BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Aufnahmeprüfung – 19. Juli 2021

Schriftliche Prüfung in Informatik

1. Gegeben sei der folgende Unteralgorithmus, der als Eingangsparameter die natürliche Zahl $\boldsymbol{n}$ nimmt und eine natürliche Zahl zurückgibt.

```
Subalgorithm berechnung(n):
    E ← 1
    P ← 1
    i ← 2
    While i ≤ n do
        P ← (-1) * P * i
        E ← E + P
        i ← i + 1
    EndWhile
    return E
EndSubalgorithm
```

Welcher ist der vom Unteralgorithmus zurückgegebene Wert für $\boldsymbol{n} \geq 1$?

A. $1! - 2! + 3! - 4! + ... + (-1)^{\boldsymbol{n}+1} \cdot \boldsymbol{n}!$
B. $1 - 1! + 2! - 3! + ... + (-1)^{\boldsymbol{n}} \cdot \boldsymbol{n}!$
C. $1 - 1 \cdot 2 + 1 \cdot 2 \cdot 3 - 1 \cdot 2 \cdot 3 \cdot 4 + ... + (-1)^{\boldsymbol{n}+1} \cdot 1 \cdot 2 \cdot ... \cdot \boldsymbol{n}$
D. $1 + 1 \cdot 2 - 1 \cdot 2 \cdot 3 + 1 \cdot 2 \cdot 3 \cdot 4 + ... + (-1)^{\boldsymbol{n}} \cdot 1 \cdot 2 \cdot ... \cdot \boldsymbol{n}$

2. Eine Excel-Datei enthält $\boldsymbol{n}$ Datensätze, die von 1 bis $\boldsymbol{n}$ nummeriert sind. Diese Datensätze müssen in eine Word-Datei kopiert werden, in der die Datensätze in $\boldsymbol{r}$ Zeilen und $\boldsymbol{c}$ Spalten auf jeder Seite (außer der ersten und letzten Seite) angeordnet werden. Auf der ersten Seite des Word-Dokuments ist aufgrund des Vorhandenseins einer Kopfzeile die Anzahl der Zeilen $\boldsymbol{r_1}$, $\boldsymbol{r_1} < \boldsymbol{r}$ (die Anzahl der auf der ersten Seite vorhandenen Zeilen ist geringer).

Die Datensätze werden in der Word-Datei auf jeder Seite von oben nach unten in jeder Spalte angeordnet, wobei die Spalten von links nach rechts gefüllt werden: Wenn der erste Datensatz auf einer Seite die Sequenznummer $\boldsymbol{i}$ hat, befindet sich darunter der Datensatz mit der Sequenznummer ($\boldsymbol{i}$ + 1), und der Datensatz mit der Sequenznummer ($\boldsymbol{i}$ + $\boldsymbol{r}$) ist der erste Datensatz in Spalte 2 auf dieser Seite und so weiter.

Für $\boldsymbol{n} = 5000$, $\boldsymbol{r} = 46$, $\boldsymbol{r_1} = 12$ und $\boldsymbol{c} = 2$ auf welcher Seite des Word-Dokuments und in welcher Spalte wird der Datensatz mit der Sequenznummer $\boldsymbol{i} = 3245$ erscheinen?

A. Seite 36, letzte Spalte
B. Seite 37, erste Spalte
C. Seite 37, letzte Spalte
D. Seite 38, erste Spalte

3. Gegeben sei der Unteralgorithmus `was(m)`, mit $\boldsymbol{m}$ natürliche Zahl ($10 \leq \boldsymbol{m} \leq 10000$).

```
Subalgorithm was(m):
    If m = 0 then
        return 0
    EndIf
    If m MOD 9 = 0 then
        return 9
    EndIf
    return m MOD 9
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus gibt den Rest der Division der Zahl ***m*** durch 9 zurück.
B. Der Unteralgorithmus gibt die Anzahl der Teiler der Zahl ***m***, die durch 9 teilbar sind zurück.
C. Der Unteralgorithmus gibt die Prüfziffer der Zahl ***m*** (die Summe ihrer Ziffern, dann die Summe der Ziffern dieser Summe, bis die erhaltene Summe eine einstellige Zahl ist) zurück.
D. Der Subalgorithmus gibt die Prüfziffer der Zahl ***m*** (die Summe ihrer Ziffern, dann die Summe der Ziffern dieser Summe, bis die erhaltene Summe eine einstellige Zahl ist) zurück genau dann, wenn die Zahl ***m*** durch 9 teilbar ist.

4. Um Zahlen mit ***n*** Ziffern zu generieren, Ziffern die nur 0, 2, 9 sein können, wird ein Algorithmus benutzt, der für ***n*** = 2 die Zahlen 20, 22, 29, 90, 92, 99 in aufsteigender Reihenfolge generiert.
Für ***n*** = 4 und unter Benutzung vom selben Algorithmus, welche ist die Zahl, die gleich nach der Zahl 2009 generiert sein wird?

A. 2022
B. 2090
C. 2010
D. Keine der anderen Varianten

5. Gegeben sei der Unteralgorithmus `suchen(n)`, wobei ***n*** eine natürliche Zahl ($0 \leq n \leq 1000000$) ist.

```
Subalgorithm suchen(n):
    v ← 0
    If n = 0 then
        return 1
    else
        m ← n
        While m > 0 do
            If m MOD 10 = 0 then
                v ← v + 1
            EndIf
            m ← m DIV 10
        EndWhile
        return v
   EndIf
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus ermittelt und gibt zurück, wie viele Stellen die Zahl ***n*** hat.
B. Der Unteralgorithmus gibt 1 zurück, wenn die Zahl ***n*** eine Potenz von 10 ist und sonst 0.
C. Der Unteralgorithmus gibt 1 zurück, wenn die Zahl ***n*** auf die Ziffer 0 endet und sonst 0.
D. Der Unteralgorithmus ermittelt und gibt die Anzahl der Stellen 0 in der Zahl ***n*** zurück.

6. Gegeben sei der Unteralgorithmus `abc(a, n, p)`, mit ***n*** eine natürliche Zahl ($1 \leq n \leq 10000$), ***p*** ist eine ganze Zahl ($-10000 \leq p \leq 10000$), und ***a*** ist eine Zeichenfolge aus ***n*** von Null verschiedene natürliche Zahlen (***a***[1], ***a***[2], …, ***a***[n]).

```
Subalgorithm abc(a, n, p):
    If n < 1 then
        return 0
    else
        If (1 ≤ p) AND (p ≤ n) then
            return a[p]
        else
            return -1
        EndIf
    EndIf
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus gibt -1 zurück genau dann, wenn ***p*** negativ oder größer als ***n*** ist.
B. Der Unteralgorithmus gibt das Element an der Stelle ***p*** zurück, wenn ***p*** größer als 0 und kleiner oder gleich der Länge der Zeichenfolge ist.
C. Der Unteralgorithmus gibt niemals 0 für Parameterwerte zurück, die die Vorbedingungen in der Anweisung erfüllen.
D. Der Unteralgorithmus gibt das Element an der Stelle ***p*** zurück, wenn ***p*** größer oder gleich 0 und streng kleiner als die Länge der Zeichenfolge ist. Wenn ***p*** nicht zwischen 1 und ***n*** liegt, wird -1 zurückgegeben.

7. Welche der folgenden Sequenzen bestimmen in der Variablen ***i*** die Länge einer Zeichenkette, die mit dem Zeichen '*' (Sternchen) endet? Das erste Zeichen steht auf Index 1 und das Sternchenzeichen ist Teil der Zeichenfolge.

A.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
```
B.
```
i ← 1
While x[i] = '*' do
    i ← i + 1
EndWhile
i ← i - 1
```
C.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
i ← i + 1
```
D.
```
i ← 1
While x[i] ≠ '*' do
    i ← i + 1
EndWhile
i ← i - 1
```

8. Gegeben sei der folgende Unteralgorithmus, der als Parameter die von Null verschiedene natürliche Zahl ***n*** nimmt und eine natürliche Zahl zurückgibt.

```
Subalgorithm f(n):
    j ← n
    While j > 1 do
       i ← 1
       While i ≤ n do
           i ← 2 * i
       EndWhile
       j ← j DIV 3
    EndWhile
    return j
EndSubalgorithm
```

In welche der folgenden Komplexitätsklassen fällt die Zeitkomplexität des Algorithmus?

A. $O(\log_2 \boldsymbol{n})$
B. $O(\log_2^2 \boldsymbol{n})$
C. $O(\log_3^2 \boldsymbol{n})$
D. $O(\log_2 \log_3 \boldsymbol{n})$

9. Der Unteralgorithmus `wieViele(n, m)` nimmt als Parameter die natürlichen Zahlen ***n*** und ***m***.

```
Subalgorithm wieViele(n, m):
    If n ≤ m then
        If (n MOD 2 = 0) AND (n MOD 3 ≠ 0) then
             return 1 + wieViele(n + 1, m)
        else
             return wieViele(n + 1, m)
        EndIf
    else
       return 0
    EndIf
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Falls ***n*** = 0 und ***m*** = 1, gibt der Unterlagorithmus den Wert 0 zurück.
B. Falls ***n*** = 4 und ***m*** = 21, gibt der Unterlagorithmus den Wert 6 zurück.
C. Falls ***n*** = 7 und ***m*** = 120, gibt der Unterlagorithmus den Wert 36 zurück.
D. Falls ***n*** = 1 und ***m*** = 215, gibt der Unterlagorithmus den Wert 72 zurück.

