

BABEŞ-BOLYAI TUDOMÁNYEGYETEM
MATEMATIKA-INFORMATIKA KAR

Mintatételek a 2021-es BBTE Matek-Infó versenyre és alapképzési felvételire
Informatika írásbeli

1. A generál(n) algoritmus egy n természetes számot dolgoz fel ($0 < n < 100$).

```
Algoritmus generál( $n$ ):  
    szám  $\leftarrow 0$   
    Minden  $i \leftarrow 1, 1801$  végezd el  
        használt $_i \leftarrow hamis$   
    vége(minden)  
    Amíg nem használt $_n$  végezd el  
        összeg  $\leftarrow 0$ ; használt $_n \leftarrow igaz$   
        Amíg  $n \neq 0$  végezd el  
            számjegy  $\leftarrow n \text{ MOD } 10$   
            összeg  $\leftarrow összeg + számjegy * számjegy *$   
                számjegy  
             $n \leftarrow n \text{ DIV } 10$   
        vége(amíg)  
         $n \leftarrow összeg$ ; szám  $\leftarrow szám + 1$   
    vége(amíg)  
    térítsd szám  
Vége(algoritmus)
```

Állapítsátok meg a fenti algoritmus hatását.

- A. ismételten kiszámítja az n szám számjegyei köbének összegét, amíg az összeg egyenlővé nem válik az n számmal és visszatéríti a végrehajtott ismétlések számát
- B. kiszámítja az n szám számjegyei köbének összegét és visszatéríti ezt az összeget
- C. kiszámítja az n szám számjegyei köbének összegét, felülírja n értékét ezzel az összeggel, és visszatéríti ezt az összeget
- D. meghatározza, hogy hányszor lesz felülírva az n szám a számjegyei köbének összegével, ameddig egy már kiszámolt értéket vagy magát a számot kapja és visszatéríti ezt a számot

2. Adott a k elemű s sorozat, amelynek elemei logikai (boolean) típusúak és a kiértékelés(s, k, i) algoritmus, ahol k és i természetes számok ($0 \leq i \leq k \leq 100$).

```
Algoritmus kiértékelés( $s, k, i$ )  
    Ha  $i \leq k$  akkor  
        Ha  $s_i$  akkor  
            térítsd  $s_i$   
        különben  
            térítsd ( $s_i$  vagy kiértékelés( $s, k, i + 1$ ))  
        vége(ha)  
    különben  
        térítsd  $hamis$   
    vége(ha)  
Vége(algoritmus)
```

Határozzátok meg, hányszor hívja meg önmagát a kiértékelés(*s*, *k*, *i*) algoritmus a következő programrészlet végrehajtásának következtében:

```
s ← (hamis, hamis, hamis, hamis, hamis, hamis, igaz, hamis, hamis, hamis)
k ← 10
i ← 3
kiértékelés(s, k, i)
```

- A. 3-szor
- B. ugyanannyiszor, mint a következő programrészlet esetében:

```
s ← (hamis, hamis, hamis, hamis, hamis, hamis, hamis, igaz)
k ← 8
i ← 4
kiértékelés(s, k, i)
```

- C. 6-szor
- D. végtelenszer

3. Adott a kifejezés(*n*) algoritmus, ahol *n* egy természetes szám ($1 \leq n \leq 10000$).

```
Algoritmus kifejezés(n):
  Ha n > 0 akkor
    Ha n MOD 2 = 0 akkor
      térítsd -n * (n + 1) + kifejezés(n - 1)
    különben
      térítsd n * (n + 1) + kifejezés(n - 1)
    vége(ha)
  különben
    térítsd 0
  vége(ha)
Vége(algoritmus)
```

Állapítsátok meg az $E(n)$ kifejezésnek azt a matematikai alakját, amelyet a fenti algoritmus számít ki:

- A. $E(n) = 1 * 2 - 2 * 3 + 3 * 4 + \dots + (-1)^{n+1} * n * (n + 1)$
- B. $E(n) = 1 * 2 - 2 * 3 + 3 * 4 + \dots + (-1)^n * n * (n + 1)$
- C. $E(n) = 1 * 2 + 2 * 3 + 3 * 4 + \dots + (-1)^{n+1} * n * (n + 1)$
- D. $E(n) = 1 * 2 - 2 * 3 - 3 * 4 - \dots - (-1)^n * n * (n + 1)$

4. A következő intervallumok közül melyikhez tartozhat egy *x* biten ábrázolt egész adattípussal rendelkező érték (*x* – szigorúan pozitív természetes szám)?

