

Más pontosítások hiányában feltételezhetjük, hogy az aritmetikai műveleteket végtelen adattípusokon végezzük, vagyis nincs túlcsordulás és alulcsordulás. Minden sorozatot 1-től számozzunk

(2022. júliusi felvételi, 24. feladat)

Adott a $\text{det}(a, n, m)$ algoritmus, ahol a egy n elemű, természetes számokból álló sorozat ($a[1], a[2], \dots, a[n]$ ha $n \geq 1$) vagy az üres sorozat, ha $n = 0$. n és m természetes számok ($0 \leq n \leq 100, 0 \leq m \leq 10^6$).

```
1. Algorithm det(a, n, m):
2.   For i ← 1, n - 1 execute
3.     For j ← i + 1, n execute
4.       If a[i] > a[j] then
5.         tmp ← a[i]
6.         a[i] ← a[j]
7.         a[j] ← tmp
8.       EndIf
9.     EndFor
10.  EndFor
11.  i ← 1
12.  j ← n
13.  b ← False
14.  While i < j execute
15.    If a[i] + a[j] = m then
16.      b ← True
17.    EndIf
18.    If a[i] + a[j] < m then
19.      i ← i + 1
20.    else
21.      j ← j - 1
22.    EndIf
23.  EndWhile
24.  return b
25. EndAlgorithm
```

A következő állítások közül melyek igazak?

- A Az algoritmus *True*-t térít vissza, ha az a sorozatban létezik egy számpár, amelynek összege m .
- B Az algoritmus mindig *False*-t térít vissza.
- C Az algoritmus *False*-t térít vissza, ha $n = 0$.
- D Az 2., 3., ..., 10. sorokban az algoritmus növekvő sorrendbe rendezi az a sorozatot.

(2022 szept. felvételi, 1. feladat)

Adott a $\text{dönt}(n, x)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10000$) és x egy n elemű, természetes számokat tároló vektor ($x[1], x[2], \dots, x[n], -100 \leq x[i] \leq 100$, ahol $i = 1, 2, \dots, n$):

```
Algorithm dönt(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
```

Mikor fog az algoritmus *True*-t téríteni vissza?

- A Ha az x vektor a következő elemeket tárolja: 1, 2, 3, ..., 10
- B Ha az x vektor szigorúan növekvő.
- C Ha az x vektornak nincsenek negatív elemei.
- D Ha az x vektornak vannak pozitív elemei, amelyek a negatív elemek előtt találhatók.

(2022 szept. felvételi, 16. feladat)

Legyen egy program, amely növekvő sorrendben generál minden, pontosan 5 különböző számjegyből álló természetes számot, amelyek a 2, 3, 4, 5, 6 számjegyekből építhetők fel.

Adjátok meg azt a két számot, amelyet a program közvetlenül az alábbi sorozat előtt és közvetlenül utána generál: 34256, 34265, 34526, 34562.

A. 32645 és 34625 B. 32654 és 34655 C. 32654 és 34625 D. 32645 és 34655

(2022 szept. felvételi, 19. feladat)

Adottak az a és n természetes számok ($1 \leq a, n \leq 1000$), valamint az n elemű, természetes számokat tároló V vektor ($V[1], V[2], \dots, V[n]$) és a $\text{one}(a, n, V)$ és $\text{two}(a, n, V)$ algoritmusok:

```
Algorithm one(a, n, V):  
  p ← 1; i ← 1;  
  While (i ≤ n) AND (a > V[p]) execute  
    p ← p + 1  
    i ← i + 1  
  EndWhile  
  return p  
EndAlgorithm
```

```
Algorithm two(a, n, V):  
  p ← 1; i ← 1;  
  While i ≤ n execute  
    If a > V[i] then  
      p ← p + 1  
    EndIf  
    i ← i + 1  
  EndWhile  
  return p  
EndAlgorithm
```

Mely tulajdonsággal rendelkezhet a V vektor ahhoz, hogy bármely n és adott tulajdonságú V esetében a két algoritmus azonos értéket térítsen vissza tetszőleges a értékre?

- A. A V vektor minden eleme egyenlő.
- B. A V vektor minden eleme különböző és növekvően rendezett.
- C. A V vektor minden eleme különböző és csökkenően rendezett.
- D. A V vektor növekvően rendezett, de az elemek nem feltétlenül különbözők.

(2022 szept. felvételi, 28. feladat)

Adott az n különböző természetes számot tároló, szigorúan növekvő a sorozat ($a[1], a[2], \dots, a[n], 2 \leq n \leq 1000$). Egy sorozatban *lokális csúcsnak* nevezünk egy olyan számot, amely szigorúan nagyobb, mint az előző pozíción levő szám, valamint a következő pozíción található szám. A sorozat első és utolsó eleme nem lehet lokális csúcs. Azt szeretnénk, hogy az $\text{átrendezés}(a, n)$ algoritmus rendezze át a sorozatban található értékeket úgy, hogy a sorozatnak maximális számú lokális csúcsa legyen, majd térítse vissza az új sorozatot. A b lokális változó sorozat. A következő algoritmusok közül melyek helyesek?

