

Aufnahmeprüfung –9. September 2021
Schriftliche Prüfung in Informatik

1. Gegeben sei der Unteralgorithmus $\text{was}(n)$, mit n eine natürliche Zahl ($1 \leq n \leq 10000$).

```
Subalgorithm was(n):  
  nr ← 0  
  For d ← 1, n execute  
    If n MOD d = 0 then  
      nr ← nr + 1  
    EndIf  
  EndFor  
  If nr = 2 then  
    return true  
  else  
    return false  
  EndIf  
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus gibt *wahr* zurück, falls die Zahl n ungerade ist.
- B. Der Unteralgorithmus gibt *wahr* zurück, falls die Zahl n gerade ist.
- C. Der Unteralgorithmus gibt *wahr* zurück, falls die Zahl n eine Primzahl ist.
- D. Der Unteralgorithmus gibt *wahr* zurück, falls die Zahl n eine Quadratzahl ist.

2. Bekannt ist, dass $x < y$ (x und y sind reelle Zahlen). Welche der folgenden Ausdrücke hat den Wahrheitswert *wahr* genau dann, wenn die in t gespeicherte Zahl (t reelle Zahl) gehört NICHT dem Intervall (x, y) ?

- A. $(t > x)$ ODER $(t < y)$
- B. $(t \leq x)$ ODER $(t \geq y)$
- C. $(t \leq x)$ UND $(t \geq y)$
- D. $(t > x)$ UND $(t < y)$

3. Gegeben sei der Unteralgorithmus $f(n)$, mit n eine natürliche Zahl ($1 \leq n \leq 10000$).

```
Subalgorithm f(n):  
  r ← 0  
  While n > 0 execute  
    r ← r + (n MOD 10) * (n MOD 2)  
    n ← n DIV 10  
  EndWhile  
  return r  
EndSubalgorithm
```

Wählen Sie die Variante, die den unterstrichenen Teil des untenstehenden Unteralgorithmus korrekt ausfüllt ist, so dass die beiden Unteralgorithmen immer denselben Wert zurückgeben.

```
Subalgorithm fr(n):  
  If n > 0 then  
    return _____  
  EndIf  
  return 0  
EndSubalgorithm
```

- A. $(n \text{ MOD } 2) * (n \text{ MOD } 10) + fr(n \text{ DIV } 10)$
- B. $(n \text{ MOD } 2) * (n \text{ MOD } 10) * fr(n \text{ DIV } 10)$
- C. $(n \text{ MOD } 10) + fr(n \text{ DIV } 10)$
- D. $(n \text{ MOD } 2) * (n \text{ MOD } 10) + fr(n \text{ MOD } 10)$

4. Gegeben sei der Unteralgorithmus $f(n)$, mit n eine natürliche Zahl ($1 \leq n \leq 10000$).

```

Subalgorithm f(n):
  For i ← 1, n execute
    For j ← 1, 2 * i - 1 execute
      write '*'
    EndFor
  EndFor
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Für $n = 3$ zeigt der Subalgorithmus 3 Sternchen an
- B. Für $n = 3$ zeigt der Subalgorithmus 9 Sternchen an
- C. Damit der Subalgorithmus 1154 Sternchen anzeigt, muss der Wert von n 34 sein
- D. Damit der Subalgorithmus 289 Sternchen anzeigt, muss der Wert von n 17 sein

5. Der untenstehende Unteralgorithmus hat als Eingangsparameter ein Vektor v bestehend aus n natürlichen Zahlen ($v[1], v[2], \dots, v[n]$) und die ganze Zahl n ($2 \leq n \leq 10000$). Der Operator $/$ ist die reelle Division (z.B. $3/2=1,5$). Der Vektor v enthält mindestens eine gerade Zahl und mindestens eine ungerade Zahl.

```

Subalgorithm fn(v, n):
  a ← 0
  b ← 0
  For i ← 1, n execute
    If v[i] MOD 2 = 0 then
      a ← a + v[i]
      b ← b + 1
    EndIf
  EndFor
  return a / b
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus gibt die Anzahl gerader Elemente des Vektors v zurück.
- B. Der Unteralgorithmus gibt den Mittelwert der geraden Elemente des Vektors v zurück.
- C. Der Unteralgorithmus gibt die Summe der geraden Elemente des Vektors v zurück.
- D. Der Unteralgorithmus gibt den Mittelwert der ungeraden Elemente des Vektors v zurück.

6. Der untenstehende Unteralgorithmus hat als Eingangsparameter ein Vektor v bestehend aus n natürlichen Zahlen ($v[1], v[2], \dots, v[n]$) und die ganze Zahl n ($1 \leq n \leq 10000$).

```

Subalgorithm fn(v, n):
  a ← 0
  For i ← 1, n execute
    ok ← 1
    b ← v[i]
    While (b ≠ 0) AND (ok = 1) execute
      If b MOD 2 = 0 then
        ok ← 0
      EndIf
      b ← b DIV 10
    EndWhile
    If ok = 1 then
      a ← a + v[i]
    EndIf
  EndFor
  return a
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus gibt die Summe der ungeraden Elemente des Vektors v zurück.
- B. Der Unteralgorithmus gibt die Summe derjenigen Elemente des Vektors v , die Potenzen von 2 sind zurück.
- C. Der Unteralgorithmus gibt die Summe derjenigen Elemente des Vektors v , die nur aus geraden Ziffern bestehen zurück.
- D. Der Unteralgorithmus gibt die Summe derjenigen Elemente des Vektors v , die nur aus ungeraden Ziffern bestehen zurück.

7. Geben Sie an welche der untenstehenden Algorithmen den Betrag (Absolutwert) einer ganzen Zahl berechnet. Wir nehmen an, dass ein logischer Ausdruck den Wert 1 annimmt, falls er wahr ist und 0 falls er falsch ist.

A.
Subalgorithm modul(n):
 return n * (-2 * (n < 0) + 1)
EndSubalgorithm

B.
Subalgorithm modul(n):
 If n < 0 then
 return n * (-1)
 EndIf
 return n
EndSubalgorithm

C.
Subalgorithm modul(n):
 If n < 0 then
 return n * (-1)
 else
 return n
 EndIf
EndSubalgorithm

D.
Subalgorithm modul(n):
 If n > 0 then
 return n * (-1)
 else
 return n
 EndIf
EndSubalgorithm

8. Welcher ist der Wert des untenstehenden Ausdrucks, falls $x = 15$ und $y = 17$?

$(\text{NOT } (x \text{ MOD } 10 = 0)) \text{ UND } (y \text{ MOD } 2 = 0) \text{ UND } (x < y)$

- A. *wahr* B. *falsch* C. Fehlermeldung D. Der Ausdruck kann nicht evaluiert werden

9. Gegeben sei der rekursive Unteralgorithmus $\text{was}(n, i)$, wobei n eine natürliche Zahl ($2 \leq n \leq 1000$) ist.

```
Subalgorithm was(n, i):  
  If i = 1 then  
    return i  
  else  
    If n MOD i = 0 then  
      return i + was(n, i - 1)  
    else  
      return was(n, i - 1)  
    EndIf  
  EndIf  
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen für den Aufruf $\text{was}(n, n)$ wahr sind.

- A. Der Unteralgorithmus gibt den Nachfolger des größten Teilers von n zurück
- B. Der Unteralgorithmus gibt die Summe der natürlichen Zahlen, die keine Primzahlen sind, bis n (inklusive) zurück
- C. Der Unteralgorithmus gibt die Summe der echten Teiler der Zahl n zurück
- D. Der Unteralgorithmus gibt die Summe der echten und unechten Teiler der Zahl n zurück

10. Der Unteralgorithmus $\text{magic}(s, n)$ hat als Eingangsparameter eine Folge s mit n Zeichen ($s[1], s[2], \dots, s[n]$) und die ganze Zahl n ($1 \leq n \leq 10000$).

```

Subalgorithm magic(s, n)
  i ← 1
  f ← 1
  While i ≤ n DIV 2 execute
    If s[i] ≠ s[n - i + 1] then
      f ← 0
    EndIf
    i ← i + 1
  EndWhile
  return f
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus gibt 1 zurück, falls s eine gerade Anzahl von Zeichen hat.
- B. Der Unteralgorithmus gibt 1 zurück, falls s eine ungerade Anzahl von Zeichen hat.
- C. Der Unteralgorithmus gibt 1 zurück, falls s ein Palindrom ist.
- D. Der Unteralgorithmus gibt 1 zurück, falls s nur voneinander verschiedene Zeichen hat.

11. Welche der folgenden Ausdrücke haben den Wahrheitswert *wahr* genau dann, wenn x eine ungerade und negative Zahl ist? Wir bezeichnen mit $|x|$ den Betrag von x .

- A. $(|x| \text{ MOD } 2 = 1) \text{ UND } (x < 0)$
- B. $\text{NOT } ((|x| \text{ MOD } 2 = 0) \text{ UND } (x \geq 0))$
- C. $\text{NOT } ((|x| \text{ MOD } 2 = 0) \text{ ODER } (x \geq 0))$
- D. $(|x| \text{ MOD } 2 \neq 0) \text{ ODER } (x < 0)$

12. Der Unteralgorithmus $\text{was}(n)$ hat als Eingangsparameter eine natürliche Zahl n ($0 \leq n \leq 10000$).

```

Subalgorithm was(n):
  s ← 0
  While n > 0 execute
    c ← n MOD 10
    If c MOD 2 ≠ 0 then
      s ← s + c
    EndIf
    n ← n DIV 10
  EndWhile
  return s
EndSubalgorithm

```

Was gibt der Aufruf $\text{was}(1234)$ zurück?

- A. 4
- B. 10
- C. 60
- D. 0

13. Gegeben sei eine Zeichenfolge und eine Funktion f , die ein Zeichen als Parameter nimmt und 1 zurückgibt, wenn dieses Zeichen eine Ziffer ist, und ansonsten 0. Welche der folgenden Ansätze ermittelt, ob die Zeichenfolge nur aus Ziffern besteht?

- A. Wir prüfen, dass die Funktion f , angewendet auf jedes Zeichen der Zeichenfolge, immer 1 zurückgibt.
- B. Wir prüfen, ob die Summe der von f zurückgegebenen Werte, angewandt auf jedes Zeichen der Zeichenfolge, der Länge der Zeichenfolge entspricht.
- C. Wir prüfen, ob die Funktion f , angewendet auf jedes Zeichen der Zeichenkette, mindestens einmal 1 zurückgibt.
- D. Wir wenden die Funktion f auf zufällig ausgewählte Zeichen aus der Zeichenfolge an, bis eine Anzahl von Werten gleich 1 zurückgegeben wird, die der Länge der Zeichenfolge entspricht.

14. Welche der folgenden Algorithmen können so implementiert werden, dass sie eine lineare Zeitkomplexität ($O(n)$) haben?

- A. Algorithmus zur sequentiellen Suche eines Elementes in einem Vektor von n Zahlen
- B. Algorithmus zur Einfügungssortierung eines eindimensionalen Arrays von n Zahlen
- C. Algorithmus zum Finden der maximalen Zahl in einem unsortierten Vektor von n Zahlen
- D. Algorithmus zur Bestimmung der Summe der Elemente auf der Hauptdiagonalen einer quadratischen Matrix mit n Zeilen und n Spalten.

15. Gegeben sei der Unteralgorithmus $f(a, b)$, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10000$).

```

Subalgorithm f(a, b):
  m ← a
  While b MOD m > 0 execute
    m ← m + 1
  EndWhile
  return m
EndSubalgorithm

```

Bei welchem der folgenden Aufrufe wird der Körper der **while**-Schleife höchstens einmal ausgeführt?

- A. $f(10, 11)$ B. $f(10, 10)$ C. $f(10, 9)$ D. $f(10, 15)$

16. Gegeben sei der Unteralgorithmus $f(a, b)$, wobei a und b natürliche Zahlen sind ($1 \leq a, b \leq 10000$).

```

Subalgorithm f(a, b):
  c ← 0
  d ← 0
  p ← 1
  While a + b + c > 0 execute
    c ← a MOD 10 + b MOD 10 + c
    d ← d + (c MOD 10) * p
    p ← p * 10
    a ← a DIV 10
    b ← b DIV 10
    c ← c DIV 10
  EndWhile
  return d
EndSubalgorithm

```

Was wird der Aufruf $f(493, 1836)$ zurückgeben?