- A. $[0, 2^x]$
- B. $[0, 2^{x-1}-1]$

- C. $[-2^{x-1}, 2^{x-1}-1]$
- D. $[-2^x, 2^x-1]$

5. Legyen az $f(a, b)$ algoritmus:

```
Algoritmus  $f(a, b)$ :  
  Ha  $a > 1$  akkor  
    térítsd  $b * f(a - 1, b)$   
  különben  
    térítsd  $b * f(a + 1, b)$   
  vége(ha)  
Vége(algoritmus)
```

Határozzátok meg, hányszor hívja meg önmagát az $f(a, b)$ algoritmus a következő programrészletben található hívás következtében:

```
 $a \leftarrow 4$   
 $b \leftarrow 3$   
 $c \leftarrow f(a, b)$ 
```

- A. 4-szer
- B. 3-szor
- C. végtelenszer
- D. egyszer sem

6. Legyen a következő logikai kifejezés: $(\text{NOT } Y \text{ OR } Z) \text{ OR } (X \text{ AND } Y)$. Válasszátok ki X, Y, Z értékeit úgy, hogy a kifejezés kiértékelésének eredménye legyen igaz:

- A. $X \leftarrow \text{hamis}; Y \leftarrow \text{hamis}; Z \leftarrow \text{hamis};$
- B. $X \leftarrow \text{hamis}; Y \leftarrow \text{igaz}; Z \leftarrow \text{hamis};$
- C. $X \leftarrow \text{igaz}; Y \leftarrow \text{hamis}; Z \leftarrow \text{igaz};$
- D. $X \leftarrow \text{hamis}; Y \leftarrow \text{igaz}; Z \leftarrow \text{igaz};$

7. Állapítsátok meg, hogy a következő kifejezések közül melyiknek lesz az értéke akkor és csakis akkor igaz, ha az n természetes szám osztható 3-mal és az utolsó számjegye 4 vagy 6:

- A. $n \text{ DIV } 3 = 0$ és $(n \text{ MOD } 10 = 4 \text{ vagy } n \text{ MOD } 10 = 6)$
- B. $n \text{ MOD } 3 = 0$ és $(n \text{ MOD } 10 = 4 \text{ vagy } n \text{ MOD } 10 = 6)$
- C. $(n \text{ MOD } 3 = 0 \text{ és } n \text{ MOD } 10 = 4)$ vagy $(n \text{ MOD } 3 = 0 \text{ és } n \text{ MOD } 10 = 6)$
- D. $(n \text{ MOD } 3 = 0 \text{ és } n \text{ MOD } 10 = 4)$ vagy $n \text{ MOD } 10 = 6$

8. Adott a következő alprogram:

```
Algoritmus f(a):  
    Ha  $a \neq 0$  akkor  
        térítsd  $a + f(a - 1)$   
    különben  
        térítsd 0  
    vége(ha)  
Vége(algoritmus)
```

Az alábbi kijelentések közül melyik hamis?

- A. ha a negatív, az alprogram 0-t térít vissza
- B. az f által kiszámított érték $a * (a + 1) / 4$
- C. az alprogram kiszámolja az a -nál kisebb vagy egyenlő természetes számok összegét
- D. az $f(-5)$ hívás végtelen ciklust eredményez

9. Legyen a következő algoritmus:

```
Algoritmus SA9(a):  
    Ha  $a < 50$  akkor  
        Ha  $a \bmod 3 = 0$  akkor  
            térítsd  $SA9(2 * a - 3)$   
        különben  
            térítsd  $SA9(2 * a - 1)$   
        vége(ha)  
    különben  
        térítsd  $a$   
    vége(ha)  
Vége(algoritmus)
```

Az a bemeneti paraméter mely értékeire térít a fenti algoritmus 61-et?

- A. 16
- B. 61
- C. 4
- D. 31

10. Adott a $\text{feldolgoz}(v, k)$ algoritmus, ahol v egy k elemű, természetes számokat tároló sorozat ($1 \leq k \leq 1000$).

```
Algoritmus feldolgoz(v, k)  
     $i \leftarrow 1; n \leftarrow 0$   
    Amíg  $i \leq k$  és  $v_i \neq 0$  végezd el  
         $y \leftarrow v_i; c \leftarrow 0$ 
```

```

Amíg y > 0 végezd el
    Ha y MOD 10 > c akkor
        c ← y MOD 10
    vége(ha)
    y ← y DIV 10
vége(amíg)
n ← n * 10 + c; i ← i + 1
vége(amíg)
térítsd n
Vége(algoritmus)

```

Állapítsátok meg, v és k mely értékeire térít vissza az algoritmus 928-at.

- A. $v = (194, 121, 782, 0)$ és $k = 4$
- B. $v = (928)$ és $k = 1$
- C. $v = (9, 2, 8, 0)$ és $k = 4$
- D. $v = (8, 2, 9)$ és $k = 3$

11. Legyen a következő logikai kifejezés: $(X \text{ OR } Z) \text{ AND } (\text{NOT } X \text{ OR } Y)$. Válasszátok ki X, Y, Z értékeit úgy, hogy a kifejezés értéke legyen TRUE:

- A. $X \leftarrow \text{FALSE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{TRUE};$
- B. $X \leftarrow \text{TRUE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{FALSE};$
- C. $X \leftarrow \text{FALSE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{FALSE};$
- D. $X \leftarrow \text{TRUE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{TRUE};$

