DAY 4
C#

LINQ

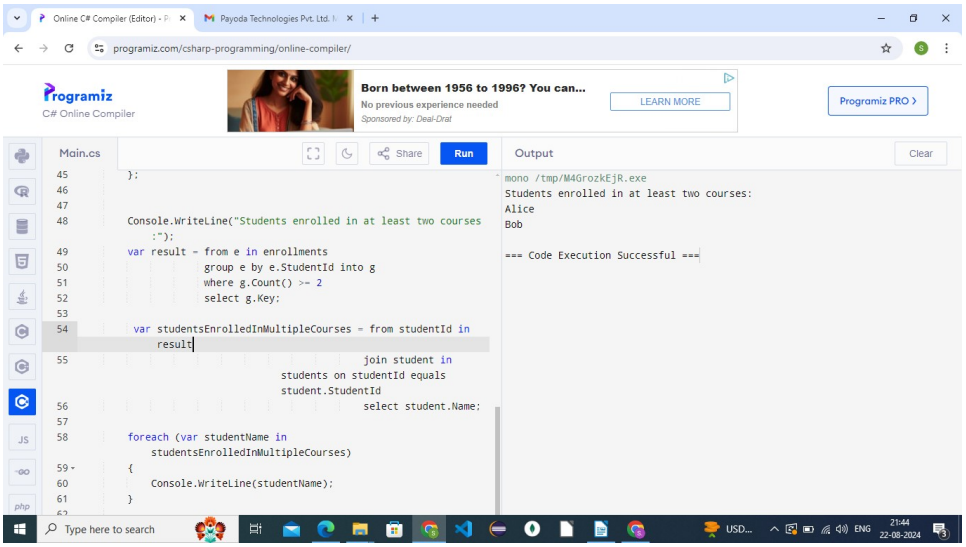
1. Filter: Retrieve a list of students who are enrolled in at least two courses.

CODE:

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

public class Student
{
  public int StudentId{set; get;}
  public string Name{set; get;}
}
public class Course
{
  public int CourseId{set; get;}
  public string Title{set; get;}
}
public class Enrollment{
  public int StudentId{set; get;}
  public int CourseId{set; get;}
}

public class Program{
    public static void Main(string[] args){
        var students = new List<Student>
        {
         new Student { StudentId = 1, Name = "Alice" },
         new Student { StudentId = 2, Name = "Bob" },
         new Student { StudentId = 3, Name = "Charlie" },
         new Student { StudentId = 4, Name = "David" }
        };

        var courses = new List<Course>
        {
         new Course { CourseId = 1, Title = "Math" },
         new Course { CourseId = 2, Title = "Science" },
         new Course { CourseId = 3, Title = "History" }
        };

        var enrollments = new List<Enrollment>
        {
         new Enrollment { StudentId = 1, CourseId = 1 },
         new Enrollment { StudentId = 1, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 3 },
         new Enrollment { StudentId = 3, CourseId = 1 },
```

```
        new Enrollment { StudentId = 4, CourseId = 2 }
        };


        Console.WriteLine("Students enrolled in at least two courses:");
        var result = from e in enrollments
                     group e by e.StudentId into g
                     where g.Count() >= 2
                     select g.Key;

        var studentsEnrolledInMultipleCourses = from studentId in result
                                                join student in students on studentId equals student.StudentId
                                                select student.Name;

        foreach (var studentName in studentsEnrolledInMultipleCourses)
        {
            Console.WriteLine(studentName);
        }


    }
}
```

OUTPUT:



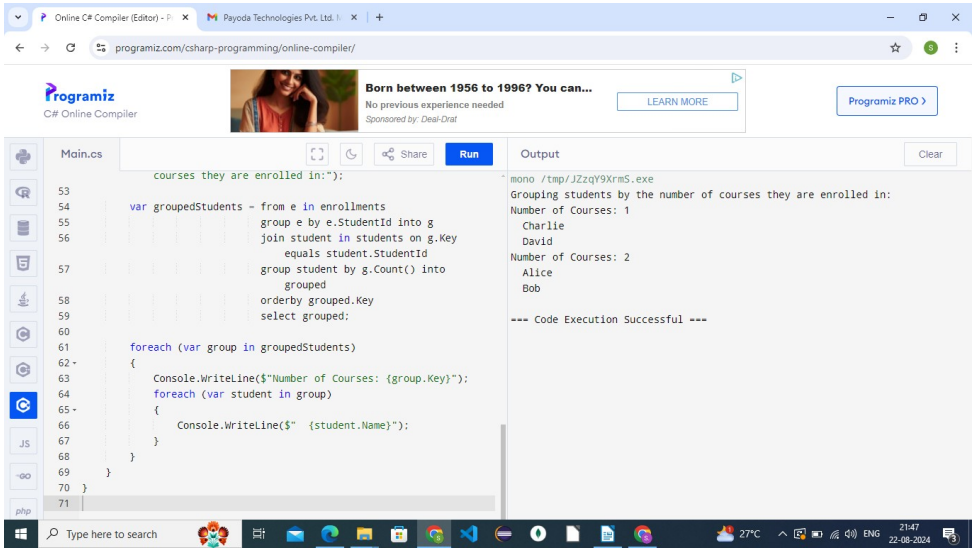
2. Group: Group the students by the number of courses they are enrolled in.

CODE:

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

public class Student
{
   public int StudentId { set; get; }
   public string Name { set; get; }
}

public class Course
{
   public int CourseId { set; get; }
   public string Title { set; get; }
}

public class Enrollment
{
   public int StudentId { set; get; }
   public int CourseId { set; get; }
}

public class Program
{
   public static void Main(string[] args)
   {
      var students = new List<Student>
      {
         new Student { StudentId = 1, Name = "Alice" },
         new Student { StudentId = 2, Name = "Bob" },
         new Student { StudentId = 3, Name = "Charlie" },
         new Student { StudentId = 4, Name = "David" }
      };

      var courses = new List<Course>
      {
         new Course { CourseId = 1, Title = "Math" },
         new Course { CourseId = 2, Title = "Science" },
         new Course { CourseId = 3, Title = "History" }
      };

      var enrollments = new List<Enrollment>
      {
         new Enrollment { StudentId = 1, CourseId = 1 },
         new Enrollment { StudentId = 1, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 3 },
         new Enrollment { StudentId = 3, CourseId = 1 },
```

```
            new Enrollment { StudentId = 4, CourseId = 2 }
        };

        Console.WriteLine("Grouping students by the number of courses they are enrolled
in:");

        var groupedStudents = from e in enrollments
                              group e by e.StudentId into g
                              join student in students on g.Key equals student.StudentId
                              group student by g.Count() into grouped
                              orderby grouped.Key
                              select grouped;

        foreach (var group in groupedStudents)
        {
            Console.WriteLine($"Number of Courses: {group.Key}");
            foreach (var student in group)
            {
                Console.WriteLine($"  {student.Name}");
            }
        }
    }
}
```

OUTPUT:



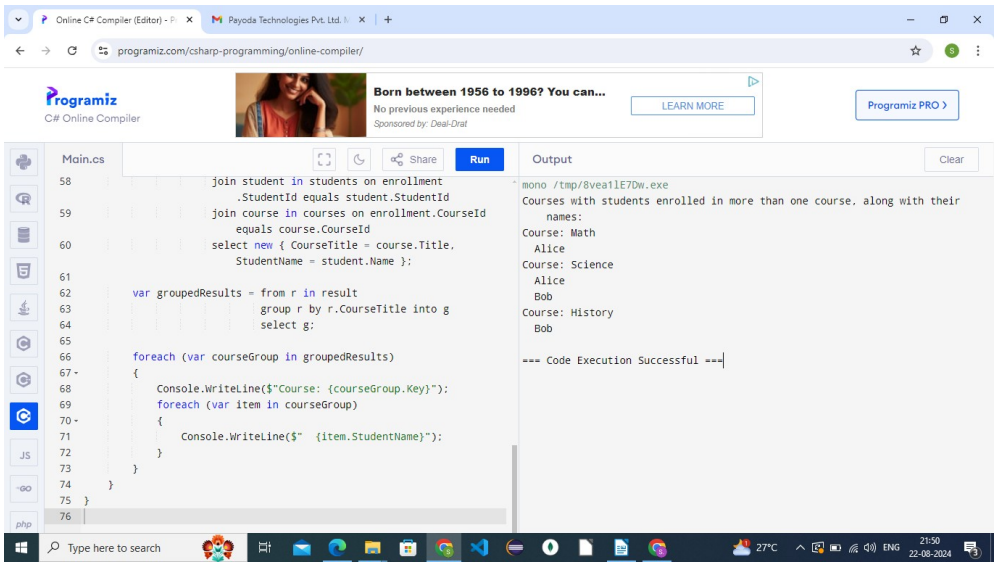
3. Join: Find the list of courses that have students enrolled in more than one course, along with the names of those students.

CODE:

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

public class Student
{
   public int StudentId { set; get; }
   public string Name { set; get; }
}

public class Course
{
   public int CourseId { set; get; }
   public string Title { set; get; }
}

public class Enrollment
{
   public int StudentId { set; get; }
   public int CourseId { set; get; }
}

public class Program
{
   public static void Main(string[] args)
   {
      var students = new List<Student>
      {
         new Student { StudentId = 1, Name = "Alice" },
         new Student { StudentId = 2, Name = "Bob" },
         new Student { StudentId = 3, Name = "Charlie" },
         new Student { StudentId = 4, Name = "David" }
      };

      var courses = new List<Course>
      {
         new Course { CourseId = 1, Title = "Math" },
         new Course { CourseId = 2, Title = "Science" },
         new Course { CourseId = 3, Title = "History" }
      };

      var enrollments = new List<Enrollment>
      {
         new Enrollment { StudentId = 1, CourseId = 1 },
         new Enrollment { StudentId = 1, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 3 },
```

```
            new Enrollment { StudentId = 3, CourseId = 1 },
            new Enrollment { StudentId = 4, CourseId = 2 }
        };

        Console.WriteLine("Courses with students enrolled in more than one course, along with their names:");

        var result = from e in enrollments
                     group e by e.StudentId into g
                     where g.Count() > 1
                     from enrollment in g
                     join student in students on enrollment.StudentId equals student.StudentId
                     join course in courses on enrollment.CourseId equals course.CourseId
                     select new { CourseTitle = course.Title, StudentName = student.Name };

        var groupedResults = from r in result
                             group r by r.CourseTitle into g
                             select g;

        foreach (var courseGroup in groupedResults)
        {
            Console.WriteLine($"Course: {courseGroup.Key}");
            foreach (var item in courseGroup)
            {
                Console.WriteLine($"  {item.StudentName}");
            }
        }
    }
}
```

CODE:



4. Order: Sort the courses by the number of students enrolled in descending order.

CODE:

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

public class Student
{
   public int StudentId { set; get; }
   public string Name { set; get; }
}

public class Course
{
   public int CourseId { set; get; }
   public string Title { set; get; }
}

public class Enrollment
{
   public int StudentId { set; get; }
   public int CourseId { set; get; }
}

public class Program
{
   public static void Main(string[] args)
   {
      var students = new List<Student>
      {
         new Student { StudentId = 1, Name = "Alice" },
         new Student { StudentId = 2, Name = "Bob" },
         new Student { StudentId = 3, Name = "Charlie" },
         new Student { StudentId = 4, Name = "David" }
      };

      var courses = new List<Course>
      {
         new Course { CourseId = 1, Title = "Math" },
         new Course { CourseId = 2, Title = "Science" },
         new Course { CourseId = 3, Title = "History" }
      };

      var enrollments = new List<Enrollment>
      {
         new Enrollment { StudentId = 1, CourseId = 1 },
         new Enrollment { StudentId = 1, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 2 },
         new Enrollment { StudentId = 2, CourseId = 3 },
         new Enrollment { StudentId = 3, CourseId = 1 },
```

```
            new Enrollment { StudentId = 4, CourseId = 2 }
        };

        Console.WriteLine("Courses sorted by the number of students enrolled (descending):");

        var sortedCourses = from e in enrollments
                            group e by e.CourseId into g
                            join course in courses on g.Key equals course.CourseId
                            orderby g.Count() descending
                            select new { CourseTitle = course.Title, StudentCount = g.Count() };

        foreach (var course in sortedCourses)
        {
            Console.WriteLine($"{course.CourseTitle}: {course.StudentCount} students");
        }
    }
}
```

OUTPUT: