

Mathe-Info-Wettbewerb - 20. April 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.
- Wenn mehrere aufeinanderfolgende Zuweisungsanweisungen in der gleichen Zeile erscheinen, werden sie durch ";" voneinander getrennt.

1. Betrachten wir den Algorithmus $\text{calcul}(v, n)$, wobei n eine natürliche Zahl ($1 \leq n \leq 10^4$) und v ein Vektor mit n Elementen natürliche Zahlen ist ($v[1], v[2], \dots, v[n], 1 \leq v[i] \leq 10^4$, für $i = 1, 2, \dots, n$):

```
Algorithm calcul(v, n):
  i ← 1; j ← n
  While i < j execute
    While i < j AND v[i] MOD 2 = 1 execute
      i ← i + 1
    EndWhile
    While i < j AND v[j] MOD 2 = 1 execute
      j ← j - 1
    EndWhile
    If v[i] ≠ v[j] then
      Return False
    EndIf
    i ← i + 1
    j ← j - 1
  EndWhile
  Return True
EndAlgorithm
```

Für welche der folgenden Situationen liefert der Algorithmus *True*?

- Wenn der Vektor v aus den Werten $[1, 11, 2, 4, 3, 4, 7, 6, 4, 21, 23, 25, 2]$ und $n = 13$ besteht.
- Wenn der Vektor v aus den Werten $[1, 11, 2, 4, 3, 7, 6, 4, 21, 23, 25, 2]$ und $n = 12$ besteht.
- Wenn und nur wenn der Betrag der Differenz zwischen zwei geraden Elementen des Vektors v , zwischen denen mindestens ein ungerades Element liegt, gleich 2 ist.
- Wenn der Vektor, der aus den geraden Elementen des Vektors v gebildet wird, der von links nach rechts durchlaufen wird, gleich dem Vektor ist, der aus den geraden Elementen des Vektors v gebildet wird, der von rechts nach links durchlaufen wird.

2. Gegeben sei der Algorithmus $g(a, b)$, wobei a und b natürliche Zahlen sind ($0 \leq a, b \leq 10^4$):

```
Algorithm g(a, b):
  If a = b then
    Return a
  EndIf
  If a > b then
    Return g(a - b, b)
  Else
    Return g(a, b - a)
  EndIf
EndAlgorithm
```

Welche der folgenden Aussagen ist zutreffend?

- Für den Aufruf $g(2, 2)$ liefert der Algorithmus 2.
- Wenn $a = b$ ist, wird der Algorithmus nie sich selbst aufrufen.
- Wenn $a = 0$ und $0 \leq b \leq 10^4$ ist, ruft sich der Algorithmus selbst einmal auf.
- Wenn $a \neq 0, b \neq 0$ und $a \neq b$ ist, ruft sich der Algorithmus selbst $a + b - 1$ mal auf.

3. Ein gerichteter Graph hat 8 Knoten, nummeriert von 1 bis 8, und Kanten $(1, 7), (1, 8), (3, 5), (3, 7), (4, 3), (4, 7), (6, 3), (6, 5), (6, 7), (6, 8), (8, 5), (8, 7)$. Die Anzahl der Knoten, die den Ausgangsgrad Null haben, ist:

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

4. Welchen Wert hat der Ausdruck $\text{NOT} ((x \text{ MOD } 2 = 0) \text{ AND } (\text{NOT} ((y > x) \text{ AND } (x \text{ MOD } 7 \neq 5))))$, wenn $x = 12$ und $y = 23$ ist?

