

# NYELVFÜGGETLEN PROGRAMOZÁS

7. 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 segítőndő, 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: 40:00/40:00

Elért pontszám: 0/8

Indítás utáni csatolmányok

## 1. feladat 0/1 pont

### Bumm

A "bumm" játékban 1-től kezdve soroljuk a pozitív egész számokat, de a 7-tel osztható vagy 7-es számjegyet tartalmazó számok helyett "Bumm"-ot mondunk. Ilyen számok pl. a 7, a 17 és a 49.

Példa: 1, 2, 3, 4, 5, 6, bumm, 8, 9, 10, 11, 12, 13, bumm, 15, 16, bumm, 18, 19, 20, bumm, 22, ...

(Amelyik szám mindkét feltételnek megfelel, ahelyett is természetesen csak 1 bummot mondunk. Amelyik számban több 7-es számjegy is van, ahelyett is szintén csak 1 db bummot mondunk.)

Számoljuk ki, hogy hányszor mondunk 1-től N-ig bummot, ahol N pozitív egész szám!

A bemeneti fájl N-et tartalmazza.

Mi a **bumm1.in.txt**-hez tartozó output?

A helyes válasz:

14675

## Magyarázat

A feladat nehezebb, mint azt első látásra gondolnánk. A brute force módszer kis számokra működik, de pl. az utolsó tesztesetet már lehetetlen így megoldani. A 7-tel osztható számokat triviális N-ig megszámolni. A 7-es számjegyet tartalmazóakat kicsit már nehezebb, itt a komplementert érdemes számolni (7-est nem tartalmazóak) és utána kivonni. De ha csak hozzáadnánk ezek számához a 7-tel osztható számok számát, akkor sok számot kétszer számolnánk. Azt kiszámolni, hogy hány olyan szám van N-ig, amely osztható 7-tel, de nem tartalmaz 7-es számjegyet, nem triviális. Az alábbi kódban implementált dinamikus programozási algoritmust tudjuk használni erre (nNumbersDivisibleBy7AndNotHaving7Until függvény).

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

# 1..N-ig a 7-est tartalmazó számok száma.
def nNumbersHaving7Until(n:int) -> int:
    if n < 7:
        return 0

    strN = str(n)
    L = len(strN)
    firstDigitOfN = int(strN[0])

    result = 0
    for firstDigit in range(0, firstDigitOfN):
        result += (10**(L-1)) if firstDigit == 7 else (10**(L-1) - 9**(L-1))

    n2 = int(strN[1:]) if L > 1 else 0
    result += (n2+1) if firstDigitOfN == 7 else nNumbersHaving7Until(n2)

    return result

# A 7-es számjegyet nem tartalmazó, de 7-tel osztható számok száma 1..N-ig.
# Először 0..N-ig számolunk az egyszerűség kedvéért, aztán 1-et levonunk a 0 miatt.
# Dinamikus programozás. A részfeladat az L hosszú, K-val kezdődő, M 7-es maradékú számok száma, és még van e
def nNumbersDivisibleBy7AndNotHaving7Until(n:int) -> int:
    strN = str(n)
    maxL = len(strN)
    allowedDigits = tuple(i for i in range(10) if i != 7)
    x = {}
    for L in range(1, maxL+1):
        digitOfN = int(strN[-L])
        unit = 7 - (10**(L-1)) % 7

        for K in allowedDigits:
            for M in range(7):
                # Tetszőleges számok (B==False)
                if L > 1:
                    value = 0
                    M2 = (M + unit*K) % 7
                    for K2 in allowedDigits:
                        value += x[L-1, K2, M2, False]
                else:
                    value = int(K%7 == M)
            x[L, K, M, False] = value

    # Számok, amelyek legfeljebb akkorák, mint N-nek az L hosszúságú suffixe (B==True)
    if L > 1:
```

```

        value = 0
        if K <= digitOfN:
            M2 = (M + unit*K) % 7
            for K2 in allowedDigits:
                value += x[L-1, K2, M2, K == digitOfN]
        else:
            value = int(K%7 == M and K <= digitOfN)
        x[L, K, M, True] = value

    result = 0
    for K in allowedDigits:
        result += x[maxL, K, 0, True]

    # Eddig egyszerűbb volt a 0-val együtt számolni, most levonjuk.
    result -= 1

    return result

def solve(n:int) -> int:
    A = nNumbersHaving7Until(n)
    B = nNumbersDivisibleBy7AndNotHaving7Until(n)
    return A + B

def solveFile(fn:str, fOut):
    with open(fn) as f:
        n = int(f.readline().strip())
    assert n

    result = solve(n)

    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, 3):
            solveFile("bumm.pelda%s.in.txt" % (i,), fOut)
        for i in range(1, 6):
            solveFile("bumm%s.in.txt" % (i,), fOut)

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

if __name__ == "__main__":
    main()

```

**Megoldás C++-ban:**

[brute force\(1\)](#)

## 2. feladat 0/1 pont

Mi a **bumm2.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

3905695429

### Magyarázat

Ld. fent.

### 3. feladat 0/1 pont

Mi a **bumm3.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

717958959620

### Magyarázat

Ld. fent.

### 4. feladat 0/2 pont

Mi a **bumm4.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

1923027915771716130

### Magyarázat

Ld. fent.

## 5. feladat 0/3 pont

Mi a `bumm5.in.txt`-hez tartozó output?

### Válasz

A helyes válasz:

73256749926411914220683141269344007395851094480081

### Magyarázat

Ld. fent.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos