1. Find out the Details on System.Console and Environment classes and try to print the   
   Back Ground Color  
   Foreground Color  
   Make the console Beep for 3 times  
   Print the machine Name  
   Print the Logged in User name  
   Print whether the Machine is 64 Bit or not.  
   Use the appropriate classes in the .net framework to achieve this.
2. Using the DirectoryInfo and The File classes in the System.IO namespace refer the documentation and perform the following operations  
   - Check the system whether a Directory exists  
   - If not exists create the Directory   
   - Create 3 text files with Your name , Company Name and Place of work inside the newly created directory  
   - Get the details of all the newly created file Like Creation Time, FullName, Size of the file   
   - Display all the information on the console.  
   Once completed submit both the code file and the screenshot after execution.
3. Using the System.Environment class , add support to your existing Dashboard application to display   
   - CommandLine  
   - UserName  
   - UserDomainName  
   - MachineName  
   Write the code that displays these details in a separate library called MyUtilities.dll and use this dll to display the above details. MyUtilities.DisplayEnvironment()  
     
   Note: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /reference:MyLibrary.dll HelloDashboard.cs  
     
   C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /target:library MyLibrary.cs
4. Referring to the below given method definition , understand the various parameter passing conventions and invoke the method using   
   - No arguments  
   - Positional arguments followed by a sequence of remaining arguments  
   - Calling by Named Parameter convention  
   static void SimpleMethod(int number = 555, string str ="DEFAULT", bool option = true, params string[] remaining) {  
               number = 100;  
               str = "Changed in Method";  
               Console.WriteLine($"Str in Method{str}- Number in Method {number} The Bool Value is {option}");  
               Console.WriteLine($"The all trailing params got here is {remaining.Length}");  
           }
5. Provide overloads of the addNumbers method so that the following calls are correct  
   addNumbers(0,0);  
     
   addNumbers(1.0f,1.0f);  
     
   addNumbers(1.0,1.0);  
     
   addNumbers(1,1.0f);  
     
   addNumbers(1.0f,1);  
     
   addNumbers(1.0f, 1.0f, "This is with additional inputs");  
     
     
     
   public <type> addNumbers(int input1, int input2)  
           {  
                Console.WriteLine($"Add Numbers called with  {input1.GetType()} and {input2.GetType()}");  
     
              // Processing  
               <type> output = input1 + input2;  
               return output;  
           }