

```

using System.Runtime.InteropServices;
using System.Collections;
using System.Globalization;

public class Program
{
    static void Main()
    {
        Console.WriteLine("== RuntimeInformation ==");

        Console.WriteLine($"OS Architecture : {RuntimeInformation.OSArchitecture}");
        Console.WriteLine($"Process Architecture : {RuntimeInformation.ProcessArchitecture}");
        Console.WriteLine($"Framework Description : {RuntimeInformation.FrameworkDescription}");
        Console.WriteLine($"OS Description : {RuntimeInformation.OSDescription}");
        Console.WriteLine($"Runtime Identifier : {RuntimeInformation.RuntimeIdentifier}");
        Console.WriteLine($"Is OS Windows : {RuntimeInformation.IsOSPlatform(OSPlatform.Windows)}");
        Console.WriteLine($"Is OS Linux : {RuntimeInformation.IsOSPlatform(OSPlatform.Linux)}");
        Console.WriteLine($"Is OS macOS : {RuntimeInformation.IsOSPlatform(OSPlatform.OSX)}");

        Console.WriteLine("-----");

        Console.WriteLine("== Environment ==");

        Console.WriteLine($"CLR Version : {Environment.Version}");
        Console.WriteLine($"Command Line : {Environment.CommandLine}");
        Console.WriteLine($"Current Directory : {Environment.CurrentDirectory}");
        Console.WriteLine($"Managed Thread ID : {Environment.CurrentManagedThreadId}");
        Console.WriteLine($"Exit Code : {Environment.ExitCode}");
        Console.WriteLine($"OS Version : {Environment.OSVersion}");
        Console.WriteLine($"Machine Name : {Environment.MachineName}");
        Console.WriteLine($"User Name : {Environment.UserName}");
        Console.WriteLine($"User Domain : {Environment.UserDomainName}");
        Console.WriteLine($"System Directory : {Environment.SystemDirectory}");
        Console.WriteLine($"Processor Count : {Environment.ProcessorCount}");
        Console.WriteLine($"Is 64 Bit OS : {Environment.Is64BitOperatingSystem}");
        Console.WriteLine($"Is 64 Bit Process : {Environment.Is64BitProcess}");
        Console.WriteLine($"System Uptime (ms) : {EnvironmentTickCount64}");
        Console.WriteLine($"App Base Directory : {AppContext.BaseDirectory}");
        Console.WriteLine($"Temp Path : {Path.GetTempPath()}`);

        Console.WriteLine("== Environment Variables ==");

        foreach (DictionaryEntry item in Environment.GetEnvironmentVariables())
        {
            Console.WriteLine($"{item.Key} = {item.Value}");
        }

        Console.WriteLine($"Target Framework : {AppContext.TargetFrameworkName}");
        Console.WriteLine($"App Context BaseDir : {AppContext.BaseDirectory}");

        Console.WriteLine($"Current Culture : {CultureInfo.CurrentCulture}");
        Console.WriteLine($"Current UI Culture : {CultureInfo.CurrentUICulture}");
    }
}

```