A.

```
Algorithm átrendezés(a, n):  
  i ← n  
  For p ← 2, n, 2 execute  
    b[p] ← a[i]  
    i ← i - 1  
  EndFor  
  For p ← 1, n, 2 execute  
    b[p] ← a[i]  
    i ← i - 1  
  EndFor  
  return b  
EndAlgorithm
```

B.

```
Algorithm átrendezés(a, n):  
  i ← n  
  For p ← 2, n, 2 execute  
    b[p] ← a[i]  
    i ← i - 1  
    b[p - 1] ← a[i]  
    i ← i - 1  
  EndFor  
  If n MOD 2 = 1 then  
    b[n] ← a[i]  
  EndIf  
  return b  
EndAlgorithm
```

C.

```
Algorithm átrendezés(a, n):  
  i ← n
```

D.

```
Algorithm átrendezés(a, n):  
  i ← n
```

```

For p ← 2, n, 2 execute
  b[p] ← a[i]
  i ← i - 1
EndFor
i ← 1
For p ← 1, n, 2 execute
  b[p] ← a[i]
  i ← i + 1
EndFor
return b
EndAlgorithm

```

```

For p ← 2, n, 3 execute
  b[p] ← a[i]
  i ← i - 1
  b[p - 1] ← a[i]
  i ← i - 1
  If p + 1 ≤ n then
    b[p + 1] ← a[i]
    i ← i - 1
  EndIf
EndFor
If n MOD 3 = 1 then
  b[n] ← a[i]
EndIf
return b
EndAlgorithm

```

Megfigyelések: meg kell érteni, hogy melyik algoritmus mit végez...

A – beteszi a páros pozíciókra a legnagyobb számokat (csökkenő sorrendbe), majd a páratlan pozíciókra a további értékeket, szintén csökkenő sorrendben.

B – 2. pozícióra a legnagyobb, elé a következő legnagyobbat, aztán a 4. pozícióra ismét a csökkenő sorban következőt, stb.

C – a páros indexekre beteszi csökkenő sorrendben a legnagyobb értékeket, majd a páratlan indexekre a legkisebbeket, növekvő sorrendben.

D – hármasával tesz be kisebb-nagyobb-kisebb értékeket

(2022 matinfó verseny, 6. feladat)

6. Ahhoz, hogy olyan számokat generáljunk, amelyeknek n számjegyük van a $\{0, 6, 7\}$ halmazból, egy olyan algoritmust futtatunk, amely $n = 2$ esetében növekvő sorrendben generálja a 60, 66, 67, 70, 76, 77 számokat. Ha $n = 4$, melyik számot generálja ugyanaz az algoritmus közvetlenül a 6767 után?

- A. 7667
- B. 6760
- C. 6776
- D. Egyik változat sem helyes

(2022 matinfó verseny, 16. feladat)

Legyen minden $l \in \{1, 2, 3\}$ hosszú sorozat, amelyeknek elemei az $\{a, b, c, d, e\}$ halmazhoz tartoznak. Hány olyan sorozat található ezek között, amelyeknek elemei abécé sorrend szerint szigorúan növekvő sorrendben találhatók és páratlan darabszámú mássalhangzót tartalmaznak? (b, c és d mássalhangzók.)

- A. 14
- B. 13
- C. 26
- D. 81

(2021 matinfó verseny, 16. feladat)

A `költöztetNullákat(a, n)` algoritmus bemeneti paraméterei az n elemű a sorozat, amelynek elemei egész számok ($a[1], a[2], \dots, a[n]$) és az n ($1 \leq n \leq 10000$) egész szám. Az algoritmus elköltözteti a 0 értékű elemeket a sorozat végére úgy, hogy megőrzi a 0-tól különböző elemek eredeti sorrendjét. Például, ha $a = [4, 0, 2, 5, 1, 0, 7, 11, 0, 3]$, az algoritmus végrehajtása után $a = [4, 2, 5, 1, 7, 11, 3, 0, 0, 0]$.

Az alábbi `költöztetNullákat(a, n)` algoritmusok közül melyek helyesek?

A.

```
Algoritmus költöztetNullákat(a, n):  
  s ← igaz  
  Amíg s = igaz végezd el  
    s ← hamis  
    Minden i ← 1, n - 1 végezd el  
      Ha a[i] = 0 akkor  
        tmp ← a[i]  
        a[i] ← a[i + 1]  
        a[i + 1] ← tmp  
      s ← igaz  
    vége(ha)  
  vége(minden)  
vége(amíg)  
Vége(algoritmus)
```

B.

```
Algoritmus költöztetNullákat(a, n):  
  c ← 0  
  Minden i ← 0, n végezd el  
    Ha a[i] = 0 akkor  
      c ← c + 1  
    vége(ha)  
  vége(minden)  
  i ← n  
  Amíg c > 0 végezd el  
    a[i] ← 0  
    i ← i - 1  
    c ← c - 1  
  vége(amíg)  
Vége(algoritmus)
```

C.

```
Algoritmus költöztetNullákat(a, n):  
  d ← 0  
  i ← 1  
  Amíg i + d ≤ n végezd el  
    Amíg (i + d ≤ n) ÉS (a[i + d] = 0) végezd el  
      d ← d + 1  
    vége(amíg)  
    Ha i + d ≤ n akkor  
      a[i] ← a[i + d]  
      i ← i + 1  
    vége(ha)  
  vége(amíg)  
  Amíg i ≤ n végezd el  
    a[i] ← 0  
    i ← i + 1  
  vége(amíg)  
Vége(algoritmus)
```

D.

```
Algoritmus költöztetNullákat(a, n):  
  i ← 1  
  f ← n  
  Amíg i < f végezd el  
    Amíg (i < f) ÉS (a[i] ≠ 0) végezd el  
      i ← i + 1  
    vége(amíg)  
    Amíg (i < f) ÉS (a[f] = 0) végezd el  
      f ← f - 1  
    vége(amíg)  
    Ha i < f akkor  
      tmp ← a[i]  
      a[i] ← a[f]  
      a[f] ← tmp  
    vége(ha)  
  vége(amíg)  
Vége(algoritmus)
```