j] \leq 10^3$, für $i = 1, 2, \dots, m; j = 1, 2, \dots, n$.

Welcher der folgenden Algorithmen gibt die Summe der Elemente der Spalte k ($1 < k \leq n$) zurück?

A. **Algorithm** suma(A, m, n, k):
 $s \leftarrow 0$
For $i \leftarrow n, 1, -1$ **execute**
 $s \leftarrow s + A[i][k]$
EndFor
Return s
EndAlgorithm

B. **Algorithm** suma(A, m, n, k):
 $s \leftarrow 0; i \leftarrow 1$
While $i \leq m$ **execute**
 $s \leftarrow s + A[i][k]$
 $i \leftarrow i + 1$
EndWhile
Return s
EndAlgorithm

C. **Algorithm** suma(A, m, n, k):
 $s \leftarrow 0$
For $i \leftarrow 1, m$ **execute**
 $s \leftarrow s + A[k][i]$
EndFor
Return s
EndAlgorithm

D. **Algorithm** suma(A, m, n, k):
 $s \leftarrow 0; k \leftarrow 1$
While $k \leq n$ **execute**
 $s \leftarrow s + A[k][k]$
 $k \leftarrow k + 1$
EndWhile
Return s
EndAlgorithm

12. Gegeben sei der Algorithmus $F(n)$, wobei n die natürliche Zahl n ist ($1 \leq n \leq 10^6$). Der Algorithmus $\text{sqrt}(n)$ gibt die quadratische Wurzel von n zurück und hat die Komplexität $O(1)$. Die Notation $[a]$ steht für den ganzen Teil des Wertes von a . Der Operator $"/$ steht für reelle Division, z. B.: $3 / 2 = 1,5$.

```

Algorithm F(n):
  If n = 1 then
    Return 1
  EndIf
  i ← [n / sqrt(n)]
  Return 1 + F(i)
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Algorithmus $F(n)$ hat die Zeitkomplexität $O(\log \log \log n)$.
- B. Der Aufruf von $F(200)$ gibt den Wert 4 zurück.
- C. Der Aufruf von $F(250)$ gibt den Wert 5 zurück.
- D. Der Algorithmus $F(n)$ hat die Zeitkomplexität $O(1)$.

13. Gegeben sei der Algorithmus $\text{check}(n, x)$, wobei n eine natürliche Zahl ($1 \leq n \leq 10^4$) 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 check(n, x):
  If n < 3 then
    Return False
  EndIf
  p ← select(n, x)
  If p = 1 OR p = n then
    Return False
  EndIf
  For i ← 2, p execute
    If x[i] ≥ x[i - 1] then
      Return False
    EndIf
  EndFor
  For i ← p + 1, n - 1 execute
    If x[i] ≥ x[i + 1] then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

```

Algorithm select(n, x):
  r ← 0
  v ← x[1]
  For i ← 2, n execute
    If x[i] < v then
      r ← i
      v ← x[i]
    EndIf
  EndFor
  Return r
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Wenn der Vektor x absteigend geordnet ist und mindestens 3 Elemente hat, gibt der Algorithmus $\text{check}(n, x)$ *True* zurück.
- B. Wenn der Vektor $x = [12, 10, 8, 5, 9, 11, 15, 18]$ und $n = 8$ ist, gibt der Algorithmus $\text{check}(n, x)$ *True* zurück.
- C. Wenn der Vektor $x = [20, 10, 5, 1, 2, 4, 6, 10, 8]$ und $n = 9$ ist, gibt der Algorithmus $\text{check}(n, x)$ *False* zurück.
- D. Wenn der Vektor x streng aufsteigend geordnet ist und mindestens 3 Elemente hat, gibt der Algorithmus $\text{check}(n, x)$ *True* zurück.

14. Gegeben sei der Algorithmus $f(x, n)$, wobei n eine natürliche Zahl ($3 \leq n \leq 10^4$) und x ein Vektor aus n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^4$, für $i = 1, 2, \dots, n$). Mit $[\]$ bezeichnen wir einen leeren Vektor, und mit $[a, b]$ bezeichnen wir einen Vektor mit 2 Elementen a und b .

```

Algorithm f(x, n):
  If n < 2 then
    Return []
  EndIf
  If n = 2 then
    If x[1] > x[2] then
      Return [x[1], x[2]]
    Else
      Return [x[2], x[1]]
    EndIf
  EndIf
  y ← f(x, n - 1)
  If x[n] > y[1] then
    Return [x[n], y[1]]
  Else
    If x[n] > y[2] then
      Return [y[1], x[n]]
    Else
      Return y
    EndIf
  EndIf
EndAlgorithm

```

Was gibt der Algorithmus für den Aufruf $f([4, 15, 5, 8, 10, 18, 16, 19, 1, 12], 10)$ zurück?

- A. $[19, 18]$
- B. $[18, 19]$
- C. $[16, 19]$
- D. $[19, 16]$

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

```

Algorithm numere(x, n, e):
  i ← 1
  c ← 0
  b ← True
  If n MOD 2 = 0 then
    Return False
  EndIf
  While (i ≤ n) AND b execute
    If x[i] < e then
      c ← c + 1
    Else
      b ← False
    EndIf
    i ← i + 1
  EndWhile
  Return c = (n - i + 1)
EndAlgorithm

```

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

- A. Wenn der Vektor eine gerade Anzahl von Elementen hat und absteigend geordnet ist, bis einschließlich des Elements mit dem Wert e , das sich an der Position $n \text{ DIV } 2$ befindet.
- B. Wenn der Vektor eine ungerade Anzahl von Elementen hat und streng aufsteigend geordnet ist, bis einschließlich des Elements mit dem Wert e , das sich an der Position $n \text{ DIV } 2 + 1$ befindet.
- C. Wenn der Vektor eine ungerade Anzahl von Elementen hat und absteigend geordnet ist, bis einschließlich des Elements mit dem Wert e , das sich an der Position $n \text{ DIV } 2 + 1$ befindet.
- D. Wenn der Vektor eine ungerade Anzahl von Elementen hat und der Wert e an der Position $n \text{ DIV } 2 + 1$ steht und es vor e nur kleinere Werte und nach e nur größere Werte gibt.

16. Welcher der folgenden Algorithmen zeigt die Darstellung der Zahl a zur Basis b , wobei a, b gegebene natürliche Zahlen zur Basis 10 sind ($1 \leq a \leq 10^4, 2 \leq b \leq 9, a > b$)?

A.

```

Algorithm afiseaza(a, b):
  If a ≠ 0 then
    Write a MOD b
    afiseaza(a DIV b, b)
  EndIf
EndAlgorithm

```

B.

```

Algorithm afiseaza(a, b):
  If a ≠ 0 then
    afiseaza(a DIV b, b)
    Write a MOD b
  EndIf
EndAlgorithm

```

C.

```

Algorithm afiseaza(a, b):
  While a > 0 execute
    Write a MOD b
    a ← a DIV b
  EndWhile
EndAlgorithm

```

D.

```

Algorithm afiseaza(a, b):
  nrNou ← 0
  putere ← 1
  While a > 0 execute
    nrNou ← nrNou + (a MOD b) * putere
    a ← a DIV b
    putere ← putere * b
  EndWhile
  Write nrNou
EndAlgorithm

```

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

```

Algorithm f(x, y):
  If x = y then
    Write "start: "
  Else
    If x MOD y = 0 then
      f(x + 1, y + 2)
    Else
      If (x DIV y) MOD 2 = 0 then
        f(x + 2, y + 1)
        Write "*"
      Else
        f(x - 1, y + 1)
        Write "#"
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Die folgenden Aufrufe $f(12, 15)$ und $f(8, 12)$ zeigen nicht dieselbe Zeichenfolge an
- B. Als Ergebnis der Aufrufe $f(15, 12)$ und $f(12, 8)$ wird die gleiche Zeichenfolge angezeigt
- C. Nach dem Aufruf $f(17, 23)$ wird kein "#" -Zeichen angezeigt
- D. Als Ergebnis des Aufrufs von $f(23, 17)$ enthält die angezeigte Zeichenkette mindestens ein "#" Zeichen

18. Gegeben sei der Algorithmus $\text{decide}(n, x, t)$, wobei x ein Vektor aus n natürlichen Zahlen ist ($2 \leq n \leq 10^4$, $1 \leq x[i] \leq 10^4$, $i = 1, 2, \dots, n$) und t eine natürliche Zahl ist ($1 \leq t \leq 10^4$).

