Odpovědi pište na zvláštní odpovědní list s vaším jménem a fotografií. Pokud budete odevzdávat více než jeden list s řešením, tak se na 2. a další listy nezapomeňte podepsat. Do zápatí všech listů vždy napište i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

Společná část pro otázky označené X

Předpokládejte následující program:

```
A1? a1 = new E1();
Console.WriteLine(a1.f());

I1? i1 = a1;
Console.WriteLine(i1.f());

Console.WriteLine(a1 is B1);
a1 = null;
Console.WriteLine(a1 is A1);

interface I1 {
    public char f();
}

class A1 : I1 {
    public virtual char f() => 'A';
}

class B1 : A1 {
    public override char f() => 'B';
}

class C1 : B1 {
    public virtual char f() =>
        (char) (base.f() + 10);
}

class D1 : C1, I1 {
}

class E1 : D1 {
    public new char f() =>
        (char) (base.f() + 5);
}
```

Otázka č. 1 (X)

Co uvedený program vypíše na 1. řádku výstupu? Detailně vysvětlete proč – jako součást vysvětlení nakreslete VMT tříd `A1`, `B1`, `C1`, `D1`, `E1`.

Otázka č. 2 (X)

Co uvedený program vypíše na 2. řádku výstupu? Detailně vysvětlete proč – jako součást vysvětlení nakreslete interfacovou tabulku metod pro interface `I1` u tříd `A1`, `B1`, `C1`, `D1`, `E1`.

Otázka č. 3 (X)

Co uvedený program vypíše na 3. a na 4. řádku výstupu? Detailně vysvětlete proč.

Otázka č. 4

Co níže uvedený program vypíše? Detailně vysvětlete proč!

```
E e1 = E.One;
Console.WriteLine(e1);
e1++;
Console.WriteLine(e1);

enum E { One = 1, Four = 4, Five };
```

Společná část pro otázky označené Y

Předpokládejte následující program:

```
interface I5 {
    public int X { get; set; }
}

struct S(int a, int b, int c) : I5 {
    public int X { get; set; } = a;
    public int Y { get; } = a;

    public int f() {
        return a + b + c;
    }
}

class Program {
    private static void Update(I5 i) {
        i.X *= 10;
    }

    private static void Main(string[] args) {
        S s1 = new S(11, 12, 13);
        I5 i1 = s1;
        Update(i1);
        Console.WriteLine(s1.X);
        Console.WriteLine(i1.X);

        S[] sa = new S[3];
        Console.WriteLine(sa[0].X);
    }
}
```

Otázka č. 5 (Y)

Detailně vysvětlete, jaká je velikost proměnné `s1` a jakou má v paměti strukturu, jaká je velikost instance `S`, a jak instance `S` souvisí se samotnou proměnnou `s1`.

Otázka č. 6 (Y)

Co uvedený program vypíše na 1. a na 2. řádku výstupu? Detailně vysvětlete proč.

Otázka č. 7 (Y)

Co uvedený program vypíše na 3. řádku výstupu? Detailně vysvětlete proč.

Otázka č. 8

Předpokládejte následující program:

```
A? a2, a3;
A? a1 = new A {
    Id = 1,
    Next = a2 = new A {
        Id = 2,
        Next = a3 = new A {
            Id = 3
        }
    }
};

A? a4 = new A { Id = 4, Next = a2 };
A? a5 = new A { Id = 5, Next = a2 };
a3.Next = a5;

// Not interested in few instances anymore:

a1 = null;
a2 = null;
a3 = null;
a4 = null;

// Force 1st GC gen0:
GC.Collect(0);

// Force 2nd GC gen0:
GC.Collect(0);

// Force 3rd GC gen0:
GC.Collect(0);

Console.WriteLine(a5.Id);

record class A {
    public int Id { get; init; }
    public A? Next { get; set; }
}
```

Předpokládejte, že volání `GC.Collect(0)` vynutí běžné provedení GC **gen0**, a neskončí do kompletního dokončení běhu GC. Dále pro jednoduchost předpokládejte, že jiné než v kódu explicitně viditelné alokace na GC haldě **neprobíhají**, a že JIT **neprovádí žádné** optimalizace.

Odhlédněte od rozdělení haldy na regiony nebo segmenty, berte ji pro jednoduchost pouze jako jeden lineární souvislý úsek virtuální paměti. Pro volací zásobník předpokládejte, že roste k nižším adresám.

Nakreslete stav haldy a zásobníku těsně **před 1.** voláním `GC.Collect(0)` s využitím následující notace:

- Každou lokální proměnnou nakreslete jednoduše jako obdélníček se jménem, ze kterého vede šipka, nebo je v něm `null`.
- Každý objekt na haldě nakreslete jen zjednodušeně jako obdélníček s hodnotou `Id`, a se šipkou vedoucí z `Next` nebo s `null` (tj. ignorujte overhead spojený s objekty na haldě).

Dále pro každý objekt označte, v jaké generaci bude:

a) **před 1.** voláním `GC.Collect(0)`,
b) **mezi 1. a 2.** voláním `GC.Collect(0)`,
c) **mezi 2. a 3.** voláním `GC.Collect(0)`,
d) a **po 3.** volání `GC.Collect(0)`.

Společná část pro otázky označené Z

Předpokládejte, že implementujeme zjednodušený farming simulátor inspirovaný hrou *Stardew Valley*. Ve svém řešení co nejefektivněji použijte možností jazyka C#, a pečlivě využívejte vhodné jazykové konstrukty pro vyjádření vašeho záměru v návrhu architektury.

Otázka č. 9 (Z)

Napište implementaci základního, ale současně co nejlépe rozšiřitelného, udržovatelného a testovatelného, objektového návrhu herních objektů zapojitelných do uvedené hry. Váš návrh by měl splňovat všechna následující pravidla:

- Hra probíhá realtime, ale v pravidelných intervalech nastává událost *„konec dne“* – chceme mít možnost, aby **každý** objekt ve hře mohl při konci dne provést nějakou akci (ale nemusí chtít provést žádnou).
- Ve hře chceme mít typ `Rabbit` pro králíky:

  a typy `WhiteChicken` a `BrownChicken` pro bílé a hnědé slepice:

- Předpokládejte, že do budoucna **nebudeme** přidávat zvláštní druhy **králíků**, ale **budeme** přidávat další druhy **slepic**.
- **Každé** zvíře ve hře můžeme nakrmit (*feed*) – krmení je pouze provedení operace bez parametrů. Opakované krmení ve stejném dni nemá žádný efekt.
- **Každé** zvíře ve hře má jako svůj vnitřní stav, zda je dospělé (*mature*) nebo nikoliv. Králík se stává dospělým, pokud byl nakrmen v 6 souvislých dnech po sobě. Slepice **každého druhu** se stává dospělou, pokud byla nakrmena ve 3 souvislých dnech po sobě.

Otázka č. 10 (Z)

Doplňte vaše **řešení otázky 9** o následující:

- **Některé** typy objektů ve hře je možné prodávat. Pro konkrétní objekt „prodejného“ typu můžeme vždy zjistit jeho aktuální prodejní cenu (danou pouze stavem a typem objektu).
- **Každé** zvíře ve hře je možné vždy prodat.
- Cena za dospělého králíka je vždy 1 tisíc jednotek, cena za dospělou bílou slepici je vždy 2 tisíc jednotek, cena za dospělou hnědou slepici je vždy 3400 jednotek.
- Cena za nedospělé zvíře je **vždy polovina** ceny za dospělé zvíře.
