

Aufnahmeprüfung - 8. September 2023
Schriftliche Prüfung in Informatik

WICHTIGER HINWEIS:

Falls nicht anders erklärt:

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

1. Gegeben sei der Algorithmus `ceFace(a, b)`, wobei a und b natürliche Zahlen sind ($0 \leq a, b \leq 10^4$).

```

Algorithm ceFace(a, b):
  c ← 0
  bc ← b
  While bc ≠ 0 execute
    c ← c * 10 + bc MOD 10
    bc ← bc DIV 10
  EndWhile
  If c ≠ a then
    Return ceFace(a - 1, b - 1)
  EndIf
  Return a
EndAlgorithm

```

Welche Auswirkung hat der Aufruf `ceFace(a, a)`?

- Der Algorithmus liefert das kleinste Palindrom, das größer oder gleich a ist.
- Der Algorithmus liefert das größte Palindrom, das kleiner oder gleich a ist.
- Der Algorithmus liefert das kleinste Palindrom, das größer als a ist.
- Der Algorithmus liefert die größte gerade Zahl, die kleiner oder gleich a ist.

2. Gegeben sei der Algorithmus `createTablou(n, m, x)`, wobei n, m natürliche Zahlen sind ($1 \leq n, m \leq 100$) und x ein zweidimensionales Array mit $n * m$ ganzzahligen Elementen ist ($x[1][1], x[1][2], \dots, x[n][m], 0 \leq x[i][j] \leq 10^4$, für $i = 1, 2, \dots, n; j = 1, 2, \dots, m$).

```

Algorithm createTablou(n, m, x):
  k ← 0
  For i ← 1, n execute
    For j ← 1, m execute
      If k MOD 2 ≠ 0 then
        x[i][j] ← k * k
      EndIf
      Write x[i][j], " "
      k ← k + 1
    EndFor
    Write new line
  EndFor
EndAlgorithm

```

Was zeigt dieser Algorithmus an, wenn die Elemente des Arrays x mit 0 initialisiert werden?

- Der Algorithmus zeigt die Elemente des zweidimensionalen Arrays x an, in dem es Werte gleich 0 und die ersten $(n * m) \text{ DIV } 2$ ungeraden perfekten Quadrate gibt.
- Der Algorithmus zeigt die Elemente des zweidimensionalen Arrays x an, in dem es Werte gleich 0 und die ersten geraden perfekten Quadrate gibt.
- Der Algorithmus zeigt die Elemente des zweidimensionalen Arrays x an, in dem die Folge der ersten $(n * m) \text{ DIV } 2$ geraden perfekten Quadrate liegt.
- Der Algorithmus zeigt die Elemente des zweidimensionalen Arrays x an, in der - würde man die Elemente zeilenweise anordnen - die ungeraden perfekten Quadrate in aufsteigender Reihenfolge erscheinen würden, möglicherweise mit Werten gleich 0 davor und/oder danach.

3. Gegeben sei der Algorithmus `something(n, x)`, wobei n eine natürliche Zahl ($1 \leq n \leq 10^4$) und x ein Vektor aus n natürlichen Zahlen ist ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^6$, für $i = 1, 2, \dots, n$).

```

Algorithm something(n, x):
  s ← 0
  For i ← 1, n execute
    nr ← 1
    While x[i] > 9 execute
      nr ← nr + 1
      x[i] ← x[i] DIV 10
    EndWhile
    s ← s + nr
  EndFor
  Return s
EndAlgorithm

```

Was ergibt der Aufruf von `something(5, [222, 2043, 29, 2, 20035])`?

- 16
- 10
- 11
- 15

4. Gegeben sei der Algorithmus $\text{ceFace}(n, v, a)$, wobei n und v zwei natürliche Zahlen sind ($1 \leq n, v \leq 10^4$) und a eine Folge natürlicher Zahlen mit n Elementen ($a[1], a[2], \dots, a[n]$) ist.

```

Algorithm ceFace(n, v, a):
  For i ← 1, n execute
    d ← v
    If a[i] ≠ 0 then
      gäsit ← False
      While (d ≤ v * a[i]) AND (NOT gäsit) execute
        If ((d DIV a[i]) * a[i] = d) AND ((d DIV v) * v = d) then
          gäsit ← True
        Else
          d ← d + 1
        EndIf
      EndWhile
    EndIf
    v ← d
  EndFor
  Return v
EndAlgorithm

```

Welchen Wert liefert der Algorithmus, wenn $n = 4$, $v = 3$ und $a = [5, 4, 2, 10]$?