12. Legyen a következő program:

C	C++	Pascal
<pre> #include<stdio.h> intfeldVektor(intv[], int *n) { int s = 0; i = 2; while (i <= *n) { s = s + v[i] - v[i - 1]; if (v[i] == v[i - 1]) *n = *n - 1; i++; } return s; } intmain(){ intv[8]; v[1] = 1; v[2] = 4; v[3] = 2; v[4] = 3; v[5] = 3; v[6] = 10; v[7] = 12; int n = 7; interedmeny= feldVektor(v, &n); printf("%d;%d", n, eredmeny); return 0; } </pre>	<pre> #include<iostream> usingnamespacestd; intfeldVektor(intv[], int&n) { int s = 0; i = 2; while (i <= n) { s = s + v[i] - v[i - 1]; if (v[i] == v[i - 1]) n--; i++; } return s; } intmain(){ intv[8]; v[1] = 1; v[2] = 4; v[3] = 2; v[4] = 3; v[5] = 3; v[6] = 10; v[7] = 12; int n = 7; interedmeny = feldVektor(v, n); cout<< n<<"<<eredmeny; return 0; } </pre>	<pre> typevector=array[1..10] of integer; functionfeldVektor(v: vector; var n: integer): integer; var s, i: integer; begin s := 0; i := 2; while (i <= n) dobegin s := s + v[i] - v[i - 1]; if (v[i] = v[i - 1]) then n := n - 1; i := i + 1; end; feldVektor := s; end; var n, eredmeny:integer; v:vector; begin n := 7; v[1] := 1; v[2] := 4; v[3] := 2; v[4]:=3; v[5] := 3; v[6] :=10; v[7]:=12; eredmeny := feldVektor(v,n); write(n, ';', eredmeny); end. </pre>

Állapítsátok meg, mit ír ki a program a végrehajtás eredményeként.

- A. 7;11
- B. 6;9
- C. 7;9
- D. 7;12

13. Adott az alábbi pszeudokódban írt algoritmus:

```
beolvas a
Minden i=1, a-1 végezd el:
    Minden j=i+2, a végezd el:
        Ha i+j>a-1 akkor
            kiír a, ' ', i, ' ', j
            új sorba lépés
        vége(ha)
    vége(minden)
vége(minden)
```

Hány megoldáspárt ír ki az algoritmus ha $a=8$?

- A. 13
- B. 15
- C. 20
- D. egyik válasz sem helyes

14. Melyik alábbi algoritmus téríti az a természetes számnak azt a legnagyobb többszörösét, amely kisebb vagy egyenlő a b természetes számmal ($0 < a < 10\,000$, $0 < b < 10\,000$, $a < b$)?

A.

```
Algoritmus f(a, b):
    c ← b
    Amíg c MOD a ≠ 0 végezd el
        c ← c - 1
    vége(amíg)
    térítsd c
Vége(algoritmus)
```

B.

```
Algoritmus f(a, b):
    Ha a < b akkor
        térítsd f(2 * a, b)
    különben
        Ha a = b akkor
            térítsd a
        különben
            térítsd b
    vége(ha)
```

```

vége(ha)
Vége(algoritmus)

```

C.

```

Algoritmus f(a, b):
  térítsd (b DIV a) * a
Vége(algoritmus)

```

D.

```

Algoritmus f(a, b):
  Ha b MOD a = 0 akkor
    térítsd b
  vége(ha)
  térítsd f(a, b-1)
Vége(algoritmus)

```

15. Adott az összes $l \in \{1, 2, 3\}$ hosszúságú sorozat, mely az $\{a, b, c, d, e\}$ halmazból való betűket tartalmaz. Ezek közül a sorozatok közül, hány olyan van, melynek elemei szigorúan csökkenő sorrendbe rendezettek és páratlan számú magánhangzót tartalmaznak? (a és e a magánhangzók)

- A. 14
- B. 7
- C. 81
- D. 78

16. Adottnak tekintjük az $\text{eleme}(x, a, n)$ algoritmust, amely eldönti, hogy az x természetes szám eleme-e az n elemű a halmaznak; a egy n elemű sorozat, amely egy természetes számokat tartalmazó halmazt ábrázol ($1 \leq n \leq 200, 1 \leq x \leq 1000$).

Legyenek az alább megadott $\text{egyesítés}(a, n, b, m, c, p)$ és $\text{számol}(a, n, b, m, c, p)$ algoritmusok, ahol a, b és c sorozatok, amelyek természetes számokat tároló és rendre n, m és p elemű halmazokat ábrázolnak ($1 \leq n \leq 200, 1 \leq m \leq 200, 1 \leq p \leq 400$). A bemeneti paraméterek a, n, b, m és p , kimeneti paraméterek pedig c és p .