- Wahr
- Falsch
- Gleicher Wert wie $\text{NOT} ((x \text{ MOD } 2 = 0) \text{ AND } (\text{NOT} ((x > y) \text{ AND } (x \text{ MOD } 7 \neq 5))))$
- Gleicher Wert wie $\text{NOT} ((y \text{ MOD } 2 = 0) \text{ AND } (\text{NOT} ((x > y) \text{ AND } (y \text{ MOD } 7 \neq 5))))$

5. Gegeben sei der Algorithmus $ghici(n)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^9$):

```

Algorithm ghici(n):
  f ← 0
  y ← -1
  For c ← 0, 9 execute
    x ← n
    k ← 0
    While x > 0 execute
      If x MOD 10 = c then
        k ← k + 1
      EndIf
      x ← x DIV 10
      If k > f then
        f ← k
        y ← c
      EndIf
    EndWhile
  EndFor
  Return y
EndAlgorithm

```

Geben Sie an, was der Algorithmus zurückgibt:

- A. Anzahl der Ziffern der Zahl n
- B. Maximale Häufigkeit der Ziffernhäufigkeiten in der Zahl n
- C. Eine der Ziffern mit der größten Häufigkeit in der Zahl n
- D. Eine der Ziffern mit maximalem Wert in der Zahl n

6. Gegeben sei der Algorithmus $divizori(n)$, wobei n eine ganze Zahl ist ($-10^3 \leq n \leq 10^3$).

```

Algorithm divizori(n):
  nr ← 0; d ← 1
  While d * d ≤ n execute
    If n MOD d = 0 then
      nr ← nr + 1
    EndIf
    d ← d + 1
  EndWhile
  Return 2 * nr
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Ist $n = 5$, gibt der Algorithmus 2 zurück.
- B. Wenn $n > 1$ ist, gibt der Algorithmus die Anzahl aller (echten und unechten) Teiler der Zahl n zurück.
- C. Ist $n = 0$, gibt der Algorithmus 0 zurück.
- D. Wenn $n < 0$ ist, gibt der Algorithmus die Anzahl aller (echten und unechten) Teiler zurück, die dem Betrag von n entsprechen.

7. Gegeben sei der Algorithmus $ceReturneaza(a, b)$, wobei a und b natürliche Zahlen sind ($0 \leq a, b \leq 10^3$):

```

Algorithm ceReturneaza(a, b):
  If a > b then
    c ← a; a ← b; b ← c
  EndIf
  d ← 0
  For i ← a, b execute
    If i MOD 2 = 0 then
      d ← d + 1
    EndIf
  EndFor
  Return d
EndAlgorithm

```

In welcher der folgenden Situationen ist das zurückgegebene Ergebnis 0?

- A. $a = 11, b = 11$
- B. $a = 4, b = 8$
- C. $a = 12, b = 12$
- D. $a = 0, b = 0$

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

```

Algorithm ceFace(n):
  k ← 0
  s ← 0
  While k ≠ n execute
    k ← k + 1
    s ← s + 2 * k - 1
    Write s, " "
  EndWhile
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Für $n = 3$ zeigt der Algorithmus: 0 9
- B. Für $n = 10$ ist der vorletzte Wert, der der Variablen s zur Laufzeit zugewiesen wird, 81
- C. Der Algorithmus zeigt die Quadrate der natürlichen Zahlen 1, 2, ..., n
- D. Für $n = 4$ zeigt der Algorithmus: 1 4 8 16

9. Gegeben seien die Algorithmen `verificare_aux(a, b)` und `verificare(a, b)`, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10^9$):

```

Algorithm verificare_aux(a, b):
  c ← b
  While a > 0 execute
    While (c > 0) AND (a MOD 10 ≠ c MOD 10) execute
      c ← c DIV 10
    EndWhile
    If c = 0 then
      Return False
    EndIf
    c ← b
    a ← a DIV 10
  EndWhile
  Return True
EndAlgorithm

```

```

Algorithm verificare(a, b):
  Return verificare_aux(a, b) AND verificare_aux(b, a)
EndAlgorithm

```

Für welche der folgenden Bedingungen liefert der Algorithmus `verificare(a, b)` *True*?

- A. Wenn a und b die gleiche Anzahl von Ziffern haben.
- B. Wenn $a = 1001$ und $b = 10$.
- C. Wenn der Häufigkeitsvektor der Ziffern von a mit dem Häufigkeitsvektor der Ziffern von b identisch ist.
- D. Wenn $a = 123$ und $b = 321$.

