

# Partitioning Lab

## Lab 1: Skip() Method

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

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

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

// TODO: Write Your Query Here

// Display data
foreach (Song song in list) {
    Console.WriteLine(song);
}

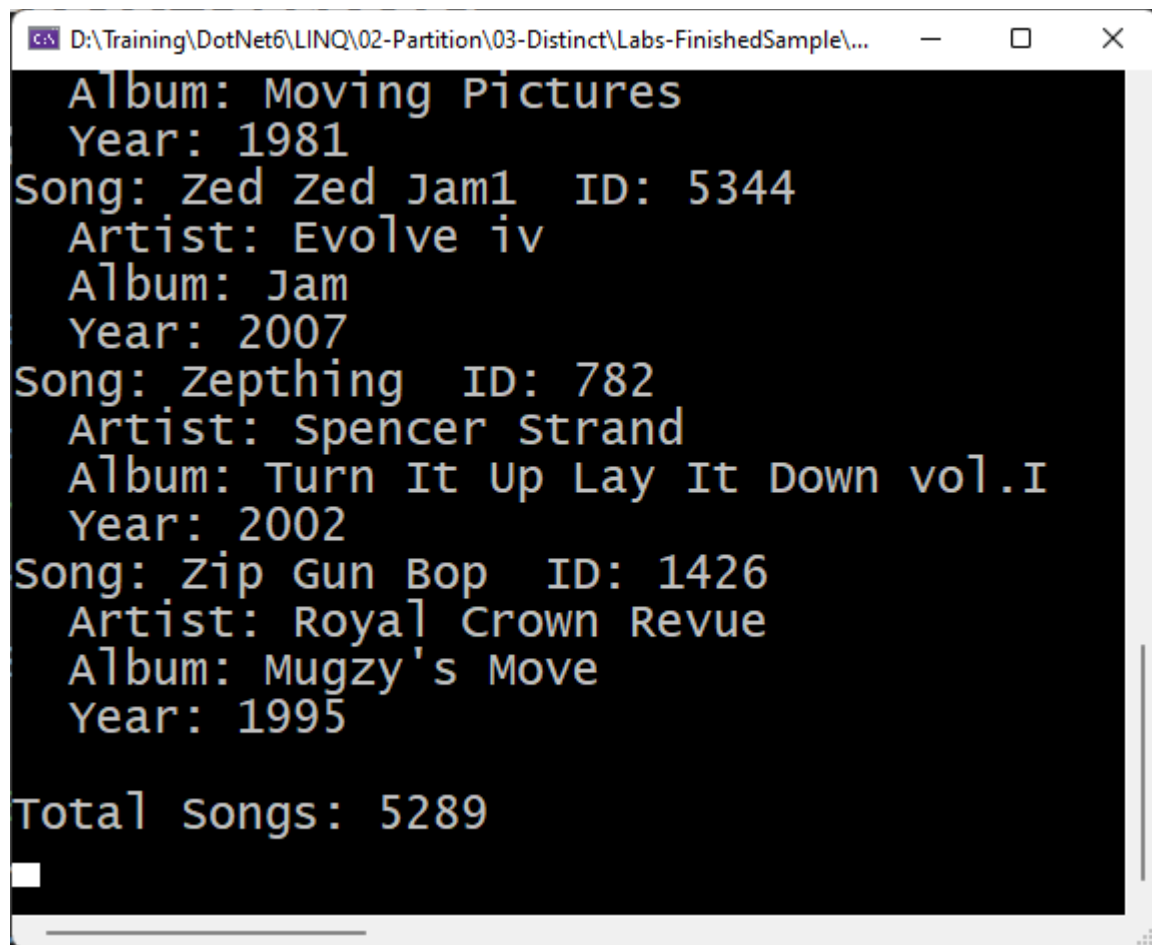
// Display Total Count
Console.WriteLine();
Console.WriteLine($"Total Songs: {list.Count}");

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

Write a query that selects all rows, orders all songs by the **SongName** property, and skips the first 1000 rows.

## Try it Out

Run the application and you should see the following:

