

Comparisons Lab

Lab 1: SequenceEqual() Method

Open the **Program.cs** file and replace the code with the following:

```
using LINQLab.EntityClasses;
using LINQLab.RepositoryClasses;

// Declare variables and fill data
List<Song> songs1 = SongRepository.GetAll();
List<Song> songs2 = SongRepository.GetAll();
bool value = false;

// TODO: Write Your Query Here

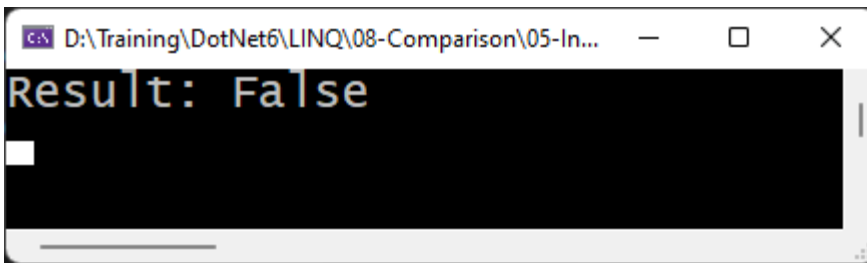
// Display the Results
Console.WriteLine($"Result: {value}");

// Pause for results
Console.ReadKey();
```

Write a query to compare these two song collections using the `SequenceEqual()` method.

Try it Out

Run the application and view the results:



Lab 2: Sequence Equal with Comparer

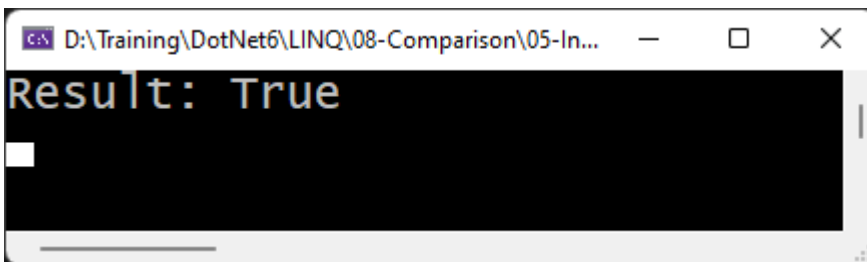
Open the **Program.cs** file and add an instance of a `SongComparer` class.

```
SongComparer sc = new();
```

Write a query to use the `SongComparer` class with the `SequenceEqual()` method to compare the ***SongId*** properties from the two collections.

Try it Out

Run the application and view the results:



Lab 3: Except() Method

Open the **Program.cs** file and replace the code with the following:

```

using LINQLab.EntityClasses;
using LINQLab.RepositoryClasses;

// Declare variables and fill data
List<Song> songs = SongRepository.GetAll();
List<MusicGenre> genres = MusicGenreRepository.GetAll();
List<int> list = new();

// TODO: Write Your Query Here

// Display the Results
foreach (var value in list) {
    Console.WriteLine(value);
}

// Display Count
Console.WriteLine($"Result: {list.Count}");

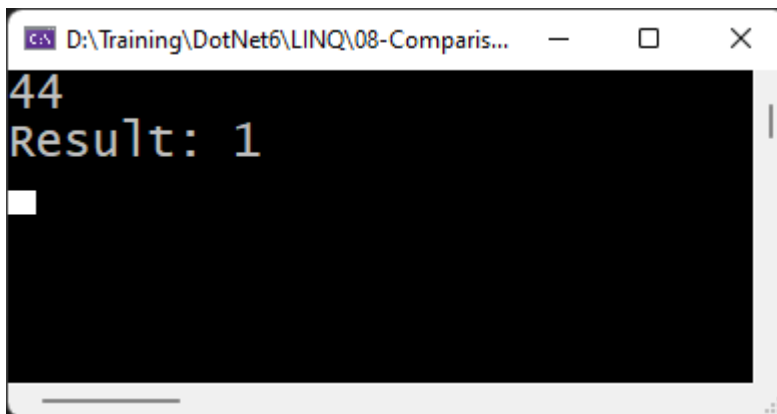
// Pause for results
Console.ReadKey();

```

Write a query to select all **GenreId** properties that are not assigned to any songs.