10. Gegeben sei der Unteralgorithmus `überprüfen(n)`, mit ***n*** eine natürliche Zahl (1 ≤ ***n*** ≤ 100000).

```
Subalgorithm überprüfen(n):
  While n > 0 do
    If (n MOD 3) > 1 then
      return 0
    EndIf
    n ← n DIV 3
  EndWhile
  return 1
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus gibt 1 zurück, wenn ***n*** eine Potenz von 3 ist, sonst 0.
B. Der Unteralgorithmus gibt 1 zurück, wenn die Basis-3-Schreibung von ***n*** nur die Ziffern 0 und/oder 1 enthält, sonst 0.
C. Der Unteralgorithmus gibt 1 zurück, wenn ***n*** als Summe von jeweils verschiedenen Potenzen von 3 geschrieben werden kann, sonst 0.
D. Der Unterbalgorithmus gibt 1 zurück, wenn die Basis-3-Schreibung von ***n*** nur die Ziffer 2 enthält, sonst 0.

11. Für eine natürliche Zahl ***nr*** (1000 ≤ ***nr*** ≤ 1000000) definieren wir die dekrementierende Operation wie folgt: Wenn die letzte Stelle von ***nr*** nicht 0 ist, subtrahieren wir 1 von ***nr***, andernfalls dividieren wir ***nr*** durch 10 und behalten nur den ganzzahligen Teil. Welcher der folgenden Unteralgorithmen gibt beim Aufruf `dekrementieren(nr, k)` die Zahl zurück, die man erhält, wenn man ***k***-mal (0 ≤ ***k*** ≤ 100) die Dekrement-Operation auf die Zahl ***nr*** anwendet? Zum Beispiel muss für ***nr*** = 15243 und ***k*** = 10 das Ergebnis 151 sein.

A.
```
Subalgorithm dekrementieren(nr, k):
    If k = 0 then
        return nr
    else
        If nr MOD 10 ≠ 0 then
            return dekrementieren(nr DIV 10, k - 1)
        else
            return dekrementieren(nr - 1, k - 1)
        EndIf
    EndIf
EndSubalgorithm
```

B.

```
Subalgorithm dekrementieren(nr, k):
    While k > 0 do
        If nr MOD 10 = 0 then
            nr ← nr DIV 10
        else
            nr ← nr - 1
        EndIf
    EndWhile
    return nr
EndSubalgorithm
```

C.

```
Subalgorithm dekrementieren(nr, k):
    For i ← 1, k do
        If nr MOD 10 > 0 then
            nr ← nr - 1
        else
            nr ← nr DIV 10
        EndIf
    EndFor
    return nr
EndSubalgorithm
```

D.

```
Subalgorithm dekrementieren(nr, k):
    If k = 0 then
        return nr
    else
        If k > nr MOD 10 then
            nr1 ← nr DIV 10
            return dekrementieren(nr1,  k - nr MOD 10 - 1)
        else
            return dekrementieren(nr - k, 0)
        EndIf
    EndIf
EndSubalgorithm
```

12. Gegeben sei der folgende Unteralgorithmus der als Eingangsparameter eine Zeichenfolge ***x*** aus ***n*** natürlichen Zahlen (***x***[1], ***x***[2], …, ***x***[***n***]) und die ganze Zahl ***n*** nimmt.

```
Subalgorithm f(x, n):
    If n = 1 then
        return 100
    else
        If x[n] > f(x, n - 1) then
            return x[n]
        else
            return f(x, n - 1)
        EndIf
    EndIf
EndSubalgorithm
```

Welches wird das Ergebnis der Ausführung des Unteralgorithmus für ***x*** = [101, 7, 6, 3] und ***n*** = 4 sein?

A. 101
B. 3
C. 100
D. 7

13. Der untenstehende Unteralgorithmus nimmt als Eingangsparameter eine Zeichenfolge $\boldsymbol{a}$ aus $\boldsymbol{n}$ natürlichen Zahlen ($\boldsymbol{a}[1]$, $\boldsymbol{a}[2]$, …, $\boldsymbol{a}[\boldsymbol{n}]$) und die natürliche Zahl $\boldsymbol{n}$ ($2 \leq \boldsymbol{n} \leq 10000$).

```
Subalgorithm h(a, n):
    If n ≤ 0 then
        return 0
    EndIf
    If (n MOD 2 = 0) AND (a[n] MOD 2 = 0) then
        return h(a, n - 1) + a[n]
    EndIf
    return h(a, n - 1) - a[n]
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus gibt die Differenz zwischen der Summe der Elemente, die die gleiche Parität haben wie die Position, an der sie stehen, und der Summe der Elemente, die eine andere Parität haben als die Position, an der sie in der Zeichenfolge $\boldsymbol{a}$ stehen, zurück.
B. Der Unteralgorithmus gibt die Differenz zwischen der Summe der geraden Elemente an den geraden Stellen und der Summe der ungeraden Elemente an den ungeraden Stellen in der Zeichenfolge $\boldsymbol{a}$ zurück.
C. Der Unteralgorithmus gibt die Differenz zwischen der Summe der geraden Elemente und der Summe der ungeraden Elemente in der Zeichenfolge $\boldsymbol{a}$ zurück.
D. Der Unteralgorithmus gibt die Differenz zwischen der Summe der geraden Elemente auf den geraden Positionen und der Summe der anderen Elemente in der Zeichenfolge $\boldsymbol{a}$ zurück.

