3.7-Byte, Short, Long Primitive Datatypes

Program

```
byte variable_byte = 50;

Console.WriteLine(variable_byte);


sbyte SignedByteVariable = -50;

Console.WriteLine(SignedByteVariable);


short variable_short = 500;

Console.WriteLine(variable_short);


ushort unsignedShortVariable = 200;

Console.WriteLine(unsignedShortVariable);


long variable_long = 5000;

Console.WriteLine(variable_long);


ulong variable_ulong = 2000000000000000000;

Console.WriteLine(variable_ulong);
```

Exercise

Write a C# program in which initialize and assigns values to various numeric variables in C#.

It begins by declaring a byte variable named variable_byte and assigns it the value of 50.

Subsequently, a sbyte variable named Signed Byte Variable is declared with a value of -50. Following this, a short variable named variable_short is declared and set to 500. An unsigned ushort variable, unsignedShortVariable, is declared with a value of 200.

Then, a long variable named variable_long is initialized to 5000. Finally, a ulong variable named variable_ulong is declared and assigned the value 2000000000000000000. Each variable's value is printed to the console using Console.WriteLine() for display.

Hint

Variable Declarations and Initialization: Declare variables of appropriate data types (byte, sbyte, short, ushort, long, ulong).

Initialize each variable with the specified values (50, -50, 500, 200, 5000, 2000000000000000000).

Print Statements: Use Console.WriteLine() statements to display the values of each variable. Ensure each variable's value is printed on a separate line.

Data Type Selection: Choose the correct data type for each variable based on the range of values it can hold.

For unsigned values, use ushort and ulong.

Explanation

Each variable represents a different range of values, from smaller integers (byte, sbyte) to larger ones (short, ushort) and very large integers (long, ulong).

By using appropriate data types, the code ensures efficient memory usage and accurate representation of the assigned values.