

# NYELVFÜGGETLEN PROGRAMOZÁS

2. forduló



A kategória támogatója: SAP Hungary Kft.

## Ismertető a feladathoz

### Útmutató:

- A **radio button-os kérdésekre** egy helyes válasz van.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Badge-ke**t a 4.forduló után kapsz majd először.
- Az **adatbekérős kérdéseknél** igyekeztünk minden variációt megadni (kisbetű, nagybetű, szóköz), de ha mégis eltérést tapasztalsz a megoldásokban, kérjük, jelezd felénk!

+1: Azért szólunk, hogy senkit ne a végén érjen meglepetés: a játék nem tipp-mix és csapatkategória sincs! Természetesen akinek nem inge...

**Jó versenyzést kívánunk!**

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

- 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

## Halmaz

Válasszunk ki az első  $N$  db pozitív egész szám közül minél többet úgy, hogy ne legyen olyan  $x$ , melyre  $x$  és  $2x$  is a halmazban van. A bemeneti fájl  $N$ -et tartalmazza, a kimenet pedig a maximális ilyen halmaz mérete legyen.

Mi a **halmaz1.in.txt**-hez tartozó output?

## Válasz

A helyes válasz:

5

## Magyarázat

Építsünk (képzeletben) gráfot az  $1..N$  számokból, és kössük össze az  $(x, 2x)$  párokat. A gráf diszjunkt utakból fog állni, minden út egy páratlan számmal kezdődik. Az optimum eléréséhez minden útból ki kell választani minden második pontot, a kezdőponttal kezdve. Azaz ki kell választanunk a páratlan számokat, utána a páratlan számok 4-szereseit, utána a páratlan számok 16-szorosait, és így tovább. Az alábbi megoldás kiszámolja minden  $L$ -re, hogy hány db  $L$  pontból álló út van ( $O(\log N)$  különféle  $L$  létezik).

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

def isOdd(x):
    return x % 2 == 1

def nEvenNumbersIn(a:int, b:int):
    if isOdd(a):
        a += 1
    if not isOdd(b):
        b += 1
    if a > b:
        return 0
    return (b-a+1) // 2

# Végpontokat beleértjük az intervallumba.
def nOddNumbersIn(a:int, b:int):
    assert 0 <= a <= b
    return nEvenNumbersIn(a+1, b+1)

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

    L = 1
    result = 0
    while True:
        a = n // (2**L) + 1
        b = n // (2**(L-1))
        if b == 0:
            break
        result += ((L+1)//2) * nOddNumbersIn(a, b)
        L += 1

    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 main():
    with open("out.txt", "w") as fOut:
        for i in range(1, 6):
            solveFile("halmaz%s.in.txt" % (i,), fOut)
        for i in range(1, 3):
            solveFile("halmaz.pelda%s.in.txt" % (i,), fOut)

if __name__ == "__main__":
    main()

```

Egy lelkes versenyző megoldása C#-ban:

```

public string Feladat10pt(BigInteger maxNumberInclusive)
{
    BigInteger solution = 0;
    int maxPowCountExclusive = (int)Math.Ceiling(BigInteger.
Log(maxNumberInclusive, 2)); // define least power count for 2 which produces greater number than maxNumberIn
for (int power = 2; power <= maxPowCountExclusive; power += 2) // loop through all allowed powers that genera
{
    var baseNumber = BigInteger.Pow(2, power);
    BigInteger denominator = maxNumberInclusive / baseNumber; // find highest allowed denominator
    solution += denominator % 2 == 0 ? denominator / 2 : denominator / 2 + 1; // even numbers cannot be used for
}
    solution += maxNumberInclusive % 2 == 0 ? maxNumberInclusive / 2 : maxNumberInclusive / 2 + 1; // add odd num
    return solution.ToString("N0");
}

```

## 2. feladat 0/1 pont

Mi a **halmaz2.in.txt**-hez tartozó output?

Válasz

A helyes válasz:

12

Magyarázat

Ld. fent.

## 3. feladat 0/1 pont

Mi a **halmaz3.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

353

### Magyarázat

Ld. fent.

## 4. feladat 0/2 pont

Mi a **halmaz4.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

242193040

### Magyarázat

Ld. fent.

## 5. feladat 0/3 pont

Mi a **halmaz5.in.txt**-hez tartozó output?

### Válasz

A helyes válasz:

4687331540247958765070549119336099934009

### Magyarázat

Ld. fent.



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

© 2023 Human Priority Kft.

KÉSZÍTETTE  **cone**

Megjelenés

 Világos 