

# IPAR 4.0 C# .NET ALAPOKON

3. forduló



A kategória támogatója: Semilab Zrt.

RENDELKEZÉSRE ÁLLÓ IDŐ:

30:00

## Ismertető a feladathoz



### Fontos információk

Ezután a forduló után automatikusan jár a [kitartóknak szóló garantált ajándékunk](#), érdemes kitöltened a feladatlapot! :)

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.

A kérdésekre **mindig van helyes válasz**, olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

### Harmadik forduló

A gyár további folyamatok elvégzéséhez kéri a segítségünket, részletes leírások a feladatokban.

Felhasznált idő: 02:07/30:00

Elért pontszám: 0/7

## 1. feladat 0/2 pont

Mit ír ki az alábbi kódrészlet?

```
6 references
internal class IntWrapper
{
    5 references
    public int N { get; set; }

    1 reference
    public IntWrapper(int n)
    {
        N = n;
    }

    0 references
    public static bool operator ==(IntWrapper n1, IntWrapper n2)
    {
        return n1.N == n2.N;
    }

    0 references
    public static bool operator !=(IntWrapper n1, IntWrapper n2)
    {
        return n1.N != n2.N;
    }
}

0 references
private static void Main(string[] args)
{
    List<int> numbers = new List<int> { 1, 2, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10 };
    var result = numbers.Select(n => new IntWrapper(n)).Distinct();
    Console.WriteLine(result.Count());
}
```

### Válaszok

A helyes válasz:

13

### Magyarázat

Hiába van felülírva az == és != operátor, a Distinct() helyes működéséhez az Equals és GetHashCode függvényeket kell felülírni.

## 2. feladat 0/5 pont

A gyártó különböző méretű szilícium wafereket gyárt, melyeken CPU-kat helyeznek el. Jelenleg 5 wafer mérettel dolgoznak, ezekre **1000, 541, 224, 25 és 1** CPU helyezhető. A legköltséghatékonyabb nagy wafereket gyártani, viszont egy adott

rendelésnél nem lehet túllépni a kért CPU-k számát, a gyártási folyamat pedig nem engedi meg, hogy egy adott waferre a mérethez tartozó CPU számnál kevesebb kerüljön, így a legnagyobbra például mindig 1000 kerül.

Például egy vevő 17594 CPU-t rendel tőlük. Ezeket leginkább költséghatékony a gyártónak 17 db 1000-es, 1 db 541-es, 2 db 25-ös és 3 db 1-es méretű waferre legyártani ( $17 \times 1000 + 1 \times 541 + 2 \times 25 + 3 \times 1 = 17594$ ).

Készítsük el a `GetTotalWaferCount(int cpuCount)` függvényt, amely azt adja meg, hogy összesen hány wafert használnak fel a gyártás során, ha a fentiek szerint költséghatékonyaságra törekednek, és pontosan annyi CPU-t gyártanak, mint ami a rendelésben szerepel.

A **3\_fordulo\_2\_feladat.txt** fájlban vesszővel elválasztva találhatóak a vevők CPU igényei. Minden igényt külön kezelnek, egy waferre nem kerülhet két különböző megrendeléshez tartozó CPU. Adjuk meg, hogy összesen hány wafert használunk fel a megrendelés költséghatékony teljesítéséhez!

Teszteléshez a `3_fordulo_2_feladat_test.txt` eredménye: 7201.

## Válaszok

A helyes válasz:

6959

5957

## Magyarázat

```
static List<int> pieces = new List<int> { 1000, 541, 224, 25, 1 };
private static int GetTotalWaferCount(int n)
{
    int sum = 0;
    int index = 0;
    while (n > 0)
    {
        while (n >= pieces[index])
        {
            n -= pieces[index];
            sum++;
        }
        index++;
    }
    return sum;
}

private static int GetTotalWaferCount(string inputFile)
{
    var cpuRequirements = File.ReadAllText(inputFile)
        .Split(',')
        .Select(int.Parse)
        .ToList();

    return cpuRequirements.Sum(cpuRequirement => GetTotalWaferCount(cpuRequirement));
}
```

A számítás szintén helyes, ha tesztadatnak az 6259-et vesszük alapul, ekkor az eredmény 5957.

[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

 Világos 