- A. 2329 B. 2229 C. 2430 D. 3292

17. Gegeben sei der Unteralgorithmus $\text{print}(M, n)$, der als Parameter eine Folge M aus n ($n \leq 10$) ganzen Zahlen ($M[1], M[2], \dots, M[n]$) nimmt, die eine Menge darstellen.

```

Subalgorithm print(M, n):
  nr ← 2n
  k ← 0
  While k < nr execute
    curent ← k
    write '{'
    For j = 1, n execute
      r ← curent MOD 2
      curent ← curent DIV 2
      If r = 1 then
        write M[j]
      EndIf
    EndFor
    write '}'
    write new line
    k ← k + 1
  EndWhile
EndSubalgorithm

```

Geben Sie an, welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus zeigt alle Permutationen der Menge M an.
- B. Der Unteralgorithmus zeigt alle Kombinationen von i Elemente der Menge M an, $i = 0, 1, \dots, n$ (nicht unbedingt in dieser Reihenfolge).
- C. Der Unteralgorithmus zeigt alle Variationen von i Elemente der Menge M an, $i = 0, 1, \dots, n$ (nicht unbedingt in dieser Reihenfolge).
- D. Der Unteralgorithmus zeigt alle Teilmengen der Menge M an.

18. Gegeben sei der Unteralgorithmus $s(a, b, c)$, wobei a, b, c natürliche positive Zahlen sind ($1 \leq a, b, c \leq 10000$).

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

Gegen Sie an welche der folgenden Aussagen für $a = b$ und $a < c$ wahr sind:

- A. Der Unteralgorithmus berechnet und gibt $c!$ zurück
- B. Der Unteralgorithmus berechnet und gibt $c! / (c - a + 1)!$ zurück
- C. Der Unteralgorithmus berechnet und gibt $c! / (c - a - 1)!$ zurück
- D. Der Unteralgorithmus berechnet und gibt Variationen von $(a - 1)$ aus c zurück

19. Der untenstehende Unteralgorithmus nimmt als Eingangsparameter eine Zeichenfolge A mit n natürlichen Zahlen ($A[1], A[2], \dots, A[n]$) und die natürliche Zahl n ($1 \leq n \leq 10000$). Für die natürlichen Zahlen x und y , x^y bedeutet x hoch y (x^y).

```
Subalgorithm h(A, n):
  If n = 0 then
    return 0
  else
    return A[n] * (-1)^(1 - A[n] MOD 2) + h(A, n - 1)
  EndIf
EndSubalgorithm
```

Geben Sie an welche der folgenden Aussagen wahr sind:

- A. Der Unteralgorithmus gibt die Differenz zwischen der Summe der Elemente an geraden Positionen und der Summe der Elemente an ungeraden Positionen in der Zeichenfolge A zurück
- B. Der Unteralgorithmus gibt die Differenz zwischen der Summe der geraden Elemente und der Summe der ungeraden Elemente in der Zeichenfolge A zurück
- C. Der Unteralgorithmus gibt die Differenz zwischen der Summe der ungeraden Elemente und der Summe der geraden Elemente in der Zeichenfolge A zurück
- D. Keine der anderen Antworten ist richtig

20. Eine Excel-Datei enthält n Datensätze mit den Sequenznummern 1 bis n . Diese Datensätze sollen in eine Word-Datei kopiert werden, in der die Datensätze in maximal r Zeilen und genau c Spalten auf jeder Seite angeordnet werden. Es ist gewährleistet, dass der Wert von n immer die Anordnung in genau c Spalten erlaubt.

Wir bezeichnen mit x_1, \dots, x_c die Anzahl der Datensätze, die in jede Spalte auf einer bestimmten Seite kopiert werden.

Auf der ersten Seite des Word-Dokuments ist aufgrund des Vorhandenseins einer Kopfzeile die Anzahl der Zeilen r_1 , $r_1 < r$ (die Anzahl der auf der ersten Seite vorhandenen Zeilen ist geringer), d. h. $x_p = r_1$, $\forall 1 \leq p \leq c$.

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 i hat, befindet sich darunter der Datensatz mit der Sequenznummer $(i + 1)$, und der Datensatz mit der Sequenznummer $(i + x_1)$ ist der erste Datensatz in Spalte 2 auf dieser Seite und so weiter.

Auf der letzten Seite des Word-Dokuments soll die Anzahl der Datensätze auf allen Spalten ausgeglichen sein, d. h. die Differenz zwischen der Anzahl der Datensätze auf zwei beliebigen Spalten ist höchstens 1 ($|x_j - x_k| \leq 1$, $\forall 1 \leq j, k \leq c, j \neq k$).

Bei den anderen Seiten (außer der ersten und letzten) ist $x_p = r$, $\forall 1 \leq p \leq c$.

Für $n = 5883$, $r = 46$, $r_1 = 12$ und $c = 2$ auf welcher Zeile der Seite kann der letzte Datensatz im Dokument (der mit der Sequenznummer $i = 5883$) gefunden werden?

- A. 29 B. 30 C. 31 D. 32

21. Gegeben sei der Unteralgorithmus `verarbeitung(a, b, c, d, e)`, der als Eingangsparameter fünf ganze Zahlen a, b, c, d und e ($1 \leq a, b \leq 10000$, $2 \leq c \leq 16$, $1 \leq d < c$) nimmt.

```

Subalgorithm verarbeitung(a, b, c, d, e):
  If a = 0 AND b = 0 then
    If e = 0 then
      return 1
    else
      return 0
    EndIf
  EndIf
  If (a MOD c = d) AND (b MOD c = d) then
    return verarbeitung(a DIV c, b DIV c, c, d, e)
  EndIf
  If a MOD c = d then
    return verarbeitung(a DIV c, b DIV c, c, d, e + 1)
  EndIf
  If b MOD c = d then
    return verarbeitung(a DIV c, b DIV c, c, d, e - 1)
  else
    return verarbeitung(a DIV c, b DIV c, c, d, e)
  EndIf
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen für den Aufruf `verarbeitung(a, b, c, d, 0)` wahr sind:

- A. Gibt 1 zurück, wenn die c -Basis-Darstellungen der Zahlen a und b die Ziffer d gleich oft enthalten, sonst 0
- B. Gibt 1 zurück, wenn die Ziffer d in der c -Basis-Darstellung der Zahl a und in der c -Basis-Darstellung der Zahl b vorkommt, sonst 0
- C. Gibt 1 zurück, wenn die Ziffer d in der c -Basis-Darstellung der Zahl a oder in der c -Basis-Darstellung der Zahl b vorkommt, sonst 0
- D. Gibt 1 zurück, wenn die Ziffer d in den c -Basis-Darstellungen der Zahlen a und b überhaupt nicht vorkommt, sonst 0

22. Betrachten wir die Unteralgorithmen $\text{val}(p, s, i, n, x)$ und $\text{val_exp}(p, n, x)$, deren Parameter folgende Spezifikation haben: eine Zeichenfolge p mit n ganzen Zahlen ($p[1], p[2], \dots, p[n]$), die natürlichen Zahlen s, i und n ($n \leq 1000, n = 2^k, k < 10$), und die reelle Zahl x . Die Werte der Zeichenfolge p sind die Koeffizienten des Ausdrucks in aufsteigender Reihenfolge der Exponenten, wobei der maximale Exponent gleich $n - 1$ ist, in einem Ausdruck der Form $p[1] + p[2] \cdot x + p[3] \cdot x^2 + \dots + p[n] \cdot x^{n-1}$

Beispiel: $p = [1, 2, 3, 4]$ entspricht dem Ausdruck $E(x) = 1 + 2x + 3x^2 + 4x^3$.