```

Algorithm decide(n, x, t):
  left ← 1; right ← n
  While x[left] + x[right] ≠ t execute
    If x[left] + x[right] < t then
      left ← left + 1
    Else
      right ← right - 1
    EndIf
  EndWhile
  Return left, right
EndAlgorithm

```

In welcher der folgenden Situationen bestimmt der Algorithmus $\text{decide}(n, x, t)$ die Indizes $left, right$ ($1 \leq left < right \leq n$) so, dass $x[left] + x[right] = t$ ist?

- A. Genau dann, wenn der Vektor x unterschiedliche Zahlen enthält.
- B. Genau dann, wenn der Vektor x aufsteigend geordnet ist.
- C. Wenn der Vektor x absteigend geordnet ist und unterschiedliche Zahlen enthält.
- D. Wenn der Vektor x aufsteigend geordnet ist und mindestens ein Paar von Elementen mit der Summe t im Vektor vorhanden ist.

19. Gegeben sei der Algorithmus $\text{perechi}(x, y)$, wobei x und y natürliche Zahlen ungleich Null sind ($1 \leq x, y \leq 100$):

```

Algorithm perechi(x, y):
  nr ← 0; d ← 2
  While d ≤ x AND d ≤ y execute:
    If (x MOD d = 0) AND (y MOD d = 0) then
      nr ← nr + 1
      x ← x DIV d
      y ← y DIV d
    Else
      d ← d + 1
    EndIf
  EndWhile
  Write nr, " ", x, " ", y
EndAlgorithm

```

In welcher der folgenden Antwortmöglichkeiten gibt es nur Zahlenpaare (x, y) , für die der Algorithmus $\text{perechi}(x, y)$ die Werte 1 7 11 anzeigt.

- A. (14, 22), (21, 33), (35, 55), (49, 77)
- B. (7, 11), (14, 22), (21, 33), (28, 44)
- C. (1, 7), (1, 11)
- D. (2, 2), (3, 3), (4, 4), (5, 5)

20. Gegeben sei der Algorithmus $\text{first}(x, n)$, wobei n eine von Null verschiedene natürliche Zahl ist ($2 \leq n \leq 10^4$) und x ein Vektor aus n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n]$, wobei $1 \leq x[i] \leq 10^4$, für $i = 1, 2, \dots, n$).

```

01. Algorithm first(x, n):
02.   f1 ← False
03.   f2 ← False
04.   For i ← 1, n execute
05.     If x[i] = 1 then
06.       f1 ← True
07.     EndIf
08.     If x[i] = n then
09.       f2 ← True
10.     EndIf
11.     If x[i] ≥ n then
12.       x[i] ← 1
13.     EndIf
14.   EndFor
15.   If NOT f1 then
16.     Return 1
17.   EndIf
18.   For i ← 1, n execute
19.     ...
20.   EndFor
21.   If f2 then
22.     x[n] ← n
23.   EndIf
24.   For i ← 1, n execute
25.     ...
26.     Return i
27.   EndIf
28. EndFor
29. Return n + 1
30. EndAlgorithm

```

Was sollte man in die Zeilen 19 und 25 ersetzen, damit der Algorithmus die kleinste natürliche Zahl liefert, die nicht im Vektor x enthalten ist?

- A.
 - 19: $x[x[i] \text{ MOD } (n + 1)] \leftarrow x[x[i] \text{ MOD } (n + 1)] + n$
 - 25: **If** $x[i] \text{ DIV } (n + 1) = 0$ **then**
- B.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow x[x[i] \text{ MOD } n] + n$
 - 25: **If** $x[i] \text{ DIV } n = 0$ **then**
- C.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow 1$
 - 25: **If** $x[i] = 1$ **then**
- D.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow x[x[i] \text{ MOD } n] + n$
 - 25: **If** $x[i] \text{ MOD } n = 0$ **then**

21. Gegeben sei der Algorithmus $\text{ceFace}(\text{arr}, n)$, wobei arr ein Vektor mit n ($1 \leq n \leq 100$) ganzzahligen Elementen ist ($\text{arr}[1], \text{arr}[2], \dots, \text{arr}[n]$), wobei $-10^5 \leq \text{arr}[i] \leq 10^5$ für $i = 1, 2, \dots, n$).

