

Ismertető a feladathoz

Valahol Amerika középnnyugati részén, egy elhagyatottnak tűnő kastélyban nem minden az, aminek látszik. Elsőre csak egy fényűző, régimódi villának tűnik, de az alagsorában berendezett laboratóriumban jelenleg is titkos tevékenység zajlik. A kastély tulajdonosa a milliárdos Oswald E. Spencer. Az ő vezetésével egy sereg tudós dolgozik egy halálos vírus kifejlesztésén, amely reményeik szerint forradalmasíthatja a hadiipart. A kutatást a hatóságok már leállították, de a munkálatokat Spencer illegálisan tovább folytatja. Te egy átlagos informatikus vagy, aki a villában dolgozik, és egyre több furcsa dolgot tapasztalsz. Kollégák tűnnek el minden nyom nélkül, furcsa rémséges hangok szűrődnek ki az alsóbb lezárt szintekről, és a levegőt átjárja minden nap a nyugtalanság, a félelem. A hatóságok felkerestek pár napja, hogy szabotáld Spencer informatikai rendszerét, és legyél kettős ügynök. Hogy véget vess a kastélyban folyó szörnyű dolgoknak, belementél az ajánlatba...

Spencer laborja egy modern C# .NET-alapú informatikai infrastruktúrára épül. A megfelelő modulok átkalibrálásával és új kódsorok beillesztésével úgy lesz felfedhető a hatóságoknak az illegális vírusfejlesztés minden részlete, hogy megúszod ép bőrrel. Már rettegésben telik minden napod a kastélyban, mert félsz, hogy te is eltűnsz, mint több kollégád a múltban. Ez az akció az egyetlen esélyed. Ha mindent ügyesen csinálsz, 7 hét alatt rács mögé juttathatod Oswald E. Spencert.

A hatóság éjjel kopogtat az ajtódon pontosan a megbeszélt időpontban. A kettős ügynök munkáddal kapcsolatban hoztak részleteket. Az ügynök mogorván, egy gyors kérdéssel tesz próbára, hogy meggyőződjön róla, hogy valóban informatikus vagy-e, ahogy azt állítod magadról. Három programrészletet ad a kezvedbe, és arra kíváncsi, hogy mit ír ki a program.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 1 pont

Mit ír ki az alábbi kódrészlet?

```
public void Challenge1A()
{
    int x = 5;
    if (x > 0)
    {
        Console.WriteLine('P');
    }
    if (x < 0)
    {
        Console.WriteLine('N');
    }
    else
    {
        Console.WriteLine('0');
    }
}
```

A megoldás:  
P0

**Magyarázat a megoldáshoz**

Az első elágazás lefut, mivel x pozitív, és mivel a másik elágazás nem bele van ágyazva, ezért annak a 'különben' ága is le fog futni.

2. feladat 0 / 2 pont

Adott az alábbi kódrészlet. Azt szeretnénk elérni, hogy a kódban lévő törtszámot kerekítse a legközelebbi egész értékre (101). Válaszd ki a felsorolt lehetőségek közül az(oka)t, amellyel (amelyekkel) a ??-et a kódban helyettesítve ez történik!

```
public void Challenge1B()
{
    Console.WriteLine($"{100.62:??}");
}
```

- ☐ K0
- ☒ N0
- ☒ F0
- ☐ P0

**Magyarázat a megoldáshoz**

Az N0 vagy n0 vagy F0 vagy f0 formázó kifejezések bármelyike felfelé kerekíti a törtet.

3. feladat 0 / 3 pont

Mi lesz az alábbi függvény eredménye, ha paraméter megadása nélkül hívjuk meg?

```
public IEnumerable<int> Challenge1C(int count = 5)
{
    int pr = 1;
    int cr = 1;
    for (int i = 0; i < count; i++)
    {
        yield return pr;
        int t = pr + cr;
        pr = cr;
        cr = t;
    }
}
```

- ☐ 0,1,1,2,3
- ☐ 1,2,3,4,5
- ☒ 1,1,2,3,5
- ☐ 1,2,3,5,8

**Magyarázat a megoldáshoz**

A yield az eredményeket egy sorozatba pakolja bele. Először a pr, majd a cr értéke kerül bele (1, 1), ezt követően mindig az előző két elem összege (Fibonacci-sorozat).

