

Felvételi verseny – 2023 szeptember 8
Informatika írásbeli

FONTOS MEGJEGYZÉS:

Más pontosítások hiányában:

- Feltételezhetjük, hogy az aritmetikai műveleteket végtelen adattípusokon végezzük, vagyis nincs *túlcsordulás* és *alulcsordulás*.
- Minden sorozatot 1-től sorszámozunk (indexelünk).
- Az aktuális paraméterek értékeire vonatkozó megszorítások az első hívás pillanatában érvényesek.
- Egy vektor tömbszakaszát a vektor egymás utáni pozícióin található elemek alkotják.

1. Legyen a $\text{ceFace}(a, b)$ algoritmus, ahol a és b természetes számok ($0 \leq a, b \leq 10^4$).

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

Mi a hatása a $\text{ceFace}(a, a)$ hívásnak?

- Az algoritmus azt a legkisebb palindromszámot téríti vissza, amely a -nál nagyobb, vagy a -val egyenlő.
- Az algoritmus azt a legnagyobb palindromszámot téríti vissza, amely a -nál kisebb, vagy a -val egyenlő.
- Az algoritmus azt a legkisebb palindromszámot téríti vissza, amely a -nál nagyobb.
- Az algoritmus azt a legnagyobb páros számot téríti vissza, amely a -nál kisebb, vagy a -val egyenlő.

2. Legyen a $\text{createTablou}(n, m, x)$ algoritmus, ahol n, m természetes számok ($1 \leq n, m \leq 100$), x egy $n * m$ elemű, egész számokat tároló kétdimenziós tömb ($x[1][1], x[1][2], \dots, x[n][m], 0 \leq x[i][j] \leq 10^4$, ha $i = 1, 2, \dots, n, j = 1, 2, \dots, m$).

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

Mit fog kiírni az algoritmus, ha az x tömb elemei 0-val inicializáltak?

- Az algoritmus kiírja az x kétdimenziós tömb elemeit, amelyek között vannak 0 értékűek és az első $(n * m) \text{ DIV } 2$ páratlan négyzetszám.
- Az algoritmus kiírja az x kétdimenziós tömb elemeit, amelyek között vannak 0 értékűek és az első páros négyzetszámok.
- Az algoritmus kiírja az x kétdimenziós tömb elemeit, amelyek között megtalálható az első $(n * m) \text{ DIV } 2$ páros négyzetszám sorozata.
- Az algoritmus kiírja az x kétdimenziós tömb elemeit, amelyben, – ha sort sor után helyeznénk – a páratlan négyzetszámok növekvő sorrendben lennének, esetleg 0 értékű elemekkel előttük és/vagy utánuk.

3. Legyen a $\text{something}(n, x)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), x egy n elemű, természetes számokat tároló vektor ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^6$, ha $i = 1, 2, \dots, n$).

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

Mit térít vissza a $\text{something}(5, [222, 2043, 29, 2, 20035])$ hívás?

- 16
- 10
- 11
- 15

4. Legyen a $\text{ceFace}(n, v, a)$ algoritmus, ahol n és v természetes számok ($1 \leq n, v \leq 10^4$), a egy n elemű, természetes számokat tároló sorozat ($a[1], a[2], \dots, a[n]$).

```

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

```

Mely értéket téríti vissza az algoritmus, ha $n = 4$, $v = 3$ és $a = [5, 4, 2, 10]$?

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

5. Legyen a $\text{calcul}(v, n)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), v egy n elemű, természetes számokat tároló vektor ($v[1], v[2], \dots, v[n]$, $1 \leq v[i] \leq 10^4$, ha $i = 1, 2, \dots, n$):

```

Algorithm calcul(v, n):
  i ← 1
  While i ≤ n DIV 2 execute
    p ← 0
    While v[i] ≠ 0 execute
      p ← p + 1
      v[i] ← v[i] DIV 10
    EndWhile
    q ← 0
    While v[n + 1 - i] ≠ 0 execute
      q ← q + 1
      v[n + 1 - i] ← v[n + 1 - i] DIV 10
    EndWhile
    If p ≠ q then
      Return False
    EndIf
    i ← i + 1
  EndWhile
  Return True
EndAlgorithm

```

A következő esetek közül melyekben térít vissza az algoritmus *True*-t?