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

```

Algorithm verifica(n):
  a ← n MOD 10
  n ← n DIV 10
  While n > 0 execute
    b ← n MOD 10
    If a ≤ b then
      Return False
    EndIf
    a ← b
    n ← n DIV 10
  EndWhile
  Return True
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Nach dem Aufruf `verifica(2024)` gibt der Algorithmus *False* zurück.
- B. Der Algorithmus gibt nur dann *True* zurück, wenn n eine Zahl ist, bei der die Ziffern in streng aufsteigender Reihenfolge stehen.
- C. Der Algorithmus gibt nur dann *True* zurück, wenn n eine Zahl ist, bei der die Ziffern in streng absteigender Reihenfolge stehen.
- D. Der Algorithmus gibt nur dann *True* zurück, wenn die höchstwertige Ziffer der Zahl n kleiner ist als ihre niedrigstwertige Ziffer.

11. Gegeben sei der Algorithmus `F(x, n, i, S, k)`, wobei x ein Vektor von n ($1 \leq n \leq 10^4$) ganzen Zahlen ist ($x[1], x[2], \dots, x[n]$), $-10^3 \leq x[i] \leq 10^3$ für $i = 1, 2, \dots, n$), S eine reelle Zahl ist und i und k natürliche Zahlen sind. Der Operator `/` steht für reelle Division, z.B. $3 / 2 = 1.5$.

```

Algorithm F(x, n, i, S, k):
  If n < i then
    If k = n then
      Return 0
    Else
      Return S / (n - k)
    EndIf
  Else
    If x[i] MOD 2 = 0 then
      Return F(x, n, i + 1, S + x[i], k)
    Else
      Return F(x, n, i + 1, S, k + 1)
    EndIf
  EndIf
EndAlgorithm

```

Wenn der Algorithmus in der Form `F(x, n, 1, 0.0, 0)` aufgerufen wird, welche der folgenden Aussagen sind dann wahr?

- A. Der Algorithmus liefert die Summe der geraden Zahlen im Vektor x geteilt durch die Anzahl der ungeraden Zahlen im Vektor.
- B. Wenn $n = 10$ und $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, liefert der Algorithmus den Wert 6.0.
- C. Der Algorithmus liefert das arithmetische Mittel der geraden Zahlen im Vektor x .
- D. Wenn $n = 10$ und $x = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]$, gibt der Algorithmus 0 zurück.

12. Man betrachte eine quadratische Matrix x der Dimension n mit Elementen paarweise verschiedene natürlichen Zahlen ($2 \leq n \leq 50, x[1][1], \dots, x[1][n], x[2][1], \dots, x[2][n], \dots, x[n][1], \dots, x[n][n], 1 \leq x[i][j] \leq 10^4$, für $i = 1, 2, \dots, n, j = 1, 2, \dots, n$). Die Elemente jeder Zeile und die Elemente jeder Spalte sind aufsteigend geordnet. Der Algorithmus `cauta(n, x, v)` sucht nach einem Wert v in der Matrix x und gibt das Paar zurück, das aus dem Zeilenindex und dem Spaltenindex der Position besteht, an der sich der Wert v in der Matrix befindet, oder `(-1, -1)`, wenn sich der Wert v nicht unter den Elementen der Matrix befindet. Wir gehen davon aus, dass der Algorithmus `cautareBinara(t, n, v)` den binären Suchalgorithmus implementiert, um festzustellen, ob eine Zahl v in dem Vektor t mit n aufsteigend geordneten Elementen vorhanden ist. Wenn v nicht in der i -ten Zeile der Matrix steht, liefert der Aufruf von `cautareBinara(x[i], n, v)` den Wert -1.

Welche der folgenden Algorithmen sind am effizientesten in Bezug auf die Zeitkomplexität und erfüllen die beschriebenen Anforderungen?

A.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  For i ← 1, n execute
    For j ← 1, n execute
      If x[i][j] = v then
        a ← i
        b ← j
      EndIf
    EndFor
  EndFor
  Return a, b
EndAlgorithm