- A. 20 B. 120 C. 60 D. 15

5. Gegeben sei der Algorithmus $\text{calcul}(v, n)$, wobei n eine natürliche Zahl ($1 \leq n \leq 10^4$) und v ein Vektor bestehend aus n Elemente 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
  While i ≤ n DIV 2 execute
    p ← 0
    While v[i] ≠ 0 execute
      p ← p + 1
      v[i] ← v[i] DIV 10
    EndWhile
    q ← 0
    While v[n + 1 - i] ≠ 0 execute
      q ← q + 1
      v[n + 1 - i] ← v[n + 1 - i] DIV 10
    EndWhile
    If p ≠ q then
      Return False
    EndIf
    i ← i + 1
  EndWhile
  Return True
EndAlgorithm

```

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

- A. Wenn der Vektor v aus den Werten [12, 12, 2, 5466, 3, 111, 1, 3, 44] besteht und $n = 9$.
- B. Wenn der Vektor v aus den Werten [12, 345, 2, 5466, 3, 111, 10] besteht und $n = 7$.
- C. Wenn die Elemente des Vektors v die gleiche Anzahl von Ziffern haben.
- D. Wenn der Vektor, der durch die Anzahl der Ziffern der Elemente des Vektors v gebildet wird, ein Palindrom bildet; z. B. aus $v = [8, 37, 3]$ wird der Vektor [1, 2, 1] gebildet, der ein Palindrom ist.

6. Gegeben sei der Algorithmus $\text{alg}(n)$, wobei n eine natürliche Zahl ist ($0 \leq n \leq 10^4$).

```

Algorithm alg(n):
  If n = 0 then
    Return 0
  Else
    If n MOD 2 = 0 then
      Return alg(n DIV 10) + n MOD 10
    Else
      Return alg(n DIV 10)
    EndIf
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Aufruf $\text{alg}(123)$ ergibt 6.
- B. Der Algorithmus berechnet die Summe der Ziffern an geraden Stellen der angegebenen Zahl.
- C. Der Algorithmus berechnet die Summe der geraden Ziffern der angegebenen Zahl.
- D. Der Algorithmus berechnet die Summe der Ziffern der angegebenen Zahl.

7. Gegeben sei der Algorithmus $f(x)$, wobei x eine von Null verschiedene natürliche Zahl ist ($1 \leq x \leq 10^5$).

```

Algorithm f(x):
  If x > 0 then
    x ← x DIV 2
    f(x)
    Write x, " "
    x ← x DIV 2
    f(x)
  EndIf
EndAlgorithm

```

Geben Sie an, was nach dem Aufruf von $f(10)$ angezeigt wird.

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

8. Gegeben sei die quadratische Matrix M der Dimension n , die natürliche Zahlen enthält, wobei n eine von Null verschiedene natürliche Zahl ist ($1 \leq n \leq 10^4$, $M[1][1], \dots, M[1][n], M[2][1], \dots, M[2][n], \dots, M[n][1], \dots, M[n][n]$, $1 \leq M[i][j] \leq 10^4$, für $i = 1, 2, \dots, n, j = 1, 2, \dots, n$). Betrachten Sie den folgenden Algorithmus:

```

Algorithm what(M, n):
  up ← 1
  down ← n
  left ← 1
  right ← n
  While left ≤ right AND up ≤ down execute
    For i ← left, right execute
      Write M[up][i], " "
    EndFor
    up ← up + 1
    For i ← up, down execute
      Write M[i][right], " "
    EndFor
    right ← right - 1
    For i ← right, left, -1 execute
      Write M[down][i], " "
    EndFor
    down ← down - 1
    For i ← down, up, -1 execute
      Write M[i][left], " "
    EndFor
    left ← left + 1
  EndWhile
EndAlgorithm

```

Was wird für die folgende Matrix M angezeigt?

