

## Assignment 1: 'Running'

(20 points)

Create a Console project named '**assignment1**' in a solution named 'Programming1-PracticeExam'.  
(later, assignment 2, 3, 4 and 5 are added as projects to the same solution)

Create a program that calculates the speed/pace of a runner.

- Write a method `double GetPace(double distance, int time)` that receives the distance in kilometers and the time in minutes, and returns the pace in minutes/kilometer.
- Write a method `double GetSpeed(double distance, int time)` that receives the distance in kilometers and the time in minutes, and returns the speed in kilometers/hour.
- In the `Start` method ask the user to enter the running distance (in kilometers) and the running time (in minutes).
- Next, ask the user to calculate the pace (min/km) or the speed (km/hr). Depending on the user's input, call method `GetPace` or `GetSpeed` and display the result. Display an error if the user does not enter "pace" or "speed".

Some output examples:

```
Assignment 1
Enter the running distance (in kilometers): 5.5
Enter the time taken (in minutes): 25

Do you want to calculate pace (min/km) or speed (km/hr)? pace
Your pace is 4.55 min/km
```

```
Assignment 1
Enter the running distance (in kilometers): 5.5
Enter the time taken (in minutes): 25

Do you want to calculate pace (min/km) or speed (km/hr)? speed
Your speed is 13.20 km/hr
```

```
Assignment 1
Enter the running distance (in kilometers): 21.1
Enter the time taken (in minutes): 108

Do you want to calculate pace (min/km) or speed (km/hr)? peace
Incorrect input
```

## Assignment 2: 'Prime factors'

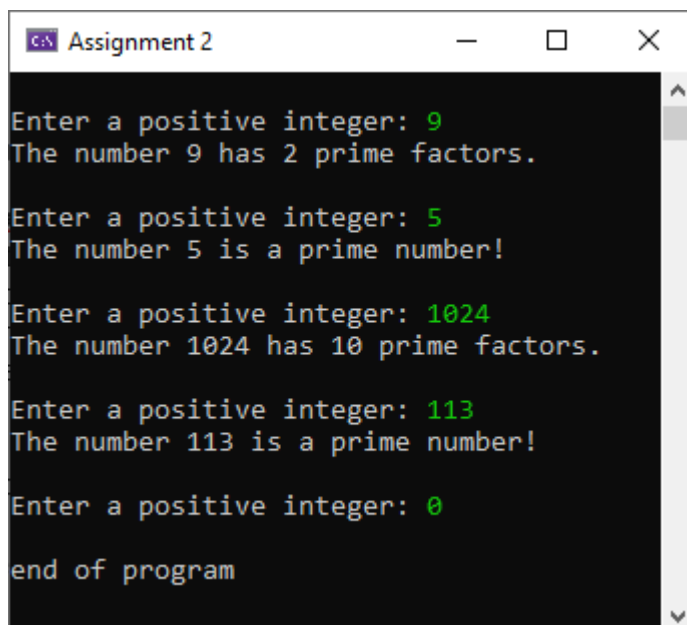
(20 points)

Add a Console project named '**assignment2**' to the solution 'Programming1-PracticeExam'.

Create a program that repeatedly asks the user to enter a positive integer and determines the number of prime factors of each entered number.

- Write a method `int CountPrimeFactors(int number)` that returns the number of prime factors for a given number. To count the prime factors of a number, follow the next procedure:
  - start with divisor = 2 and counter = 0;
  - while the number is bigger than 1: if the number is a multiple of divisor then divide the number by the divisor (and increase the counter), else increase the divisor by 1.
- In the `Start` method repeatedly prompt the user to enter a positive integer, call the `CountPrimeFactors` method to get the number of prime factors, and display the result (if the count is 1, display that the number is a prime number). Stop the program when the user enters a 0 or a negative number.

An output example:



```
Assignment 2
Enter a positive integer: 9
The number 9 has 2 prime factors.

Enter a positive integer: 5
The number 5 is a prime number!

Enter a positive integer: 1024
The number 1024 has 10 prime factors.

Enter a positive integer: 113
The number 113 is a prime number!

Enter a positive integer: 0
end of program
```

## Assignment 3: 'Random numbers'

(20 points)

Add a Console project named '**assignment3**' to the solution 'Programming1-PracticeExam'.

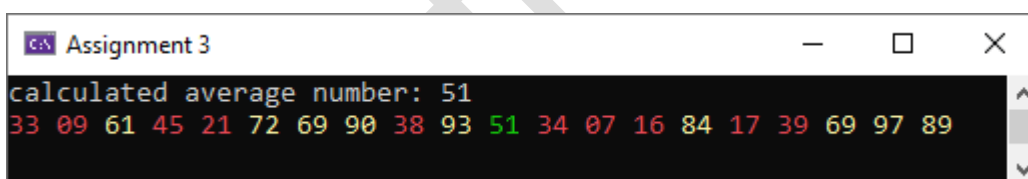
Create a program that fills and displays an array with random integer numbers.

- Write a method `void FillNumbers(int[] numbers)` that fills the array with random numbers (1 as lowest possible value and 99 as highest possible value).
- Write a method `int GetAverage(int[] numbers)` that returns the average (integer) value of all array numbers.
- Write a method `void DisplayNumbers(int[] numbers, int average)` that displays all numbers in the array using the following colors:
  - red for numbers lower than the average
  - yellow for numbers higher than the average
  - green for numbers equal to the average

Example: to display a text in green, use: `Console.ForegroundColor = ConsoleColor.Green.`

- In the `Start` method create an `int` array with 20 elements, and fill it by using method `FillNumbers`.
- Get the average via method `GetAverage` and display all array numbers using method `DisplayNumbers`.

An output example:



```
C:\> Assignment 3
calculated average number: 51
33 09 61 45 21 72 69 90 38 93 51 34 07 16 84 17 39 69 97 89
```

## Assignment 4: 'Ordering drinks'

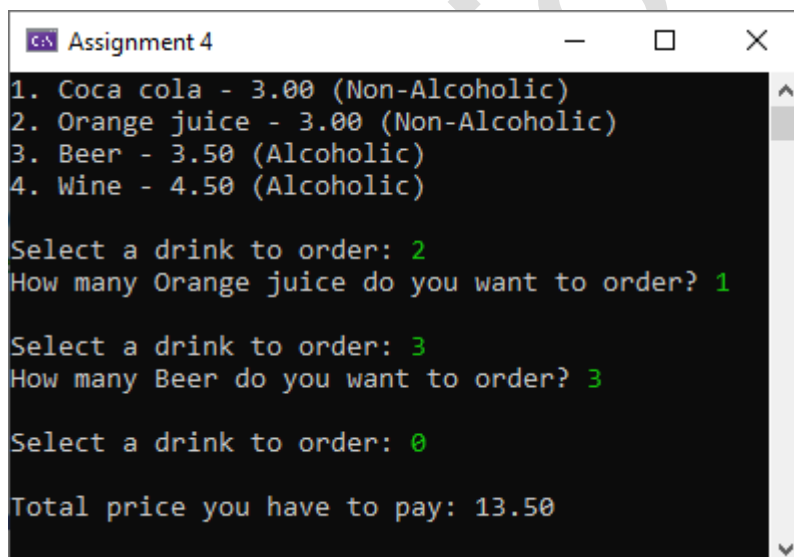
(20 points)

Add a Console project named '**assignment4**' to the solution 'Programming1-PracticeExam'.

Create a program to process drink orders.

- Create a Drink class with public fields `string` Name, `double` Price and `bool` IsAlcoholic.
- Give class Drink a constructor that initializes all fields: `public Drink(string name, double price, bool isAlcoholic)`.
- (in class Drink) Write a method `string` GetDescription() that returns a complete description for the drink (name, price and alcohol/non-alcohol).
- (in class Program) Write a method `void` DisplayDrinks(Drink[] drinks) that displays all drinks in the given array. Use object method GetDescription for each drink.
- In the Start method create 4 (hard coded) drinks and store these in an array of Drink objects.
- Display all available drinks with their details using method DisplayDrinks,
- Repeatedly ask the user to select a drink (0 stops the loop) and how many he/she wants to order.
- At the end, display the total amount to pay.

An output example:



```
Assignment 4
1. Coca cola - 3.00 (Non-Alcoholic)
2. Orange juice - 3.00 (Non-Alcoholic)
3. Beer - 3.50 (Alcoholic)
4. Wine - 4.50 (Alcoholic)

Select a drink to order: 2
How many Orange juice do you want to order? 1

Select a drink to order: 3
How many Beer do you want to order? 3

Select a drink to order: 0

Total price you have to pay: 13.50
```

## Assignment 5: 'Machine'

(20 points)

Add a Console project named '**assignment5**' to the solution 'Programming1-PracticeExam'.

Create a program that that uses a class `Machine` with several properties.

- Create a `Machine` class with a readonly property `string` `Name`.
- Give class `Machine` a read/write property `int` `NumberOfWorkingHours` that uses a backing field. Silently ignore negative hours.
- Give class `Machine` a constructor that initializes all properties: `public Machine(string name, int numberOfWorkingHours)`.
- Give class `Machine` a calculated property `bool` `NeedsMaintenance` that returns `true` if the number of workings hours is at least 10.000, else it returns `false`.
- (in class `Machine`) Write a method `void DisplayInfo()` that displays the details of a machine.
- (in class `Program`) Write a method `Machine ReadMachine()` that creates a `Machine` object based on user input, and returns this machine.
- In the `Start` method create a machine object (using method `ReadMachine`) and display this machine (using object method `DisplayInfo`).

Some output examples:

```
Assignment 5
Enter machine name: 3D printer
Enter number of working hours: 3500

Machine name: 3D printer
Number of working hours: 3500
Needs maintenance: no
```

```
Assignment 5
Enter machine name: CNC Router
Enter number of working hours: 12450

Machine name: CNC Router
Number of working hours: 12450
Needs maintenance: yes
```