```

Algorithm ceFace(arr, n):
  sum ← 0
  For i ← 1, n execute
    sum ← sum + arr[i]
  EndFor
  If sum MOD 2 ≠ 0 then
    Return False
  EndIf
  Return auxiliar(arr, n, sum DIV 2)
EndAlgorithm

```

```

Algorithm auxiliar(arr, n, sum):
  If sum = 0 then
    Return True
  EndIf
  If n = 1 AND sum ≠ 0 then
    Return False
  EndIf
  If arr[n - 1] > sum then
    Return auxiliar(arr, n - 1, sum)
  EndIf
  Return auxiliar(arr, n - 1, sum) OR
    auxiliar(arr, n - 1, sum - arr[n - 1])
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- Der Aufruf $\text{ceFace}([11, 5, 6, 22, 0, 7, 6, 13], 8)$ gibt *True* zurück.
- Der Aufruf $\text{ceFace}([-5, -6, -22, -7, -6, -6, -13], 6)$ liefert **NICHT** *True*.
- Wenn der Vektor arr nur negative Werte enthält, wird der $\text{auxiliar}(\text{arr}, n, \text{sum})$ in einen unendlichen Zyklus eintreten.
- Genau dann, wenn die Elemente in arr in zwei Gruppen aufgeteilt werden können, so dass der Mittelwert der Elemente in den beiden Gruppen gleich ist, gibt der Algorithmus $\text{ceFace}(\text{arr}, n)$ *True* zurück.

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

- Algorithm $f(n, x)$:
- $s1 \leftarrow h(n, x)$
- For $i \leftarrow 1, 2 * n$ execute
- $x[((i + 1) \text{ MOD } n) + 1] \leftarrow g(x[(i \text{ MOD } n) + 1], x[((i + 1) \text{ MOD } n) + 1])$
- EndFor
- $s2 \leftarrow h(n, x)$
- Return $x[n]$
- EndAlgorithm

```

Algorithm g(a, b):
  If a * b = 0 then
    Return a + b
  EndIf
  If a = b then
    Return a
  EndIf
  If a > b then
    Return g(a - b, b)
  EndIf
  Return g(a, b - a)
EndAlgorithm

```

```

Algorithm h(n, x):
  s ← 0
  For i ← 1, n execute
    s ← s + x[i]
  EndFor
  Return s
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- Im Falle des Aufrufs $f(6, [12, 16, 80, 40, 28, 144])$ gibt der Algorithmus den Wert 4 zurück.
- Für jeden Eingabevektor ist der Wert $s1$ (berechnet in Zeile 2 des Algorithmus $f(n, x)$) strikt größer als der Wert $s2$ (berechnet in der Zeile 6 des Algorithmus $f(n, x)$).
- Ersetzt man im Algorithmus $f(n, x)$ die Anweisungen in den Zeilen 3, 4 und 5 durch die untenstehende Sequenz, so hat der Vektor x am Ende des Algorithmus $f(n, x)$ den gleichen Inhalt wie im ursprünglichen Algorithmus.


```

For j ← 1, 2 execute
  For i ← 1, n - 1 execute
    x[i + 1] ← g(x[i], x[i + 1])
  EndFor
EndFor

```
- Es gibt einen Eingabevektor mit n Elementen, für den die Zeitkomplexität des Algorithmus $f(n, x)$ $O(n)$ ist.

23. Gegeben sei eine gerade n ($2 \leq n \leq 12$) natürliche Zahl n . Wir wollen in der Zeichenkette x alle Zeichenketten erzeugen, die aus n runden Klammern bestehen, die sich korrekt öffnen und schließen. Der Algorithmus $\text{paranteze}(i, \text{desc}, \text{inc}, x, n)$ wird als $\text{paranteze}(2, 1, 0, x, n)$ aufgerufen, wobei bekannt ist, dass die Initialisierungen $x[1] \leftarrow '('$ und $x[n] \leftarrow ')'$ aufgetreten sind. Der Algorithmus $\text{afisare}(n, x)$ zeigt die Zeichenkette x der Länge n an.

```

1. Algorithm paranteze(i, desc, inc, x, n):
2.   If i = n then
3.     afisare(n, x)
4.   Else
5.     If ____ then
6.       x[i] ← '('
7.       paranteze(i + 1, desc + 1, inc, x, n)
8.     EndIf
9.     If ____ then
10.      x[i] ← ')'
11.      paranteze(i + 1, desc, inc + 1, x, n)
12.    EndIf
13.  EndIf
14. EndAlgorithm

```

Was muss in die folgenden Zeilen eingetragen werden, damit der Algorithmus nur die richtigen Zeichenfolgen anzeigt?

- A. Die Zeile 5 muss mit $desc < n$ und die Zeile 9 muss mit $inc < desc$ vervollständigt werden.
- B. Die Zeile 5 muss mit $desc < n \text{ DIV } 2$ und die Zeile 9 muss mit $inc < desc$ vervollständigt werden.
- C. Die Zeile 5 muss mit $desc < n$ und die Zeile 9 muss mit $inc < n \text{ DIV } 2$ vervollständigt werden.
- D. Unabhängig von den Vergleichen, mit denen die Zeilen 5 und 9 abgeschlossen werden, zeigt der Algorithmus nicht alle Zeichenfolgen wie gewünscht an.

24. Im Sportunterricht stehen n Kinder nebeneinander und schauen ihrem Lehrer zu, als dieser sie auffordert, sich nach links zu drehen. Einige Kinder drehen sich nach links, und einige Kinder drehen sich versehentlich nach rechts. In einer Zeiteinheit t drehen sich alle Kinder, die sich gegenüberstehen, nach links um (d. h. um 180°), wobei jedes Kind höchstens eine Umdrehung macht. Die Bewegung wird fortgesetzt, bis sich keine Kinder mehr gegenüberstehen. Geben Sie an, welcher der folgenden Algorithmen $\text{intoarceri}(n, c)$ die Anzahl der Zeiteinheiten t bestimmt, die vergehen, bis sich keine Kinder mehr gegenüberstehen. Die Variable n ist eine natürliche Zahl ($1 \leq n \leq 100$), und die Zeichenkette c hat n Elemente, wobei $c[i]$ entweder "s" (für "links") oder "d" (für "rechts") ist, je nachdem, in welche Richtung sich das i -te Kind nach dem Befehl des Lehrers gedreht hat. *Beispiele:* wenn $n = 6$ und $c = \text{'sdssd'}$, dann $t = 3$; wenn $n = 3$ und $c = \text{'sdd'}$, dann $t = 0$. Der Algorithmus $\text{copiază}(a, n)$ gibt eine Kopie des Vektors a mit n Elementen zurück.

A. Algorithm $\text{intoarceri}(n, c)$:

```

t ← 0; aux ← copiază(c, n); ok ← False
While NOT ok execute
  ok ← True
  For i ← 1, n - 1 execute
    If (aux[i] = 'd') AND (aux[i + 1] = 's') then
      c[i] ← 's'; c[i + 1] ← 'd'
      ok ← False
    EndIf
  EndFor
  aux ← copiază(c, n)
  If NOT ok then
    t ← t + 1
  EndIf
EndWhile
Return t
EndAlgorithm

```

C. Algorithm $\text{intoarceri}(n, c)$:

```

stop ← False; t ← 0
While NOT stop execute
  i ← 1
  stop ← True
  While i < n execute
    If (c[i] = 'd') AND (c[i + 1] = 's') then
      c[i] ← 's'; c[i + 1] ← 'd'
      i ← i + 2
      stop ← False
    Else
      i ← i + 1
    EndIf
  EndWhile
  If NOT stop then
    t ← t + 1
  EndIf
EndWhile
Return t
EndAlgorithm

```

B. Algorithm $\text{intoarceri}(n, c)$:

```

t ← 0; dr ← 0; st ← 0
For i ← 1, n execute
  If c[i] = 'd' then
    dr ← dr + 1
    If st > 0 then
      st ← st - 1
    EndIf
  Else
    If dr > 0 then
      t ← dr + st
      st ← st + 1
    EndIf
  EndIf
EndFor
Return t
EndAlgorithm

```

D. Algorithm $\text{intoarceri}(n, c)$:

```

t ← 0; dr ← 0; st ← 0
For i ← 1, n execute
  If c[i] = 'd' then
    dr ← dr + 1
    If st > 0 then
      t ← dr + st
      st ← st - 1
    EndIf
  Else
    If dr > 0 then
      t ← dr + st
      dr ← dr + 1
    EndIf
  EndIf
EndFor
Return t
EndAlgorithm

```


Aufnahmeprüfung – 6. September 2024

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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