<pre> 1. Algoritmus egyesítés(a, n, b, m, c, p): 2. Ha n = 0 akkor 3. Minden i ← 1, m végezd el 4. p ← p + 1 5. c_p ← b_i 6. vége(minden) 7. különben 8. Ha nem eleme(a_n, b, m) akkor 9. p ← p + 1 10. c_p ← a_n 11. vége(ha) 12. egyesítés(a, n - 1, b, m, c, p) 13. vége(ha) 14. Vége(algoritmus) </pre>	<pre> 1. Subalgoritmus számol(a, n, b, m, c, p): 2. p = 0 3. reuniune(a, n, b, m, c, p) 4. SfSubalgoritmus </pre>
---	---

A következő állítások közül melyek bizonyulnak mindig igaznak?

- A. ha az **a** halmaz egyetlen elemet tartalmaz, a számol(a, n, b, m, c, p) algoritmus meghívása végtelen ciklust okoz
- B. ha az **a** halmaznak négy eleme van, a számol(a, n, b, m, c, p) algoritmus meghívása maga után vonja az egyesítés algoritmus 12. sorában található utasítás végrehajtását négyszer
- C. ha az **a** halmaznak öt eleme van, a számol(a, n, b, m, c, p) algoritmus meghívása maga után vonja az egyesítés algoritmus második sorában található utasítás végrehajtását ötször
- D. ha az **a** és **b** halmazok elemei azonosak, a számol(a, n, b, m, c, p) algoritmus végrehajtása után a **c** halmaznak ugyanannyi eleme lesz, mint az **a** halmaznak

17. Adott a számol(n) algoritmus, ahol **n** egy természetes szám ($1 \leq n \leq 10000$).

```

Algoritmus számol(n):
  x ← 0, z ← 1
  Amíg z ≤ n végezd el
    x ← x + 1
    z ← z + 2 * x
    z ← z + 1
  vége(amíg)
  térítsd x
Vége(algoritmus)

```

Az alábbi válaszok közül melyek **hamisak**?

- A. Ha **n** < 8, akkor a számol(n) 3-at térít vissza
- B. Ha **n** ≥ 85 és **n** < 100, akkor a számol(n) 9-et térít vissza
- C. Az algoritmus kiszámítja és visszatéríti az **n**-nél kisebb, szigorúan pozitív négyzetszámok darabszámát
- D. Az algoritmus kiszámítja és visszatéríti az **n** szám négyzetgyökének egész részét

18. Legyen az **n** × **n** méretű négyzetes **mat** tömb (**n** – páratlan természetes szám, $3 \leq n \leq 100$). A tegyélB(mat, n, i, j) algoritmus 'b' betűket tesz a **mat** tömb bizonyos pozícióira. Az **i** és **j** paraméterek természetes számok ($1 \leq i \leq n, 1 \leq j \leq n$).

```

Algoritmus tegyélB(mat, n, i, j):
  Ha i ≤ n DIV 2 akkor
    Ha j ≤ n - i akkor
      mat[i][j] ← 'b'
      tegyélB(mat, n, i, j + 1)
    különben
      tegyélB(mat, n, i + 1, i + 2)
  vége(ha)
  vége(ha)
Vége(algoritmus)

```

Határozzátok meg, hányszor hívja meg önmagát a tegyélB(mat, n, i, j) algoritmus a következő programrészlet végrehajtásának következtében:

```

n ← 7, i ← 2, j ← 4
tegyélB(mat, n, i, j)

```

- A. 5-ször
- B. ugyanannyiszor, mint a következő programrészlet esetében:
 $n \leftarrow 9, i \leftarrow 3, j \leftarrow 5$
 tegyélB(mat, n, i, j)
- C. 10-szer
- D. végtelenszer

19. Legyen a számol(*a*, *b*) algoritmus, amelynek bemeneti paraméterei az *a* és *b* pozitív természetes számok, ahol $1 \leq a \leq 1000, 1 \leq b \leq 1000$.

```

1.      Algoritmus számol(a, b):
2.          Ha a ≠ 0 akkor
3.              térítsd számol(a DIV 2, b + b) + b * (a MOD 2)
4.          vége(ha)
5.          térítsd 0
6.      Vége(algoritmus)
    
```

Az alábbi válaszok közül melyek hamisak?

- A. ha *a* és *b* egyenlők, az algoritmus *a* értékét téríti
- B. ha *a* = 1000 és *b* = 2, az algoritmus 10-szer hívja meg önmagát
- C. az algoritmus által kiszámított és térített érték egyenlő $(a / 2 + 2 * b)$ -vel
- D. az 5. sorban található utasítás egyszer hajtódik végre

20. Legyen a törzsTényezők(*n*, *d*, *k*, *x*) algoritmus, amely meghatározza az *n* természetes szám *k* darab törzstényezőjét, a törzstényezők keresését a *d* értéktől kezdve. Bemeneti paraméterek az *n*, *d* és *k* számok, kimeneti paraméterek az *x* sorozat, amely a *k* törzstényezőt tartalmazza ($1 \leq n \leq 10000, 2 \leq d \leq 10000, 0 \leq k \leq 10000$).

```

Algoritmus törzsTényezők(n, d, k, x):
    Ha n MOD d = 0 akkor
        k ← k + 1
        x[k] ← d
    vége(ha)
    Amíg n MOD d = 0 végezd el
        n ← n DIV d
    vége(amíg)
    Ha n > 1 akkor
        törzsTényezők(n, d + 1, k, x)
    vége(ha)
Vége(algoritmus)
    
```

