

# NYELVFÜGGETLEN PROGRAMOZÁS

5. forduló



A kategória támogatója: SAP Hungary Kft.

## Ismertető a feladathoz

**A feladatlap több csatolmányt is tartalmaz, ezért a megoldását asztali gépen javasoljuk!**

- Minden fordulóban két algoritmikus feladat lesz.
- Minden feladat esetében 5 "éles" inputra kell előállítanod az outputokat, amelyeket aztán a versenyfelületen a megfelelő szövegmezőbe kell illesztened.
- A megoldásod ellenőrzését segítő, minden feladathoz tartozik 2 db példa input és output.
- Pl. ha egy feladat címe "Cica", akkor a cica.peldaX.in.txt-ben lesz a példa input, a cica.peldaX.out.txt-ben pedig az ehhez tartozó példa output (X egy egész szám). A cicaX.in.txt fájlokban lesznek a pontokért megoldandó inputok, ahol X: 1..5.
- Mindezeket a txt fájlokat a csatolt tömörített archívum tartalmazza, melyet a feladatsor indítása után tölthetsz le.
- A megoldásokat bármilyen programnyelven elkészítheted.
- A forráskódot nem kell beküldeni, csak az outputokat.

Jó szórakozást!

Felhasznált idő: 39:51/40:00

Elért pontszám: 0/8

Indítás utáni csatolmányok

## 1. feladat 0/1 pont

### Összeg

Adott három nemnegatív egész szám: A, B és C. Határozzuk meg a legkisebb olyan K számot, melyre  $A \leq K \leq B$ , és K számjegyeinek összege C!

A bemenet egyetlen sorból áll, mely az A, B és C számokat tartalmazza szóközzel elválasztva. A kimenet K legyen. Ha nincs ilyen K szám, akkor a NINCS szót kell kiírni (csupa nagybetűvel).

Mi az **osszeg1.in.txt**-hez tartozó output?

### Válasz

133950

A helyes válasz:

133059

## Magyarázat

Még dinamikus programozás sem kell a megoldáshoz. Az első ötlet, hogy először felső korlát nélkül oldjuk meg a feladatot, aztán csekkeljük, hogy az eredmény legfeljebb B-e. A következő algoritmust használjuk ezután, amelyről bebizonyítható, hogy mindig helyes eredményt ad: Jelölje  $\text{sum}(A)$  A számjegyeinek összegét. Ha  $C = \text{sum}(A)$ , akkor készen vagyunk, A a megoldás. Egyébként A-ból indulunk ki. Ha  $C > \text{sum}(A)$ , akkor addig növeljük a legutolsó nem 9-es számjegyet, amíg el nem érjük C-t. Ha elfogynak a számjegyek (azaz minden számjegy 9-es lett), akkor a szám elé írunk egy 0-t, és úgy folytatjuk. Végül, ha  $C < \text{sum}(A)$ , akkor a megoldás úgy fog kinézni, hogy A egy számjegyet (ami potenciálisan egy, a szám elé írandó 0) megnöveljük 1-gyel, a tőle jobbra lévő számjegyeket 0-ra állítjuk, majd az előző algoritmust (utolsó számjegy növelgetése) végezzük el a kapott számra.

```
#!/usr/bin/env python3

# b = végtelen
def solve(a, c):
    strA = str(a)
    digits = list(map(int, strA))
    sumOfA = sum(digits)

    if sumOfA == c:
        return a

    if sumOfA < c:
        diff = c - sumOfA
        assert diff > 0
        pos = -1
        while diff:
            assert diff > 0
            if pos + len(digits) < 0:
                digits.insert(0, 0)

            increment = min(diff, 9 - digits[pos])
            assert increment >= 0

            digits[pos] += increment
            diff -= increment
            pos -= 1

        assert sum(digits) == c
        assert max(digits) <= 9
        assert min(digits) >= 0

        result = int(''.join(map(str, digits)))
        assert result >= a
        return result

    diff = sumOfA - c
    assert diff > 0
    pos = -1
    while diff > -1 or digits[pos] == 9:
        assert pos + len(digits) >= 0
        diff -= digits[pos]
        digits[pos] = 0
        pos -= 1

    if pos + len(digits) < 0:
        digits.insert(0, 0)
    assert pos < -1
    digits[pos] += 1
```

```

diff += 1

assert diff <= 0
assert sum(digits) == c + diff
d = solve(0, -diff)
assert d < 10**(-pos - 1)
newDigits = list(map(int, str(d)))
assert pos + len(newDigits) < 0
digits[-len(newDigits):] = newDigits
assert sum(digits) == c
result = int("".join(map(str, digits)))
assert result >= a
return result

def solveFile(fn:str, fOut):
    with open(fn) as f:
        a, b, c = map(int, f.readline().strip().split())

    k = solve(a, c)
    if k > b:
        k = "NINCS"

    result = k
    message = "Output for %s: %s" % (fn, result)
    print(message)
    fOut.write(message+"\n")

if "pelda" in fn:
    fnPeldaOut = fn.replace(".in.", ".out.")
    assert fnPeldaOut != fn
    with open(fnPeldaOut, "w") as fPeldaOut:
        fPeldaOut.write(str(result))

def solveFiles(fnOut:str):
    with open(fnOut, "w") as fOut:
        for i in range(1, 6):
            solveFile("osszeg%s.in.txt" % (i,), fOut)
        for i in range(1, 3):
            solveFile("osszeg.pelda%s.in.txt" % (i,), fOut)

def main():
    solveFiles(fnOut = "out.txt")

if __name__ == "__main__":
    main()

```

## 2. feladat 0/1 pont

Mi az `osszeg2.in.txt`-hez tartozó output?

### Válasz

A helyes válasz:

89444029

## Magyarázat

Ld. fent.

## 3. feladat 0/1 pont

Mi az `osszeg3.in.txt`-hez tartozó output?

### Válasz

A helyes válasz:

42659520000000

## Magyarázat

Ld. fent.

## 4. feladat 0/2 pont

Mi az `osszeg4.in.txt`-hez tartozó output?

### Válasz

A helyes válasz:

247373836950000000

## Magyarázat

Ld. fent.

## 5. feladat 0/3 pont

Mi az `osszeg5.in.txt`-hez tartozó output?

### Válasz

**5322991872892999**

## Ld.fent