1	2	3
8	9	4
7	6	5

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

9. Gegeben sei der Algorithmus $ce_face(a, b)$, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10^4$).

```

Algorithm ce_face(a, b):
  If a = 1 then
    Return 1
  Else
    If a MOD b = 0 then
      Return ce_face(a DIV b, b)
    Else
      Return 0
    EndIf
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr?

- A. Im Fall des Aufrufs $ce_face(1, 2)$ liefert der Algorithmus 1.
- B. Im Fall des Aufrufs $ce_face(24, 2)$ liefert der Algorithmus 0.
- C. Im Fall des Aufrufs $ce_face(2024, 4)$ liefert der Algorithmus 4.
- D. Im Fall des Aufrufs $ce_face(8, 3)$ liefert der Algorithmus 2.

10. Gegeben seien die Algorithmen $decide(n)$ und $compute(m)$, wobei n und m natürliche Zahlen ungleich Null sind ($1 \leq n, m \leq 10^4$):

```

Algorithm decide(n):
  result ← -1
  m ← 0
  While n > 0 execute
    m ← m * 10 + n MOD 10
    n ← n DIV 10
  EndWhile
  If m MOD 3 = 0 then
    result ← 1
  EndIf
  Return result
EndAlgorithm

```

```

Algorithm compute(m):
  cnt ← 0
  For k ← 0, m - 1 execute
    cnt ← cnt + decide(k)
  EndFor
  Return cnt
EndAlgorithm

```

Für welche Werte von m liefert der Algorithmus $compute(m)$ -33?

- A. 100
- B. 99
- C. 98
- D. 101

11. Gegeben sei der Algorithmus $f(n, x)$, wobei n und x natürliche Zahlen sind ($1 \leq n \leq 10^5$, $2 \leq x \leq 10$):

```

Algorithm f(n, x):
  If n > 0 then
    f(n DIV x, x)
    Write n MOD x
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Algorithmus zeigt die Darstellung der Zahl n in der Zahlenbasis x an.
- B. Der Algorithmus zeigt den Rest der ganzzahligen Division der Zahl x durch die Zahl n an.
- C. Der Algorithmus zeigt die Anzahl der Ziffern der Darstellung der Zahl n in der Basis x an.
- D. Der Algorithmus prüft, ob die Zahl n durch x teilbar ist.

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

```

Algorithm ceFace(n):
  If n ≤ 9 then
    If n MOD 2 = 0 then
      Return n
    Else
      Return -1
    EndIf
  EndIf
  x ← n MOD 10
  y ← ceFace(n DIV 10)
  If x MOD 2 ≠ 0 then
    Return y
  EndIf
  If x > y then
    Return x
  EndIf
  Return y
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Der Algorithmus liefert eine einstellige Zahl oder -1.
- B. Der Algorithmus liefert eine ungerade Zahl.
- C. Der Algorithmus liefert die maximale ungerade Ziffer der Zahl n oder -1.
- D. Der Algorithmus liefert die höchste gerade Ziffer der Zahl n oder -1.

13. 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 ($x[1], x[2], \dots, x[n], -100 \leq x[i] \leq 100$, für $i = 1, 2, \dots, n$):

```

Algorithm decide(n, x):
  b ← True
  i ← 1
  While b = True AND i < n execute
    If x[i] < x[i + 1] then
      b ← True
    Else
      b ← False
    EndIf
    i ← i + 1
  EndWhile
  Return b
EndAlgorithm

```

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

- A. Wenn der Vektor $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ und $n = 10$ sind
- B. Wenn $n > 1$ und die Elemente des Vektors x in streng aufsteigender Reihenfolge sind
- C. Wenn der Vektor x keine negativen Elemente hat
- D. Wenn der Vektor x positive Elemente vor den negativen Elementen hat

14. Es seien x und y zwei positive natürliche Zahlen mit den Eigenschaften: x ist eine Potenz von 2 und y ist ein Vielfaches von 3.

Es sei der logische Ausdruck $((x * y + 3) \text{ DIV } 6 = 10) \text{ OR } (((x * y) \text{ MOD } 6 = 0) \text{ AND } ((x + y) \text{ MOD } 4 = 0))$

Welche der folgenden Aussagen trifft für Zahlenpaare zu, die die in der Aussage genannten Eigenschaften erfüllen?

- A. Es gibt ein Paar (x, y) , für das der Ausdruck wahr ist.
- B. Es gibt ein Paar (x, y) , für das der Ausdruck falsch ist.
- C. Es gibt Paare (x_1, y_1) und (x_2, y_2) , mit $x_1 \neq x_2$ und $y_1 \neq y_2$, so dass der Ausdruck für beide Paare wahr ist.
- D. Der Ausdruck ist für jedes Paar (x, y) falsch.

15. Man betrachte zwei natürliche Zahlen n und m ($1 \leq n, m \leq 256$) bzw. die Zeichenketten a mit n Elemente ($a[1], a[2], \dots, a[n]$) und die Zeichenkette b mit m Elemente ($b[1], b[2], \dots, b[m]$).

Welcher der folgenden Algorithmen liefert *True*, wenn die Zeichenkette a aus der Zeichenkette b gebildet werden kann, indem einige Zeichen entfernt werden, ohne die relative Position der verbleibenden Zeichen zu verändern, und andernfalls *False*. Zum Beispiel kann die Zeichenkette "ace" durch Entfernen von Zeichen aus der Zeichenkette "abcde" gebildet werden, aber die Zeichenkette "aec" kann durch dieses Verfahren nicht gebildet werden.

A.