A screenshot of a Windows command prompt window. The title bar shows the file path: D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\... The window contains the following text:

```
Album: Moving Pictures
Year: 1981
Song: Zed Zed Jam1 ID: 5344
Artist: Evolve iv
Album: Jam
Year: 2007
Song: Zepthing ID: 782
Artist: Spencer Strand
Album: Turn It Up Lay It Down vol.I
Year: 2002
Song: Zip Gun Bop ID: 1426
Artist: Royal Crown Revue
Album: Mugzy's Move
Year: 1995

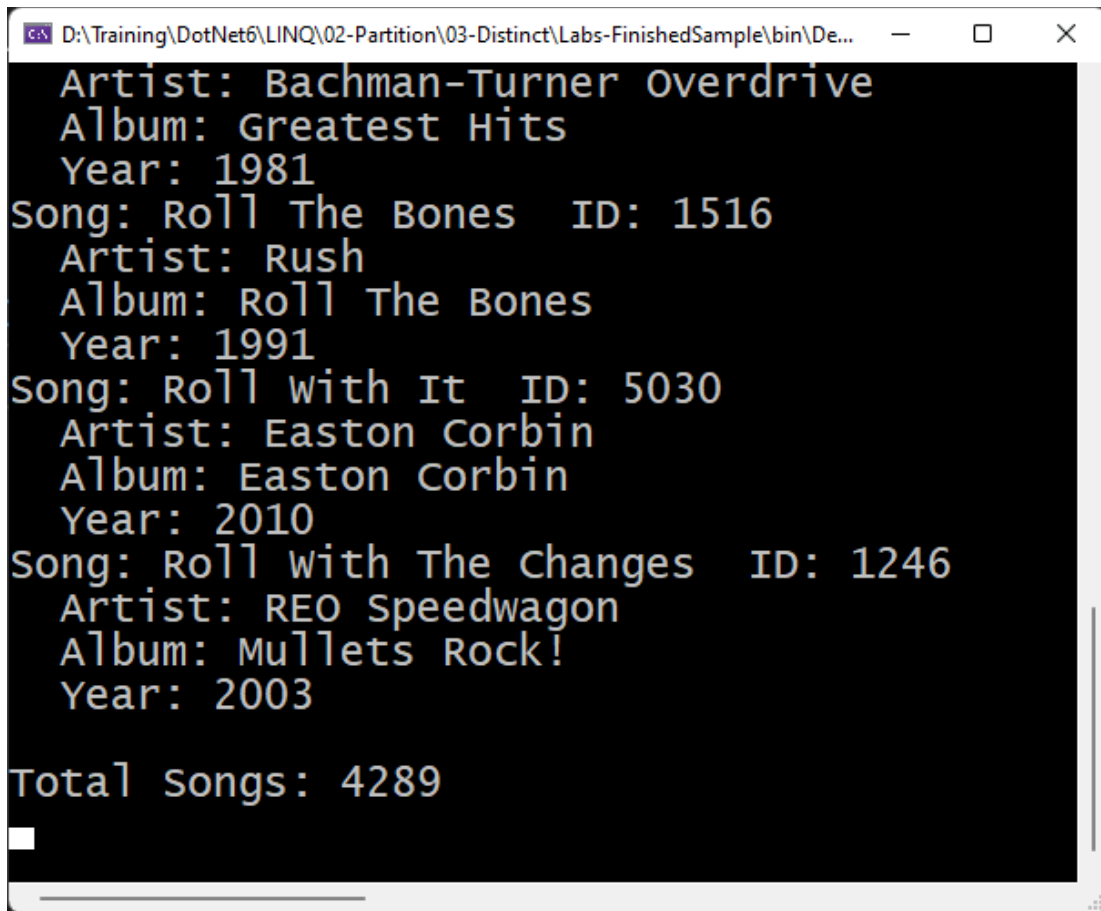
Total songs: 5289
```

## Lab 2: SkipLast() Method

Modify the query to skip the last 2000 rows.

### Try it Out

Run the application and you should see the following:

A screenshot of a Windows console application window. The title bar shows the file path: D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\bin\De... The console output displays song information for four different tracks, each with its artist, album, year, and song title followed by an ID. The data is as follows:

Artist	Album	Year	Song	ID
Bachman-Turner Overdrive	Greatest Hits	1981	Roll The Bones	1516
Rush	Roll The Bones	1991	Roll with It	5030
Easton Corbin	Easton Corbin	2010	Roll with The Changes	1246
REO Speedwagon	Mullets Rock!	2003		

At the bottom of the console output, it states "Total Songs: 4289".