Try it Out

Run the application and view the results:



Lab 4: Except using Comparer

Open the **Program.cs** file and replace the code with the following:

```
using LINQLab.EntityClasses;
using LINQLab.RepositoryClasses;

// Declare variables and fill data
List<Song> songs1 = SongRepository.GetAll()
    .Where(row => row.TrackNumber == "3").ToList();
List<Song> songs2 = SongRepository.GetAll()
    .Where(row => row.GenreId == 30).ToList();
SongComparer sc = new();
List<Song> list = new();

// TODO: Write Your Query Here

// Display the Results
foreach (var value in list) {
    Console.WriteLine(value);
}

// Display Count
Console.WriteLine($"Result: {list.Count}");

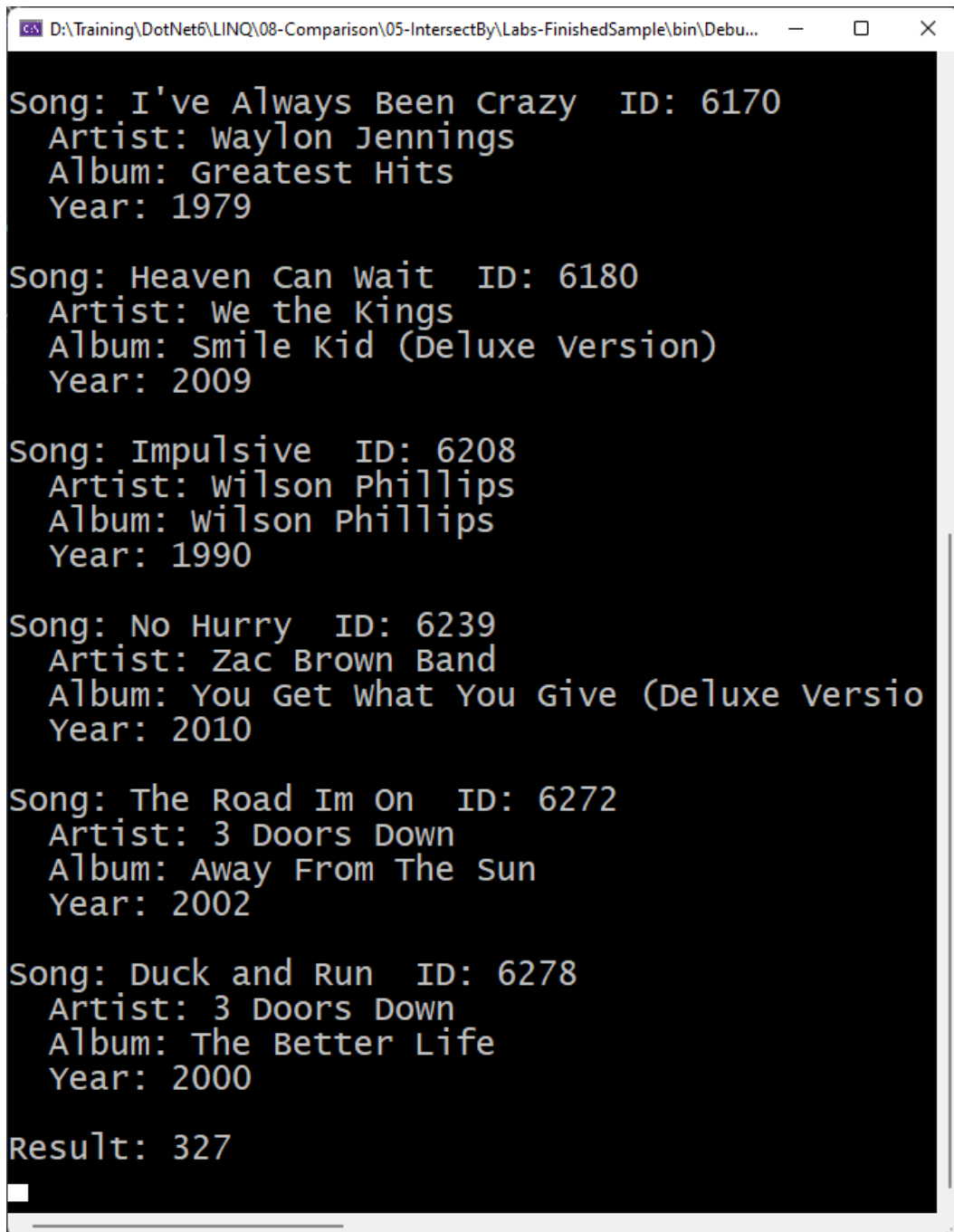
// Pause for results
Console.ReadKey();
```

The above code creates one variable to hold all songs where the **TrackNumber** property is equal to 3. It creates another variable to hold all songs where the **GenreId** property is equal to 30.

Write a query using the **Except()** method and the **SongComparer** class to determine those songs that have **TrackNumber** equal to 3, and **GenreId** not equal to 30.

Try it Out

Run the application and view the results:



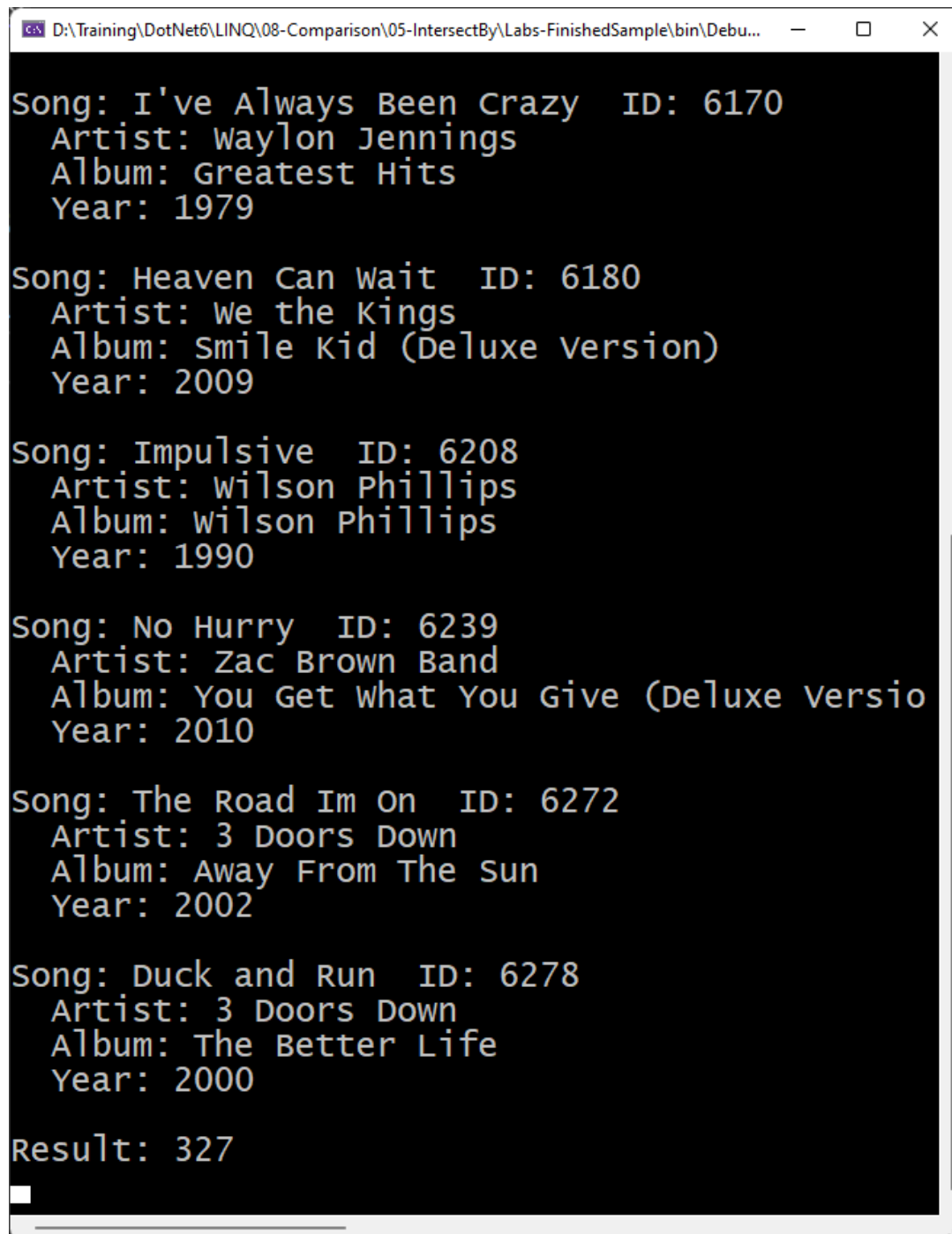
```
D:\Training\DotNet6\LINQ\08-Comparison\05-IntersectBy\Labs-FinishedSample\bin\Debu...  
Song: I've Always Been Crazy ID: 6170  
Artist: Waylon Jennings  
Album: Greatest Hits  
Year: 1979  
  
Song: Heaven Can Wait ID: 6180  
Artist: We the Kings  
Album: Smile Kid (Deluxe Version)  
Year: 2009  
  
Song: Impulsive ID: 6208  
Artist: Wilson Phillips  
Album: Wilson Phillips  
Year: 1990  
  
Song: No Hurry ID: 6239  
Artist: Zac Brown Band  
Album: You Get What You Give (Deluxe Versio  
Year: 2010  
  
Song: The Road Im On ID: 6272  
Artist: 3 Doors Down  
Album: Away From The Sun  
Year: 2002  
  
Song: Duck and Run ID: 6278  
Artist: 3 Doors Down  
Album: The Better Life  
Year: 2000  
  
Result: 327
```