- A. Ha a v vektor elemei $[12, 12, 2, 5466, 3, 111, 1, 3, 44]$ és $n = 9$.
 B. Ha a v vektor elemei $[12, 345, 2, 5466, 3, 111, 10]$ és $n = 7$.
 C. Ha a v vektor elemei azonos darabszámú számjegyből állnak.
 D. Ha az a vektor, amit úgy kapunk, hogy egymás után helyezük a v vektor elemeinek számjegydarabszámait, palindrom tulajdonságú; például, a $v = [8, 37, 3]$ vektor alapján kialakul az $[1, 2, 1]$ vektor, amely palindrom.

6. Legyen az $\text{alg}(n)$ algoritmus, ahol n egy természetes szám ($0 \leq n \leq 10^4$).

```

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

```

A következő állítások közül melyek igazak?

- A. Az $\text{alg}(123)$ hívás 6-ot térít vissza.
 B. Az algoritmus kiszámítja az adott számban páros pozíciókon található számjegyek összegét.
 C. Az algoritmus kiszámítja az adott szám páros számjegyeinek összegét.
 D. Az algoritmus kiszámítja az adott szám számjegyeinek összegét.

7. Legyen az $f(x)$ algoritmus, ahol x egy nemnulla természetes szám ($1 \leq x \leq 10^5$).

```

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

```

Állapítsátok meg, mit ír ki az algoritmus, ha $f(10)$ alakban hívjuk meg:

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

8. Legyen az n méretű M négyzetes mátrix, amelynek elemei természetes számok, ahol n egy nemnulla természetes szám ($1 \leq n \leq 10^4$, $M[1][1], \dots, M[1][n], M[2][1], \dots, M[2][n], \dots, M[n][1], \dots, M[n][n]$, $1 \leq M[i][j] \leq 10^4$, ha $i = 1, 2, \dots, n$ és $j = 1, 2, \dots, n$). Legyen a következő algoritmus:

```

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

```

Mit fog kiírni az algoritmus a következő M mátrix esetében?

1	2	3
8	9	4
7	6	5

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

9. Legyen a $ce_face(a, b)$ algoritmus, ahol a és b természetes számok ($1 \leq a, b \leq 10^4$).

```

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

```

Határozzátok meg az igaz kijelentéseket:

- A. A $ce_face(1, 2)$ hívás esetében az algoritmus 1-et térít vissza.
- B. A $ce_face(24, 2)$ hívás esetében az algoritmus 0-át térít vissza.
- C. A $ce_face(2024, 4)$ hívás esetében az algoritmus 4-et térít vissza.
- D. A $ce_face(8, 3)$ hívás esetében az algoritmus 2-t térít vissza.

10. Legyenek a $decide(n)$ és a $compute(m)$ algoritmusok, ahol n és m nemnulla természetes számok ($1 \leq n, m \leq 10^4$):

```

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

```

```

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

```

Az m mely értékeire térít vissza a $compute(m)$ algoritmus -33-at?

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

11. Legyen az algoritmus $f(n, x)$, ahol n és x természetes számok ($1 \leq n \leq 10^5$, $2 \leq x \leq 10$):

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

A következő állítások közül melyek igazak?

- A. Az algoritmus kiírja az n számot az x számrendszerben ábrázolva.
- B. Az algoritmus kiírja az n szám x -szel való egész osztási maradékát.
- C. Az algoritmus kiírja az n szám x számrendszerben való ábrázolásának számjegyeinek darabszámát.
- D. Az algoritmus megvizsgálja, hogy osztható-e az n szám x -szel.

12. Legyen a $ceFace(n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^9$).

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

A következő állítások közül melyek igazak?

- A. Az algoritmus egy olyan számot térít vissza, amelynek egy számjegye van vagy -1-et.
- B. Az algoritmus egy páratlan számot térít vissza.
- C. Az algoritmus az n szám legnagyobb páratlan számjegyét téríti vissza vagy -1-et.
- D. Az algoritmus az n szám legnagyobb páros számjegyét téríti vissza vagy -1-et.

13. Legyen a $decide(n, x)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^4$), x egy n elemű, egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, ha $i = 1, 2, \dots, n$):

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

A következő esetek közül melyekben térít vissza az algoritmus *True*-t?

- A. Ha $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ és $n = 10$
- B. Ha $n > 1$ és az x vektor elemei szigorúan növekvő sorrendben vannak
- C. Ha az x vektornak nincsenek negatív elemei
- D. Ha az x vektornak vannak pozitív elemei, amelyek a negatív elemek előtt találhatók

14. Legyen x és y két pozitív természetes szám, ahol x kettőhatvány és y 3-nak többszöröse. Legyen a következő logikai kifejezés: $((x * y + 3) \text{ DIV } 6 = 10) \text{ OR } ((x * y) \text{ MOD } 6 = 0) \text{ AND } ((x + y) \text{ MOD } 4 = 0)$

Az alábbi állítások közül melyek igazak, ha a számpárok rendelkeznek a kijelentésben levő tulajdonságokkal?

- A. Létezik egy olyan (x, y) számpár, amelyre a kifejezés igaz.
- B. Létezik egy olyan (x, y) számpár, amelyre a kifejezés hamis.
- C. Léteznek az (x_1, y_1) és (x_2, y_2) számpárok, ahol $x_1 \neq x_2$ és $y_1 \neq y_2$ úgy, hogy a kifejezés igaz, mindkét számpár esetében.
- D. A kifejezés hamis bármely (x, y) számpár esetében.

15. Legyenek az n és m természetes számok ($1 \leq n, m \leq 256$), az n elemű a karakterlánc $(a[1], a[2], \dots, a[n])$ és az m elemű b karakterlánc $(b[1], b[2], \dots, b[m])$. A következő algoritmusok közül melyik térít vissza *True*-t, ha az a karakterlánc kialakítható a b karakterláncból úgy, hogy egyes karaktereket kizárunk anélkül, hogy módosítanánk a megmaradt karakterek egymáshoz viszonyított pozícióit, és *False*-t különben. Például, az "ace" kialakítható, ha kizárunk karaktereket az "abcde" karakterláncból, de az "aec" nem kapható meg ugyanazzal az eljárással.

A.

```

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

```

C.

```

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

```

B.

```

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

```

D.

```

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

```

16. Legyen a $\text{ceva}(x, n, e)$ algoritmus, ahol x egy n elemű, különböző egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $1 \leq n \leq 10^3$ és $x[i] \neq x[j]$, ha $1 \leq i < j \leq n$) és e egy egész szám. Az algoritmus keresi az e elemet az x vektorban és ha megtalálja, elköltözteti az elemet a vektor első pozíciójára és True -t térít vissza, anélkül, hogy a többi elem egymáshoz viszonyított helyét módosítaná. Ha e nem található meg az x vektorban, az algoritmus False -t térít vissza és nem módosítja az x vektor tartalmát. Például, ha az x vektor elemei $[-100, 2, 71, 31, -62, 51]$ és $e = 31$, az algoritmus True -t térít vissza és az x vektor $[31, -100, 2, 71, -62, 51]$ lesz.

A $\text{ceva}(x, n, e)$ algoritmus következő kódjainak melyik változata helyes, és időbonyolultsága $O(n)$?

A.

```

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

```

B.

```

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

```

C.

```

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

```

D.

Az A, B, C változatok egyike sem

17. Legyen az $\text{expresie}(x, y, z)$ algoritmus, ahol x, y, z természetes számok ($0 \leq x, y, z \leq 10^4$):

```

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

```

Állapítsátok meg, hogy az alábbi kifejezések közül melyiknek értékét számítja ki és téríti vissza az algoritmus:

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

18. Legyen a $\text{ceFace}(v, a, b)$ algoritmus, ahol v egy vektor, amelynek n eleme van a $\{0, 1\}$ halmazból ($1 \leq n \leq 10^4$, $v[1], \dots, v[n]$), a és b nemnulla természetes számok. A v vektor növekvően rendezett.

```

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

```

A következő állítások közül melyek igazak tudva, hogy az eredeti hívás alakja $\text{ceFace}(v, 1, n)$?