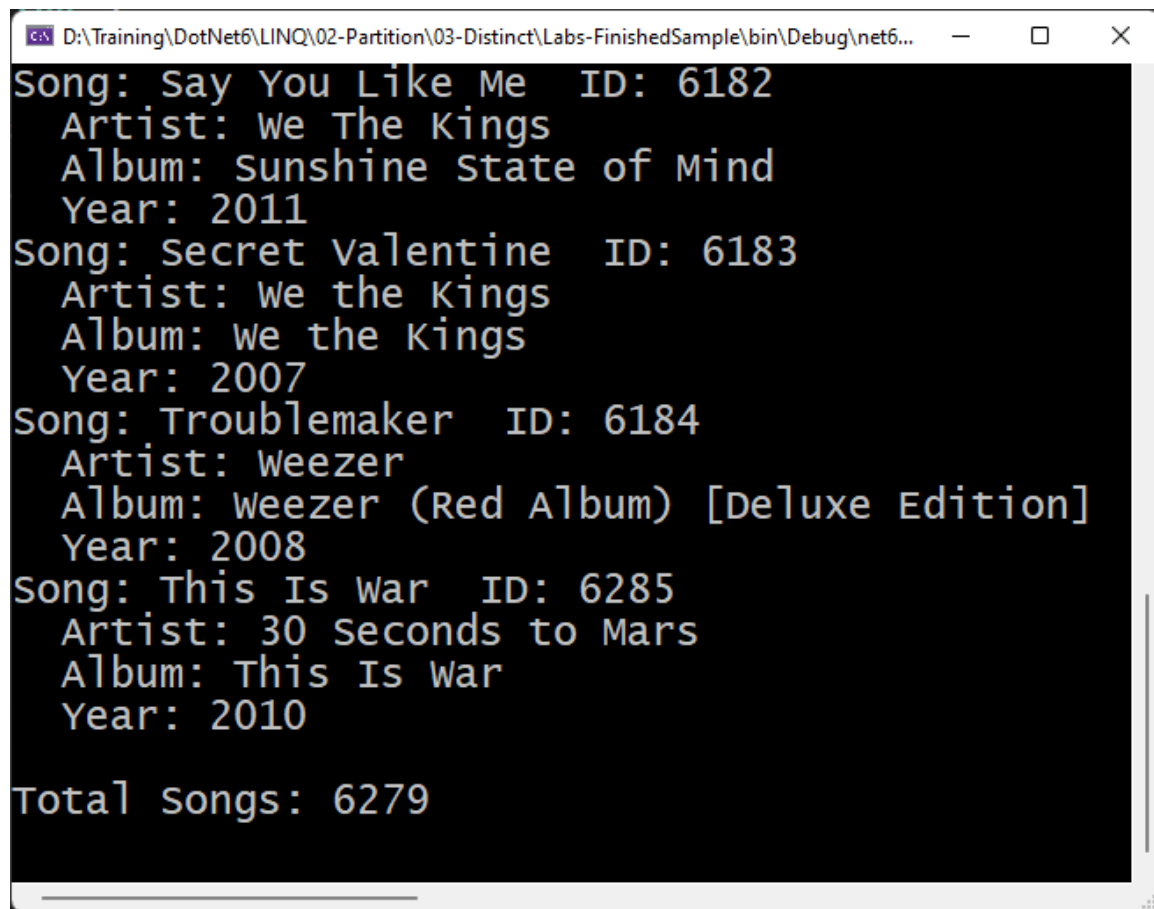
## Lab 3: SkipWhile() Method

Write a query that selects all rows and orders all songs by the **GenreId** property.

Skip all rows from the beginning of the collection where the **GenreId** is equal to 1.

### Try it Out

Run the application and notice that 10 songs are missing. This is the number of songs that have a **GenreId** property set to 1.

A screenshot of a Windows console window. The title bar shows the file path: D:\Training\DotNet6\Linq\02-Partition\03-Distinct\Labs-FinishedSample\bin\Debug\net6... The console output lists five songs with their details: Song: Say You Like Me ID: 6182, Artist: We The Kings, Album: sunshine state of mind, Year: 2011; Song: Secret Valentine ID: 6183, Artist: we the Kings, Album: we the Kings, Year: 2007; Song: Troublemaker ID: 6184, Artist: Weezer, Album: weezer (Red Album) [Deluxe Edition], Year: 2008; Song: This Is War ID: 6285, Artist: 30 Seconds to Mars, Album: This Is War, Year: 2010; and Total Songs: 6279.

