BABEŞ-BOLYAI TUDOMÁNYEGYETEM
MATEMATIKA-INFORMATIKA KAR

MATEK-INFO verseny – 2023 március 26

Informatika írásbeli

FONTOS MEGJEGYZÉS:
Más pontosítások hiányában:

- feltételezhetjük, hogy az aritmetikai műveleteket végtelen adattípusokon végezzük, vagyis nincs túlcsordulás és alulcsordulás.
- Minden sorozatot/vektort 1-től sorszámozunk (indexelünk).
- Az aktuális paraméterek értékeire vonatkozó minden megszorítás a kezdeti hívás pillanatára vonatkozik. .

1. Legyen az `f(a, b)` algoritmus, ahol $\boldsymbol{a}$ és $\boldsymbol{b}$ nullától különböző természetes számok ($1 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10^9$).

```
1: Algorithm f(a, b):
2:     If a = b then
3:         return a
4:     EndIf
5:     If a > b then
6:         return f(a - b, b)
7:     EndIf
8:     return f(a, b - a)
9: EndAlgorithm
```

Állapítsátok meg, hogy a következő állítások közül melyek igazak:

A. Az `f(2000, 21)` hívás eredményeként az algoritmus 1-et térít vissza.
B. Az `f(2000, 21)` hívás eredményeként az algoritmus nem fejezi be a végrehajtást a 2. sorban található feltétel következtében.
C. Ahhoz, hogy az algoritmus $\boldsymbol{a}$ és $\boldsymbol{b}$ legnagyobb közös osztóját térítse vissza, a 8. sort a következőképpen kellene módosítanunk: `return f(b - a, b)`.
D. Ahhoz, hogy az algoritmus az `f(2000, 21)` hívás eredményeként 1-et térítsen vissza, a 8. sort a következőképpen kellene módosítanunk: `return f(b - a, b - a)`.

2. Legyen a következő algoritmusrészlet, ahol $\boldsymbol{a}$ egy $\boldsymbol{n}$ elemű, természetes számokból álló vektor ($\boldsymbol{a}[1], \boldsymbol{a}[2], ..., \boldsymbol{a}[\boldsymbol{n}]$, $1 \leq \boldsymbol{a}[\boldsymbol{i}] \leq 10^4$, ha $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$), és $\boldsymbol{n}$ egy nemnulla természetes szám ($1 \leq \boldsymbol{n} \leq 10^4$):

```
For i ← 1, n - 1 execute
    poz ← i
    For j ← i + 1, n execute
        If a[j] < a[poz] then
            poz ← j
        EndIf
    EndFor
    If poz ≠ i then
        temp ← a[i]
        a[i] ← a[poz]
        a[poz] ← temp
    EndIf
EndFor
```

A következő állítások közül melyek igazak abban a pillanatban, amikor $\boldsymbol{i}$ értéke 2-vé válik?

A. $\boldsymbol{a}[1] \leq \boldsymbol{a}[\boldsymbol{k}]$ bármely $\boldsymbol{k} \in \{1, 2, ..., \boldsymbol{n}\}$ esetén
B. $\boldsymbol{a}[\boldsymbol{n}] \leq \boldsymbol{a}[\boldsymbol{k}]$ bármely $\boldsymbol{k} \in \{1, 2, ..., \boldsymbol{n}\}$ esetén
C. $\boldsymbol{a}[1] \geq \boldsymbol{a}[\boldsymbol{k}]$ bármely $\boldsymbol{k} \in \{1, 2, ..., \boldsymbol{n}\}$ esetén
D. $\boldsymbol{a}[\boldsymbol{k}] \leq \boldsymbol{a}[\boldsymbol{k} + 1]$ bármely $\boldsymbol{k} \in \{1, 2, ..., \boldsymbol{n} - 1\}$ esetén

3. Legyen az `alg(n)` algoritmus, ahol $\boldsymbol{n}$ egy természetes szám ($0 \leq \boldsymbol{n} \leq 10^9$).

```
Algorithm alg(n):
    If n MOD 2 = 0 then
        Return n + alg(n - 1)
    Else
        Return n
    EndIf
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. Ha $\boldsymbol{n} = 4$, az algoritmus által visszatérített érték 7.
B. Az algoritmus az $\boldsymbol{n}$-nél kisebb természetes számok összegét téríti vissza.
C. Az algoritmus az $\boldsymbol{n}$-nél kisebb vagy egyenlő természetes számok összegét téríti vissza.
D. Ha $\boldsymbol{n} = 7$, az algoritmus által visszatérített érték 7.

4. Legyen az `f(nr)` algoritmus, ahol ***nr*** egy egész szám ($-10^4 \le$ ***nr*** $\le 10^4$).