```

Subalgorithm val(p, s, i, n, x):
  If s + i ≤ n then
    _____
  else
    return p[s]
  EndIf
EndSubalgorithm

Subalgorithm val_exp (p, n, x):
  return val(p, 1, 1, n, x)
EndSubalgorithm

```

Welche der folgenden Möglichkeiten füllt die unterstrichene Stelle korrekt aus, so dass der Unteralgorithmus $\text{val_exp}(p, n, x)$ den Wert des Ausdrucks $E(x)$ liefert?

- A. `return p[s] + x * val(p, s + i, i * 2, n, x * x)`
- B. `return val(p, s, i * 2, n - i, x * x) + x * val(p, s + i, i * 2, n, x * x)`
- C. `return val(p, s + i, i * 2, n, x * x) + x * val(p, s, i * 2, n - i, x * x)`
- D. `return p[s] + x * val(p, s + i, i, n, x)`

23. Gegeben sei der Unteralgorithmus $f(a)$, der eine natürliche Zahl a ($2 \leq a < 1000000$) als Parameter nimmt und *true* zurückgibt, wenn es eine natürliche Zahl d , $1 < d < a$ mit der Eigenschaft gibt, dass d die Zahl a teilt, und ansonsten *false*. Wir bezeichnen mit $[x]$ den ganzen Teil der Zahl x .

Welche der folgenden Varianten des Unteralgorithmus $f(a)$ sind korrekt?

A.

```

Subalgorithm f(a):
  If a = 2 then
    return false
  EndIf
  If a MOD 2 = 0 then
    return true
  EndIf
  For d ← 3, [√a] - 1, 2 execute
    If a MOD d = 0 then
      return true
    EndIf
  EndFor
  return false
EndSubalgorithm

```

C.

```

Subalgorithm f(a):
  If a ≤ 2 then
    return false
  EndIf
  If a MOD 2 = 0 then
    return true
  EndIf
  For d ← 3, [√a], 2 execute
    If a MOD d = 0 then
      return true
    EndIf
  EndFor
  return false
EndSubalgorithm

```

B.

```

Subalgorithm f(a):
  For d ← 2, [√a] execute
    If a MOD d = 0 then
      return true
    EndIf
  EndFor
  return false
EndSubalgorithm

```

D.

```

Subalgorithm f(a):
  d ← a - 1
  While true execute
    If a MOD d = 0 then
      return true
    EndIf
    d ← d - 1
  EndWhile
  return false
EndSubalgorithm

```

24. Gegeben sei der untenstehende Ausdruck, mit $1 < A < 2021$ und $1 < n < 10202110$.

$$E(A, n) = (A + A^2 + A^3 + \dots + A^n) \text{ MOD } 2021$$

Welcher der folgenden Unteralgorithmen berechnet korrekt den Wert von $E(A, n)$ und hat die angegebene Zeitkomplexität?

Gehen Sie davon aus, dass alle Berechnungen mit 32-Bit-Datentypen durchgeführt werden. Nehmen Sie an, dass x^k in $O(\log k)$ berechnet wird.

A.

```
Subalgorithm E(A, n):  
    return (A * (An - 1) DIV (A - 1)) MOD 2021  
EndSubalgorithm
```

Zeitkomplexität: $O(\log n)$

B.

```
Subalgorithm E(A, n):  
    return ((A * (An - 1)) MOD 2021) DIV ((A - 1) MOD 2021)  
EndSubalgorithm
```

Zeitkomplexität: $O(\log n)$

C.

```
Subalgorithm E1(A, n):  
    If n = 1 then  
        return (A, A) //gibt ein Wertepaar zurück  
    EndIf  
    If n MOD 2 = 1 then  
        (t1, t2) ← E1(A, n - 1)  
        p ← (t1 * A) MOD 2021  
        return (p, (p + t2) MOD 2021)  
    else  
        (t1, t2) ← E1(A, n DIV 2)  
        p ← (t1 * t1) MOD 2021  
        return (p, ((1 + t1) * t2) MOD 2021)  
    EndIf  
EndSubalgorithm
```

```
Subalgorithm E(A, n):  
    (aux1, aux2) ← E1(A, n)  
    return aux2  
EndSubalgorithm
```

Zeitkomplexität: $O(\log n)$

D.