14. Gegeben sei der Unteralgorithmus `was(n)`, der Parameter $\boldsymbol{n}$ ist eine von Null verschiedene natürliche Zahl.

```
Subalgorithm was(n):
    i ← 1
    While n > 0 do
        If n MOD 2 ≠ 0 then
            write i
        EndIf
        i ← i + 1
        n ← n DIV 2
    EndWhile
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus zeigt die Sequenz 12345 für $\boldsymbol{n} = 31$ an.
B. Der Unteralgorithmus zeigt die Sequenz 234 für $\boldsymbol{n} = 14$ an.
C. Der Subalgorithmus zeigt 1 am Anfang der Sequenz an, für ungerades $\boldsymbol{n}$.
D. Der Subalgorithmus zeigt eine einzelne Zahl für $\boldsymbol{n} = 2^{\boldsymbol{k}}$ an, wobei $\boldsymbol{k}$ eine natürliche Zahl ist.

15. Gegeben sei eine Menge $\boldsymbol{S}$ mit $\boldsymbol{n}$ Intervallen, die durch das linke Ende $\boldsymbol{s_i}$ und das rechte Ende $\boldsymbol{d_i}$ ($\boldsymbol{s_i} < \boldsymbol{d_i}\ \forall\ \boldsymbol{i} = 1...\boldsymbol{n}$) spezifiziert sind. Es soll eine Teilmenge $\boldsymbol{S'} \subseteq \boldsymbol{S}$ von $\boldsymbol{m}$ Elementen so bestimmt werden, dass sich keine zwei Intervalle in $\boldsymbol{S'}$ schneiden und $\boldsymbol{m}$ den größtmöglichen Wert hat.
Welche der folgenden Strategien löst das Problem richtig?

A. Die Intervalle aus der Menge $\boldsymbol{S}$ werden aufsteigend nach dem linken Ende sortiert. Das erste Intervall aus der sortierten Zeichenfolge wird $\boldsymbol{S'}$ hinzugefügt. Man gehe die anderen Elemente der Zeichenfolge in der sortierten Reihenfolge durch und wenn ein Intervall gefunden wird, das sich nicht mit dem Intervall überschneidet, das zuletzt zu $\boldsymbol{S'}$ hinzugefügt wurde, füge man es ebenfalls zu $\boldsymbol{S'}$ hinzu.
B. Man sortiert die Intervalle in der Menge $\boldsymbol{S}$ aufsteigend nach dem rechten Ende. Das erste Intervall aus der sortierten Zeichenfolge wird $\boldsymbol{S'}$ hinzugefügt. Man gehe die anderen Elemente der Zeichenfolge in der sortierten Reihenfolge durch und wenn ein Intervall gefunden wird, das sich nicht mit dem Intervall überschneidet, das zuletzt zu $\boldsymbol{S'}$ hinzugefügt wurde, füge man es ebenfalls zu $\boldsymbol{S'}$ hinzu.

C. Man sortiere die Intervalle in der Menge ***S*** aufsteigend nach der Intervall Länge. Das erste Intervall aus der sortierten Zeichenfolge wird ***S'*** hinzugefügt. Man gehe die anderen Elemente der Zeichenfolge in der sortierten Reihenfolge durch und wenn ein Intervall gefunden wird, das sich nicht mit dem Intervall überschneidet, das zuletzt zu ***S'*** hinzugefügt wurde, füge man es ebenfalls zu ***S'*** hinzu.

D. Man sortiere die Intervalle der Menge ***S*** aufsteigend nach der Anzahl der Intervalle in ***S***, mit denen sie sich schneiden. Das erste Intervall aus der sortierten Zeichenfolge wird ***S'*** hinzugefügt. Man gehe die anderen Elemente der Zeichenfolge in der sortierten Reihenfolge durch und wenn ein Intervall gefunden wird, das sich nicht mit dem Intervall überschneidet, das zuletzt zu ***S'*** hinzugefügt wurde, füge man es ebenfalls zu ***S'*** hinzu.

16. Gegeben sei der Unteralgorithmus `f(a, b)`, der als Parameter zwei natürliche Zahlen ***a*** und ***b*** (1 ≤ ***a*** < ***b*** ≤ 1000) nimmt.

```
Subalgorithm f(a, b):
    m ← 0
    For n ← a, b do
        c ← 0
        For d ← 1, n do
            If n MOD d = 0 then
                c ← c + 1
            EndIf
        EndFor
        If c > m then
            m ← c
        EndIf
    EndFor

    For n ← a, b do
        c ← 0
        For d ← 1, n do
            If n MOD d = 0 then
                c ← c + 1
            EndIf
        EndFor
        If c = m then
            write n
        EndIf
    EndFor
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus zeigt das Maximum aus der Anzahl der Teiler von ***a*** und der Anzahl der Teiler von ***b*** an.

B. Der Unteralgorithmus zeigt die natürlichen Zahlen im Intervall [***a***, ***b***] an, die die Eigenschaft haben, dass sie die größte Anzahl von Teilern haben.

C. Der Unteralgorithmus zeigt die Anzahl der Teiler für jede natürliche Zahl im Intervall [***a***, ***b***] an.

D. Der Unteralgorithmus zeigt die natürlichen Zahlen im Intervall [***a***, ***b***] an, die die Eigenschaft haben, dass sie die größte Anzahl von echten Teilern haben.