.NET fejlesztés C# nyelven (DXC)  
2. forduló

Ismertető a feladathoz

Az ügynök napokkal ezelőtt megbízott a szabotázs első lépéseivel. Két programkódot tervezel beilleszteni a biztonsági réteg kritikus nyelvi szervizeibe, ezzel biztosítva a munkálataid további lépéseit. Még utoljára ellenőrzöd a kódok működését.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 4 pont

Mi lesz az **s** változó értéke az alábbi metódus futásának végén? A választ vesszőkel elválasztva, csupa nagybetűvel, szóköz nélkül kell megadni! pl.AB,CD,EF,GH

```
public void Challenge2A()
{
    string s = "CQ,CQ,CQ,DE,HA3DM,K";
    s.Replace(",", " ");
    var sl = s.Split(',').ToList();
    sl.RemoveAll(ss => ss.Substring(0, 1)
        .Equals("c", StringComparison.CurrentCultureIgnoreCase));
}
```

A megoldások:  
CQ,CQ,CQ,DE,HA3DM,K  
CQ,CQ ,CQ,DE,HA3DM,K

**Magyarázat a megoldáshoz**

Sem a Replace, sem a Split nem változtatja meg a string eredeti értékét, a megváltozott értéket bele kell tenni egy változóba.

2. feladat 0 / 5 pont

Adott az alábbi metódus:

```
public int Challenge2B(string key, int value)
{
    value /= 2;
    var x = new bool[] {
        key == "schluß",
        key.Equals("schluß"),
        key.Equals("schluß", StringComparison.Ordinal),
        key.Equals("Schluß", StringComparison.CurrentCulture),
        key.Equals("schluß", StringComparison.OrdinalIgnoreCase),
        key.Equals("schluß", StringComparison.CurrentCulture),
        key.Equals("Schluß", StringComparison.CurrentCultureIgnoreCase),
        value!=50
    };
    int sum = x.Sum(a => a ? 1 : 0);
    return sum;
}
```

Mi lesz a visszatérési érték az alábbi hívás eredményeképpen? A választ számjegyként add meg!

```
int sum = Challenge2B("schluss",101);
```

A megoldás:  
2

**Magyarázat a megoldáshoz**

Az x tartalma false, false, false, false, false, true, true, false. A sum-ban megszámoljuk, hány db true volt, ez 2.

Ismertető a feladathoz

A korábban kialakított biztonsági réssel sikerült mélyebbre jutni Spencer rendszereiben, és elérni a titkosított dokumentumok zömét. Azonban azok csak utalásokat tesznek egy szérumra, amit minden áron titokban kell tartani. Sajnos a szérumra vonatkozó minden egyéb információt offline tárolnak valahol az alsóbb, lezárt szinteken. Ahhoz, hogy megállítsd az illegális műveleteket, újabb szabotázs programkódokat készítettél. Hogy senki se szűrje ki a hibákat, próbálod őket feltűnés nélkül megtalálni. Mint mindig, utoljára még egyszer ellenőrzöd a kódot, amit írtál.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 15 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 4 pont

Milyen színű lesz a p1-es toll az alábbi kód futásának végére?

```
public void Challenge3A()
{
    System.Drawing.Point pt1 = new System.Drawing.Point(20, 30);
    System.Drawing.Point pt2 = pt1;
    pt2.X += pt2.Y;

    System.Drawing.Pen p1 = new System.Drawing.Pen(System.Drawing.Color.
    System.Drawing.Pen p2 = p1;
    p2.Color = System.Drawing.Color.FromArgb(pt1.X, pt2.Y, pt1.Y + pt2.X
}
```

- ☐ R = 20, G = 30, B = 50
- ☐ AliceBlue
- ☒ R = 20, G = 30, B = 80
- ☐ R = 50, G = 30, B = 80

Magyarázat a megoldáshoz

A Point struktúra, ezért érték típusú, az értékek másolódnak.

pt1 = 20, 30; pt2 = 50, 30 lesz. A Pen viszont objektum, ezért referencia típusú, így a p2 változtatása p1-et is változtatja.

2. feladat 0 / 8 pont

Sorold fel a v tömb elemeit az alábbi metódus futása után, vesszővel elválasztva, szóköz nélkül! pl.: 1,2,3,4

```
public void Challenge3B()
{
    int[] v = new int[5];
    for (int i = 0; i < v.Length; i++)
    {
        v[i] = i + 1;
    }
    for (int i = 0; i < v.Length; i++)
    {
        try
        {
            v[i] *= v[^i];
        }
        catch
        {
            v[i] = 0;
        }
    }
}
```

A megoldás:  
0,10,12,48,50

Magyarázat a megoldáshoz

A v tartalma kezdetben: { 1, 2, 3, 4, 5 }. A v[^i] a C# 8.0-ban v[v.Length - i], ami i = 0 esetén IndexOutOfRangeException-t dob, amit elkapunk, így v[0] = 0. Majd v[1] = 2 \* 5 = 10, v[2] = 3 \* 4 = 12, v[3] = 4 \* 12 = 48, v[4] = 5 \* 10 = 50.



Ismertető a feladathoz

Sikerült mélyebbre jutnod a villa szervereiben. Már nem sok választ el attól, hogy megnyíljon az út az alagsorhoz, ahol a titkos tevékenységet rejtegeti Spencer. A szabotázsakciódhoz először kicserélsz két hasonló kódot egymással. Amit reményeid szerint nem fognak észrevenni, de a fő feladatod a liftet működtető firmware megbabrálása, hogy észrevétlenül lejuthass az alagsorba és az offline számítógépekből kinyerhesd az információkat.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 15 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 6 pont

Mit ír ki az alábbi látható kód a lefuttatása után?

```
public void Challenge4A()
{
    var counter = 0;
    var list = new List<int> { 1, 2, 3, 4, 5 };

    var filteredList = list.Where(x =>
    {
        ++counter;
        return x == 2;
    });

    Console.WriteLine($"{counter++}, {filteredList.First() + ++counter}")
}
```

- ☐ 5, 6
- ☒ 0, 6
- ☐ 6, 6
- ☐ 5, 8
- ☐ 3, 6
- ☐ 0, 5

Magyarázat a megoldáshoz

A Where csak a First() metódus futásakor aktiválódik, akkor a counter már 1, de 0-t ír ki a postfix operátor miatt. x-et kétszer növeljük a Where-ben, egyszer meg postfix módon, és 2-t adunk hozzá, így lesz 6 a második érték.

2. feladat 0 / 8 pont

Mit ír ki az alábbi kód a lefuttatása után?

```
public void Challenge4B()
{
    var counter = 0;
    var list = new List<int> { 1, 2, 3, 4, 5 };

    var filteredList = list.Where(x =>
    {
        ++counter;
        return x == 2;
    });

    Console.WriteLine($"{counter++}, {filteredList.Count() + ++counter}")
}
```

- ☐ 5, 6
- ☐ 5, 8
- ☐ 6, 6
- ☒ 0, 8
- ☐ 3, 6
- ☐ 0, 5

Magyarázat a megoldáshoz

A Where csak a Count() metódus futásakor aktiválódik, akkor a counter már 1, de 0-t ír ki a postfix operátor miatt. x-et kétszer növeljük a Where-ben, egyszer meg postfix módon, és 5-öt adunk hozzá (a lista hossza), így lesz 8 a második érték.

3. feladat 0 / 6 pont

Mit ír ki az alábbi program a konzolra?

```
static void Main(string[] args)
{
    var st = new Szamok();
    int szam = 9;
    ref var x = ref st.SzamKeres(szam);
    x *= 2;
    Console.WriteLine(st);
    Console.ReadKey();
}

class Szamok
{
    int[] num = { 1, 4, 8, 15, 31 };

    public ref int SzamKeres(int t)
    {
        for (int i = 0; i < num.Length; i++)
        {
            if (num[i] >= t)
                return ref num[i];
        }
        return ref num[0];
    }

    public override string ToString() => string.Join(" ", num);
}
```

A megoldás:  
1 4 8 30 31

Magyarázat a megoldáshoz

Az osztályban lévő függvény a paraméterként megadott számnál (itt 9) nagyobb vagy azzal egyenlő elemének referenciáját adja vissza. Ez az adott tömbben a 15, a referenciaként átadás miatt a tömbben lévő elemet duplázza meg, így lesz 30.

Ismertető a feladathoz

A liftet sikerült átprogramoznod, így már a te belépési szintteddel is sikerült lejutnod az alagsorba. Éjfél után nem sokkal mész le az alagsorba, hogy csak az éjszakai őrseget kelljen kicselezned. Lent szörnyű dolgokat találsz. Az eltűnt embereket cellákban tartják és kísérleteket végeznek rajtuk. Sikerül észrevétlenül eljutnod egy irodáig, ahol egy elektronikus széfet találsz. Itt találsz egy offline számítógépet, és egy Linux pendrive segítségével sikerül belenézned a rajta található kódokban. Találsz egy sor különös kódrészletet, aminek segítségével elérheted a páncélszekrényen futó firmware-t. Hogy jobban megértsd a működését, szemügyre veszed a kódot.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 20 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 10 pont

Mennyi ideig fut az alábbi kódrészlet (*XUnit-ből futtatva*)?

```
[Fact(Timeout = 4000)]
public async static Task Challenge5A()
{
    var ct = new CancellationTokenSource(TimeSpan.FromSeconds(2)).Token;

    var t = new List<int>();
    await Task.Run(() =>
    {
        while (true)
        {
            Task.Delay(300);
        }
    }, ct);
}
```

- ☐ Örökké fut
- ☒ 4 mp-ig fut
- ☐ 4000 mp-ig fut
- ☐ 2 s-ig fut
- ☐ 2 ms-ig fut
- ☐ 2100 ms-ig fut
- ☐ 4200 ms-ig fut

Magyarázat a megoldáshoz

A CancellationToken nem szakítja meg a futást 2 másodperc után, úgyhogy a Fact opciójában megadott 4000 ms, azaz 4 mp után lövi le a futást.

2. feladat 0 / 12 pont

Add meg vesszővel elválasztva, szóköz nélkül, hogy milyen számok találhatóak sorrendben a t listában az alábbi kód futásának végén!

```
public async Task Challenge5B()
{
    var are = new AutoResetEvent(true);
    var t = new List<int>();
    var t1 = Task.Run(() =>
    {
        while (t.Count < 3)
        {
            are.WaitOne();
            t.Add(t.Count());
            are.Set();
        }
    });

    var t2 = Task.Run(() =>
    {
        while (t.Count < 5)
        {
            are.WaitOne();
            t.Add(t.Count());
            are.Set();
        }
    });
    Task.WaitAll(t1, t2);
}
```

A megoldások:  
0,1,2,3,4  
01234

Magyarázat a megoldáshoz

A két taszk egymást blokkolja, amíg az egyik véget nem ér, és mindkettő a tömb aktuális hosszát teszi a listába (0-tól kezdve), így eggyel növeli minden lépésben annak hosszát.

3. feladat 0 / 14 pont

Az alábbiak közül mely(ek) NEM lehet(nek) az alábbi függvény eredménye(i)?

```
public static string Challenge5C()
{
    Random r = new Random();
    string res = "";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < r.Next(2, 4); j++)
        {
            res += r.Next(100) < 80 ? r.Next(10).ToString() : ((char)('A' + r.Next(26)));
        }
        res += i != 2 ? "-" : "";
    }
    return res;
}
```

- ☐ 11-AA-B7
- ☒ 93A-ABF6-D9
- ☒ AB-CDE-EFG-H1
- ☐ 001-002-03
- ☒ R2-D2
- ☐ 22-938-283
- ☒ A8-7[A-A9

Magyarázat a megoldáshoz

A függvény x-y-z alakú kifejezést állít elő, ahol x, y, z is 2 vagy 3 darab számjegyből vagy az angol ábécé nagybetűjéből áll. "93A-ABF6-D9": ebben van egy 4 hosszú csoport, "AB-CDE-EFG-H1": ez négy részből áll, "R2-D2": ez csak két részből, "A8-7[A-A9": ebben pedig a '[' nem jöhet ki, ezért egyik sem lehetséges.

Ismertető a feladathoz

A kód bonyolultabb, mint gondoltad, egyéb részeit is szemügyre kell venned. Ahhoz, hogy letöltsd a páncélszekrény firmware-ét és kinyisd azt, valamiféle API-jellegű kommunikáció történik. Ha megérted ezen kódok működését, végre hozzáférhetsz a páncélszekrény kódjához is.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 25 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 18 pont

Az alábbi szervizen meghívják 5 alkalommal a Counter API-t GET metódussal. Sorold fel **vesszővel elválasztva, szóköz nélkül**, hogy milyen 5 db visszatérési érték keletkezik? (Az AddMyService beregisztrálásra került a program indulásakor.)

```
[ApiController]
[Route("[controller]")]
public class CounterController : ControllerBase
{
    private readonly CounterService _counterService;
    public CounterController(CounterService counterService)
    {
        _counterService = counterService;
    }
    [HttpGet]
    public IActionResult Get()
    {
        return Ok(_counterService.Counter);
    }
}

public class CounterService
{
    private ulong _counter;
    public ulong Counter { get => ++_counter; }
}

public static class Wireup
{
    public static IServiceCollection AddMyService(this IServiceCollection s)
    {
        return s.AddScoped<CounterService>();
    }
}
```

A megoldás:  
1,1,1,1,1

**Magyarázat a megoldáshoz**

A Scoped regisztrálás miatt minden futtatás alkalmával új példány jön létre a szervizből, ezért a \_counter értéke mindig 1 lesz.