```

C.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  i ← 1; j ← n
  While i ≤ n AND j > 0 execute
    If x[i][j] = v then
      a ← i
      b ← j
    EndIf
    If x[i][j] > v then
      j ← j - 1
    Else
      i ← i + 1
    EndIf
  EndWhile
  Return a, b
EndAlgorithm

```

B.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  For i = 1, n execute
    j ← cautareBinara(x[i], n, v)
    If j ≠ -1 then
      a ← i
      b ← j
    EndIf
  EndFor
  Return a, b
EndAlgorithm

```

D.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  i ← 1; j ← 1
  While i ≤ n AND x[i][j] < v execute
    i ← i + 1
  EndWhile
  While j ≤ n AND x[i][j] < v execute
    j ← j + 1
  EndWhile
  If x[i][j] = v then
    a ← i
    b ← j
  EndIf
  Return a, b
EndAlgorithm

```

13. Welche der folgenden Codesequenzen implementiert bei einer quadratischen Matrix M mit 3×3 Elementen korrekt eine trigonometrische Drehung der Matrix um 90 Grad um das Element an der Position (2, 2)?

A.

```

For i ← 0, 1 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

C.

```

For i ← 1, 2 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

B.

```

For i ← 0, 2 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

D.

```

For i ← 1, 3 execute
  X ← M[1][1]
  M[1][1] ← M[1][i]
  M[1][i] ← M[1][3]
  M[1][3] ← M[i][3]
  M[i][3] ← M[3][3]
  M[3][3] ← M[3][i]
  M[3][i] ← M[3][1]
  M[3][1] ← M[i][1]
  M[i][1] ← X
EndFor

```

14. Gegeben sei der Algorithmus `rearanjeaza(x, n)`, wobei n eine natürliche Zahl ($1 \leq n \leq 200$) und x ein Vektor aus n verschiedenen ganzen Zahlen ist ($x[1], x[2], \dots, x[n], -100 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$). Der Algorithmus `interschimba(x, i, j)` vertauscht die Elemente $x[i]$ und $x[j]$.

```

Algorithm rearanjeaza(x, n):
  v ← x[n]
  i ← 0
  j ← 1
  While j ≤ n - 1 execute
    If x[j] ≤ v then
      i ← i + 1
      interschimba(x, i, j)
    EndIf
    j ← j + 1
  EndWhile
  i ← i + 1
  interschimba(x, i, n)
  Return i
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Der Algorithmus sortiert die Elemente des Vektors x in aufsteigender Reihenfolge.
- B. Wenn der Vektor x aufsteigend sortiert ist, ändert sich die Reihenfolge der Elemente im Vektor nicht.
- C. Der Vektor x wird so umgeordnet, dass das letzte Element des ursprünglichen Vektors auf der linken Seite nur Elemente mit niedrigerem Wert und auf der rechten Seite nur Elemente mit höherem Wert hat.
- D. Der Algorithmus gibt den Anfangsindex des Elements mit dem kleinsten Wert im Vektor x zurück.

15. Gegeben sei der Algorithmus `calcul(v, n)`, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^4$) und v ein Vektor mit n Elementen natürliche Zahlen ist ($v[1], v[2], \dots, v[n], 1 \leq v[i] \leq 200$, für $i = 1, 2, \dots, n$):

```

Algorithm calcul(v, n):
  If n = 1 then
    Return v[1]
  EndIf
  If v[1] MOD v[n] = 0 then
    v[1] ← v[n]
    n ← n - 1
    Return calcul(v, n)
  Else
    aux ← v[n]
    v[n] ← v[1] MOD v[n]
    v[1] ← aux
    Return calcul(v, n)
  EndIf
EndAlgorithm

```

Für welchen der folgenden Parameter liefert der Algorithmus den Wert 12?

- A. $v = [60, 96, 120, 84], n = 4$
- B. $v = [75, 24, 12, 84], n = 4$
- C. $v = [75, 24, 49, 80], n = 4$
- D. $v = [60, 24, 12, 84], n = 4$

16. Gegeben sei der Algorithmus `ceFace(n)`, wobei n eine ganze Zahl ist ($-10^4 \leq n \leq 10^4$):

