

## TASK 1: GENERAL TESTING

- How could you determine whether the editor is displaying text in the correct typeface?
  - I would first gather test data on the most commonly used typefaces like Helvetica, Arial, Times New Roman etc and some uncommon ones too
  - Create a set of test cases with the predefined text/fonts gathered from the test data
  - Launch the text editor and select one of the fonts from the test data
  - Paste the predefined text and check that the font and font size are set to the correct values
  - Cross check the displayed text from the text editor and compare it with the expected font
  - Record the results including any discrepancies between them
  - Repeat the tests with each font selected for testing
- How could you determine whether the editor is displaying text in the right size?
  - Similar to the previous question. I would do the same process to determine whether the editor is displaying text in the right size
  - 
  - Create a set of test cases with the predefined text/fonts and sizes with a range from small to large
  - Launch the text editor and select one of the fonts and fonts sizes
  - Paste the predefined text and check that the font and font size are set to the correct values.
  - Cross check the displayed text from the text editor and compare it with the expected font size.
  - Record the results including any discrepancies between them
  - Repeat the tests with each font selected for testing
- How would you test whether the editor displays text correctly (in any font that the user specifies)?
  - To test this, I would use a sample text that includes different type of characters (lower/upper case letters, numbers, special characters). Then I would use the data gathered from the tests used for the typeface and text size and test them.

Overall my main scope would be to target the most common used fonts and sizes. However, I would also include a diverse sets of fonts to ensure edge cases are covered too.

If a user happens to use a font that was not included in our tests and reports an issue, or if we are able to track the error using an error tracker software like Sentry, I would manually check the font and size and add it to our test suite to ensure it is covered in future tests. However, as mentioned in the task question, it is not possible to test every font and size. Therefore, I would focus on testing the most commonly used fonts, while also including some additional fonts and sizes to cover as many scenarios as possible.

Additionally, I would use a font validator to identify whether the problem is coming from the font file itself or the program.