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).

Otázka č. 1

Předpokládejte následující kód, který se bez chyb přeloží:

```
class Prg1 {
  public int A { get; init; }
  private int B { get; init; }
  internal int C { get; init; }

  class Helper {
    public void Process(Prg1 p) { }
  }

  static void _Main() {
    var prg = new Prg1();
    new Helper().Process(prg);
} }
```

Které vlastnosti objektu třídy `Prg1` může číst metoda `Helper.Process`, pokud ho získá skrz argument `p`? A do kterých vlastností může takto zapisovat? Vysvětlete proč.

[1 bod]

Otázka č. 2

Předpokládejte následující kód:

```
public static class Utils {
   public static IEnumerable<string>
      Range(int from, int to) {
         for (int i = from; i <= to; i++) {
            yield return new string('+', i);
         }
   }
}
```

Naprogramujte co nejvhodnější implementaci třídy `Utils` bez použití konceptu `yield return`, ale se zachováním stejného API.

[1 bod]

Otázka č. 3

Předpokládejte následující dvě metody:

```
static void P1(IList<object> list) {
  foreach (var item in list) Console.Write(item);
}
static void P2<T>(IList<T> list) {
  foreach (var item in list) Console.Write(item);
}
```

(A) Je mezi uvedenými metodami nějaký praktický rozdíl? Lze obě využít ve stejných scénářích? Je jejich implementace stejně efektivní? Vše detailně vysvětlete.

[1 bod]

(B) Jsou dané metody vzhledem ke své funkci deklarovány nejvhodněji nebo by bylo lepší nějak změnit jejich parametry? Vše detailně vysvětlete.

[1 bod]

Otázka č. 4

Předpokládejte, že chceme naimplementovat následující třídu `MyManualResetEventSlim`, jejíž základní API odpovídá standardnímu chování třídy `System.Threading.ManualResetEventSlim`:

```
public class MyManualResetEventSlim {
   /// <summary>Initializes a new instance of the
   /// class with an initial state of nonsignaled
   /// </summary>
   public MyManualResetEventSlim() {
   }

   public void Set() { }

   public void Reset() { }

   public void Wait() { }
}
```

Napište implementaci takové třídy `MyManualResetEventSlim` bez použití `ManualResetEventSlim` i bez použití `WaitHandle` a jejích libovolných potomků.

[1 bod]

Otázka č. 5

Immutable struktura `System.Drawing.Color` má pouze bezparametrický konstruktor a následující factory metody:

```
public static Color FromArgb(
  int red, int green, int blue
)
public static Color FromArgb(
  int alpha, int red, int green, int blue
)
```

a veřejné getter-only `byte` vlastnosti `A`, `R`, `G`, `B`.

Doplňte následující zdrojový soubor tak, aby se správně přeložil a výsledný program po spuštění vypsal (počítejte s tím, že struktura `Color` má již vhodně overridovanou metodu `ToString`) na standardní výstup `Color [A=105, R=217, G=0, B=13]`. **POZOR: máte zakázáno měnit tělo metody `Main`, její obsah musí zůstat přesně takový, jak je zapsáno níže**:

```
class Prg5 {
  static void Main() {
    Color c = 217.AsRed()
      .With(13).Blue()
      .With(105).Alpha();
    Console.WriteLine(c);
  }
}
```

[1 bod]

Otázka č. 6

Předpokládejte následující program:

```
using C = System.Console;

interface IW {
    void m1(); void m2(); void m3();
}
class X : IW {
    public virtual void m1() { C.WriteLine(11); }
    public virtual void m2() { C.WriteLine(12); }
    public void m3() { C.WriteLine (13); }
}
class Y : X, IW {
    public virtual void m1() { C.WriteLine(21); }
    public override void m2() { C.WriteLine(22); }
    public virtual void m3() { C.WriteLine(23);  }
    void IW.m3() { C.WriteLine(-23);  }
}
class Z : Y {
    public override void m1() { C.WriteLine(31); }
    public override void m3() { C.WriteLine(33); }
}
class Prg6 {
    static void Main() {
        IW iw = new Z();
        X x = (X) iw;
        ((IW) x).m1();
        ((IW) x).m2();
        ((IW) x).m3();
}   }
```

Detailně vysvětlete (nakreslete tabulky VMT), co a proč vypíše po svém spuštění.
[1 bod]

Otázka č. 7

Předpokládejte následující třídu, která se bez chyb přeloží:

```
class Prg7 {
  static Func<double, Func<double, double>>
  Add(int c)
  {
     return
        x => (y => --x + ++y + (c *= 10));
  }

  static void Main() {
     int a = 1;
     int b = 10;
     var F = Add(4);
     var G = F(a);

     G(b);
     G(b);

     Console.WriteLine(G(b));
} }
```

Je chování takového kódu jasně definované, nebo mohou různé překladače C# v této situaci generovat zcela rozdílný kód? Pokud se kód vždy chová stejně, tak napište, co program vypíše na standardní výstup. Chování programu detailně vysvětlete, a vaše odpovědi zdůvodněte.
[1 bod]

Otázka č. 8

Předpokládejte následující kód, který se bez chyb přeloží:

```
class Prg8 {
  static void Main() {
    var list = new List<int> { 2, 9, 4, 1 };
    var ordered = from x in list
                  orderby x select x;
    for (int i = 0; i < ordered.Count(); i++) {
      Console.WriteLine(
        $"{list[i]} > {ordered.Skip(i).First()}"
      );
} } }
```

Co program vypíše na standardní výstup? Je daná implementace nejvhodnější? Pokud ne, tak napište, jak byste ji vylepšili (přepsali). Vše detailně vysvětlete.
[1 bod]

Otázka č. 9

Předpokládejte následující kód, který je součástí nějaké třídy, která se bez chyb přeloží:

```
class Exchange {
  public string Id { get; init; }
  public string Url { get; init; }
  public Func<string, decimal> ParsePrice {
    get; init;
} }

static HttpClient s_httpClient;

static async Task<decimal> GetAvgRateAsync(
  List<Exchange> list)
{
  Monitor.Enter(list);
  decimal sum = 0;
  foreach (var ex in list) {
    string json =
      await s_httpClient.GetStringAsync(ex.Url);
    sum += ex.ParsePrice(json);
  }
  var avg = sum / list.Count;
  Monitor.Exit(list);
  return avg;
}
```

K předanému seznamu mohou paralelně přistupovat jiná vlákna – všechna taková vlákna používají pro vzájemné vyloučení přístupu k seznamu zámek přímo na instanci `List<Exchange>`. Je kód uvedené funkce správně (i vzhledem k tomu, kdy jaká vlákna vykonávají části uvedeného kódu)? Pokud získáme seznam odkazující se na řádově jednotky různých webových služeb (každá na jiném serveru), je uvedená implementace nejefektivnější? Vše detailně vysvětlete. Pokud řešení není správné nebo efektivní, tak načrtněte, jak ho opravit.
Poznámka: Pokud neznáte třídu `HttpClient`, tak funguje podobně jako `WebClient`, tj. umožňuje připojení k webovému serveru přes protokol HTTP, a stáhnout z něj data na základě požadavku. Třída je thread-safe, a umožňuje spuštění více paralelních požadavků dalšími voláními metody `GetStringAsync` z libovolného vlákna před dokončením operací vyžádaných předchozími voláními `GetStringAsync`.
[1 bod]