Evaluative Report – Text Summarisation Tool

Strengths & Limitations

Some of the strengths for this application include the ability to input any text file that you would like summarising, meaning that it can be used for a wide range of subject areas to get the key points of any text. Another strength of this tool is that if the summarisation factor that the user wanted isn’t physically possible by keeping all the words whole, the console will output the summarisation factor actually used, this maintains integrity for the user as they may need to know the precise amount the file was summarised by.

Some limitations for this tool are that the output is only in sentences; the paragraphs don’t flow from one sentence to the next as it is displaying the sentences with the highest count of ‘important’ words. Another limitation for this tool is that some of the characters that are used for breaking sentences are used within words meaning that it breaks a word up and it doesn’t read naturally. An example of this is the use of ‘.’ for writing ‘Dr.Name’ this would split the word into two sentences as the full stop is being used. If I had more time to work on the program I would have removed the count of the key word from the start of the output sentence.

I used a variety of lists, arrays and dictionaries to store different variables within this project, I mainly used lists as these were what I have had most experience with, ultimately this was a good choice as the lists worked throughout. I attempted to use a temporary dictionary instead of saving the temporary variables to a file, but this got too confusing when I tried to access just the words (Keys) to write out the final summarisation. I used arrays to hold the characters that split the words and sentences as there was no need to add to this string of characters later in the program.

When calculating the summarisation factor, I tried to use as little ‘Parsing’ as possible as I found my original attempt, which contained multiple parsing, to run slightly slower, this also made the code look messy and confusing for someone reading it later. One problem did arise from the messiness of my code in the later stages of development, I found that I had an unnecessary function when creating the list of sentences meaning I was setting the sentences into a new list without changing anything then trying to access the empty list later in the code. This meant my output was empty as the sentence list was empty. To fix this I simply removed the meaningless function and renamed the variables later in the code.

*Main*

char [] splitForWords

string [] splitForSentences

string tempString

Streamreader sr

list wordlist

dictionary countOfWords

GetUserInput()

Process()

Output()