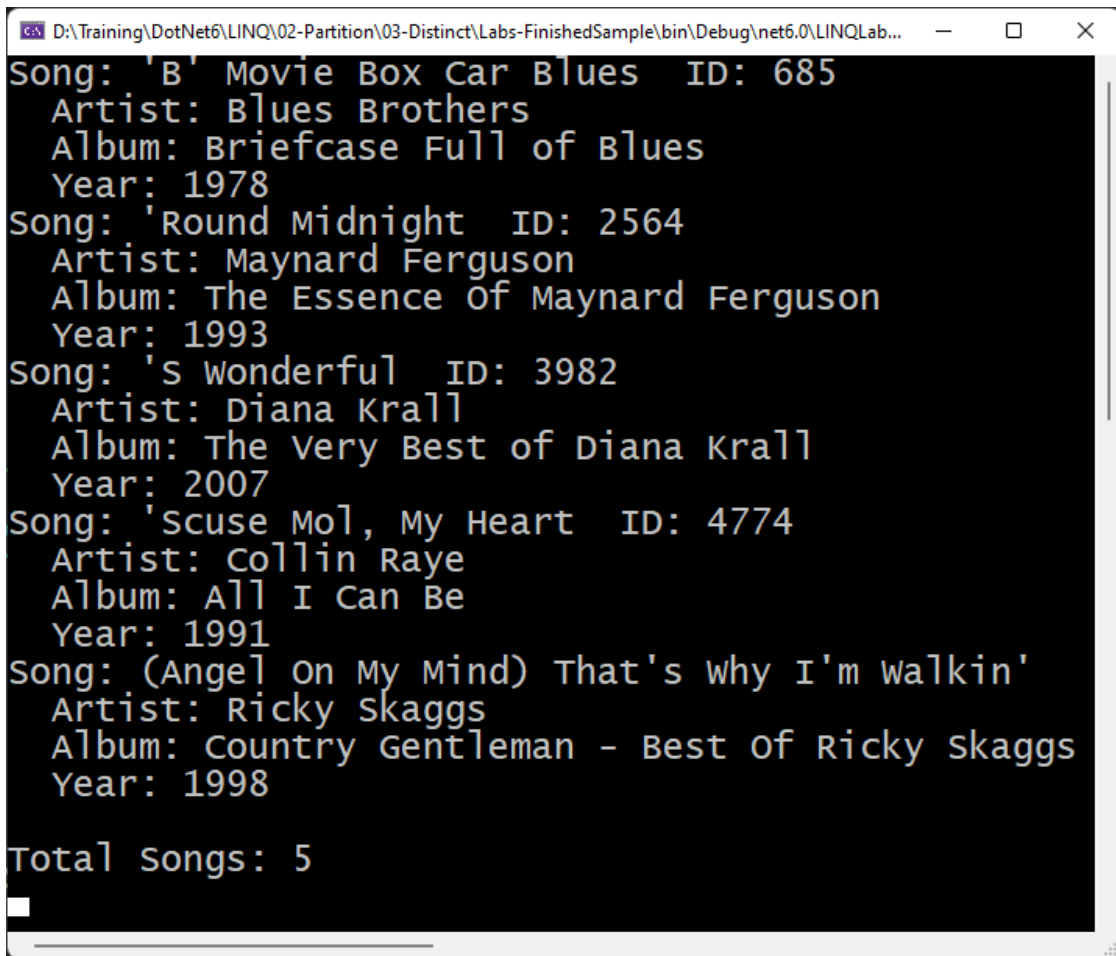
```
Song: Say You Like Me ID: 6182
Artist: We The Kings
Album: sunshine state of mind
Year: 2011
Song: Secret Valentine ID: 6183
Artist: we the Kings
Album: we the Kings
Year: 2007
Song: Troublemaker ID: 6184
Artist: Weezer
Album: weezer (Red Album) [Deluxe Edition]
Year: 2008
Song: This Is War ID: 6285
Artist: 30 Seconds to Mars
Album: This Is War
Year: 2010
Total Songs: 6279
```

## Lab 4: Take() Method

Write a query that selects all songs, orders the songs by the **SongName** property, and takes just the first 5 rows from the collection.