17. Gegeben seien die natürlichen Zahlen ***a*** und ***b***, mit ***b*** ≠ 0. Welche der folgenden Varianten berechnet:

- `a DIV b`, falls `a MOD b = 0`
- `(a / b)` aufgerundet auf die nächste ganze Zahl, wenn `a MOD b ≠ 0`

A. `(a - 1) DIV b`

B. `(a + b + 1) DIV b`

C. `(a + b - 1) DIV b`

D. `((a + 2 * b - 1) DIV b) - 1`

18. Ionel muss den binären Suchalgorithmus eines Elementes ***a*** in einer Folge ***V*** bestehend aus ***n*** ($1 \leq$ $n \leq 1000$) wachsend geordnete ganze Zahlen (***V***[1], ***V***[2], …, ***V***[***n***]) implementieren. Er schreibt den folgenden Unteralgorithmus:

```
Subalgorithm BinäreSuche(a, n, V):
     st ← 1
     dr ← n
     While dr – st > 1 execute
          m ← (st + dr) DIV 2
          If a ≤ V[m] then
               dr ← m
          else
               st ← m
          EndIf
     EndWhile
     return dr
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Für ***n*** = 1 ist der vom Unteralgorithmus zurückgegebene Wert immer 1.
B. Für jedes ***n*** ≥ 1, der von Ionel geschriebene Unteralgorithmus gibt 1 zurück, falls ***a*** kleiner als alle Elemente der Folge ist.
C. Wenn das Element ***a*** in der Zeichenkette erscheint, gibt der von Ionel geschriebene Unteralgorithmus NICHT immer die Position (Index im Vektor ***V***) zurück, an der es erscheint.
D. Für ein beliebiges ***n*** > 1 gibt der von Ionel geschriebene Unteralgorithmus den Wert ***n*** zurück, wenn ***a*** größer als alle Elemente der Zeichenfolge ist.

19. Gegeben sei der Unteralgorithmus `berechnung(x, n)`, mit Eingangsparameter natürliche Zahlen ***n*** und ***x***, mit $1 \leq x \leq n < 10$.

```
Subalgorithm berechnung(x, n):
    b ← 1
    For i ← 1, n - x do
        b ← b * i
    EndFor
    a ← b
    For i ← n – x + 1, n do
        a ← a * i
    EndFor
    return a DIV b
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Für ***n*** = 5 und ***x*** = 2, gibt der Unteralgorithmus 20 zurück.
B. Für ***n*** = 3 und ***x*** = 2, gibt der Unteralgorithmus 6 zurück.
C. Der Unteralgorithmus gibt die Anzahl der Elemente der Menge $\{\overline{c_1 c_2 \ldots c_x}\ : c_i \neq c_j\ \forall\ 1 \leq i, j \leq x, i \neq j,\ 1 \leq c_i \leq n\}$ zurück
D. Der Unteralgorithmus durchführt ***n*** Multiplikationen.

20. Gegeben sei der Unteralgorithmus `was(n,k)`, der als Parameter zwei natürliche, von Null verschiedene Zahlen ***n*** und ***k*** ($1 \leq n, k \leq 1000000$) nimmt.

```
Subalgorithm was(n, k):
    While n ≥ 1 execute
        If k ≤ n then
            i ← k
        else
            i ← n
        EndIf
        n ← n - i
```

```
                x ← 1
                While i ≥ 1 execute
                    Write x,' '
                    x ← x + 1
                    i ← i - 1
                EndWhile
            EndWhile
        EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Für ***n*** = 8 und ***k*** = 3 zeigt der Unteralgorithmus die Folge 1 2 3 1 2 3 1 2 an
B. Für ***k*** = 2 ist der kleinste Wert von ***n***, für den der Wert 1 dreimal auf dem Bildschirm angezeigt wird, ***n*** = 3.
C. Für ***k*** = 5 ist der kleinste Wert von ***n***, für den der Wert 2 37-mal auf dem Bildschirm angezeigt wird, ***n*** = 182.
D. Für ***n*** = 7 und ***k*** = 3 zeigt der Unteralgorithmus 1 2 3 1 2 3 an

21. Gegeben sei der Unteralgorithmus `berechnung(a, b, c)`, mit Eingabeparametern natürliche Zahlen ungleich Null, der den größten gemeinsamen Teiler der drei Zahlen berechnet.
Welche der folgenden Implementationen sind korrekte Implementierungen des Unteralgorithmus:

A.

```
Subalgorithm berechnung(a, b, c):
    While (a ≠ b) OR (a ≠ c) OR (b ≠ c) do
        x ← a
        If a ≠ x then
            a ← a - x
        EndIf
        If b ≠ x then
            b ← b - x
        EndIf
        If c ≠ x then
            c ← c - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

B.

```
Subalgorithm berechnung(a, b, c):
    x ← a
    y ← b
    While x ≠ y do
        If x > y then
            x ← x - y
        else
            y ← y - x
        EndIf
    EndWhile
    z ← c
    While x ≠ z do
        If x > z then
            x ← x - z
        else
            z ← z - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

C.

```
Subalgorithm berechnung(a, b, c):
    While (a ≠ b) OR (a ≠ c) OR (b ≠ c) do
        x ← a
        If b < x then
            x ← b
        EndIf
        If c < x then
            x ← c
        EndIf
        If a ≠ x then
            a ← a - x
        EndIf
        If b ≠ x then
            b ← b - x
        EndIf
        If c ≠ x then
            c ← c - x
        EndIf
    EndWhile
    return x
EndSubalgorithm
```

D.

