```
using System;
using System.Collections.Generic;
using System.Linq;

class Employee
{
    public int EmployeeID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Title { get; set; }
    public DateTime DOB { get; set; }
    public DateTime DOJ { get; set; }
    public string City { get; set; }
}

class Program
{
    static void Main()
    {

        List<Employee> empList = new List<Employee>
        {
            new Employee
            {
                EmployeeID = 1001,
                FirstName = "Malcolm",
                LastName = "Daruwalla",
                Title = "Manager",
                DOB = DateTime.Parse("1984-11-16"),
                DOJ = DateTime.Parse("2011-06-08"),
                City = "Mumbai"
```

```
 },
new Employee
{
   EmployeeID = 1002,
   FirstName = "Asdin",
   LastName = "Dhalla",
   Title = "AsstManager",
   DOB = DateTime.Parse("1984-08-20"),
   DOJ = DateTime.Parse("2012-07-07"),
   City = "Mumbai"
},
new Employee
{
   EmployeeID = 1003,
   FirstName = "Madhavi",
   LastName = "Oza",
   Title = "Consultant",
   DOB = DateTime.Parse("1987-11-14"),
   DOJ = DateTime.Parse("2015-04-12"),
   City = "Pune"
},
new Employee
{
   EmployeeID = 1004,
   FirstName = "Saba",
   LastName = "Shaikh",
   Title = "SE",
   DOB = DateTime.Parse("1990-06-03"),
   DOJ = DateTime.Parse("2016-02-02"),
   City = "Pune"
},
new Employee
{
   EmployeeID = 1005,
   FirstName = "Nazia",
   LastName = "Shaikh",
   Title = "SE",
   DOB = DateTime.Parse("1991-03-08"),
   DOJ = DateTime.Parse("2016-02-02"),
   City = "Mumbai"
},
new Employee
{
   EmployeeID = 1006,
   FirstName = "Amit",
   LastName = "Pathak",
   Title = "Consultant",
   DOB = DateTime.Parse("1989-11-07"),
   DOJ = DateTime.Parse("2014-08-08"),
   City = "Chennai"
},
new Employee
{
   EmployeeID = 1007,
   FirstName = "Vijay",
   LastName = "Natrajan",
   Title = "Consultant",
   DOB = DateTime.Parse("1989-12-02"),
   DOJ = DateTime.Parse("2015-06-01"),
   City = "Mumbai"
},
new Employee
```

```
    {
        EmployeeID = 1008,
        FirstName = "Rahul",
        LastName = "Dubey",
        Title = "Associate",
        DOB = DateTime.Parse("1993-11-11"),
        DOJ = DateTime.Parse("2014-11-06"),
        City = "Chennai"
    },
    new Employee
    {
        EmployeeID = 1009,
        FirstName = "Suresh",
        LastName = "Mistry",
        Title = "Associate",
        DOB = DateTime.Parse("1992-08-12"),
        DOJ = DateTime.Parse("2014-12-03"),
        City = "Chennai"
    },
    new Employee
    {
        EmployeeID = 1010,
        FirstName = "Sumit",
        LastName = "Shah",
        Title = "Manager",
        DOB = DateTime.Parse("1991-04-12"),
        DOJ = DateTime.Parse("2016-01-02"),
        City = "Pune"
    }
};
```

```
        while(true)

        {

            Console.WriteLine("Enter a choice");

           Console.WriteLine("1.Display detail of all employee");

           Console.WriteLine("2.Display detail of all employee whose location is not mumbai");

           Console.WriteLine("3.Display detail of all employee whose title is AsstManager");

           Console.WriteLine("4.Display detail of all employee whose Last Name start with S");

           Console.WriteLine("5.Display list of all employee who have joined before 1/1/2015");

           Console.WriteLine("6.Display list of all employee whose date of birth is after 1/1/1990");

           Console.WriteLine("7.Display list of all employee whose designation is Consultant and
Associate");

           Console.WriteLine("8.Display total number of employees");

           Console.WriteLine("9.Display total number of employees belonging to 'Chennai'");

           Console.WriteLine("10.Display highest employee id from the list");

           Console.WriteLine("11.Display total number of employee who have joined after 1/1/2015");

           Console.WriteLine("12.Display total number of employee whose designation is not
'Associate'");

           Console.WriteLine("13.Display total number of employee based on City");

          Console.WriteLine("14.Display total number of employee based on City and title");

          Console.WriteLine("15.Display total number of employee who is youngest in the list");


          int ch=int.Parse(Console.ReadLine());

          switch(ch)

          {

            case 1:

              Console.WriteLine("All Employees:");

              foreach (var emp in empList)

              {

                Console.WriteLine($"{emp.EmployeeID}, {emp.FirstName},{emp.LastName},{emp.Title},
{emp.DOB.ToShortDateString()}, {emp.DOJ.ToShortDateString()},{emp.City}");

              }

