Task 1

Assume you have the following method including some complex code α processing the `resultLines` data and writing it into the file `Out.txt`:

```
void PrintResults(List<string> resultLines) {
    using (var sw = new StreamWriter("Out.txt")) {
        α
    }
}
```

(A) Can we rewrite the function to take advantage of the C# 8.0 `using` command – i.e. defining the variable without need to enclose the code α into nested curly brace block? Explain why and the rewrite is possible, then do it.
[1 point]

(B) Rewrite the method code without any variant of the `using` block or statement. Behavior of the new method variant should be exactly the same as the provided one in all possible scenarios.
[1 point]

Task 2

Explain what (if any) semantical difference is between `ArgumentOutOfRangeException` and `IndexOutOfRangeException` exception types? Provide an example of a scenario where you would use the one and the other.
[1 point]

Task 3

Assume we have the following method, that will be called in part of our web server application – the `name` argument value is received from client web browser and contains exactly the text user entered into an editable field in our web application:

```
void PrintName(TextWriter writer, string name) {
    writer.WriteLine(
        "{0}: user " + name + "!", DateTime.Now);
}
```

Explain if and why it is a good implementation of such method. If there some potential problems in the implementation, then explain them.
[1 point]

Task 4

Compare string concatenation (using + operator) to using the `StringBuilder` class – explain advantages and disadvantages.
[1 point]

Task 5

Explain what is CIL code, and what is its purpose in the context of the .NET platform. How does it relate to the C# language?
[1 point]

Task 6

Assume you have the following program:

```
class A {
    public A(int argX) {
        Console.WriteLine("0");
    }
}

class B : A {
    int y = P.m("1");

    public B()
        : this(P.m("2"))
    {
        Console.WriteLine("3");
    }

    public B(int x)
        : base(P.m("4")) {
        Console.WriteLine("5");
    }

    static B() {
        Console.WriteLine("6");
    }
}

class P {
    static void Main(string[] args) {
        Console.WriteLine("7");
        B b = new B();
    }

    public static int m(string msg) {
        Console.WriteLine(msg);
        return 1;
    }
}
```

(A) What text will the program print on the standard output? Explain why.
[1 point]

Task 7

Assume you have the following program:

```
class Prg6 {
    static void Main(string[] args) {
        int j = 0;
        for (int i = int.MaxValue - 10;
             i <= int.MaxValue; i++) {
            j++;
        }
        Console.WriteLine(j);
    }
}
```

The program will never terminate. Would it be possible to modify it, so that the primary cause of the infinity loop is "automatically" detected and "converted" into an exception? Explain how, or why not.
[1 point]