Határozzátok meg, hányszor hívja meg önmagát a törzsTényezők(*n*, *d*, *k*, *x*) algoritmus a következő programrészlet végrehajtásának következtében:

```

n ← 120
d ← 2
k ← 0
törzsTényezők(n, d, k, x)
    
```

- A. 3-szor
- B. 5-ször
- C. 6-szor
- D. ugyanannyiszor, mint a következő programrészlet esetében

```

x ← 750
d ← 2
k ← 0
törzsTényezők(n, d, k, x)

```

21. Legyenek az **mésn** természetes számok ($0 \leq m \leq 10, 0 \leq n \leq 10$), valamint az $\text{Ack}(m, n)$ algoritmus, amely kiszámítja az Ackermannfüggvény értékét **mésn** esetében.

```

AlgoritmusAck(m, n)
  Ha m = 0 akkor
    térítsd n + 1
  különben
    Ha m > 0 és n = 0 akkor
      térítsdAck(m - 1, 1)
    különben
      térítsdAck(m - 1, Ack(m, n - 1))
  vége(ha)
vége(ha)
Vége(algoritmus)

```

Állapítsátok meg, hányszor hívja meg önmagát az $\text{Ack}(m, n)$ algoritmus az alábbi utasítások végrehajtásának következtében.

```

m ← 1, n ← 2
Ack(m, n)

```

- A. 7-szer
- B. 5-ször
- C. 10-szer
- D. ugyanannyiszor, mint az alábbi utasítások végrehajtásának következtében:

```

m ← 1, n ← 3
Ack(m, n)

```

22. Definiáljuk a $\overline{c_1 c_2 \dots c_k}$ számjegyekből álló **k** természetes számra a **csenkítésműveletet**:

$$\text{csenkít}(\overline{c_1 c_2 \dots c_k}) = \begin{cases} 0, & \text{ha } k < 2 \\ \overline{c_1 c_2}, & \text{különben} \end{cases}$$

Állapítsátok meg, hogy az alábbi algoritmusok közül melyik határozza meg az **n** elemű sorozat **csenkített elemeinek összegét**. Az elemek természetes számok és kisebbek, mint 1 000 000 (n – természetes szám és $1 \leq n \leq 1\,000$).

Például, ha $n = 4$ és $x = (213, 7, 78347, 22)$, akkor a csenkített elemek összege $21 + 0 + 78 + 22 = 121$.

A.

```

Algoritmus csonkítottSzámokÖsszege(n, x)
  s ← 0
  Amíg n > 0 végezd el
    Ha x[n] > 9 akkor
      Amíg x[n] > 99 végezd el
        x[n] ← x[n] DIV 10
      vége(amíg)
      s ← s + x[n]
    vége(ha)
    n ← n - 1
  vége(amíg)
  térítsd s
Vége(algoritmus)

```

B.

```

Algoritmus csonkítottSzámokÖsszege(n, x)
  s ← n
  Amíg n > 0 végezd el
    Ha x[n] > 9 akkor
      Amíg x[n] > 99 végezd el
        x[n] ← x[n] DIV 10
      vége(amíg)
      s ← s + x[n]
    vége(ha)
    n ← n - 1
  vége(amíg)
  térítsd s
Vége(algoritmus)

```

C.

```

Algoritmus csonkítottSzámokÖsszege(n, x)
  s ← 0
  Amíg n > 0 végezd el
    Ha x[n] > 9 akkor
      Amíg x[n] > 99 végezd el
        x[n] ← x[n] DIV 10
      s ← s + x[n]
    vége(amíg)
    vége(ha)
    n ← n - 1
  vége(amíg)
  térítsd s
Vége(algoritmus)

```

D.

```

Algoritmus csonkítottSzámokÖsszege(n, x)
  s ← 0
  Amíg x[n] > 99 végezd el
    x[n] ← x[n] DIV 10
  vége(amíg)
  s ← s + x[n]
  térítsd s
Vége(algoritmus)

```

23. Legyen az s egytermészetes számokat tároló sorozat, ahol $s_i = \begin{cases} x, & \text{dacă } i = 1 \\ x + 1, & \text{dacă } i = 2 \\ s_{(i-1)}@s_{(i-2)} & \text{dacă } i > 2 \end{cases}, (i = 1, 2, \dots)$. A@művelet a bal és a jobb operandus számjegyeitkonkatenálja(egymás után ragasztja)ebben a sorrendben, azxpedig egytermészetes szám ($1 \leq x \leq 99$). Például, $hax = 3$, azssorozat elemei a következők: 3, 4, 34, 434, 34434,Állapítsátok meg, hány számjegye van az s sorozat azon elemének, amely a k számjegyű elem előtt található ($1 \leq k \leq 30$).

2, ...). A@művelet a bal és a jobb operandus számjegyeitkonkatenálja(egymás után ragasztja)ebben a sorrendben, azxpedig egytermészetes szám ($1 \leq x \leq 99$). Például, $hax = 3$, azssorozat elemei a következők: 3, 4, 34, 434, 34434,Állapítsátok meg, hány számjegye van az s sorozat azon elemének, amely a k számjegyű elem előtt található ($1 \leq k \leq 30$).

- A. $hax = 15$ és $k = 6$, az s sorozatban a k számjegyű elem előtt levő elem számjegyeinek száma 5.
 B. $hax = 2$ és $k = 8$, az s sorozatban a k számjegyű elem előtt levő elem számjegyeinek száma 5.
 C. $hax = 14$ és $k = 26$, az s sorozatban a k számjegyű elem előtt levő elem számjegyeinek száma 16.
 D. $hax = 5$ és $k = 13$, az s sorozatban a k számjegyű elem előtt levő elem számjegyeinek száma 10.

24. Legyen az n elemű x sorozat, amely természetes számokat tárol ($3 \leq n \leq 10000$) és ak természetes szám ($1 \leq k < n$). AkörkörösPerm(n, k, x) algoritmus az x sorozat körkörös permutációját kellene generálja, k pozícióval balra, például, a (4, 5, 2, 1, 3) sorozat az (1, 3, 4, 5, 2) sorozat körkörös permutációja két pozícióval balra. Sajnos, akörkörösPerm(n, k, x) algoritmus nem helyes, mivel néskbizonyos értékeire a generált eredmény hibás.

```

AlgoritmuskörkörösPerm( $n, k, x$ )
   $c \leftarrow k$ 
  Minden  $j = 1, c$  végezd el
     $hova \leftarrow j$ 
     $sza\acute{m} \leftarrow x[hova]$ 
    Minden  $i = 1, n / c - 1$  végezd el
       $honna\acute{n} \leftarrow hova + k$ 
      H $ahonna\acute{n} > n$  akkor
         $honna\acute{n} \leftarrow honna\acute{n} - n$ 
      vége( $ha$ )
       $x[hova] \leftarrow x[honna\acute{n}]$ 
       $hova \leftarrow honna\acute{n}$ 
    vége(minden)
     $x[hova] \leftarrow szá\acute{m}$ 
  vége(minden)
Vége(algoritmus)

```

Válasszátok kin, k és x értékeit, amelyeknek esetében akörkörösPerm(n, k, x) algoritmus az x sorozat körkörös permutációját generálja, k pozícióval balra:

- A. $n = 6, k = 2, x = (1, 2, 3, 4, 5, 6)$
 B. $n = 8, k = 3, x = (1, 2, 3, 4, 5, 6, 7, 8)$
 C. $n = 5, k = 3, x = (1, 2, 3, 4, 5)$
 D. $n = 8, k = 4, x = (1, 2, 3, 4, 5, 6, 7, 8)$

25. Egy nullától különböző x természetes számot *szerencsés*nek nevezzük, ha a négyzete felírható x darab egymás utáni természetes szám összegeként. Például, a 7 szerencsés szám, mivel $7^2 = 4 + 5 + 6 + 7 + 8 + 9 + 10$.

A következő algoritmusok közül, melyik dönti el az x természetes számról ($2 \leq x \leq 1000$), hogy *szerencsés szám*? Minden algoritmus bemeneti paramétere az x szám, kimeneti paraméterei pedig a nullától különböző *start* természetes szám és a *szerencsés* logikai változó. Ha az x szerencsés szám, akkor *szerencsés* = *igaz* és *start* értéke az összeg első tagjának értéke (például, ha $x = 7$, akkor *start* = 4); ha x nem szerencsés szám, akkor *szerencsés* = *hamis* és *start* értéke -1.

A.

```
Algoritmus szerencsésSzám(x, start, szerencsés):  
    xNégyzet  $\leftarrow x * x$   
    szerencsés  $\leftarrow hamis$   
    start  $\leftarrow -1$ , k  $\leftarrow 1$ , s  $\leftarrow 0$   
    Amíg k  $\leq$  xNégyzet - x és nem szerencsés végezd el  
        Minden i  $\leftarrow k$ , k + x - 1 végezd el  
            s  $\leftarrow s + i$   
        vége(minden)  
        Ha s = xNégyzet akkor  
            szerencsés  $\leftarrow igaz$   
            start  $\leftarrow k$   
        vége(ha)  
    vége(amíg)  
Vége(algoritmus)
```