### Try it Out

Run the application and it should look like the following



```
D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\bin\Debug\net6.0\LINQLab...
Song: 'B' Movie Box Car Blues ID: 685
Artist: Blues Brothers
Album: Briefcase Full of Blues
Year: 1978
Song: 'Round Midnight ID: 2564
Artist: Maynard Ferguson
Album: The Essence Of Maynard Ferguson
Year: 1993
Song: 'S Wonderful ID: 3982
Artist: Diana Krall
Album: The Very Best of Diana Krall
Year: 2007
Song: 'Scuse Mo', My Heart ID: 4774
Artist: Collin Raye
Album: All I Can Be
Year: 1991
Song: (Angel On My Mind) That's why I'm walkin'
Artist: Ricky Skaggs
Album: Country Gentleman - Best of Ricky Skaggs
Year: 1998

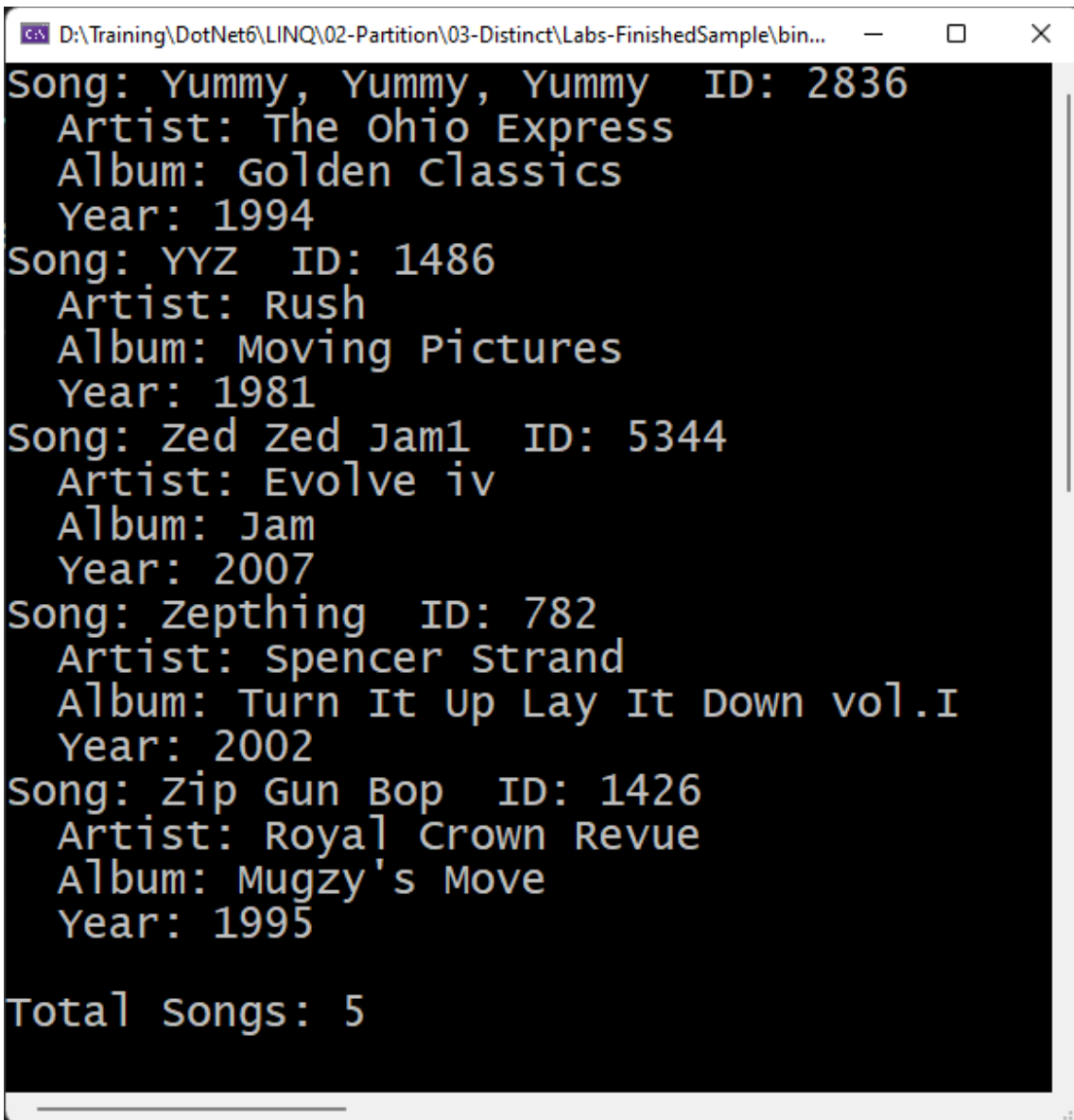
Total Songs: 5
```

## Lab 5: TakeLast() Method

Write a query that selects all rows, orders the songs by the **SongName** property, and takes just the last 5 rows from the collection.