```

Algorithm hasProperty(a, b, n, m):
  If n = 0 then
    Return True
  EndIf
  If m = 0 then
    Return False
  EndIf
  If a[n] = b[m] then
    Return hasProperty(a, b, n - 1, m - 1)
  EndIf
  Return hasProperty(a, b, n, m - 1)
EndAlgorithm

```

B.

```

Algorithm hasProperty(a, b, n, m):
  i ← 1
  j ← 1
  While i ≤ n AND j ≤ m execute
    If a[i] = b[j] then
      i ← i + 1
    EndIf
    j ← j + 1
  EndWhile
  If i > n then
    Return True
  Else
    Return False
  EndIf
EndAlgorithm

```

C.

```

Algorithm hasProperty(a, b, n, m):
  i ← n
  j ← m
  While i * j > 0 execute
    If a[i] = b[j] then
      i ← i - 1
    EndIf
    j ← j - 1
  EndWhile
  If i = 0 then
    Return True
  Else
    Return False
  EndIf
EndAlgorithm

```

D.

```

Algorithm hasProperty(a, b, n, m):
  If n > m then
    Return False
  EndIf
  i ← 1
  j ← 1
  While i < n execute
    If a[i] = b[j] then
      i ← i + 1
    EndIf
    j ← j + 1
  EndWhile
  If i > m then
    Return True
  Else
    Return False
  EndIf
EndAlgorithm

```

16. Gegeben sei der Algorithmus $\text{ceva}(x, n, e)$, wobei x ein Vektor bestehend aus n verschiedenen ganzzahligen Zahlen ist ($x[1], x[2], \dots, x[n]$, $1 \leq n \leq 10^3$ und $x[i] \neq x[j]$, für $1 \leq i < j \leq n$) und e eine ganze Zahl ist. Der Algorithmus sucht nach dem Element e im Vektor x , und wenn er es findet, verschiebt er das Element an die erste Position im Vektor und gibt *True* zurück, ohne die relative Reihenfolge der anderen Elemente zu ändern. Wenn e nicht in x gefunden wird, gibt der Algorithmus *False* zurück und ändert den Inhalt des Vektors nicht. Beispiel: Für den Vektor x mit den Elementen $[-100, 2, 71, 31, -62, 51]$ und $e = 31$ gibt der Algorithmus *True* zurück und der Vektor x wird zu $[31, -100, 2, 71, -62, 51]$. Welche der folgenden Implementierungen ist eine korrekte Implementierung für den Algorithmus $\text{ceva}(x, n, e)$ und hat die Zeitkomplexität $O(n)$?

A.

```

Algorithm ceva(x, n, e):
  index ← 1
  While index ≤ n execute
    If x[index] = e then
      tmp ← x[index]
      x[index] ← x[1]
      x[1] ← tmp
      Return True
    EndIf
    index ← index + 1
  EndWhile
  Return False
EndAlgorithm

```

B.

```

Algorithm ceva(x, n, e):
  index ← 2
  tmp ← x[1]
  While index ≤ n execute
    If x[index] = e then
      x[1] ← e
      x[index] ← tmp
      Return True
    EndIf
    tmp2 ← x[index]
    x[index] ← tmp
    tmp ← tmp2
    index ← index + 1
  EndWhile
  Return False
EndAlgorithm

```

C.