```

Algorithm ceFace(n):
  If n = 0 then
    Return "0"
  EndIf
  If n < 0 then
    Return "-" + ceFace(-n)
  EndIf
  If n MOD 3 = 0 then
    Return ceFace(n DIV 3) + "0"
  EndIf
  If n MOD 3 = 1 then
    Return ceFace(n DIV 3) + "1"
  EndIf
  Return ceFace(n DIV 3) + "2"
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Wenn die Zahl n eine Potenz von 3 ist, enthält die zurückgegebene Zeichenkette ein einzelnes Zeichen "1".
- B. Für $n = 3$ und $n = -3$ liefert der Algorithmus `ceFace(n)` identische Werte.
- C. Ist $n = 82$, gibt der Algorithmus "010001" zurück.
- D. Wenn n eine negative Zahl ist, tritt der Algorithmus in den unendlichen Zyklus ein.

17. Gegeben sei der Algorithmus $\text{decide}(n, x)$, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^4$) und x ein Vektor bestehend aus n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n], 0 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$). Welche der folgenden Aussagen ist wahr?

- A. Bei $n = 5$ und $x = [1, 2, 1, 3, 1]$ gibt der Algorithmus 1 zurück.
- B. Bei $n = 5$ und $x = [1, 2, 2, 3, 1]$ gibt der Algorithmus -1 zurück.
- C. Für jeden beliebigen Eingabevektor gibt der Algorithmus -1 zurück.
- D. Der Algorithmus gibt das erste Element des Vektors x zurück.

```

Algorithm decide(n, x):
  a ← x[1]; i ← 2; j ← 1
  While i ≤ n execute
    If x[i] = a then
      j ← j + 1
    Else
      If j > 0 then
        j ← j - 1
      Else
        a ← x[i]
        j ← 1
    EndIf
  EndIf
  i ← i + 1
EndWhile
i ← 1; j ← 0
While i ≤ n execute
  If x[i] = a then
    j ← j + 1
  EndIf
  i ← i + 1
EndWhile
If j > (n DIV 2) then
  Return a
Else
  Return -1
EndIf
EndAlgorithm

```

18. Gegeben sei der Algorithmus $\text{ceFace}(n)$, in dem n Zahlen gelesen werden, wobei n eine natürliche Zahl ist ($1 \leq n \leq 10^9$):

```

Algorithm ceFace(n):
  nr ← 0
  Read a
  For i ← 2, n execute
    Read b
    If a ≠ b then
      nr ← nr + 1
    EndIf
    a ← b
  EndFor
  Return nr
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Der Algorithmus gibt die Anzahl der sich wiederholenden Zahlen unter den gelesenen Zahlen zurück (zum Beispiel, wenn die Zahlen 3, 34, 34, 7, 3, 34 sind, dann ist der zurückgegebene Wert 2).
- B. Der Algorithmus gibt die Länge der längsten Teilfolge mit gleichen Werten unter den gelesenen Zahlen zurück (zum Beispiel, wenn die Zahlen 2, 34, 34, 34, 5 sind, dann ist der zurückgegebene Wert 3).
- C. Der Algorithmus gibt die Anzahl der Paare von aufeinanderfolgenden Elementen mit unterschiedlichen Werten unter den gelesenen Zahlen zurück (z. B. wenn die Zahlen 2, 34, 34, 7 sind, dann sind (2, 34), (34, 7) Paare von aufeinanderfolgenden Elementen mit unterschiedlichen Werten und der zurückgegebene Wert ist 2).
- D. Der Algorithmus gibt die Anzahl der Paare von aufeinanderfolgenden Elementen mit gleichen Werten unter den gelesenen Zahlen zurück (z.B. wenn die Zahlen 2, 2, 3, 3 sind, dann sind (2, 2), (3, 3) Paare von aufeinanderfolgenden Elementen mit gleichen Werten und der zurückgegebene Wert ist 2).

19. Betrachten wir den Algorithmus $f(a)$, wobei a eine natürliche Zahl ist ($0 \leq a \leq 10^4$):