- A. Ha a v vektor tartalmaz legalább egy 1-est akkor az algoritmus a vektor hosszát téríti vissza.
- B. Ha a v vektor csak 1-eseket tartalmaz, akkor az algoritmus n értékét téríti vissza.
- C. Ha a v vektor csak 0 értékeket tartalmaz, akkor az algoritmus 0-át térít vissza.
- D. Az algoritmus a v vektorban található 1-es értékek darabszámát téríti vissza.

19. Tudjuk, hogy az n hosszúságú bináris (csak 0-ás és 1-es karaktereket tartalmazó) sorozatok darabszáma egyenlő 2^n -nel. Például, ha $n = 2$ ezek: 00, 01, 10 és 11, a darabszámuk pedig $2^2 = 4$. A 100011 karaktersorozat hossza 6 és tömbszakaszként tartalmazza mind a 4 lehetséges sorozatot, amelyeknek hossza $n = 2$. Az első pozíción kezdődik az 10, a második pozíciótól kezdődően megjelenik a 00, a negyedik pozíción kezdődik a 01 és az ötödik pozíción kezdődik az 11.

Mennyi a minimális hossza annak a sorozatnak, amely tömbszakaszként tartalmazza mind a 2^n lehetséges bináris sorozatot, amelyeknek hossza $n = 4$?

A. 18

B. 19

C. 20

D. 21

20. Legyen a $t(q, x, y)$ algoritmus, ahol q egy tetszőleges karakter, x és y nemnulla természetes számok ($1 \leq x, y \leq 100$).

```

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

```

A következő állítások közül melyek igazak?

- A. A $t('c', 33, 28)$, $t('c', 10, 6)$ és $t('c', 22, 16)$ hívások eredményeként az algoritmus ugyanazokat a karaktereket írja ki.
- B. A $t('c', 33, 28)$ és $t('c', 45, 40)$ hívások eredményeként az algoritmus nem ugyanazokat a karaktereket írja ki.
- C. A $t('c', 11, 8)$ hívás eredményeként az algoritmus a "cc" karaktereket írja ki.
- D. A $t('c', 25, 16)$ hívás eredményeként az algoritmus nem a "cccc" karaktereket írja ki.

21. Legyen a $hIndex(x, n)$ algoritmus, ahol x egy n elemű vektor ($1 \leq n \leq 10^5$), amelynek elemei nemnulla természetes számok ($x[1], x[2], \dots, x[n]$). Definíáljuk az x vektor ***h-index***-ét, mely az a legnagyobb v érték, amelynek esetében igaz, hogy az x vektorban létezik legkevesebb v darab olyan érték, amelyek nagyobbak vagy egyenlők v -vel. Például, az $x = [3, 10, 2, 7, 10, 8, 50, 1, 1, 5]$ sorozat ***h-index***-e 5.

```

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

```

A következő állítások közül melyek igazak?

- A. Amikor a 22. sor következne végrehajtásra, az x vektor csökkenően rendezett.
- B. A $hIndex(x, n)$ algoritmus visszatéríti az x vektor ***h-index***-ét, ha a 22. sorba a **Return** h utasítást írjuk.
- C. A $hIndex(x, n)$ algoritmus visszatéríti az x vektor ***h-index***-ét, ha a 22. sorba a **Return** $h - 1$ utasítást írjuk.
- D. Ha a $hIndex(x, n)$ algoritmust egy szigorúan csökkenő x vektor esetében hívjuk meg, akkor az algoritmus nem téríti vissza az x vektor ***h-index***-ét, függetlenül attól, hogy milyen utasítást írunk a 22. sorba.

