

Entregar **una solución** con **una única aplicación de consola**, desde la que se **invoquen todos los métodos pedidos** y **muestre el resultado** de los cálculos realizados por la pantalla.

### Ejercicio 1 (1,5 puntos)

**Método extensor** de IEnumerable que calcule la **diferencia** entre dos colecciones representando conjuntos, usando una implementación **lazy**. La diferencia entre A y B es el conjunto de los elementos de A que no están en B. B sería un parámetro en la llamada. Por ejemplo, si las colecciones contienen {15, 18, 6, 14, 1, 7, 8, 2, 3, 0} y {1, 2, 6, 7, 8} el resultado sería {15, 18, 14, 3, 0}. **NOTA:** el uso del método predefinido Except **no** está permitido.

### Ejercicio 2 (1,75 puntos)

Implementar un generador que reciba como parámetro un real, x. Devuelve una **secuencia potencialmente infinita** con las potencias de x de exponente 0, 1, 2, etc. Implementar de modo que se actualice el estado del generador realizando el mínimo número de cálculos. Usar dicho generador para implementar una función que calcula, usando un bucle adecuado,  $\sum_{i=n}^m \left(\frac{1}{x^i}\right)$  siendo n, m enteros, x real. Probar con los valores n=1, m=3, x=2.0, se realizaría el cálculo  $\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} = 0.875$

### Ejercicio 3 (2,25 puntos)

Partir del ejemplo de cálculo del módulo de un vector con el diseño *Master/Worker*. Modificar el proyecto de modo que se calcule la **intersección de dos vectores de enteros**. La intersección de dos vectores de enteros a y b es un vector en el que están los elementos de a que también están en b. La colección en la que se almacena y devuelve el resultado debe de ser un recurso compartido entre master y workers. Ejemplo, si las colecciones son las mismas que en el caso del ejercicio 1, el resultado sería {6, 1, 7, 8, 2}. Usar las mencionadas colecciones en la invocación, con 1 y 4 hilos. **NOTA:** el uso del método predefinido Intersect **no** está permitido.