```
Algorithm f(nr):
    If nr < 0 then
        Return f(-nr)
    EndIf
    If (nr = 0) OR (nr = 7) then
        Return 1
    EndIf
    If nr < 10 then
        Return 0
    EndIf
    Return f((nr DIV 10) - 2 * (nr MOD 10))
EndAlgorithm
```

Az ***nr*** szám mely értékeire térít vissza az algoritmus 1-et?

A. 308
B. -7
C. 7098
D. 57

5. Legyen az `afis(n)` algoritmus, ahol ***n*** egy természetes szám ($1 \le$ ***n*** $\le 10^4$):

```
Algorithm afis(n):
    If n > 9 then
        If n MOD 2 = 0 then
            afis(n DIV 100)
            Write n MOD 10, " "
        Else
            afis(n DIV 10)
        EndIf
    EndIf
EndAlgorithm
```

A következő hívások közül melyek esetében ír ki az algoritmus **`2 4`**-et (ebben a sorrendben)?

A. `afis(1234)`
B. `afis(1224)`
C. `afis(4224)`
D. `afis(4321)`

6. Legyen az `Afișare(a)` algoritmus, ahol ***a*** egy természetes szám ($1 \le$ ***a*** $\le 10^4$).

```
Algorithm Afișare(a):
    If a < 9000 then
        Write a, " "
        Afișare(3 * a)
        Write a, " "
    EndIf
EndAlgorithm
```

Mit ír ki az algoritmus az `Afișare(1000)` hívás következtében?

A. `1000 3000 9000 9000 3000 1000`
B. `1000 3000 9000 3000 1000`
C. `1000 3000 3000 1000`
D. `1000 3000 9000`

7. Legyen az `f(n, x)` algoritmus, ahol ***n*** egy természetes szám ($3 \le$ ***n*** $\le 10^4$) és ***x*** egy ***n*** elemű, természetes számokat tartalmazó vektor (***x***[1], ***x***[2], ..., ***x***[***n***], $1 \le$ ***x***[***i***] $\le 10^4$, ***i*** = 1, 2, ..., ***n***):

```
Algorithm f(n, x):
    For i ← 1, n - 2 execute
        If x[i] + x[i + 1] ≠ x[i + 2] then
            Return False
        EndIf
    EndFor
    Return True
EndAlgorithm
```

A következő hívások közül, melyeknek esetében térít vissza az algoritmus *True*-t?

A. `f(3, [10, 15, 25])`
B. `f(4, [0, 0, 0, 0])`
C. `f(5, [100, 535, 635, 1170, 1805])`
D. `f(4, [0, 1, 0, 1])`

8. Mi lesz az eredmény, ha a 10-es számrendszerben megadott $2^{10} - 2^5 – 1$ értéket kettes számrendszerbe alakítjuk át?

A. `1111011111`
B. `1010011001`
C. `1000011001`
D. Egyik sem az A, B, C válaszok közül

9. Legyenek a `one(a, b)` és `two(n, m)` algoritmusok, ahol az $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{n}$ és $\boldsymbol{m}$ bemeneti paraméterek természetes számok ($2 \leq \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{n}, \boldsymbol{m} \leq 10^6$ és $\boldsymbol{n} < \boldsymbol{m}$).

```
Algorithm one(a, b):
    s ← 0
    For i ← 1, a execute
        If a MOD i = 0 then
            s ← s + i
        EndIf
    EndFor
    For i ← 1, b execute
        If b MOD i = 0 then
            s ← s + i
        EndIf
    EndFor
    Return s
EndAlgorithm
```

```
Algorithm two(n, m):
    For i ← n, m execute
        If one(i, i) = 2 * i + 2 then
            Write i, " "
        EndIf
    EndFor
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. A `two(n, m)` algoritmus nem ír ki semmit, függetlenül a bemeneti paraméterek értékeitől.
B. A `two(n, m)` algoritmus az [$\boldsymbol{n}$, $\boldsymbol{m}$] intervallumhoz tartozó prímszámokat írja ki.
C. A `two(n, m)` algoritmus az [$\boldsymbol{n}$, $\boldsymbol{m}$] intervallumhoz tartozó, 2-vel osztható számokat írja ki.
D. Egyik sem helyes a fenti válaszok közül.