*Process*

list indexList

int maxCount

int index

int userWordLength= wordListInSentence Lists.Count

string maxSentence int maxIndex

int count

list resultStringList

int actualIndex

Gets; - summarised document

-Actual summarisation factor

*GetUserInput*

string userInput

string sumFileTemp

string sumFile

string stopFileTemp

string stopFile

string userSumFactor

string sumFactTemp

string sumFact

string sumFileTemp

Gets; - user file

-summarisation factor

*Output*

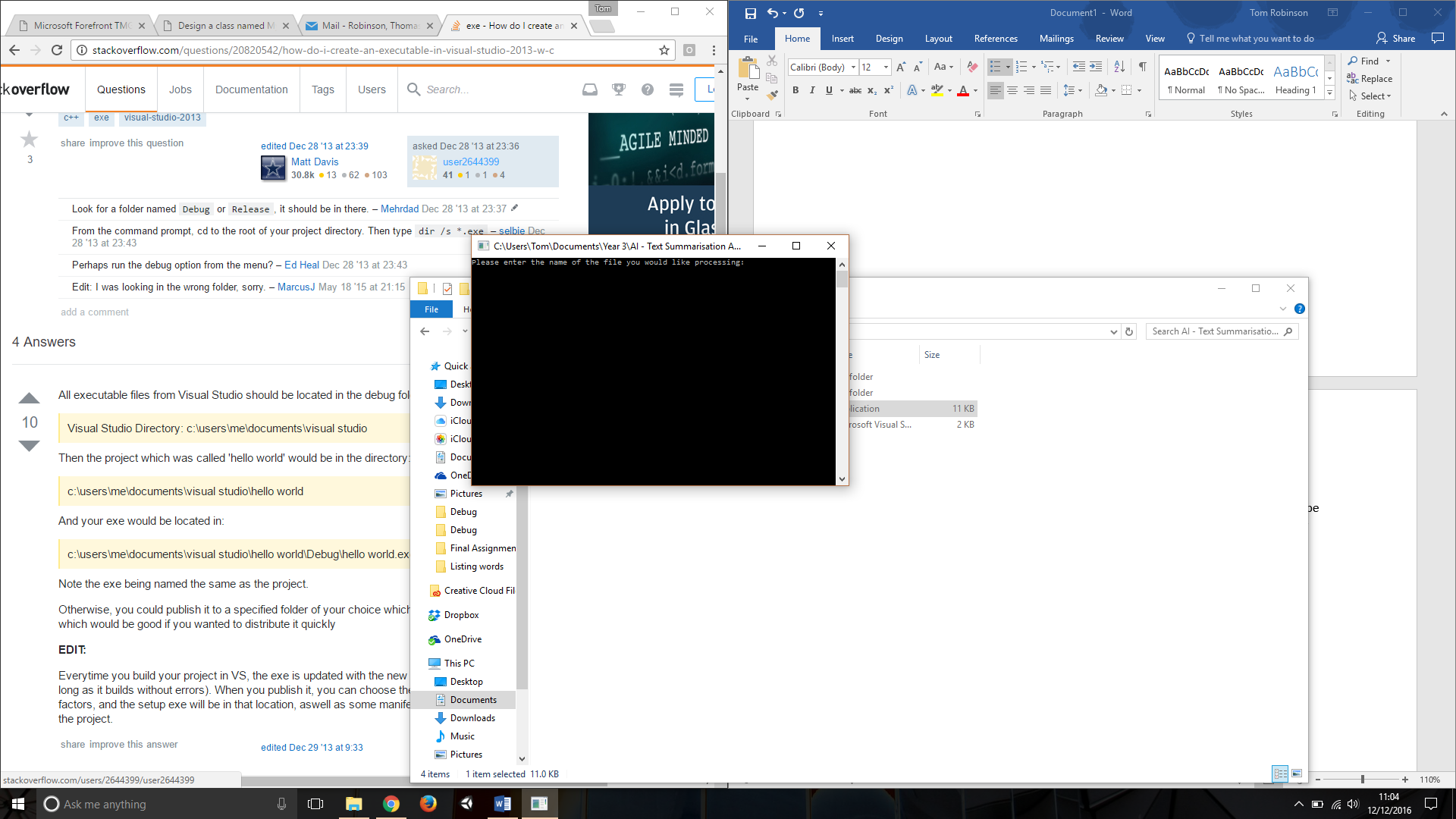
List summarisedDoc