Lab 5: ExceptBy() Method

Write a query to use the ExceptBy() method to determine those songs that have **TrackNumber** equal to 3 and **GenreId** not equal to 30.

Try it Out

Run the application and view the results:



```
D:\Training\DotNet6\LINQ\08-Comparison\05-IntersectBy\Labs-FinishedSample\bin\Debu...  
Song: I've Always Been Crazy ID: 6170  
Artist: waylon Jennings  
Album: Greatest Hits  
Year: 1979  
  
Song: Heaven Can Wait ID: 6180  
Artist: we the Kings  
Album: smile Kid (Deluxe version)  
Year: 2009  
  
Song: Impulsive ID: 6208  
Artist: wilson Phillips  
Album: wilson Phillips  
Year: 1990  
  
Song: No Hurry ID: 6239  
Artist: Zac Brown Band  
Album: You Get What You Give (Deluxe Versio  
Year: 2010  
  
Song: The Road Im On ID: 6272  
Artist: 3 Doors Down  
Album: Away From The Sun  
Year: 2002  
  
Song: Duck and Run ID: 6278  
Artist: 3 Doors Down  
Album: The Better Life  
Year: 2000  
  
Result: 327
```

The results are the same as the last lab but was accomplished without using the SongComparer class.

Lab 6: Intersect() Method

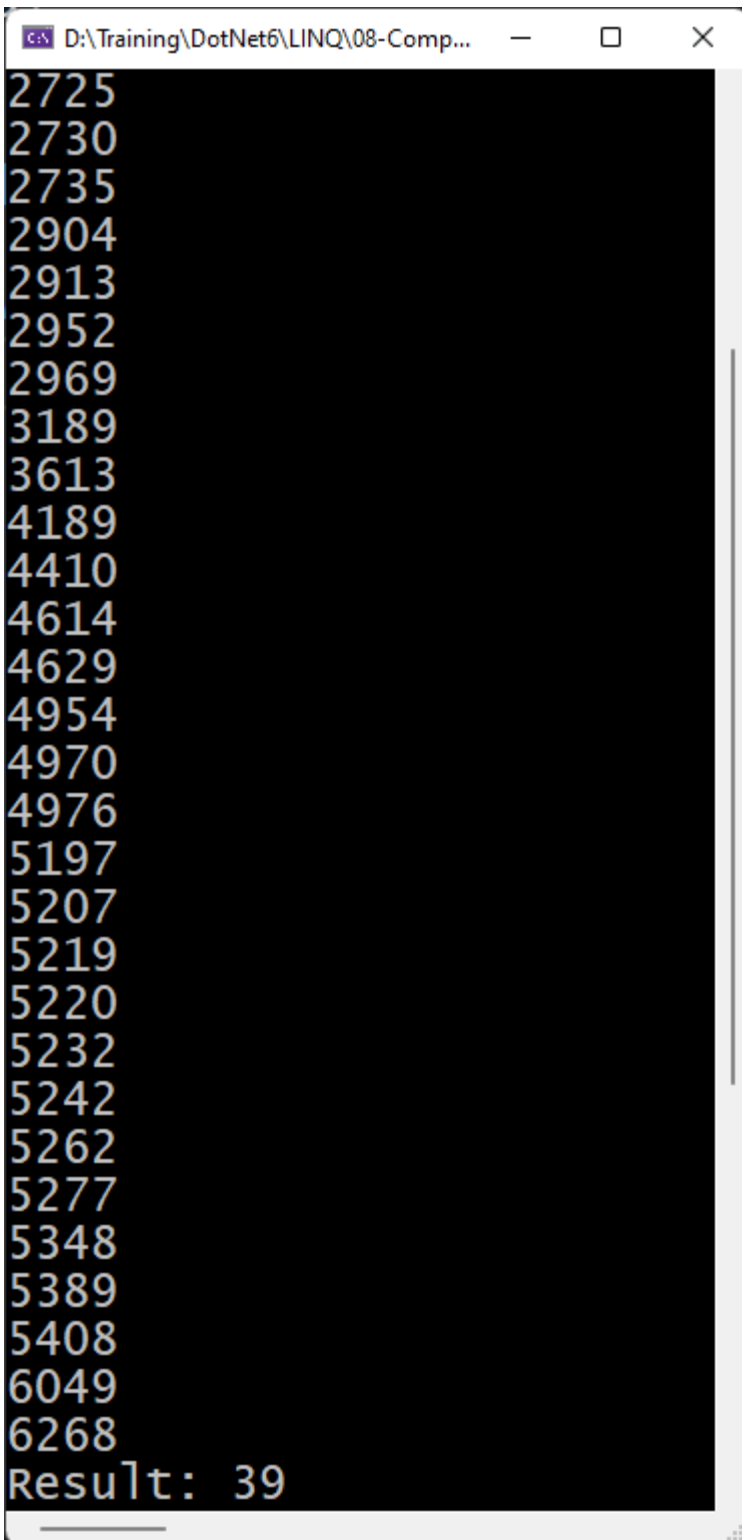
Open the **Program.cs** file and replace the declaration for the ***list*** variable to a list of integers.

```
List<int> list = new();
```

Write a query to use the Intersect() method to determine those songs that have ***TrackNumber*** equal to 3 and ***GenreId*** equal to 30.

Try it Out

Run the application and view the results:



```
D:\Training\DotNet6\LINQ\08-Comp...  
2725  
2730  
2735  
2904  
2913  
2952  
2969  
3189  
3613  
4189  
4410  
4614  
4629  
4954  
4970  
4976  
5197  
5207  
5219  
5220  
5232  
5242  
5262  
5277  
5348  
5389  
5408  
6049  
6268  
Result: 39
```