B.

```
Algoritmus szerencsésSzám(x, start, szerencsés):  
    xNégyzet  $\leftarrow x * x$   
    szerencsés  $\leftarrow hamis$   
    start  $\leftarrow -1$ , k  $\leftarrow 1$   
    Amíg k  $\leq$  xNégyzet - x és nem szerencsés végezd el  
        s  $\leftarrow 0$   
        Minden i  $\leftarrow k$ , k + x - 1 végezd el  
            s  $\leftarrow s + i$   
        vége(minden)  
        Ha s = xNégyzet akkor  
            szerencsés  $\leftarrow igaz$   
            start  $\leftarrow k$   
        vége(ha)  
        k  $\leftarrow k + 1$   
    vége(amíg)  
Vége(algoritmus)
```

C.

```
Algoritmus szerencsésSzám(x, start, szerencsés):  
    Ha x MOD 2 = 0 akkor  
        szerencsés  $\leftarrow hamis$   
        start  $\leftarrow -1$   
    különben  
        szerencsés  $\leftarrow igaz$   
        start  $\leftarrow (x + 1) \text{ DIV } 2$   
    vége(ha)  
Vége(algoritmus)
```

D.

Algoritmus szerencsésSzám(x , start, szerencsés):

```
Ha  $x \bmod 2 = 0$  akkor
    szerencsés  $\leftarrow$  hamis
    start  $\leftarrow$  -1
különben
    szerencsés  $\leftarrow$  igaz
    start  $\leftarrow$   $x \operatorname{DIV} 2$ 
vége(ha)
Vége(algoritmus)
```

26. Adott az $\text{alg}(x, b)$ algoritmus (alprogram), amelynek bemeneti paraméterei az x és b természetes számok ($1 \leq x \leq 1000$, $1 < b \leq 10$). Határozzátok meg, hogy mit csinál az algoritmus.

```
Algoritmus alg( $x$ ,  $b$ ):
     $s \leftarrow 0$ 
    Amíg  $x > 0$  végezd el
         $s \leftarrow s + x \bmod b$ 
         $x \leftarrow x \operatorname{DIV} b$ 
    vége(amíg)
    térítsd  $s \bmod (b - 1) = 0$ 
Vége(algoritmus)
```

- A. vizsgálja, hogy az x szám számjegyeinek összege a $b - 1$ számrendszerben osztható-e $(b - 1)$ -gyel
- B. vizsgálja, hogy az x szám osztható-e $(b - 1)$ -gyel
- C. vizsgálja, hogy a b számrendszerben felírt x szám számjegyeinek összege osztható-e $(b - 1)$ -gyel
- D. vizsgálja, hogy az x szám számjegyeinek összege osztható-e $(b - 1)$ -gyel

27. Legyen az (1, 2, 3, 2, 5, 2, 3, 7, 2, 4, 3, 2, 5, 11, ...) sorozat, amelyet a következőképpen hoztunk létre: kiindulva természetes számok sorozatából, azokat a számokat, amelyek nem prímszámok helyettesítettük a saját osztóikkal úgy, hogy minden d osztót csak egyszer használtunk minden szám esetében. Az alábbi algoritmusok közül melyik határozza meg a sorozat n -dik elemét (n természetes szám, $1 \leq n \leq 1000$)?

A.

```
Algoritmus beazonosítás( $n$ ):
     $a \leftarrow 1$ ,  $b \leftarrow 1$ ,  $c \leftarrow 1$ 
    Amíg  $c < n$  végezd el
         $a \leftarrow a + 1$ ,  $b \leftarrow a$ ,  $c \leftarrow c + 1$ ,  $d \leftarrow 2$ 
        Amíg  $c \leq n$  és  $d \leq a \operatorname{DIV} 2$  végezd el
            Ha  $a \bmod d = 0$  akkor
                 $c \leftarrow c + 1$ ,  $b \leftarrow d$ 
            vége(ha)
         $d \leftarrow d + 1$ 
    vége(amíg)
Vége(amíg)
```

```
térítsd b
Vége(algoritmus)
```

B.

```
Algoritmus beazonosítás(n):
  a ← 1, b ← 1, c ← 1
  Amíg c < n végezd el
    c ← c + 1, d ← 2
    Amíg c ≤ n és d ≤ a DIV 2 végezd el
      Ha a MOD d = 0 akkor
        c ← c + 1, b ← d
      vége(ha)
    d ← d + 1
  vége(amíg)
  a ← a + 1, b ← a
vége(amíg)
térítsd b
Vége(algoritmus)
```

C.

```
Algoritmus beazonosítás(n):
  a ← 1, b ← 1, c ← 1
  Amíg c < n végezd el
    a ← a + 1, d ← 2
    Amíg c < n és d ≤ a végezd el
      Ha a MOD d = 0 akkor
        c ← c + 1, b ← d
      vége(ha)
    d ← d + 1
  vége(amíg)
vége(amíg)
térítsd b
Vége(algoritmus)
```

D.

```
Algoritmus beazonosítás(n):
  a ← 1, b ← 1, c ← 1
  Amíg c < n végezd el
    b ← a, a ← a + 1, c ← c + 1, d ← 2
    Amíg c ≤ n és d ≤ a DIV 2 végezd el
      Ha a MOD d = 0 akkor
        c ← c + 1, b ← d
      vége(ha)
    d ← d + 1
  vége(amíg)
vége(amíg)
térítsd b
Vége(algoritmus)
```