2. feladat 0 / 18 pont

Hány olyan különböző socket lesz maximum egyazon időben, amelyek a kapcsolatnak **bármelyik oldalán** nyitva maradnak az alábbi program futása során egy **4 magos processzorral** rendelkező számítógépen?

```
public async void Challenge6B()
{
    Parallel.For(0,10,async (p)=>{
        using var client = new HttpClient();
        System.Console.WriteLine((await client.GetAsync("https://www...
    });
}
```

A megoldások:  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21

**Magyarázat a megoldáshoz**

Bár a using-ból egyszerre legfeljebb négy darab fut a Parallel.For miatt, azonban a HttpClient nem zárja be rögtön a socketet, alapesetben még két percig nyitva maradnak.

<https://docs.microsoft.com/en-us/dotnet/api/system.net.http.httpclient?view=net-5.0#remarks>

<https://aspnetmonsters.com/2016/08/2016-08-27-httpclientwrong/>

Ezek alapján el tudjuk fogadni nem csak a 10-et, hanem bármely értéket 10 és 21 között.



Ismertető a feladathoz

Sikerült letöltened a széf kódját! Már hallod a lifttel leérkezni az éjjeli őrséget, nem sok időd van, hogy kinyisd a pánccs szekrényt és kimenekülj az információkkal. A kód zömét sikerült nagyon hamar megértened, de egyetlen próbálkozásra van már csak időd! Átírva a kódot, utoljára ellenőrzöd azt, mielőtt felöltened a széfre.

Amennyiben a kód módosítása sikeres volt: a széf nesztelenül kinyílik. Egy köteg dokumentum van benne néhány lezárt kémcsővel, amik valamilyen ismeretlen vegyületet tartalmaztak. Ezeket épp volt időd eltenni, és a széfet visszazárni. Amíg az éjjeli őrök a cellákat ellenőrizték, te sikeresen visszajutottál a lifting és anélkül, hogy meghallották volna, kijutottál a villából. A hatóságoknak még aznap átadtad az iratokat. Védőőrizetben vártad meg, amíg felszámolták Oswald E. Spencer illegális kutatásait.

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 30 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 20 pont

Az alábbiak közül mi(k) lehet(nek) az alábbi program futásának eredménye(i)?

```
public class Program
{
    private static int _cycle = int.MaxValue;
    public static void Main(string[] args)
    {
        var printers = new List<Action>();
        var derived = new Derived(2);

        var i = 0;

        for (; i < _cycle--; i++)
        {
            printers.Add(delegate { derived.Print(i); });
        }

        foreach (var printer in printers)
        {
            printer();
        }

        Console.WriteLine(derived.Result);
    }

    public class Base
    {
        public decimal Result = 0;
        protected decimal Factor = new decimal(0.1);

        protected Base(int factor) { }
        protected Base(double factor) : this(new decimal(factor)) { }
        protected Base(decimal factor) => Factor = factor;
        public void Print(int num) { Result += --num; }
    }

    public class Derived : Base
    {
        public Derived(decimal factor) : base(factor) { }
        public Derived(int factor) : base(factor) { }
        public Derived(double factor) : base(new decimal(factor)) { }
        public new void Print(int num) { Result = num * Factor; }
    }
}
```

- ☒ OutOfMemoryException
- ☐ int.MaxValue
- ☐ int.MaxValue / 2
- ☒ int.MaxValue / 20
- ☐ 0

Magyarázat a megoldáshoz

A főprogram ciklusában az i változó minden lépésben eggyel növekszik, a \_cycle pedig eggyel csökken, így int.MaxValue / 2 alkalommal fut csak le. A Derived osztály az ős konstruktorát hívja egész értékkel (2), azaz ott kód nem fut le, de a Factor értéke 0.1 lesz. Az utolsó delegált lefutásakor a Result értéke tehát int.MaxValue / 2 \* 0.1, azaz int.MaxValue / 20 lenne, ez jelenne meg a képernyőn, azonban nagy valószínűséggel előbb elfogy a memória, így az OutOfMemoryException is helyes válasz.

2. feladat 0 / 20 pont

A széf kinyitásához egy utolsó számkombinációra van szükség, amit az alábbi program ír ki a konzolra. Mi az?

```
public class Program: IDisposable
{
    public void Dispose()
    {
        Console.WriteLine("15");
    }

    static Program()
    {
        Console.WriteLine("20");
    }

    public static void Main(string[] args)
    {
        using (var pr = new Program())
        {
            Console.WriteLine("10");
        }

        Console.ReadLine();
    }
}
```

A megoldások:  
201015  
20,10,15

Magyarázat a megoldáshoz

Először a statikus konstruktor fut le, ezután jön létre a Program új példánya, végül a Dispose akkor fut le, amikor a program végén az objektum megsemmisül.