```
Subalgorithm E(A, n):  
    raspuns ← A  
    Pentru i = 2, n execută  
        raspuns ← raspuns + Ai  
    SfPentru  
  
    returnează raspuns MOD 2021  
SfSubalgorithm
```

Zeitkomplexität: $O(n \cdot \log n)$

25. Alle Zahlen von 1 bis 1000 werden in aufsteigender Reihenfolge im Uhrzeigersinn auf einen Kreis geschrieben. Beginnend bei 1, wird jede k -te Zahl im Uhrzeigersinn gefärbt ($1, k + 1, 2 * k + 1, \dots$). Der Vorgang wird so lange fortgesetzt, bis Sie eine bereits gefärbte Zahl erreichen, wobei am Ende x Zahlen gefärbt werden. Welche der folgenden Aussagen ist richtig?

- A. Falls $k = 15$ dann $x = 300$
- B. Falls $k = 45$ dann $x = 200$
- C. Falls $k = 25$ dann $x = 40$
- D. Falls $k = 30$ dann $x = 150$

26. Gegeben sei der Unteralgorithmus $\text{was}(n, k)$, mit n und k natürliche Zahlen ($1 \leq n, k \leq 1000000$).

```

Subalgorithm was(n, k):
  nr ← 0
  p ← 1
  While (n ≠ 0) AND (k ≠ 0) execute
    if n MOD 2 ≠ 0 then
      nr ← nr + ((n DIV 10) MOD 10) * p
      p ← p * 10
    else
      k ← k - 1
    EndIf
    n ← n DIV 10
  EndWhile
  return nr
EndSubalgorithm

```

Welche der folgenden Aufrufpaare geben identische Werte zurück?

- A. $\text{was}(32345, 3)$ und $\text{was}(321458, 7)$
- B. $\text{was}(321458, 4)$ und $\text{was}(2314587, 4)$
- C. $\text{was}(2314, 3)$ und $\text{was}(23145, 4)$
- D. $\text{was}(23145, 3)$ und $\text{was}(231458, 4)$

27. Gegeben seien die Unteralgorithmen:

- $\text{potenz}(b, p)$ – Bestimmt b^p (b hoch p), b, p – natürliche Zahlen ($1 \leq b \leq 20, 1 \leq p \leq 20$);
- $\text{anzahlZiffern}(nr)$ – gibt die Anzahl der Ziffern einer natürlichen, von Null verschiedenen Zahl nr ($0 < nr \leq 1000000$) zurück, oder den Wert 0 für $nr = 0$;
- $\text{produkt}(st, dr)$ – Der untenstehende Algorithmus, wobei st, dr – natürliche Zahlen ($100 < st < 1000000, 0 \leq dr < 1000000, st$ – eine Zahl, die, wenn sie zur Basis 10 dargestellt wird, mindestens zwei von Null verschiedene Ziffern hat).

```

Subalgorithm produkt(st, dr):
  If st > 0 then
    drCrt ← _____
    stCrt ← st DIV 10
    If st * dr < stCrt * drCrt then
      return produkt(stCrt, drCrt)
    else
      return st * dr
    EndIf
  else
    return st * dr
  EndIf
EndSubalgorithm

```

Welche der folgenden Möglichkeiten füllt die unterstrichene Stelle korrekt aus, so dass der Unteralgorithmus $\text{produkt}(st, dr)$ durch Ausführung der Befehlssequenz

```

write produkt(1092, 0)
write produkt(75981, 0)

```

die Werte 920 și 73575 anzeigt?

- A. $(st \bmod 10) * \text{potenz}(10, \text{anzahlZiffern}(dr)) + dr$
- B. $(st \bmod 10) * \text{potenz}(10, dr) + dr$
- C. $(st \bmod 10) * \text{potenz}(10, \text{anzahlZiffern}(dr))$
- D. $(st \bmod 10) * \text{anzahlZiffern}(dr)$

28. Gegeben sei der Unteralgorithmus $\text{was}(a, n, i, f)$, der eine Folge a bestehend aus n ganze Zahlen $(a[1], a[2], \dots, a[n])$ und die ganze Zahlen i, f und n ($2 \leq n \leq 10000$) nimmt.