22. Legyen a $ceFace(n, k, x, p)$ algoritmus, ahol n, k és p nemnulla természetes számok ($1 \leq n, k, p \leq 10, p \leq n$), x egy $p + 1$ elemű vektor, amelynek elemei természetes számok ($x[0], x[1], \dots, x[p]$). Feltételezzük, hogy $x[0]$ kezdőértéke 0.

```

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

```

Állapítsuk meg, hogy a következő állítások közül melyek igazak.

- A. Miután az algoritmus meghívódik $ceFace(3, 1, x, 3)$ alakban, ez még 6-szor hívja meg önmagát.
- B. Ha $x[0]$ kezdőértéke egy 0-tól különböző érték, a $ceFace(5, 1, x, 3)$ hívás eredményeként a kiírt szóközők darabszáma nem 10.
- C. Ha az algoritmust a $ceFace(5, 1, x, 4)$ alakban hívjuk meg, a 1245 1234 1345 1235 2345 számokat írja ki, de más sorrendben.
- D. Ha az algoritmust a $ceFace(5, 1, x, 3)$ alakban hívjuk meg, a kiírt eredmény 123 124 125 134 135 145 234 235, ebben a sorrendben.

23. Legyen az $f(sir, s, d, p)$ algoritmus, ahol sir egy karakterlánc, s, d, p nemnulla természetes számok ($0 < s, d, p < 10^9$). A "+" operátor két karakterlánc konkatenálását (egymás után ragasztását) jelöli. A $print(a)$ algoritmus kiírja az a karakterláncot és egy sorvége jelet.

```

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

```

Állapítsátok meg, hogy a következő állítások közül melyek igazak az $f("", 0, 0, 2)$ hívás eredményeként?

- A. Két különböző sorba két karakterlánc lesz kiírva, mindkettőben 4 szám lesz, amelyeknek összege 0 (például, a "-1 1 -1 1" karakterláncban levő számok összege 0).
- B. Csak "-1 -1 1 1" lesz kiírva.
- C. Csak "-1 -1 1 1" lesz kiírva, de az algoritmus nem végzi be a végrehajtást egy hiba miatt.
- D. Ha a 2. sorban az AND operátort OR-ral helyettesítjük, akkor az algoritmus csak a "-1 -1"-et írja ki.

24. Legyen a $ceFace(a, i, n)$ algoritmus, ahol i és n természetes számok ($1 \leq i, n \leq 100$), a egy n elemű vektor, amely egész számokat tárol ($a[1], a[2], \dots, a[n], -100 \leq a[i] \leq 100$). Az a sorozatban található legalább egy pozitív szám. A $\max(x, y, z)$ algoritmus visszatéríti az x, y és z ($-10^4 \leq x, y, z \leq 10^4$) három egész szám maximumát. A $ceFace(a, i, n)$ algoritmus meghívja az $intermediar(a, i, m, n)$ algoritmust, ahol az a, i és n paraméterek jelentése a fenti, m pedig egy természetes szám ($1 \leq m \leq n$).

```

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

```

```

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

```

Állapítsátok meg, hogy a következő állítások közül melyek igazak, ha az algoritmust a $ceFace(a, i, n)$ alakban hívjuk meg:

- A. Az algoritmus beazonosít egy m pozíciót az a vektorban úgy, hogy vagy az $1, 2, \dots, m$ pozíciókon található elemek összege, vagy az $m, m + 1, \dots, n$ pozíciókon található elemek összege maximális bármely $1 \leq m \leq n$ esetében és visszatéríti ezt a maximális összeget.
- B. Az algoritmus visszatéríti azt a maximális összeget, amelyet úgy kapunk, hogy összeadjuk az a vektor értékeiből vett valamely részhalmaznak az elemeit.
- C. Az algoritmus visszatéríti azt a maximális összeget, amelyet úgy kapunk, hogy összeadjuk az a vektor egy tömbszakaszának elemeit.
- D. Ha az a vektor minden eleme pozitív, az algoritmus visszatéríti az a vektor összes elemének összegét.

Felvételi verseny – 2023 szeptember 8.

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

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