10. Legyen a `decide(n, x)` algoritmus, ahol $\boldsymbol{n}$ egy nullától különböző természetes szám ($1 \leq \boldsymbol{n} \leq 10^4$) és $\boldsymbol{x}$ egy $\boldsymbol{n}$ elemű, természetes számokat tároló vektor ($\boldsymbol{x}[1]$, $\boldsymbol{x}[2]$, ..., $\boldsymbol{x}[\boldsymbol{n}]$, $0 \leq \boldsymbol{x}[\boldsymbol{i}] \leq 100$, $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$).

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

Mikor térít vissza *True*-t a fenti algoritmus?

A. Mindig
B. Ha az $\boldsymbol{x}$ vektor elemei: [1, 2, 3]
C. Ha az $\boldsymbol{x}$ vektor elemei [1, 1, 1]
D. Ha az $\boldsymbol{x}$ vektor elemei palindromot alkotnak, vagyis $\boldsymbol{x}[\boldsymbol{i}] = \boldsymbol{x}[\boldsymbol{n} - \boldsymbol{i} + 1]$ bármely $\boldsymbol{i} = 1, 2, ..., \boldsymbol{n}$ esetében.

11. Legyen az `alg(a, b)` algoritmus, ahol $\boldsymbol{a}$ és $\boldsymbol{b}$ természetes számok ($1 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10^3$).

```
Algorithm alg(a, b):
    If b = 0 then
        return 1
    Else
        return a * alg(a, b - 1)
    EndIf
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. Az `alg(2, 3)` hívás eredményeként az algoritmus 7-et térít vissza.
B. Az `alg(2, 3)` hívás eredményeként az algoritmus 4-szer hívódik meg, beleszámítva az eredeti hívást is.
C. Az algoritmus kiszámítja és visszatéríti az $\boldsymbol{a}^{\boldsymbol{b}-1}$ értékét.
D. Az algoritmus kiszámítja és visszatéríti az $\boldsymbol{a}^{\boldsymbol{b}}$ értékét.

12. Legyen a `ceFace(a, b)` algoritmus, ahol ***a*** és ***b*** természetes számok ($1 < \boldsymbol{a}, \boldsymbol{b} \leq 10^5$). A `prim(n)` algoritmus *True*-t téríti vissza, ha az ***n*** > 1 szám prím és *False*-t különben.

```
Algorithm ceFace(a, b):
    If prim(a) = True then
        Write a, " "
    Else
        If prim(b) ≠ True then
            ceFace(a, b + 1)
        Else
            If b > a then
                Write a, " "
            Else
                If a MOD b = 0 then
                    Write b, " "
                    ceFace(a DIV b, b)
                Else
                    ceFace(a, b + 1)
                Endif
            EndIf
        EndIf
    EndIf
EndAlgorithm
```

Mit ír ki az algoritmus, ha `ceFace(100, 2)` alakban hívjuk meg?

A. `2 5 5 5`
B. `5 5 2 2`
C. `2 2 2 5`
D. `2 2 5 5`

13. Legyen az `f(n, p)` algoritmus, ahol ***n*** egy nullától különböző természetes szám ($1 \leq \boldsymbol{n} \leq 10^9$) és ***p*** természetes szám ($0 \leq \boldsymbol{p} \leq 10^9$):

```
Algorithm f(n, p):
    If n ≤ 9 then
        If n MOD 2 = 0 then
            Return 10 * p + n
        Else
            Return p
        EndIf
    Else
        If n MOD 2 = 0 then
            p ← p * 10 + n MOD 10
        EndIf
        Return f(n DIV 10, p)
    EndIf
EndAlgorithm
```

A következő hívások közül mikor térít vissza 22-t a mellékelt algoritmus?

A. `f(23572, 0)`
B. `f(23527, 0)`
C. `f(2, 0)`
D. `f(1242, 0)`

14. Legyen a `cifre(n)` algoritmus, ahol ***n*** egy természetes szám ($0 \leq \boldsymbol{n} \leq 10^3$).