```

Algorithm f(a):
  n ← 0
  While a > 1 execute
    b ← 1
    While b ≤ a execute
      b ← 3 * b
      n ← n + 1
    EndWhile
    a ← a DIV 3
  EndWhile
  Return n
EndAlgorithm

```

Welchen Wert liefert der Algorithmus, wenn er für $a = 81$ aufgerufen wird?

- A. 0
- B. 14
- C. 16
- D. 9

20. Gegeben sei der Algorithmus $h(n, a)$, wobei n eine natürliche Zahl ($1 \leq n \leq 10^3$) und a ein Vektor aus n ganzzahligen Elementen ($a[1], a[2], \dots, a[n], -10^4 \leq a[i] \leq 10^4$, für $i = 1, 2, \dots, n$) in aufsteigender Reihenfolge ist.

Welcher der folgenden Aufrufe gibt den Wert 4 zurück?

- A. $h(5, [1, 2, 3, 4, 5])$
- B. $h(6, [2, 4, 6, 10, 18, 20])$
- C. $h(7, [2, 2, 3, 4, 6, 9, 13])$
- D. $h(5, [2, 2, 2, 4, 6])$

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

Welchen Wert wird der Algorithmus zurückgeben, wenn der Aufruf die Form $f(x, 9, 41)$ hat, wobei $x = [41, 15, 5, 8, 10, 1, 16, 18, 19]$?

- A. 1
- B. 3
- C. 5
- D. 7

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

```

Algorithm select(v, x, n):
  i ← 1
  j ← n
  While i ≤ j execute
    k ← (i + j) DIV 2
    If v[k] = x then
      Return k
    EndIf
    If v[i] ≤ v[k] then
      If v[i] ≤ x AND x < v[k] then
        j ← k - 1
      Else
        i ← k + 1
      EndIf
    Else
      If v[k] < x AND x ≤ v[j] then
        i ← k + 1
      Else
        j ← k - 1
      EndIf
    EndIf
  EndWhile
  Return -1
EndAlgorithm

```

```

Algorithm h(n, a):
  t ← 0; i ← n
  While i > 2 execute
    k ← 1
    j ← i - 1
    b ← a[i]
    While k < j execute
      If a[k] + a[j] = b then
        t ← t + 1
        k ← k + 1
        j ← j - 1
      Else
        If a[k] + a[j] < b then
          k ← k + 1
        Else
          j ← j - 1
        EndIf
      EndIf
    EndWhile
    i ← i - 1
  EndWhile
  Return t
EndAlgorithm

```

```

Algorithm f(x, n, m):
  If m = 0 then
    Return 1
  EndIf
  If n = 0 then
    Return 0
  EndIf
  If x[n] > m then
    Return f(x, n - 1, m)
  Else
    Return f(x, n - 1, m) + f(x, n - 1, m - x[n])
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen ist zutreffend?

- A. Im Fall des Aufrufs $select([0, 1, 2, 4, 5, 8, 9, 10, 7, 6], 10, 10)$ liefert der Algorithmus 10.
- B. Der Algorithmus gibt die Position zurück, an der ein Element x in Vektor v erscheint, wenn und nur wenn Vektor v aufsteigend sortiert ist.
- C. Die Komplexität des Algorithmus ist $O(\log_2 n)$.
- D. Im Falle des Aufrufs $select([0, 1, 2, 4, 5, 8, 9, 10, 7, 6], 7, 10)$ liefert der Algorithmus -1.

23. Gegeben sei der Algorithmus `maiMare(n)`, wobei n eine natürliche Zahl ($1 \leq n < 10^6$) ist, die aus verschiedenen Ziffern besteht. Der Algorithmus gibt die Anzahl der Zahlen zurück, die strikt größer als n sind und aus den Ziffern von n bestehen. z.B. `maiMare(213) = 3`. Angenommen, n hat keine führenden 0-Ziffern und wir haben die folgenden Algorithmen wie angegeben implementiert:

- `factorial(n)` – liefert die Fakultät der natürlichen Zahl n ($1 \leq n \leq 10$)
- `nrCifre(n)` – liefert die Anzahl der Ziffern der natürlichen Zahl n ($1 \leq n < 10^6$)
- `imparte(n)` – liefert einen Vektor mit den Elementen die Ziffern der natürlichen Zahl n ($1 \leq n < 10^6$), in umgekehrter Reihenfolge. Zum Beispiel: `imparte(1352)` gibt den Vektor `[2, 5, 3, 1]` zurück.

```

Algorithm maiMare(n):
  cifre ← imparte(n)
  nrCif ← nrCifre(n)
  Return calculeaza(cifre, nrCif)
EndAlgorithm

1. Algorithm calculeaza(v, n):
2.   If n < 2 then
3.     Return 0
4.   EndIf
5.   mm ← 0
6.   For i ← 1, n - 1 execute
7.     If v[i] > v[n] then
8.       mm ← mm + 1
9.     EndIf
10.  EndFor
11.  ...
12. EndAlgorithm

```

Welche der folgenden Anweisungen sollte in Zeile 11 des Algorithmus `calculeaza(v, n)` geschrieben werden?

- A. `Return factorial(n) - ((n - mm - 1) * factorial(n - 1) + calculeaza(v, n - 1))`
- B. `Return calculeaza(v, n - 1) * mm + factorial(n - 1)`
- C. `Return (mm * factorial(n) + calculeaza(v, n - 1)) DIV n`
- D. `Return calculeaza(v, n - 1) + mm * factorial(n - 1)`

24. Eine Veranstaltung sollte im Saal I stattfinden, muss aber in den Saal II verlegt werden, wo die Nummerierung der Sitzplätze anders ist. In beiden Sälen gibt es L Sitzreihen ($2 \leq L \leq 50$), wobei jede Reihe in der Mitte durch einen Gang geteilt ist und K Sitze ($2 \leq K \leq 50$) auf jeder Seite des Ganges hat (der Saal enthält also insgesamt $2 * K * L$ Sitze). Im Saal II 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 Sitze in der ersten Reihe haben also die Nummern 2, 4, 6 usw. Nachdem alle Sitze einer Reihe nummeriert wurden, wird die Nummerierung in der nächsten Reihe fortgesetzt und beginnt wieder mit dem Sitz 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 mit 1, 3, 5 usw. nummeriert, beginnend mit dem Gang zum Rand des Saals.

In Saal I wird jeder Platz durch drei Werte identifiziert. Die Reihennummer (**rand** - ein Wert zwischen 1 und L einschließlich, wobei Reihe 1 die Reihe vor der Bühne ist), die Richtung des Sitzplatzes in Bezug auf den Gang (**directie** - der Wert "stanga" oder "dreapta") und die Sitzplatznummer innerhalb der Reihe (**loc** - ein Wert zwischen 1 und K einschließlich, wobei Sitz 1 der Sitz neben dem Gang ist). Da sich die Vorstellung verschiebt, müssen die Sitzplätze auf den Eintrittskarten in Saal I (**rand, loc, directie**) in gültige Sitzplätze in Saal II (einfache Nummer) umgewandelt werden.

Welcher der nachstehenden Algorithmen führt bei der Eingabe von K , **rand**, **loc**, **directie** gemäß der Aussage die Transformation korrekt durch (eine Transformation ist korrekt, wenn jeder Zuschauer einen eindeutigen Sitzplatz in Raum II hat)?

A.

```

Algorithm transforma(K, rand, loc, directie):
  If directie = "stanga" then
    rez ← 2 * (loc + K * (rand - 1))
  Else
    rez ← 2 * (loc + K * (rand - 1) + 1)
  EndIf
  Return rez
EndAlgorithm

```

B.

```

Algorithm transforma(K, rand, loc, directie):
  rez ← rand * (K - 1) * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez - 1
  EndIf
  Return rez
EndAlgorithm

```

C.

```

Algorithm transforma(K, rand, loc, directie):
  rez ← (rand - 1) * K * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez - 1
  EndIf
  Return rez
EndAlgorithm

```

D.

```

Algorithm transforma(K, rand, loc, directie):
  rez ← (rand - 1) * K * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez + 1
  EndIf
  Return rez
EndAlgorithm

```

BABEŞ-BOLYAI UNIVERSITÄT

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Mathe-Info-Wettbewerb - 20. April 2024

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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