```
Subalgorithm berechnung(a, b, c):
    x ← a
    y ← b
    r ← x MOD y
    While r ≠ 0 do
        x ← y
        y ← r
        r ← x MOD y
    EndWhile
    z ← c
    r ← y MOD z
    While r ≠ 0 do
        y ← z
        z ← r
        r ← y MOD z
   EndWhile
   return z
EndSubalgorithm
```

22. Der Unteralgorithmus `was(n)` nimmt als Parameter die natürliche Zahl $\boldsymbol{n}$ ($1 \leq \boldsymbol{n} \leq 100$).

```
Subalgorithm was(n):
    s ← 0
    If n MOD 2 = 0 then
        a ← 1
        While a < n do
            s ← s + a
            a ← a + 2
        EndWhile
    else
        b ← 2
        While b < n do
            s ← s + b
            b ← b + 2
        EndWhile
    EndIf
    return s
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Wenn ***n*** gerade ist, wird die Summe der natürlichen Zahlen streng kleiner als ***n*** zurückgegeben; wenn ***n*** ungerade ist, wird die Summe der geraden natürlichen Zahlen kleiner als ***n*** zurückgegeben.
B. Wenn ***n*** gerade ist, wird die Summe der geraden natürlichen Zahlen kleiner als ***n*** zurückgegeben; wenn ***n*** ungerade ist, wird die Summe der ungeraden natürlichen Zahlen kleiner als ***n*** zurückgegeben.
C. Wenn ***n*** gerade ist, wird die Summe der ungeraden natürlichen Zahlen kleiner als ***n*** zurückgegeben; wenn ***n*** ungerade ist, wird die Summe der geraden natürlichen Zahlen kleiner als ***n*** zurückgegeben.
D. Wenn ***n*** gerade ist, wird die Summe der geraden natürlichen Zahlen streng kleiner als ***n*** zurückgegeben; wenn ***n*** ungerade ist, wird die Summe der natürlichen Zahlen streng kleiner als ***n*** zurückgegeben.

23. Der Unteralgorithmus `was(a)` bekommt als Parameter die natürliche Zahl ***a*** ($1 \leq a \leq 100000$).

```
Subalgorithm was(a):
    b ← 0
    c ← 0
    d ← 0
    e ← 1
    While a > 0 do
        d ← a MOD 10
        If (d ≠ 4) AND (d < 7) then
            b ← b + e * (d DIV 2)
            c ← c + e * (d - d DIV 2)
        else
            b ← b + e
            c ← c + e * (d - 1)
        EndIf
        a ← a DIV 10
        e ← e * 10
    EndWhile
    write b
    write c
EndSubalgorithm
```

Welche der folgenden Wertepaare werden für keinem gültigen Eingabewert angezeigt?

A. 1112 und 11233
B. 1111 und 88888
C. 21001 und 33011
D. 3141 und 3258

24. Gegeben seien die Unteralgorithmen `f(n, c)` und `g(n, c)`, die als Parameter die natürlichen Zahlen ***n*** und ***c*** nehmen.

```
Subalgorithm f(n, c):
    If n ≤ 9 then
        If n = c then
            return 1
        else
            return 0
        EndIf
    else
        If n MOD 10 = c then
            return f(n DIV 10, c) + 1
        else
            return f(n DIV 10, c)
        EndIf
    EndIf
EndSubalgorithm
```

```
Subalgorithm g(n, c):
    If c = 0 then
        return 0
    else
        If f(n, c) > 0 then
            return g(n, c - 1) + 1
        else
            return g(n, c - 1)
        EndIf
    EndIf
EndSubalgorithm
```

Was gibt der Aufruf g(n, 9) zurück?

A. Es gibt die Anzahl der Ziffern der Zahl ***n*** zurück.
B. Es gibt die Anzahl der jeweils verschiedenen Ziffern der Zahl ***n*** zurück.
C. Es gibt die Anzahl der Ziffern größer als 1 der Zahl ***n*** zurück.
D. Keine der anderen Antworten ist richtig.

25. Auf einer Website hat jeder registrierte Benutzer anstelle eines Passworts einen Geheimcode aus ***n*** Ziffern. Um sich auf der Seite anzumelden, muss der Benutzer nicht den vollständigen Code eingeben, sondern die Seite generiert zufällig 3 verschiedene Positionen, ***p***1, ***p***2 und ***p***3, so dass 1 ≤ ***p***1 < ***p***2 < ***p***3 ≤ ***n*** ist und der Benutzer nur die Ziffern an diesen 3 Positionen eingeben muss. Wenn der Benutzercode z. B. 987654321 lautet und die Seite zufällig die Positionen 2, 5 und 7 erzeugt, muss der Benutzer die Ziffern 8, 5, 3 eingeben.
Unten sind die Werte, die von einem Benutzer für 9 Anmeldungen auf dieser Seite eingegeben wurden.
1, 2, 3
2, 9, 0
6, 3, 2
2, 0, 2
1, 4, 7
9, 3, 2
4, 4, 3
4, 3, 1
5, 6, 0

Unter der Annahme, dass alle 9 Anmeldungen gültig sind und der Benutzercode in der Zwischenzeit nicht geändert wurde, geben Sie an, welche der folgenden Aussagen wahr sind.