```
Algorithm cifre(n):
    If n ≥ 1 then
        If (n * 5) MOD 10 = 0 then
            Return cifre(n DIV 10)
        Else
            Return n MOD 10
        EndIf
    Else
        Return -1
    EndIf
EndAlgorithm
```

Melyek igazak a következő állítások közül?

A. Az algoritmus mindig egy 10-nél kisebb számot térít vissza.
B. Az algoritmus akkor és csakis akkor térít vissza -1-et, ha ***n*** kezdeti értéke 0.
C. Ha ***n*** ≥ 1, az algoritmus az ***n*** szám legkevésbé szignifikáns páratlan számjegyét téríti vissza, vagy -1-et, ha ilyen számjegy nem létezik.
D. Ha ***n*** ≥ 1, az algoritmus az ***n*** szám legszignifikánsabb páratlan számjegyét téríti vissza, vagy -1-et, ha ilyen számjegy nem létezik.

15. Legyen a `ceFace(a, b)` algoritmus, ahol ***a*** és ***b*** természetes számok ($0 \leq \boldsymbol{a}, \boldsymbol{b} \leq 10^6$).

```
Algorithm ceFace(a, b):
    c ← 0, p ← 1
    While a * b ≠ 0 execute
        If (a MOD 10) = (b MOD 10) then
            c ← (a MOD 10) * p + c
        Else
            If (a MOD 10) < (b MOD 10) then
                c ← ((b MOD 10 - a MOD 10) DIV 2) * p + c
            Else
                c ← ((a MOD 10 - b MOD 10) DIV 2) * p + c
            EndIf
        EndIf
        p ← p * 10
        a ← a DIV 10
        b ← b DIV 10
    EndWhile
    Return c
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. Ha ***a*** = 0 és ***b*** = 0, az algoritmus 1-et térít vissza.
B. Ha ***a*** = 11 és ***b*** = 111, az algoritmus 11-et térít vissza.
C. Ha ***a*** = 5678 és ***b*** = 5162738, az algoritmus 1024-et térít vissza.
D. Ha ***a*** = 112233 és ***b*** = 331122, az algoritmus 110000-et térít vissza.

16. Legyenek a `ceva(n, m)` és az `altceva(n, m)` algoritmusok, ahol ***n*** és ***m*** nullától különböző természetes számok ($1 \leq \boldsymbol{n}, \boldsymbol{m} \leq 10^{12}$ és $\boldsymbol{m} \leq \boldsymbol{n}$).

```
Algorithm ceva(n, m):
    nc ← n
    mc ← m
    While nc > 0 AND mc > 0 execute
        nc ← nc DIV 10
        mc ← mc DIV 10
    EndWhile
    If nc = mc then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

```
Algorithm altceva(n, m):
    c ← 0
    While ceva(n, m) = False execute
        m ← m * 10 + 1
        c ← c + 1
    EndWhile
    Write n, " ", m
    Return c
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. A `ceva(n, m)` algoritmus végrehajtási idejének bonyolultsága $O(\log \boldsymbol{m})$.
B. Az `altceva(n, m)` algoritmus akkor és csakis akkor térít vissza 0-át, ha ***n*** = ***m***.
C. A kijelentésben levő $\boldsymbol{m} \leq \boldsymbol{n}$ előfeltételre azért van szükség, mert ha ***m*** > ***n*** az `altceva(n, m)` algoritmus mindig végtelen ciklust eredményezne.
D. Léteznek ***n*** és ***m*** természetes számok (amelyek betartják az előfeltételeket) amelyeknek esetében az `altceva(n, m)` algoritmus két növekvő sorrendben lévő értéket ír ki.