```

Algorithm ceva(x, n, e):
  index ← n
  While index > 1 execute
    If x[index] = e then
      index2 ← index
      While index2 > 1 execute
        x[index2] ← x[index2 - 1]
        index2 ← index2 - 1
      EndWhile
      x[index2] ← e
    EndIf
    index ← index - 1
  EndWhile
  If x[1] = e then
    Return True
  Else
    Return False
  EndIf
EndAlgorithm

```

D.

Keine der Varianten A, B, C

17. Gegeben sei der Algorithmus $\text{expresie}(x, y, z)$, wobei x, y, z natürliche Zahlen sind ($0 \leq x, y, z \leq 10^4$):

```

Algorithm expresie(x, y, z):
  If x = 0 then
    Return z
  Else
    Return expresie(x - 1, y, x * x + y * y + z)
  EndIf
EndAlgorithm

```

Geben Sie den Ausdruck an, dessen Wert der Algorithmus berechnet und zurückgibt:

- A. $\sum_{i=1}^x i^2 + \sum_{i=1}^y x * y + \sum_{k=1}^z 1$
- B. $\sum_{i=1}^x i^2 + \sum_{j=1}^y j^2 + z$
- C. $\sum_{i=1}^x i^2 + x * y^2 + z$
- D. $\sum_{i=1}^x i^2 + \sum_{j=1}^y j^2 + \sum_{k=1}^z k$

18. Gegeben sei der Algorithmus $\text{ceFace}(v, a, b)$, wobei v ein Vektor mit n Elementen aus der Menge $\{0, 1\}$ ist, ($1 \leq n \leq 10, v[1], \dots, v[n]$), und a und b natürliche Zahlen ungleich Null sind. Der Vektor v ist aufsteigend geordnet.

```

Algorithm ceFace(v, a, b):
  If b - a + 1 = 0 then
    Return 0
  EndIf
  If v[a] = 1 then
    Return b - a + 1
  EndIf
  If v[b] = 0 then
    Return 0
  EndIf
  c ← (a + b) DIV 2
  Return ceFace(v, a, c) + ceFace(v, c + 1, b)
EndAlgorithm

```

Welche der folgenden Aussagen sind wahr, wenn der erste Aufruf $\text{ceFace}(v, 1, n)$ lautet?

- A. Wenn der Vektor v mindestens einen Wert gleich mit 1 enthält, wird die Länge des Vektors zurückgegeben.
- B. Wenn der Vektor v nur Werte gleich mit 1 enthält, wird der Wert von n zurückgegeben.
- C. Wenn der Vektor v nur Werte gleich mit 0 enthält, wird 0 zurückgegeben.
- D. Es wird die Anzahl der im Vektor v enthaltenen 1-Werte zurückgegeben.

19. Es ist bekannt, dass die Gesamtzahl der binären Zeichenketten (die nur die Zeichen 0 und 1 enthalten) der Länge n gleich mit 2^n ist. Für $n = 2$ sind dies zum Beispiel 00, 01, 10 und 11, und deren Anzahl ist $2^2 = 4$. Die Zeichenkette 100011 hat die Länge 6 und enthält als Teilfolge alle 4 möglichen Zeichenketten der Länge $n = 2$, denn ab der ersten Stelle erscheint 10, ab der zweiten Stelle 00, ab der vierten Stelle 01 und ab der fünften Stelle 11.

Was ist die minimale Länge einer Zeichenkette, die als Teilfolge alle 2^n möglichen binären Zeichenketten für $n = 4$ enthält?

- A. 18 B. 19 C. 20 D. 21

20. Gegeben sei der Algorithmus $t(q, x, y)$, wobei q ein beliebiges Zeichen ist, und x und y natürliche Zahlen ungleich Null sind ($1 \leq x, y \leq 100$).

```

Algorithm t(q, x, y):
  If x ≤ y then
    Write q
  Else
    If x MOD y = 0 then
      t(q, x + 1, y - 2)
    Else
      If (x DIV y) MOD 2 ≠ 0 then
        t(q, x - 1, y + 2)
        Write 'c'
      Else
        t(q, x - 1, y - 1)
        Write "cc"
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. Nach den Aufrufen $t('c', 33, 28)$, $t('c', 10, 6)$ und $t('c', 22, 16)$ werden die gleichen Zeichen angezeigt.
- B. Nach den Aufrufen $t('c', 33, 28)$ und $t('c', 45, 40)$ werden nicht die gleichen Zeichen angezeigt.
- C. Nach dem Aufruf $t('c', 11, 8)$ wird 'cc' angezeigt.
- D. Nach dem Aufruf $t('c', 25, 16)$ wird nicht "cccc" angezeigt.

21. Gegeben sei der Algorithmus $hIndex(x, n)$, wobei x ein Vektor mit n ($1 \leq n \leq 10^5$) natürlichen Zahlenelementen ungleich Null ($x[1], x[2], \dots, x[n]$) ist. Wir definieren den ***h-Index*** des Vektors x als den größten Wert v , für den gilt, dass es mindestens v Werte in x gibt, die größer oder gleich v sind. Zum Beispiel für $x = [3, 10, 2, 7, 10, 8, 50, 1, 1, 5]$ ist der ***h-Index*** 5.

```

1. Algorithm hIndex(x, n):
2.   h ← 1
3.   cont ← True
4.   While cont = True AND h ≤ n execute
5.     pos ← h
6.     For i ← h + 1, n execute
7.       If x[i] > x[pos] then
8.         pos ← i
9.       EndIf
10.    EndFor
11.    If pos ≠ h then
12.      tmp ← x[pos]
13.      x[pos] ← x[h]
14.      x[h] ← tmp
15.    EndIf
16.    If x[h] ≥ h then
17.      h ← h + 1
18.    Else
19.      cont ← False
20.    EndIf
21.  EndWhile
22.  ...
23. EndAlgorithm

```

Welche der folgenden Aussagen sind zutreffend?

- A. In dem Moment wenn Zeile 22 ausgeführt wird, ist der Vektor x absteigend sortiert.
- B. Der Algorithmus $hIndex(x, n)$ gibt den ***h-Index*** des Vektors x zurück, wenn wir die ***Return h*** Anweisung in Zeile 22 schreiben.
- C. Der Algorithmus $hIndex(x, n)$ gibt den ***h-Index*** des Vektors x zurück, wenn wir die Anweisung ***Return h - 1*** in Zeile 22 schreiben.
- D. Wenn der Algorithmus $hIndex(x, n)$ für einen streng absteigend sortierten Vektor x aufgerufen wird, gibt der Algorithmus nicht den ***h-Index*** des Vektors x zurück, egal welche Anweisung wir in Zeile 22 hinzufügen.

22. Gegeben sei der Algorithmus $ceFace(n, k, x, p)$, wobei n, k und p natürliche, von Null verschiedene Zahlen sind ($1 \leq n, k, p \leq 10, p \leq n$), und x ein Vektor mit $p + 1$ natürlichen Zahlenelementen ist ($x[0], x[1], \dots, x[p]$). Wir nehmen an, dass $x[0]$ mit 0 initialisiert wird.

```

Algorithm ceFace(n, k, x, p):
  If k > p then
    For i ← 1, p execute
      Write x[i]
    EndFor
    Write " " //un singur spațiu
  Else
    For i ← x[k - 1] + 1, n execute
      x[k] ← i
      ceFace(n, k + 1, x, p)
    EndFor
  EndIf
EndAlgorithm

```

Bitte geben Sie an, welche der folgenden Antworten richtig sind.

- A. Nachdem der Algorithmus in der Form $ceFace(3, 1, x, 3)$ aufgerufen wurde, wird er sich selbst noch 6 weitere Male aufrufen.
- B. Wenn $x[0]$ mit einem anderen Wert als 0 initialisiert wird, dann ist nach dem Aufruf $ceFace(5, 1, x, 3)$ die Anzahl der angezeigten Leerzeichen verschieden von 10.
- C. Wenn der Algorithmus als $ceFace(5, 1, x, 4)$ aufgerufen wird, werden die Zahlen 1245 1234 1345 1235 2345 angezeigt, aber in einer anderen Reihenfolge.
- D. Wenn der Algorithmus als $ceFace(5, 1, x, 3)$ aufgerufen wird, ist das angezeigte Ergebnis 123 124 125 134 135 145 234 235 in dieser Reihenfolge.