```

Subalgorithm was(a, n, i, f):
  If (i = n) AND (f = 2) then
    return true
  else
    If (i = n) then
      return false
    else
      If (f ≤ 1) AND (a[i] < a[i + 1]) then
        return was(a, n, i + 1, 1)
      EndIf
      If (1 ≤ f) AND (a[i] > a[i + 1]) then
        return was(a, n, i + 1, 2)
      EndIf
      return false
    EndIf
  EndIf
EndSubalgorithm

```

Geben Sie an welche der folgenden Aussagen wahr sind, unter der Berücksichtigung des initialen Aufrufs $\text{was}(a, n, 1, 0)$.

- A. Der Unteralgorithmus liefert *true* genau dann, wenn das Maximum der Zeichenkette a an der Position i liegt, $1 < i < n$.
- B. Der Unteralgorithmus liefert *true* genau dann, wenn $\exists k$, ($1 < k < n$), so dass $a[1] < a[2] < \dots < a[k] > a[k + 1] > \dots > a[n]$.
- C. Der Unteralgorithmus gibt *false* zurück genau dann, wenn die Zeichenfolge a streng steigend ist.
- D. Der Unteralgorithmus liefert *true* genau dann, wenn $\exists k$, ($1 < k < n$), so dass $a[k] > a[k + 1] > \dots > a[n]$.

29. 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 ≤ n4 do
      i ← 4 * i
    EndWhile
    j ← j DIV 2
  EndWhile
  return j
EndSubalgorithm

```

In welche der folgenden Komplexitätsklassen fällt die Zeitkomplexität des Algorithmus?

- A. $O(\log_2 n^2)$
- B. $O(\log_2^2 n^2)$
- C. $O(\log_4^2 n)$
- D. $O(\log_2 \log_4 n)$

30. Gegeben sei eine Zeichenkette s mit n Zeichen aus dem englischen Alphabet an, ($s[1], s[2], \dots, s[n]$). Wir wollen sein längstes Suffix finden, das palindromisch ist. Ein Zeichenketten-Suffix ist eine Teilsequenz der Zeichenkette, die das letzte Zeichen enthält. Zum Beispiel ist für die Zeichenkette *abab* das längste palindromische Suffix *bab*.

Angenommen, wir haben den folgenden Unteralgorithmus definiert:

- `ascii(c)` - gibt den ASCII-Code des Zeichens `c` zurück.

Wir nehmen an, dass die arithmetische Operationen keinen Überlauf auf der Integer-Menge erzeugen. Welche der folgenden Implementierungen gibt beim Aufruf von `suffix(s, n)` die Länge dieses Suffixes zurück?

A.
Subalgorithm `suffix(s, n)`:
`hf` \leftarrow 0
`hb` \leftarrow 0
`raspuns` \leftarrow 1
For `i` \leftarrow `n`, 1, -1 **execute**
 `hf` \leftarrow `ascii(s[i])` + 2021 * `hf`
 `hb` \leftarrow `hb` + `ascii(s[i])` * 2021^{n-i}

 If `hf` = `hb` **then**
 `raspuns` \leftarrow `n` - `i` + 1
 EndIf
EndFor
return `raspuns`
EndSubalgorithm

C.
Subalgorithm `suffix(s, n)`:
`hf` \leftarrow 0
`hb` \leftarrow 0
`raspuns` \leftarrow 1
For `i` \leftarrow `n`, 1, -1 **execute**
 `hf` \leftarrow `ascii(s[i])` + 2021 * `hb`
 `hb` \leftarrow `hf` + `ascii(s[i])` * 2021^{n-i}

 If `hf` = `hb` **then**
 `raspuns` \leftarrow `n` - `i` + 1
 EndIf
EndFor
return `raspuns`
EndSubalgorithm

B.
Subalgorithm `suffix(s, n)`:
`hf` \leftarrow 0
`hb` \leftarrow 0
`raspuns` \leftarrow 1
For `i` \leftarrow `n`, 1, -1 **execute**
 `hf` \leftarrow `ascii(s[i])` + 3 * `hf`
 `hb` \leftarrow `hb` + `ascii(s[i])` * 3^{n-i}

 If `hf` = `hb` **then**
 `raspuns` \leftarrow `n` - `i` + 1
 EndIf
EndFor
return `raspuns`
EndSubalgorithm

D. Keine der anderen Antworten ist richtig.

BABEŞ-BOLYAI UNIVERSITÄT
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Aufnahmeprüfung – 19. September 2021
Schriftliche Prüfung in Informatik
PUNKTEANZAHL & LÖSUNGEN

ANFANGSPUNKTEANZAHL: 10 punkte

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