**Lab 1: Write a C# program to create multidimensional array to store the marks of three student in different subjects. First student has marks of 3 subjects, second student has marks of 4 subjects and Third student has marks of 2 subjects, Display the subject marks and average marks for each student**

using System;

namespace R\_lab1

{

internal class Program

{

static void Main(string[] args)

{

int[][] studentMarks = new int[3][];

studentMarks[0] = new int[3]; // Student 1 - 3 subjects

studentMarks[1] = new int[4]; // Student 2 - 4 subjects

studentMarks[2] = new int[2]; // Student 3 - 2 subjects

Console.WriteLine("Enter marks for each student:");

for (int i = 0; i < studentMarks.Length; i++)

{

Console.WriteLine($"\nStudent {i + 1}:");

for (int j = 0; j < studentMarks[i].Length; j++)

{

Console.Write($"Enter mark for Subject {j + 1}: ");

studentMarks[i][j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("\n--- Student Marks and Averages ---\n");

// Displaying each student's marks and average

for (int i = 0; i < studentMarks.Length; i++)

{

Console.WriteLine($"Student {i + 1} Marks:");

int sum = 0;

for (int j = 0; j < studentMarks[i].Length; j++)

{

Console.WriteLine($"\tSubject {j + 1}: {studentMarks[i][j]}");

sum += studentMarks[i][j];

}

double average = (double)sum / studentMarks[i].Length;

Console.WriteLine($"\tAverage Marks: {average:F2}\n");

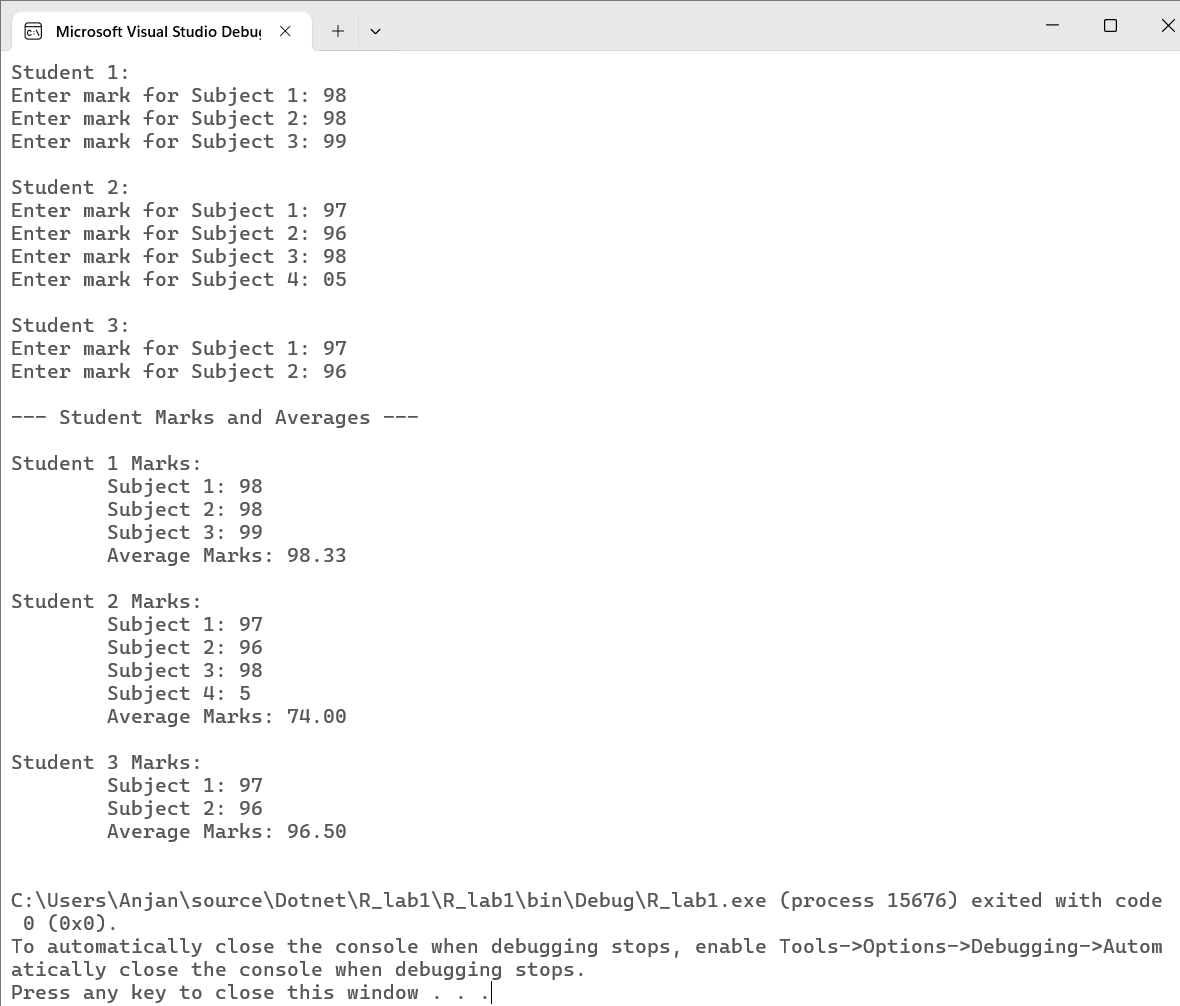
}

}

}

}

**Output:**



**Lab 2: Write a C# program to select odd and divisible by 3 number from list of numbers (1-30) using LINQ query**

using System;

using System.Collections.Generic;

using System.Linq;

namespace R\_lab2

{

internal class Program

{

static void Main(string[] args)

{

List<int> numbers = Enumerable.Range(1, 30).ToList();

// Step 2: LINQ query to get numbers that are odd and divisible by 3

var result = from num in numbers

where num % 2 != 0 && num % 3 == 0

select num;

Console.WriteLine("Numbers from 1 to 30 that are ODD and divisible by 3:");

foreach (int n in result)

{

Console.Write(n + " ");

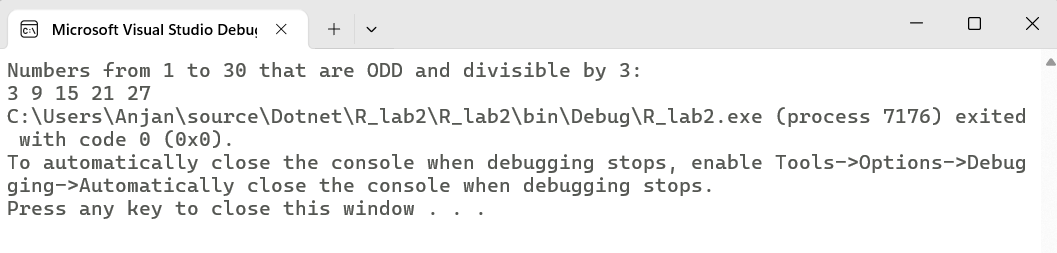
}

}

}

}

**Output:**



**Lab 3: Write a C# program to achieve dynamic binding using virtual method in C#**

using System;

namespace R\_lab3

{

internal class Program

{

class Animal

{

public virtual void Speak()

{

Console.WriteLine("The animal makes a sound.");

}

}

class Dog : Animal

{

public override void Speak()

{

Console.WriteLine("The dog barks.");

}

}

class Cat : Animal

{

public override void Speak()

{

Console.WriteLine("The cat meows.");

}

}

static void Main()

{

Animal a1 = new Dog();

Animal a2 = new Cat();

Animal a3 = new Animal();

a1.Speak();

a2.Speak();

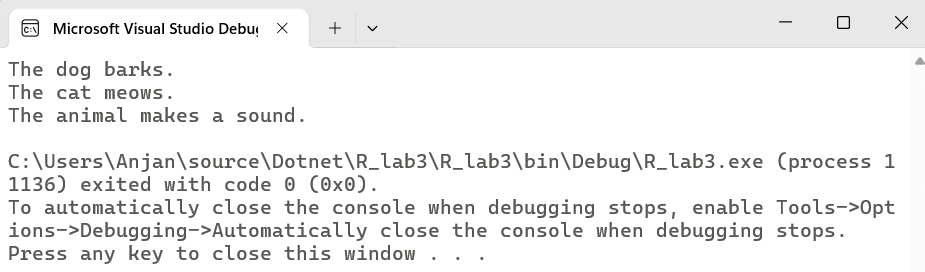
a3.Speak();

}

}

}

**Output:**

****