Lab 7: Intersect using Comparer

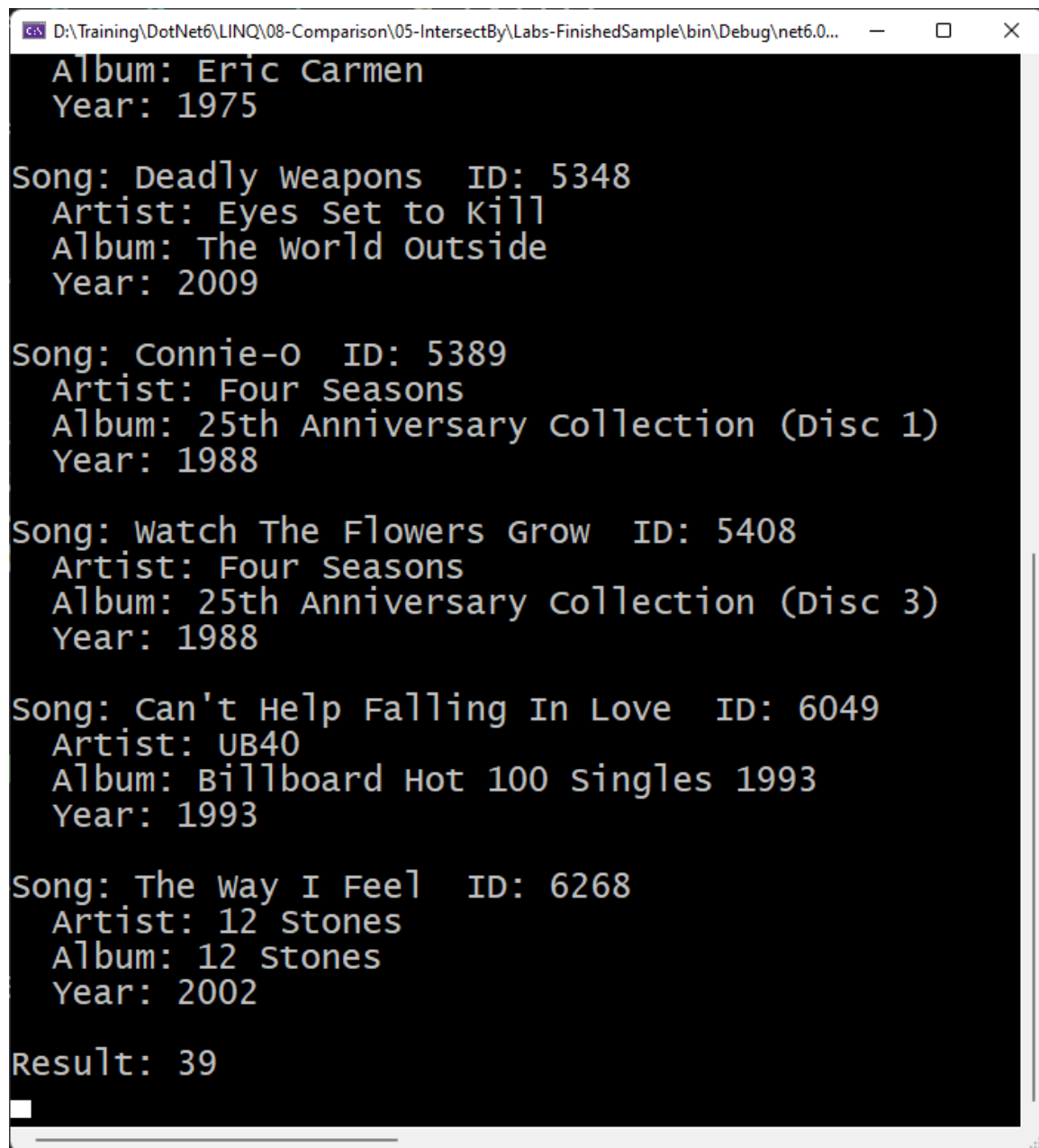
Open the **Program.cs** file and change the list of integers to a list of Song objects.

```
List<Song> list = new();
```

Write a query using the `Intersect()` method and the `SongComparer` class to determine those songs that are in common between the two song collections.

Try it Out

Run the application and view the results:



