# Pointers + unsafe + fixed

1. Create variable “text“ and fill it with random string of length 1000.  
   *Help: text += (char)(65 + rand.Next() % 40);*
2. Create function with one string parameter. This function will count and return number of “c” characters in the string.
3. Call this function for each possible substring of the string from the 1st task.
4. Create the same function as in 2nd and 3rd task, but this time use pointers to pass string parameter to the function (use unsafe and fixed).
5. Use class Stopwatch to measure time spend by execution of the first and then of the second function.

# ASP.NET Core

1. Create empty ASP.NET Core application.
2. Create and use middleware “PageMiddleware”. This middleware will be able to serve 3 different “pages” based on the current URL. Use property “Path” of “Request” to access current URL.
   1. 1st page will return HTML: „<h1>Homepage</h1>“
   2. 2nd page will return HTML: „<h1>Second page</h1>”
   3. 3rd page will throw some exception.
3. Create and use middleware “ExceptionHandlerMiddleware”. Put this middleware before the first one and implement exception handling = catch exceptions from other middlewares and generate user-friendly error page with information about what happened.
4. Create service “ErrorLogger”. This service will be able to write information about handled exceptions to the text file (log). Then use this service in your ExceptionHandlerMiddleware.