28. Az m és n hosszú oldalakkal rendelkező téglalap fel van osztva 1 oldalhosszúságú négyzetecskékre(m , n – természetes számok, $0 < m < 101$, $0 < n < 101$). Adott a téglalap(m , n) algoritmus:

```

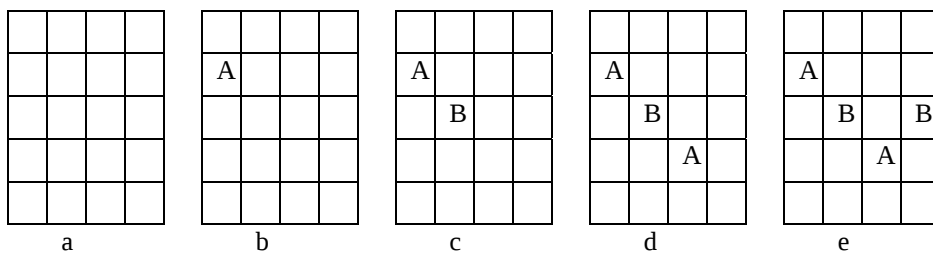
Algoritmustéglalap( $m, n$ )
   $d \leftarrow m$ 
   $c \leftarrow n$ 
  Amíg  $d \neq c$  végezd el
    Ha  $d > c$  akkor
       $d \leftarrow d - c$ 
    különben
       $c \leftarrow c - d$ 
    vége(ha)
  vége(amíg)
  térítsd  $m + n - d$ 
Vége(algoritmus)

```

Mi a hatása ennek az algoritmusnak?

- A. Kiszámítja és téríti azoknak az 1 oldalhosszúságú négyzetecskéknek a számát, amelyeket átvág a téglalap egy átlója.
- B. Meghatározza d -ben a téglalap oldalainak legnagyobb közös osztóját és téríti az oldalak összegének és d -nek a különbségét.
- C. Ha $m = 8$ és $n = 12$, a térített érték 16.
- D. Ha $m = 6$ és $n = 11$, a térített érték 15.

29. Legyen egy téglalap alakú tábla, amely föl van osztva $n \times m$ cellára (n – a sorok száma, m – az oszlopok száma, n, m – természetes számok, $2 \leq n \leq 100, 2 \leq m \leq 100$). Két játékos, A és B, felváltva lépéseket hajtanak végre: minden lépésnél a soron levő játékos megjelöl egyetlen cellát, amely átlósan szomszédos az előző lépésben, a másik játékos által megjelölt cellával és amely még nincs megjelölve. Az a játékos, akinek nincs hova lépnie, veszít. Az A játékos lép először, és megjelöl egy cellát a táblán.



Példa: a) eredeti állapot ($n = 5$ és $m = 4$), b) az első lépés utáni helyzet (lépett A), c) a második lépés után (lépett B), d) a harmadik lépés után (lépett A), e) a negyedik lépés után (lépett B)

Határozzátok meg, milyen feltételek mellett van A-nak biztos nyerési stratégiája, (vagyis nyerni fog, B lépéseitől függetlenül) és mi lehet A első lépése ahhoz, hogy nyerjen.

- A. feltétel: m páratlan szám;
az A játékos először a fenti első sorban (1-es sor) és egy páratlan sorszámu oszlopban levő cellára lép.
- B. feltétel: n páratlan szám;
az A játékos először egy páros sorszámu sorban és a tábla bal első oszlopában (1-es oszlop) levő cellára lép.
- C. feltétel: mindkét szám (n és m) páros szám;
az A játékos először a tábla bal felső sarkában (1-es sor és 1-es oszlop) levő cellára lép.

- D.** feltétel: n és m közül legalább az egyik páratlan szám;
az A játékos először a tábla bal felső sarkában (1-es sor és 1-es oszlop) levő cellára lép.

30. Egy 8 sorból álló mátrix csak 0 és 1 elemeket tartalmaz és a következő három tulajdonsággal rendelkezik:

- az első sor egyetlen 1-es elemet tartalmaz,
- a j . sor kétszer annyi nullától különböző elemet tartalmaz, mint a $j - 1$. sor, bármely $j \in \{2, 3, \dots, 8\}$ esetén,
- az utolsó sor egyetlen 0-val egyenlő elemet tartalmaz.

Összesen hány 0-val egyenlő elemet tartalmaz a mátrix?

- 777
- 769
- 528
- nem létezik ilyen mátrix

Helyes válaszok:

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

11. A, D
12. B
13. B
14. C, D
15. A
16. B, D
17. A, C
18. A, B
19. A, C
20. A, D

21. B
22. A
23. B, C
24. A, D
25. B, C
26. B, C
27. A
28. A, C
29. A, D
30. A