23. Gegeben sei der Algorithmus $f(\text{sir}, s, d, p)$, wobei sir eine Zeichenkette ist und s, d, p natürliche Zahlen ungleich Null sind ($0 < s, d, p < 10^9$). Der "+"-Operator ist der Operator für die Verkettung zweier Zeichenketten. Der Algorithmus $\text{print}(a)$ gibt die Zeichenkette a aus und springt dann in eine neue Zeile.

```

1. Algorithm f(sir, s, d, p):
2.   If s = p AND d = p then
3.     print(sir)
4.   EndIf
5.   If s < p then
6.     f(sir + "-1 ", s + 1, d, p)
7.   EndIf
8.   If s > d then
9.     f(sir + " 1 ", s, d + 1, p)
10.  EndIf
11. EndAlgorithm

```

Geben Sie an, welche der folgenden Aussagen nach dem Aufruf $f("", 0, 0, 2)$ wahr sind:

- A. In getrennten Zeilen werden zwei Zeichenfolgen angezeigt, die jeweils 4 Zahlen enthalten, deren Summe 0 ist (z. B. ist die Summe der Zahlen in der Zeichenfolge "-1 1 -1 1" 0)
- B. Es wird nur "-1 -1 1 1" angezeigt.
- C. Es wird nur "-1 -1 1 1" angezeigt, aber der Algorithmus beendet seine Ausführung nicht aufgrund eines Fehlers.
- D. Wenn in Zeile 2 der Operator AND durch OR ersetzt würde, würde nur "-1 -1" angezeigt werden.

24. Gegeben sei der Algorithmus $\text{ceFace}(a, i, n)$, wobei i und n natürliche Zahlen sind ($1 \leq i, n \leq 100$) und a ein Vektor mit n ganzzahligen Elementen ist ($a[1], a[2], \dots, a[n], -100 \leq a[i] \leq 100$). In der Zeichenkette a gibt es mindestens eine positive Zahl. Der Algorithmus $\text{max}(x, y, z)$ gibt das Maximum der drei ganzen Zahlen x, y und z zurück ($-10^4 \leq x, y, z \leq 10^4$). Algorithmus $\text{ceFace}(a, 1, n)$ ruft den Algorithmus $\text{intermediar}(a, i, m, n)$ auf, wobei die Parameter a, i und n die obige Bedeutung haben und m eine natürliche Zahl ist ($1 \leq m \leq n$).

Algorithm $\text{intermediar}(a, i, m, n)$:

```

s ← 0
left ← a[m]
For k ← m, i, -1 execute
  s ← s + a[k]
  If s > left then
    left ← s
  EndIf
EndFor
s ← 0
right ← a[m]
For i ← m, n execute
  s ← s + a[i]
  If s > right then
    right ← s
  EndIf
EndFor
Return max(left, right, left + right - a[m])
EndAlgorithm

```

Algorithm $\text{ceFace}(a, i, n)$:

```

If i ≥ n then
  Return a[i]
EndIf
m ← (i + n) DIV 2
v1 ← ceFace(a, i, m - 1)
v2 ← ceFace(a, m + 1, n)
v3 ← intermediar(a, i, m, n)
Return max(v1, v2, v3)
EndAlgorithm

```

Geben Sie an, welche der folgenden Aussagen wahr sind, wenn der Algorithmus in der Form $\text{ceFace}(a, i, n)$ aufgerufen wird:

- A. Der Algorithmus ermittelt eine Position m des Vektors a , bei der entweder die Summe der Elemente an den Positionen $1, 2, \dots, m$ oder die Summe der Elemente an den Positionen $m, m + 1, \dots, n$ das für jede $1 \leq m \leq n$ erreichbare Maximum ist, und gibt die so erhaltene Maximalsumme zurück.
- B. Der Algorithmus gibt die maximale Summe zurück, die durch Summierung der Elemente einer Teilmenge der Werte des Vektors a erhalten werden kann.
- C. Der Algorithmus liefert die maximale Summe, die für eine Teilfolge des Vektors a erhalten werden kann.
- D. Wenn alle Elemente des Vektors a positiv sind, gibt der Algorithmus die Summe aller Elemente des Vektors a zurück.

Aufnahmeprüfung – 8. September 2023

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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