17. Legyen a `h(s, d, A)` algoritmus. ahol ***s*** és ***d*** nemnulla természetes számok ($1 \leq s, d \leq 10^3$) és ***A*** egy ***n*** darab nullától különböző természetes számot tartalmazó vektor (***A***[1], ***A***[2], ..., ***A***[***n***], $1 \leq A[i] \leq 10^3$, ***i*** = 1, 2, ..., ***n***).

```
Algorithm h(s, d, A):
    If s = d then
        x ← A[s]
        y ← x MOD 10
        x ← x DIV 10
        While x > 0 execute
            z ← x MOD 10
            If z - y ≠ 2 then
                Return 0
            EndIf
            y ← z
            x ← x DIV 10
        EndWhile
        Return 1
    Else
        Return h(s, (s + d) DIV 2, A) + h((s + d) DIV 2 + 1, d, A)
    EndIf
EndAlgorithm
```

Az ***n*** szám és az ***A*** vektor mely értékeire térít vissza a `h(1, n, A)` hívás 5-öt?

A. ***n*** = 7, ***A*** = (20, 53, 10, 42, 31, 131, 42)
B. ***n*** = 10, ***A*** = (420, 75, 68, 86, 97, 975, 53, 64, 24, 57)
C. ***n*** = 10, ***A*** = (402, 75, 6, 86, 7, 9, 35, 46, 24, 57)
D. ***n*** = 10, ***A*** = (642, 97, 6, 64, 7, 9, 75, 4, 53, 31)

18. Legyen az `f(a, x)` algoritmus, ahol ***x*** nullától különböző természetes szám ($1 \leq x \leq 10^4$) és ***a*** egy 10 darab nullától különböző természetes számot tároló vektor (***a***[1], ***a***[2], ..., ***a***[10]) .

```
Algorithm f(a, x):
    i ← 1, j ← 10
    k ← 1
    While a[k] ≠ x AND i < j execute
        k = (i + j) DIV 2
        If a[k] < x then
            i ← k
        Else
            j ← k
        EndIf
    EndWhile
    If a[k] = x then
        Return True
    Else
        Return False
    EndIf
EndAlgorithm
```

A következő bemeneti adatok közül, melyeknek esetében eredményez az algoritmus futása végtelen ciklust?

A. ***a*** = [3, 3, 3, 3, 3, 3, 3, 3, 3, 3] és ***x*** > 3
B. ***a*** = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] és ***x*** < 10
C. ***a*** = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20] és 1 < ***x*** < 20, ***x*** – páratlan szám
D. ***a*** = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20] és 1 < ***x*** < 20, ***x*** – páros szám

19. Legyen az `f(a)` algoritmus, ahol ***a*** természetes szám ($1 \leq a \leq 10^9$).

```
Algorithm f(a):
    x ← a MOD 10
    If x = a then
        If x MOD 2 = 0 then
            Return a
        Else
            Return 0
        EndIf
    EndIf
    If x MOD 2 = 0 then
        Return 10 * f(a DIV 10) + x
    EndIf
    Return f(a DIV 10)
EndAlgorithm
```

A következő állítások közül melyek igazak?

A. Ha ***a*** = 253401976, az `f(a)` algoritmus 8-szor hívódik meg. Az eredeti hívást is számoljuk.
B. Ha ***a*** = 253401976 az `f(a)` algoritmus 9-szer hívódik meg. Az eredeti hívást is számoljuk.
C. Ha ***a*** = 253401976 az algoritmus 2406-ot térít vissza.
D. Ha az ***a*** szám csak páratlan számjegyekből áll, az `f(a)` algoritmus az ***a*** értékét téríti vissza.

20. Legyen az `A(k)` algoritmus, ahol a ***k*** paraméter egy nullától különböző természetes szám ($1 \le \boldsymbol{k} \le 10^9$).

```
Algorithm A(k):
    gr ← (-1 + radical(1 + 8 * k)) / 2
    If gr = [gr] then
        p ← gr
    Else
        p ← [gr] + 1
    EndIf
    Return p - (k – p * (p - 1) DIV 2 - 1)
EndAlgorithm
```

- [***gr***]-rel a ***gr*** szám egész részét jelöltük.
- A `radical(x)` algoritmus az ***x*** szám négyzetgyökét téríti vissza.
- A / műveleti jel két valós szám osztását jelöli, például: 7 / 2 = 3.5

A következő állítások közül melyek igazak?

