

# Basic Query Syntax

A basic LINQ query, in *query syntax*, has three parts:

```
string[] heroes = { "D. Va", "Lucio", "Mercy", "Soldier 76", "Pharah",  
"Reinhardt" };  
  
var shortHeroes = from h in heroes  
    where h.Length < 8  
    select h;
```

The `from` operator declares a variable to iterate through the sequence. In this case, `h` is used to iterate through `heroes`.

The `where` operator picks elements from the sequence if they satisfy the given condition. The condition is normally written like the conditional expressions you would find in an `if` statement. In this case, the condition is `h.Length < 8`.

The `select` operator determines what is returned for each element in the sequence. In this case, it's just the element itself.

The `from` and `select` operators are required, `where` is optional. In this next example, `select` is used to make a new string starting with "Hero: " for each element:

```
var heroTitles = from hero in heroes  
    select $"HERO: {hero.ToUpper()}";
```

Each element in `heroTitles` would look like `"HERO: D. VA"`, `"HERO: LUCIO"`, etc.

## ☒ Instructions

1.

Write a `from` - `where` - `select` query that selects all of the elements in `heroes` that contain the character `"i"`. Store the result in a variable named `heroesWithI`.

Hint



Make sure to use query syntax. Here's an example that finds all birds containing the letter "g":

```
var gBirds = from bird in birds
              where bird.Contains("g")
              select bird;
```

2.

Write a `from - select` query that returns the same array as `heroes`, but every space is replaced with an underscore (`_`). Store the result in a variable named `underscored`.

Hint



Call `Replace()` with each string. [Documentation here](#).