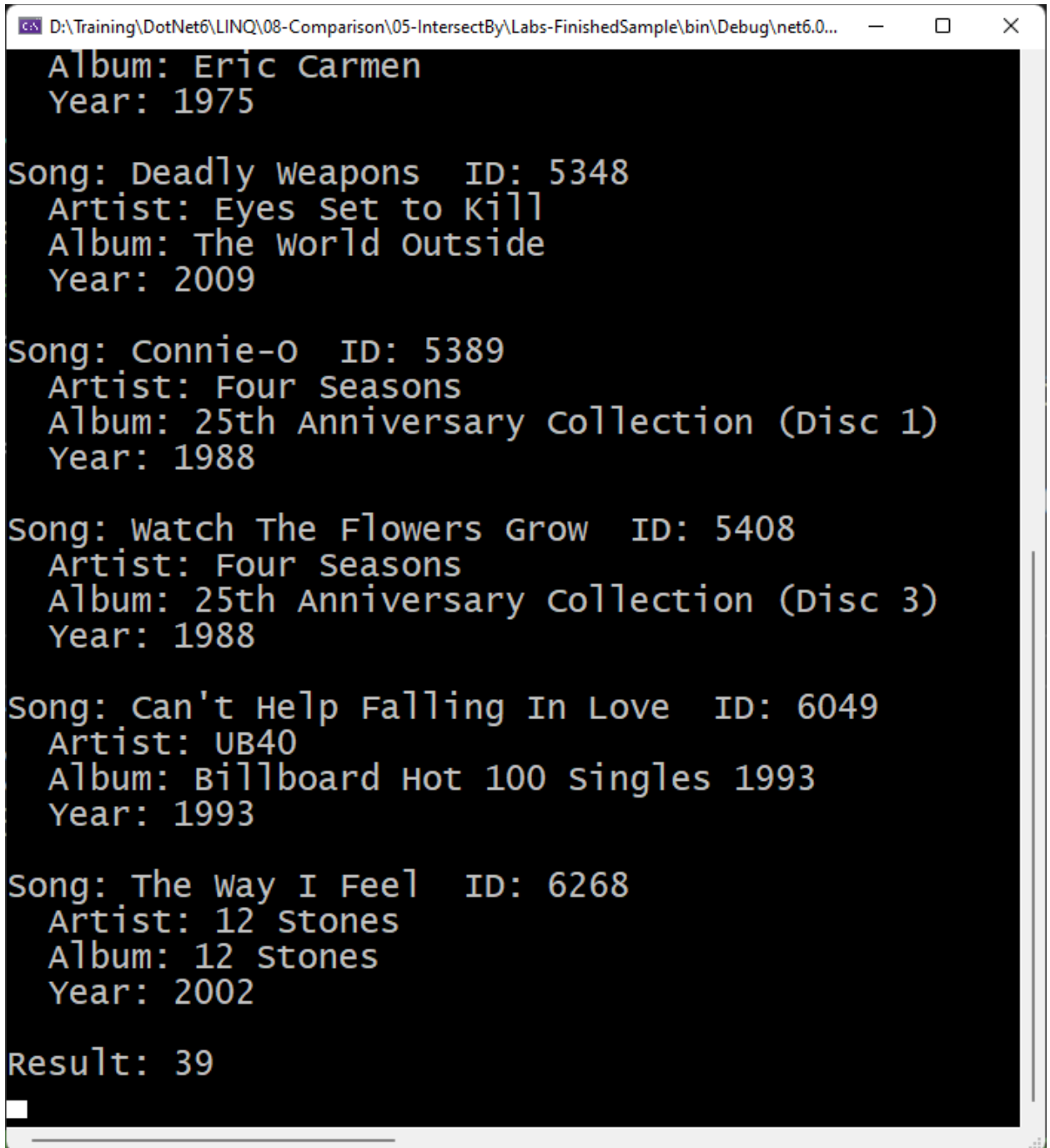
```
D:\Training\DotNet6\LINQ\08-Comparison\05-IntersectBy\Labs-FinishedSample\bin\Debug\net6.0...  
Album: Eric Carmen  
Year: 1975  
  
Song: Deadly Weapons ID: 5348  
Artist: Eyes Set to Kill  
Album: The world Outside  
Year: 2009  
  
Song: Connie-O ID: 5389  
Artist: Four Seasons  
Album: 25th Anniversary collection (Disc 1)  
Year: 1988  
  
Song: watch The Flowers Grow ID: 5408  
Artist: Four Seasons  
Album: 25th Anniversary Collection (Disc 3)  
Year: 1988  
  
Song: Can't Help Falling In Love ID: 6049  
Artist: UB40  
Album: Billboard Hot 100 singles 1993  
Year: 1993  
  
Song: The way I Feel ID: 6268  
Artist: 12 Stones  
Album: 12 Stones  
Year: 2002  
  
Result: 39
```

Lab 8: IntersectBy() Method

Write a query to use the `IntersectBy()` method to determine those songs that are in common between the two song collections based on the ***SongId*** property.

Try it Out

Run the application and view the results:



```
D:\Training\DotNet6\LINQ\08-Comparison\05-IntersectBy\Labs-FinishedSample\bin\Debug\net6.0...  
Album: Eric Carmen  
Year: 1975  
  
Song: Deadly weapons ID: 5348  
Artist: Eyes Set to Kill  
Album: The world outside  
Year: 2009  
  
Song: Connie-O ID: 5389  
Artist: Four Seasons  
Album: 25th Anniversary collection (Disc 1)  
Year: 1988  
  
Song: Watch The Flowers Grow ID: 5408  
Artist: Four Seasons  
Album: 25th Anniversary collection (Disc 3)  
Year: 1988  
  
Song: Can't Help Falling In Love ID: 6049  
Artist: UB40  
Album: Billboard Hot 100 singles 1993  
Year: 1993  
  
Song: The way I Feel ID: 6268  
Artist: 12 Stones  
Album: 12 Stones  
Year: 2002  
  
Result: 39
```

These results should match the previous lab.