### Try it Out

Run the application and you should see results that look like the following:



```
D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\bin...
Song: Yummy, Yummy, Yummy ID: 2836
Artist: The Ohio Express
Album: Golden Classics
Year: 1994
Song: YYZ ID: 1486
Artist: Rush
Album: Moving Pictures
Year: 1981
Song: Zed Zed Jam1 ID: 5344
Artist: Evolve iv
Album: Jam
Year: 2007
Song: Zepthing ID: 782
Artist: Spencer Strand
Album: Turn It Up Lay It Down vol.I
Year: 2002
Song: Zip Gun Bop ID: 1426
Artist: Royal Crown Revue
Album: Mugzy's Move
Year: 1995

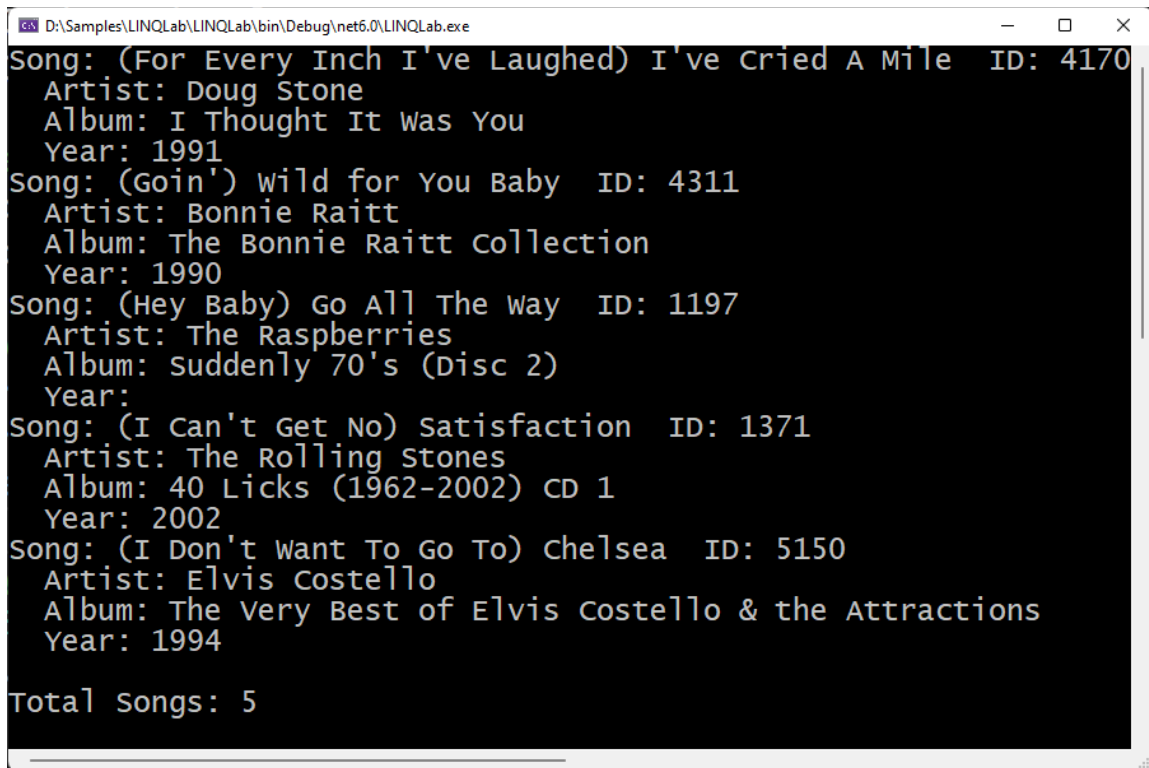
Total Songs: 5
```

## Lab 6: Take() Method Using Range Operator

Write a query that selects all rows, orders the songs by the **SongName** property, and takes just rows 5 through 9 from the collection.

### Try it Out

Run the application and you should see results that look like the following:



```
D:\Samples\LINQLab\LINQLab\bin\Debug\net6.0\LINQLab.exe
Song: (For Every Inch I've Laughed) I've Cried A Mile ID: 4170
Artist: Doug Stone
Album: I Thought It Was You
Year: 1991
Song: (Goin') wild for You Baby ID: 4311
Artist: Bonnie Raitt
Album: The Bonnie Raitt collection
Year: 1990
Song: (Hey Baby) Go All The way ID: 1197
Artist: The Raspberries
Album: Suddenly 70's (Disc 2)
Year:
Song: (I Can't Get No) Satisfaction ID: 1371
Artist: The Rolling Stones
Album: 40 Licks (1962-2002) CD 1
Year: 2002
Song: (I Don't Want To Go To) Chelsea ID: 5150
Artist: Elvis Costello
Album: The Very Best of Elvis Costello & the Attractions
Year: 1994

Total Songs: 5
```