A. Der Benutzercode enthält die Ziffer 8 bestimmt nicht.
B. Der kürzest mögliche Code hat 12 Ziffern.
C. Der kürzestmögliche Code enthält mindestens 3-mal die Ziffer 2.
D. Die Summe der Ziffern des kürzest möglichen Codes kann 44 sein.

26. Gegeben sei der Unteralgorithmus `f(x, n)` mit ***x***, ***n*** natürliche Zahlen und ***x*** > 0.

```
Subalgorithm f(x, n):
    If n = 0 then
        return 1
    EndIf
    m ← n DIV 2
    p ← f(x, m)
    If n MOD 2 = 0 then
        return p * p
    EndIf
    return x * p * p
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus gibt x^n zurück.
B. Wenn es statt “`n MOD 2`” der Ausdruck “`m MOD 2`” wäre, würde der Unteralgorithmus x^n zurückgeben.
C. Die Zeilen nach dem Autoaufruf der Funktion werden nie ausgeführt.
D. Der Unteralgorithmus gibt x^n zurück, genau dann, wenn n gerade ist.

27. Gegeben sei der Unteralgorithmus `f2(a,b)` mit Parameter ***a*** und ***b*** natürliche Zahlen, und der Unteralgorithmus `f(arr, i, n, p)` mit Parameter die Zeichenfolge ***arr*** mit ***n*** ganze Zahlen (***arr***[1], ***arr***[2], …, ***arr***[***n***]), und die ganzen Zahlen ***i*** und ***p***.

```
Subalgorithm f2(a, b):
    If a > b then
        return a
    else
        return b
    EndIf
EndSubalgorithm

Subalgorithm f(arr, i, n, p):
    If i = n then
        return 0
    EndIf
    n1 ← f(arr, i + 1, n, p)
    n2 ← 0
    If p + 1 ≠ i then
        n2 ← f(arr, i + 1, n, i) + arr[i]
    EndIf
    return f2(n1, n2)
EndSubalgorithm
```

Geben Sie das Ergebnis des Aufrufs `f(arr, 1, 9, -10)`, falls die Zeichenfolge ***arr*** die Werte (10, 1, 3, 4, 8, 12, 1, 11, 6) enthält.

A. 24
B. 37
C. 26
D. 56

28. Gegeben sei der Unteralgorithmus `überprüfe(n)`, der eine ganze Zahl ***n*** ($1 \leq n \leq 100000$) als Parameter nimmt und wahr zurückgibt, wenn ***n*** eine Ziffer enthält, die gleich der Summe der anderen Ziffern ist. Zum Beispiel gibt `überprüfe(1517)` wahr zurück, weil 7 = 1 + 5 + 1 ist.
Welche der folgenden Implementationen sind korrekte Implementierungen des Unteralgorithmus `überprüfe(n)`?

A.
```
Subalgorithm überprüfe(n):
    s ← 0
    c ← n
    r ← false
    While c > 0 do
        s ← s + c MOD 10
        c ← c DIV 10
    EndWhile
    c ← n
    While c > 0 do
        d ← c MOD 10
        If d = s - d then
            r ← true
        else
            r ← false
        EndIf
        c ← c DIV 10
    EndWhile
```

```
    return r
EndSubalgorithm
```

B.

```
Subalgorithm überprüfe(n):
    m ← -1
    c ← n
    r ← false
    While c > 0 do
        d ← c MOD 10
        c ← c DIV 10
        If d > m then
            m ← d
        EndIf
    EndWhile
    c ← n
    s ← 0
    While c > 0 do
        d ← c MOD 10
        If d ≠ m then
            s ← s + d
        EndIf
        c ← c DIV 10
    EndWhile
    If s = m then
        r ← true
    EndIf
    return r
EndSubalgorithm
```

C.

```
Subalgorithm überprüfe(n):
    v ← [0,0,0,0,0,0,0,0,0]
    r ← false
    While n > 0 do
        d ← n MOD 10
        If d > 0 then
            v[d] ← v[d] + 1
        EndIf
        n ← n DIV 10
    EndWhile
    m ← 9
    While v[m] = 0 do
        m ← m - 1
    EndWhile
    If v[m] = 1 then
        d ← m
        s ← 0
        m ← m - 1
        While m > 0 do
            s ← s + v[m] * m
            m ← m - 1
        EndWhile
        If d = s then
            r ← true
        EndIf
    EndIf
    return r
EndSubalgorithm
```

D. Keine der anderen Varianten ist richtig.

29. Gegeben sei der Unteralgorithmus `f(x, n, e, y, m)`, der als Parameter eine Zeichenfolge ***x*** mit ***n*** ganzzahligen Elementen (***x***[1], ***x***[2], ..., ***x***[***n***]), eine Zeichenfolge ***y*** mit ***m*** ganzzahligen Elementen (***y***[1], ***y***[2], ..., ***y***[***m***]) und eine ganze Zahl ***e***, die nicht zur Zeichenfolge ***y*** gehört, annimmt. Der Unteralgorithmus gibt eine Zeichenfolge und eine natürliche Zahl zurück. Gegeben sind die folgenden Unteralgorithmen:

- `(c, p) ← verkettung(a, n, b, m)` der als Eingabeparameter eine Zeichenfolge ***a*** mit ***n*** ganzzahligen Elementen und eine Zeichenfolge ***b*** mit ***m*** ganzzahligen Elementen hat und die Zeichenfolge ***c*** mit ***p*** ganzzahligen Elementen zurückgibt, die die Verkettung der beiden Zeichenketten ***a*** und ***b*** darstellt, d. h.: ***a***[1], ***a***[2], ..., ***a***[***n***], ***b***[1], ***b***[2], ..., ***b***[***m***]
- `(c, p) ← differenz(a, n, b, m)`, die als Eingabeparameter eine Zeichenfolge ***a*** mit ***n*** ganzzahligen Elementen und eine Zeichenfolge ***b*** mit ***m*** ganzzahligen Elementen hat und die Zeichenfolge ***c*** mit ***p*** ganzzahligen Elementen zurücklgibt, die alle Elemente der Zeichenfolge ***a*** enthält (die restlichen Elemente in der Zeichenfolge behalten ihre ursprüngliche Reihenfolge bei), die nicht in Zeichenfolge ***b*** enthalten sind.