              break;
            case 2:
              Console.WriteLine("\nEmployees not from Mumbai:");
              var notMumbai = empList.Where(e => e.City != "Mumbai");
              foreach (var emp in notMumbai)
              {
                Console.WriteLine($"{emp.EmployeeID}, {emp.FirstName},{emp.LastName},{emp.Title},
{emp.DOB.ToShortDateString()}, {emp.DOJ.ToShortDateString()},{emp.City}");
              }
              break;

            case 3:
              Console.WriteLine("\nEmployees with title AsstManager:");
              var asstManagers = empList.Where(e => e.Title == "AsstManager");
              foreach (var emp in asstManagers)
              {
                Console.WriteLine($"{emp.EmployeeID}, {emp.FirstName},{emp.LastName},{emp.Title},
{emp.DOB.ToShortDateString()}, {emp.DOJ.ToShortDateString()},{emp.City}");
              }
              break;

            case 4:
              Console.WriteLine("\nEmployees whose last name starts with S:");
               var lastNameS = empList.Where(e => e.LastName.StartsWith("S"));
               foreach (var emp in lastNameS)
               {
                Console.WriteLine($"{emp.EmployeeID}, {emp.FirstName},{emp.LastName},{emp.Title},
{emp.DOB.ToShortDateString()}, {emp.DOJ.ToShortDateString()},{emp.City}");
               }
               break;

            case 5:
```

```
          Console.WriteLine("\nEmployees who joined before 01/01/2015:");

           var joinedBefore2015 = empList.Where(e => e.DOJ < DateTime.Parse("2015-01-01"));

           foreach (var emp in joinedBefore2015)

           {

              Console.WriteLine($"{emp.EmployeeID},
{emp.FirstName},{emp.LastName},{emp.Title}, {emp.DOB.ToShortDateString()},
{emp.DOJ.ToShortDateString()},{emp.City}");

           }

           break;


        case 6:

          Console.WriteLine("\nEmployees born after 01/01/1990:");

           var bornAfter1990 = empList.Where(e => e.DOB > DateTime.Parse("1990-01-01"));

           foreach (var emp in bornAfter1990)

           {

              Console.WriteLine($"{emp.EmployeeID},
{emp.FirstName},{emp.LastName},{emp.Title}, {emp.DOB.ToShortDateString()},
{emp.DOJ.ToShortDateString()},{emp.City}");

           }

           break;


        case 7:

          Console.WriteLine("\nEmployees who are Consultant or Associate:");

          var consultantsOrAssociates = empList.Where(e => e.Title == "Consultant" || e.Title ==
"Associate");

           foreach (var emp in consultantsOrAssociates)

           {

             Console.WriteLine($"{emp.EmployeeID}, {emp.FirstName},{emp.LastName},{emp.Title},
{emp.DOB.ToShortDateString()}, {emp.DOJ.ToShortDateString()},{emp.City}");

           }

           break;


        case 8:
          Console.WriteLine($"\nTotal number of employees: {empList.Count}");

          break;


        case 9:

          Console.WriteLine($"Total number of employees in Chennai: {empList.Count(e => e.City == "Chennai")}");

          break;


        case 10:

          Console.WriteLine($"Highest Employee ID: {empList.Max(e => e.EmployeeID)}");

          break;


        case 11:

          Console.WriteLine($"\nTotal number of employees joined after 01/01/2015: {empList.Count(e => e.DOJ > DateTime.Parse("2015-01-01"))}");

          break;


        case 12:

          Console.WriteLine($"Total number of employees not Associate: {empList.Count(e => e.Title != "Associate")}");

          break;


        case 13:

          Console.WriteLine("\nTotal number of employees by city:");

           var empByCity = empList.GroupBy(e => e.City).Select(group => new { City = group.Key, Count = group.Count() });

           foreach (var group in empByCity)

           {

             Console.WriteLine($"{group.City}: {group.Count}");

           }

           break;
        case 14:
          Console.WriteLine("\nTotal number of employees by city and title:");
           var empByCityAndTitle = empList.GroupBy(e => new { e.City, e.Title }).Select(group =>
new { group.Key.City, group.Key.Title, Count = group.Count() });
           foreach (var group in empByCityAndTitle)
           {
           Console.WriteLine($"{group.City} - {group.Title}: {group.Count}");
           }
           break;

        case 15:
          Console.WriteLine("\nYoungest Employee:");
           var youngest = empList.OrderByDescending(e => e.DOB).First();
           Console.WriteLine($"{youngest.EmployeeID}, {youngest.FirstName}
{youngest.LastName}, {youngest.DOB.ToShortDateString()}");
           break;

        default :
          Console.WriteLine("Invalid Choice");
          break;
      }
    }
}
}
```