A. Az alábbi `A1(k)` algoritmus ekvivalens az `A(k)` algoritmussal.

```
Algorithm A1(k):
    c ← 0
    i ← 1
    While c < k execute
        j ← 1
        While j ≤ i execute
            If c < k then
                c ← c + 1
                If c = k then
                    Return j
                Else
                    j ← j + 1
                EndIf
            Else
                Return j
            EndIf
        EndWhile
        i ← i + 1
    EndWhile
EndAlgorithm
```

B. Az alábbi `A2(k)` algoritmus ekvivalens az `A(k)` algoritmussal.

```
Algorithm A2(k):
    c ← 0
    i ← 1
    While c < k execute
        j ← i
        While j ≥ 1 execute
            If c < k then
                c ← c + 1
                If c = k then
                    Return j
                Else
                    j ← j - 1
                EndIf
            Else
                Return j
            EndIf
        EndWhile
        i ← i + 1
    EndWhile
EndAlgorithm
```

C. Az `A(k)` algoritmus az [1, 2, ..., ***i***] (***i*** = 1, 2, ..., ***k***) alakú sorozatok ebben a sorrendben való egymás után ragasztásaként létrejött sorozat ***k***-dik tagjának értékét téríti vissza (vagyis a sorozat: [1, 1, 2, 1, 2, 3, 1, 2, 3, 4, ...]).

D. Az `A(k)` algoritmus az [***i***, …, 2, 1] (***i*** = 1, 2, ..., ***k***) alakú sorozatok ebben a sorrendben való egymás után ragasztásaként létrejött sorozat ***k***-dik tagjának értékét téríti vissza (vagyis a sorozat: [1, 2, 1, 3, 2, 1, 4, 3, 2, 1, ...]).

21. Legyen a `ceFace(a, lung)` algoritmus, ahol ***lung*** természetes szám ($1 \le \boldsymbol{lung} \le 10^5$) és ***a*** egy ***lung*** elemű, egész számokat tároló vektor (***a***[1], ***a***[2], ..., ***a***[***lung***]). Az ***a*** vektorban található legalább egy pozitív szám.

```
Algorithm ceFace(a, lung):
    value1 ← 0
    value2 ← 0
    For i ← 1, lung execute
        value2 ← value2 + a[i]
        If value1 < value2 then
            value1 ← value2
        EndIf
        If value2 < 0 then
            value2 ← 0
        EndIf
    EndFor
    Return value1
EndAlgorithm
```

Tudva, hogy az ***x*** = [***x***[1], ***x***[2], ..., ***x***[***n***]] vektor egy tömbszakaszát az ***x*** vektor azon elemei alkotják, amelyek egymás utáni pozíciókon találhatók (például ***y*** = [***x***[3], ***x***[4], ***x***[5], ***x***[6]] az ***x*** vektor 4 hosszúságú tömbszakasza), állapítsátok meg, hogy a következő állítások közül melyek igazak?

A. Ha az ***a*** vektorban egyetlen pozitív szám található, az algoritmus ennek az értékét téríti vissza.

B. Az algoritmus az ***a*** vektor egy olyan tömbszakaszának hosszát téríti vissza, amelynek összege maximális.

C. Az algoritmus az ***a*** vektor egy olyan tömbszakaszának összegét téríti vissza, amely maximális.

D. Az algoritmus az ***a*** vektor végén, egymás utáni pozíciókon található pozitív számok összegét téríti vissza.

22. Legyen a `ceFace(sir, a, b)` algoritmus, ahol ***sir*** egy ***n*** elemű ($1 \le n \le 100$) nemnulla, különböző természetes számokat tároló, növekvően rendezett vektor (***sir***[1], ***sir***[2], …, ***sir***[***n***]), valamint ***a*** és ***b*** természetes számok ($1 \le a, b \le n$) .

```
Algorithm ceFace(sir, a, b):
    If a > b then
        return a
    EndIf
    c ← a + (b – a) DIV 2
    If sir[c] = c then
        return ceFace(sir, c + 1, b)
    else
        return ceFace(sir, a, c – 1)
    EndIf
EndAlgorithm
```

A következő állítások közül melyek igazak, ha az algoritmus eredeti hívásának alakja: `ceFace(sir, 1, n)`?

A. Ha a ***sir*** vektorban az első ***n*** különböző természetes szám található, az algoritmus ***n*** + 1 értékét téríti vissza.
B. Az algoritmus azt a legnagyobb ***p*** pozíciót téríti vissza, amely ***n*** **DIV** 2-nél kisebb vagy vele egyenlő és amelynek esetében ***sir***[***p***] = ***p*,** vagy 1-et, ha ilyen pozíció nem létezik ($1 \le p \le n$).
C. Az algoritmus azt a legnagyobb ***p*** pozíciót téríti vissza, amely ***n*** **DIV** 2-nél kisebb vagy vele egyenlő és amelynek esetében ***sir***[***p***] ≠ ***p*,** vagy ***n*** + 1-et, ha ilyen pozíció nem létezik ($1 \le p \le n$).
D. Az algoritmus azt a legkisebb, nullától különböző természetes számot téríti vissza, amely nem jelenik meg a ***sir*** vektorban.

23. Legyen a `ceFace(s, x, c, y, n, m, k)` algoritmus, ahol ***s*** egy ***x*** elemű karakterlánc (***s***[1], ***s***[2], ..., ***s***[***x***]), ***c*** egy ***y*** elemű karakterlánc (***c***[1], ***c***[2], ..., ***c***[***y***]), ***x***, ***y***, ***n***, ***m*** és ***k*** nemnulla természetes számok ($1 \le x, y, n, m, k \le 100$).

```
1. Algorithm ceFace(s, x, c, y, n, m, k):
2.     If (n ≥ 0) AND (m ≥ 0) AND (n ≤ x) AND (m ≤ y) then
3.         If k MOD 2 = 0 then
4.             Write s[(n + k) MOD x + 1]
5.             ...
6.             ceFace(s, x, c, y, n - 1, m, k)
7.         EndIf
8.         If k MOD 2 = 1 then
9.             Write c[(m + k) MOD y + 1]
10.            ...
11.            ceFace(s, x, c, y, n, m - 1, k)
12.        EndIf
13.    EndIf
14. EndAlgorithm
```

Szeretnénk, ha a `ceFace("+-", 2, "123", 3, 2, 2, 4)` hívás eredményeként egy érvényes aritmetikai kifejezést kapnánk (vagyis egy olyan aritmetikai kifejezést, amelyben egy-egy műveleti jel és egy-egy operandus váltják egymást; kezdődhet '+'-szal vagy '-'-szal és operandussal kell végződnie. A következő állítások közül melyek **NEM** igazak?

A. Az 5. és 10. sor kiegészíthető a `k ← k + 7` utasítással.
B. Az 5. sor kiegészíthető a `k ← k + 2` utasítással, és a 10. sor a `k ← k + 5` utasítással.
C. Az 5. és 10. sor kiegészíthető a `k ← k + 2` utasítással.
D. Az 5. sor kiegészíthető a `k ← k + 7` utasítással és a 10. sor a `k ← k – 1` utasítással.

24. Adott az ***n*** természetes szám ($1 \le n \le 50$) és az ***n*** elemű, egész számokat tároló ***x*** vektor (***x***[1], ***x***[2], ..., ***x***[***n***]). Melyek igazak a következő állítások közül, függetlenül ***n*** értékétől, valamint a vektor elemeinek értékétől?

A. Létezik olyan ***k*** természetes szám ($1 \le k \le n$), amelyre az ***x***[1] + ***x***[2] + ... + ***x***[***k***] összeg osztható ***n***-nel.
B. Létezik (***i***, ***j***), $0 \le i < j \le n$ úgy, hogy az ***x***[***i*** + 1] + ***x***[***i*** + 2] + ... + ***x***[***j***] összeg osztható ***n***-nel.
C. Az A és B állítások közül egyik sem igaz.
D. Tudva, hogy az ***x*** = [***x***[1], ***x***[2], ..., ***x***[***n***]] vektor egy tömbszakaszát az ***x*** vektor azon elemei alkotják, amelyek egymás utáni pozíciókon találhatók (például ***y*** = [***x***[3], ***x***[4], ***x***[5], ***x***[6]] az ***x*** vektor 4 hosszúságú tömbszakasza), létezik olyan ***k*** természetes szám ($1 \le k \le n$), amelynek esetében az ***x*** vektorban létezik egy ***k*** elemű ($1 \le k \le n$) tömbszakasz, amelynek összege osztható ***n***-nel.

BABEŞ-BOLYAI TUDOMÁNYEGYETEM

MATEMATIKA ÉS INFORMATIKA KAR

MATEK-INFO verseny – 2023 március 26

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

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