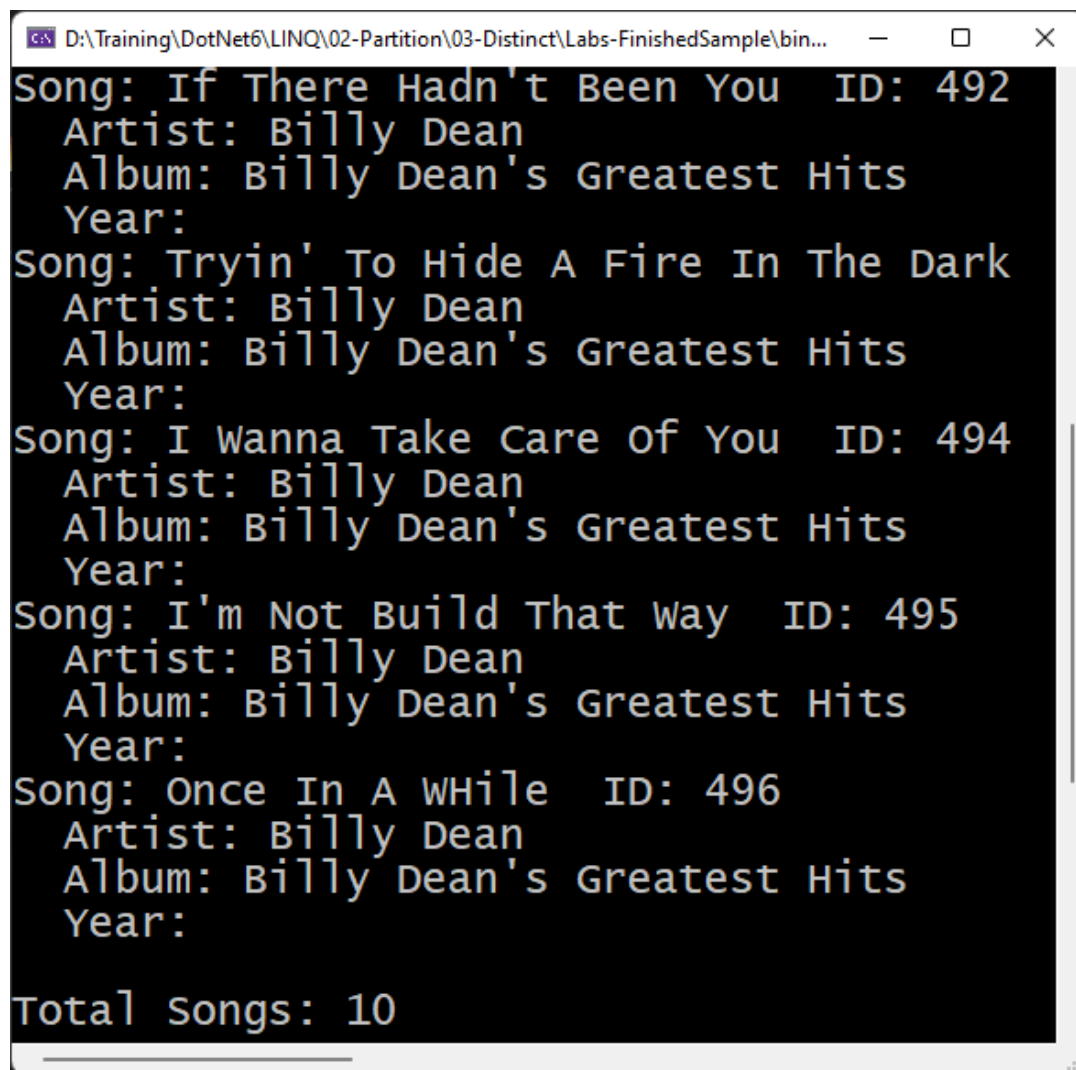
## Lab 7: TakeWhile() Method

Write a query that selects all rows and orders the songs by the **GenreId** property.

After selecting the rows, take all rows from the beginning of the collection where the **GenreId** is equal to 1.

### Try it Out

Run the application and you should see the following results:



A screenshot of a Windows console window. The title bar shows the file path: D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\bin... The console output lists five songs by Billy Dean, each with its ID and album name. The songs are: 'If There Hadn't Been You' (ID: 492), 'Tryin' To Hide A Fire In The Dark' (ID: 493), 'I Wanna Take Care Of You' (ID: 494), 'I'm Not Build That Way' (ID: 495), and 'Once In A While' (ID: 496). All songs are from the album 'Billy Dean's Greatest Hits'. The year field is empty for all entries. At the bottom, it says 'Total Songs: 10'.

```

Song: If There Hadn't Been You ID: 492
Artist: Billy Dean
Album: Billy Dean's Greatest Hits
Year:
Song: Tryin' To Hide A Fire In The Dark
Artist: Billy Dean
Album: Billy Dean's Greatest Hits
Year:
Song: I Wanna Take Care Of You ID: 494
Artist: Billy Dean
Album: Billy Dean's Greatest Hits
Year:
Song: I'm Not Build That Way ID: 495
Artist: Billy Dean
Album: Billy Dean's Greatest Hits
Year:
Song: Once In A While ID: 496
Artist: Billy Dean
Album: Billy Dean's Greatest Hits
Year:

Total Songs: 10

```

## Lab 8: Distinct() Method

Replace the code in the **Program.cs** file with the following:



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

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

// TODO: Write Your Query Here

// Display data
foreach (int? id in list) {
    Console.WriteLine(id);
}

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

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

Write a query to select all rows and order the songs by the **Rating** property.  
Select just the **Rating** property.  
Apply the Distinct() method to get a list of integer values.

## Try it Out

Run the application and you should see the following results:



```
D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-F...  
20  
40  
60  
80  
100  
  
Total Ratings: 5
```

## Lab 9: DistinctBy() Method

Modify the *list* variable to be a List<Song> values

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

Modify the foreach statement to use the Song class.