```
1.  Subalgorithm f(x, n, e, y, m):
2.      If n = 0 then
3.          return [], 0
4.      EndIf
5.      If x[1] ≠ e then
6.          s ← []
7.          s[1] ← x[1]
8.          (r1, l1) ← differenz(x, n, s, 1)
9.          (r2, l2) ← f(r1, l1, e, y, m)
10.         (r3, l3) ← verkettung(s, 1, r2, l2)
11.         return r3, l3
12.     else
13.         (r1, l1) ← f(y, m, e, x, n)
14.         s ← []
15.         s[1] ← x[1]
16.         (r2, l2) ← differenz(x, n, s, 1)
17.         (r3, l3) ← f(r2, l2, e, y, m)
18.         (r4, l4) ← verkettung(r1, l1, r3, l3)
19.         return r4, l4
20.     EndIf
21. EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

A. Der Unteralgorithmus `f(x, n, e, y, m)` konstruiert aus der Zeichenfolge ***x*** ein eindimensionales Array, in dem Vorkommen des Elements ***e*** gelöscht und anstelle jedes Vorkommens die Elemente von ***y*** eingefügt werden. Der Unteralgorithmus gibt das konstruierte Array und dessen Dimension zurück.
B. Wenn die Zeichenfolgen ***x*** und ***y*** keine gemeinsamen Elemente haben, dann enthält die vom Unteralgorithmus `f(x, n, e, y, m)` zurückgegebene Zeichenfolge nur unterschiedliche Elemente.
C. Die Länge der Zeichenfolge, die vom Unteralgorithmus `f(x, n, e, y, m)` zurückgegeben wird, mit nicht leeren ***x***- und ***y***-Zeichenfolgen als Eingabeparameter, kann kleiner als ***n*** sein.
D. Wenn wir in Zeile 18 statt ***r1*** und ***l1*** ***y*** und ***m*** hätten, würde die Funktion ein eindimensionales Array (und dessen Dimension) zurückgeben, beginnend mit den Elementen von ***y***, gefolgt von den Elementen von ***x***, wobei Vorkommen von ***e*** durch die Elemente von ***y*** ersetzt werden.

30. Gegeben sei der Unteralgorithmus `s(a, b, c)`, mit ***a***, ***b***, ***c*** von Null verschiedene natürliche Zahlen, $\boldsymbol{b} \geq \boldsymbol{a}$

```
Subalgorithm s(a, b, c):
    If c = 0 then
        return 1
    else
        If a > b then
            return (1 / a) * s(a - 1, b, c)
        else
            If a < b then
                return (1 / b) * s(a, b - 1, c)
            else
                return c * s(a - 1, b - 1, c - 1)
            EndIf
        EndIf
    EndIf
EndSubalgorithm
```

Wie muss das Verhältnis zwischen ***a***, ***b*** und ***c*** sein, um $1/C_b^a$ zu erhalten (wobei C_b^a die Kombinationen von ***a*** Elementen aus ***b*** bezeichnet)?

A. a + b = c
B. a + c = b
C. b - c = a
D. b + c = a - b

BABEŞ-BOLYAI UNIVERSITÄT

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Aufnahmeprüfung – 19. Juli 2021

Schriftliche Prüfung in Informatik

PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

1	A, C	3 punkte
2	B	3 punkte
3	C	3 punkte
4	D	3 punkte
5	D	3 punkte
6	B,C	3 punkte
7	A	3 punkte
8	B, C	3 punkte
9	A,B,D	3 punkte
10	B, C	3 punkte
11	C, D	3 punkte
12	C	3 punkte
13	D	3 punkte
14	A, B, C, D	3 punkte
15	B	3 punkte
16	B	3 punkte
17	C, D	3 punkte
18	A, C, D	3 punkte
19	A, B, C, D	3 punkte
20	A,C	3 punkte
21	B, D	3 punkte
22	C	3 punkte
23	B, D	3 punkte
24	D	3 punkte
25	B, D	3 punkte
26	A	3 punkte
27	B	3 punkte
28	D	3 punkte
29	B, C	3 punkte
30	B, C	3 punkte