Your organization has teams all around the world. You are building an application on .NET Framework that the USA team will use. You are required to get data about the USA team, but you would prefer to rely on the client's language and culture. Which of the following code fragments would you use to achieve this?

(Select all that apply)

```
1. CultureInfo th= new CultureInfo("en-US")
```

```
1. CultureInfo th= new CultureInfo("US");
```

```
1. RegionInfo MyReg = new RegionInfo("US");
```

```
1. RegionInfo MyReg = new RegionInfo( new CultureInfo("en-US",false).LCID);
```

Select the RIGHT output for the following SQL Query.

```
1. SELECT RIGHT(REPLACE('SQL Server 2008', '2008', '2012'),  4)
```

2008

SQL

2012

Server

er 2

Suppose, in an MVC application, there is an Area "Admin" with a controller "AccountController" and an action method "Index. You need an action link to point to the method from the outside area. Which is the correct syntax?

```
1. Html.ActionLink("Go To Account", "Index", "AccountController", new { Area = "Admin" }, new{})
```

```
1. Html.ActionLink("Go To Account", "Index", "Account", "Admin", new{})
```

```
1. Html.ActionLink("Go To Account", "Admin", "Index", "Account", new{})
```

```
1. Html.ActionLink("Go To Account", "Index", "Account", new { Area = "Admin" }, new{})
```

```
1. Html.ActionLink("Go To Account", "Index", "Account", new { Area = "AdminArea" }, new{})
```

Your Project is initializing a lot of classes while starting. Some of these classes are used very late in the life-cycle or sometimes might not be used, so you want the creation of those objects to defer until they are first used, how can this be achieved in C#?

```
Lazy<DataType> name = new Lazy<DataType>();
```

```
List<Lazy<DataType>> name = new List<Lazy<DataType>>();
```

```
private Lazy<DataType> name = new Lazy<DataType>(() =>
        {
            //some additional details
        }).
```

```
private List<Lazy<DataType>> name = new List<Lazy<DataType>>(() =>
        {
            //some additional details
        }).
```