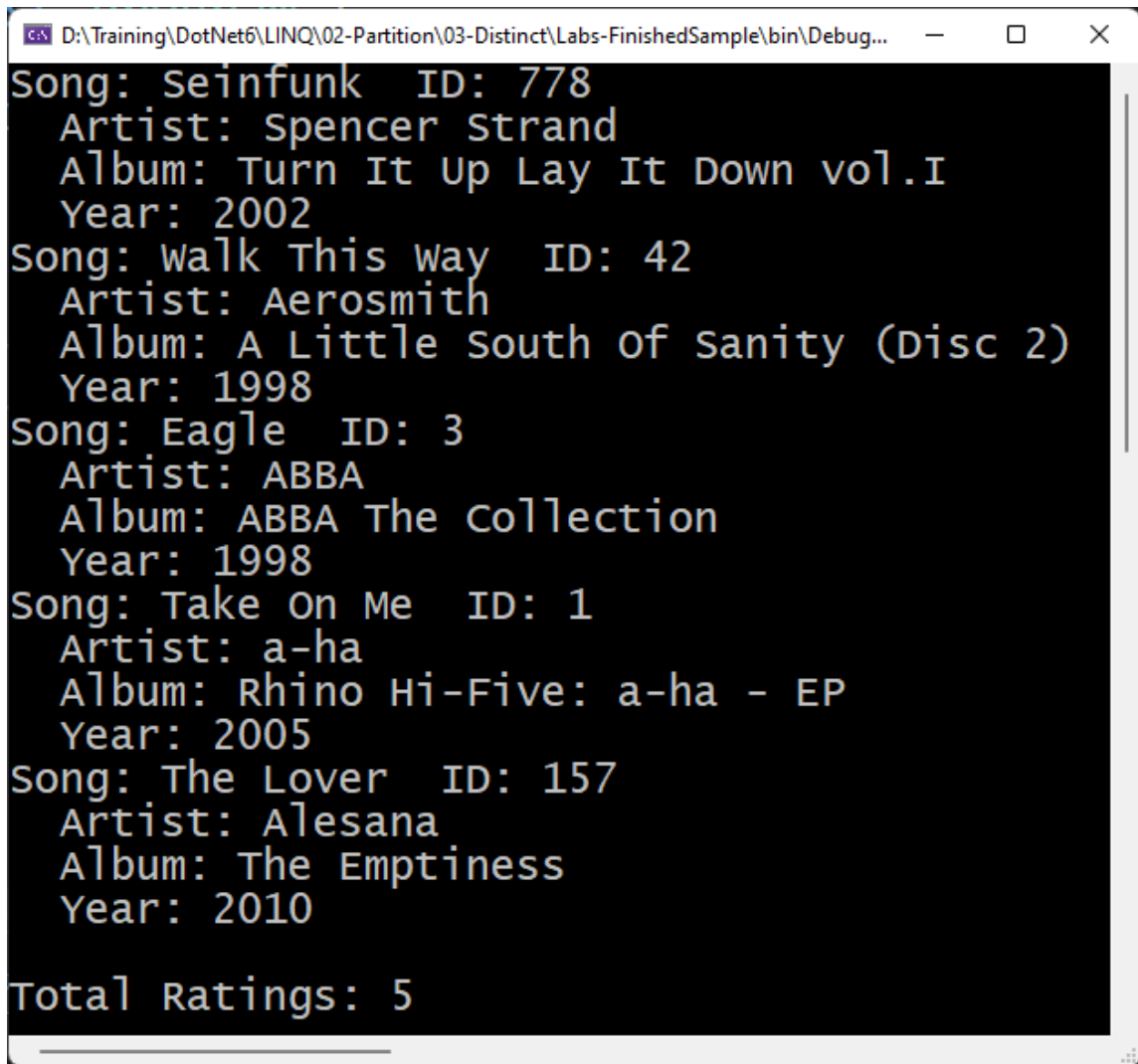
```
// Display data  
foreach (Song id in list) {  
    Console.WriteLine(id);  
}
```

Write a query to select all rows and order the songs by the *Rating* property.

Apply the DistinctBy() method to get a list of songs that are the first distinct values for each *Rating*.

### Try it Out

Run the application and you should see results that look like the following:

A screenshot of a Windows command prompt window. The title bar shows the file path: D:\Training\DotNet6\LINQ\02-Partition\03-Distinct\Labs-FinishedSample\bin\Debug... The window contains a list of five songs with their details, each preceded by a line separator. The data is as follows:

Song	ID	Artist	Album	Year
Seinfunk	778	Spencer Strand	Turn It Up Lay It Down vol.I	2002
walk This way	42	Aerosmith	A Little south of sanity (Disc 2)	1998
Eagle	3	ABBA	ABBA The collection	1998
Take On Me	1	a-ha	Rhino Hi-Five: a-ha - EP	2005
The Lover	157	Alesana	The Emptiness	2010

At the bottom of the list, it says "Total Ratings: 5".