List sentencesOut

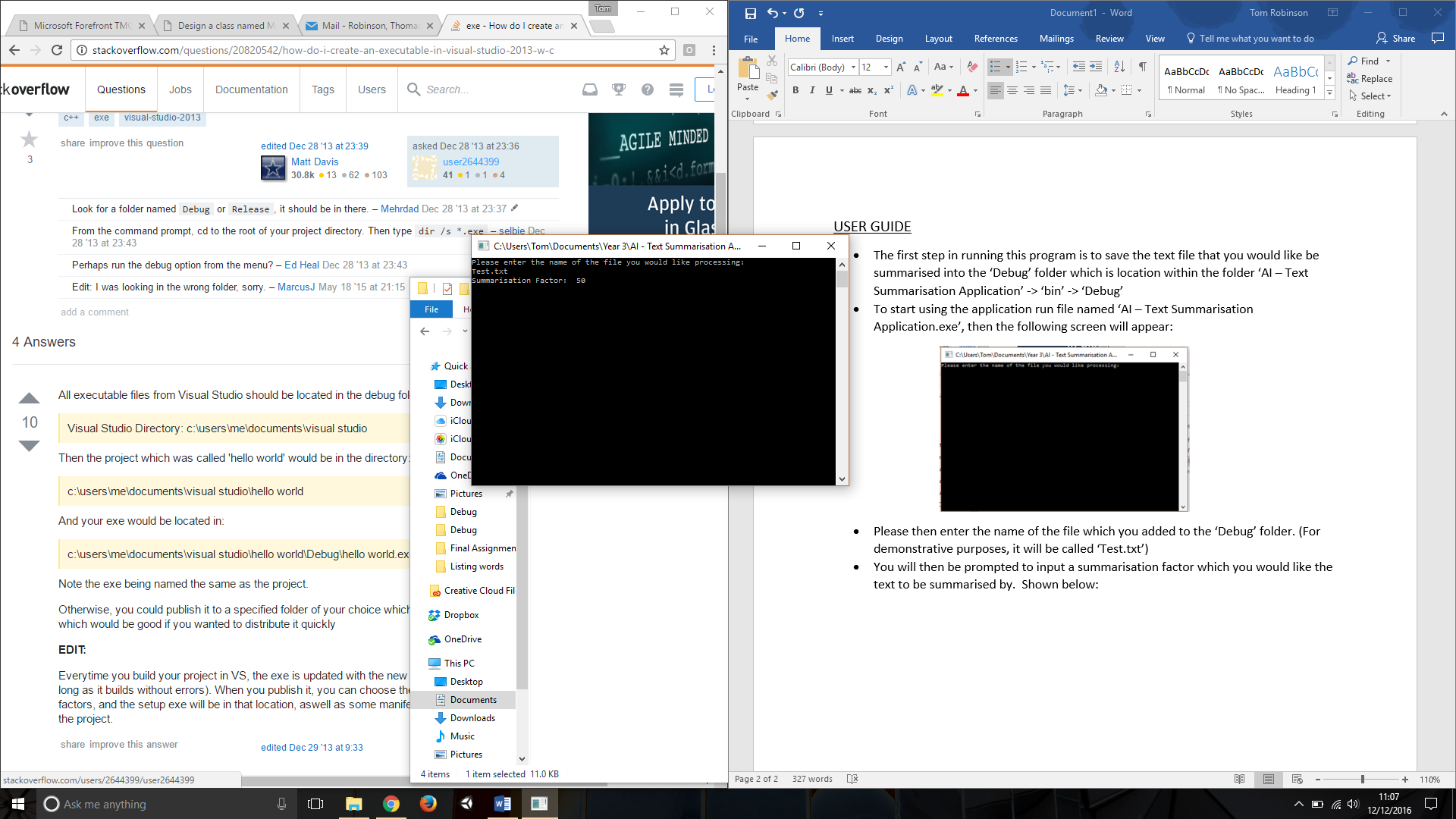
Gets; - summarised document to file

USER GUIDE

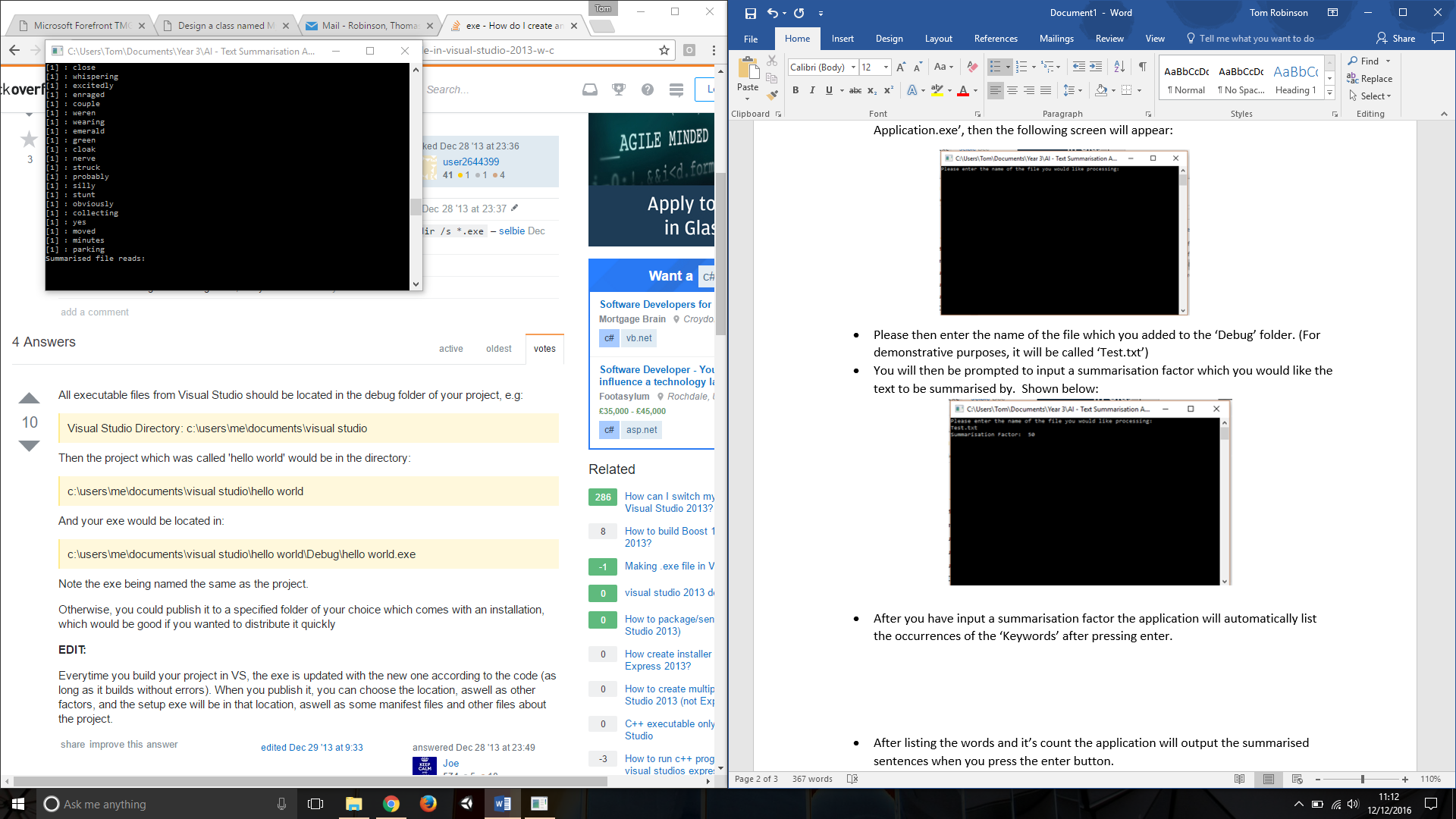
* The first step in running this program is to save the text file that you would like be summarised into the ‘Debug’ folder which is location within the folder ‘AI – Text Summarisation Application’ -> ‘bin’ -> ‘Debug’
* To start using the application run file named ‘AI – Text Summarisation Application.exe’ (this will be within the ‘Debug’ folder), then the following screen will appear:

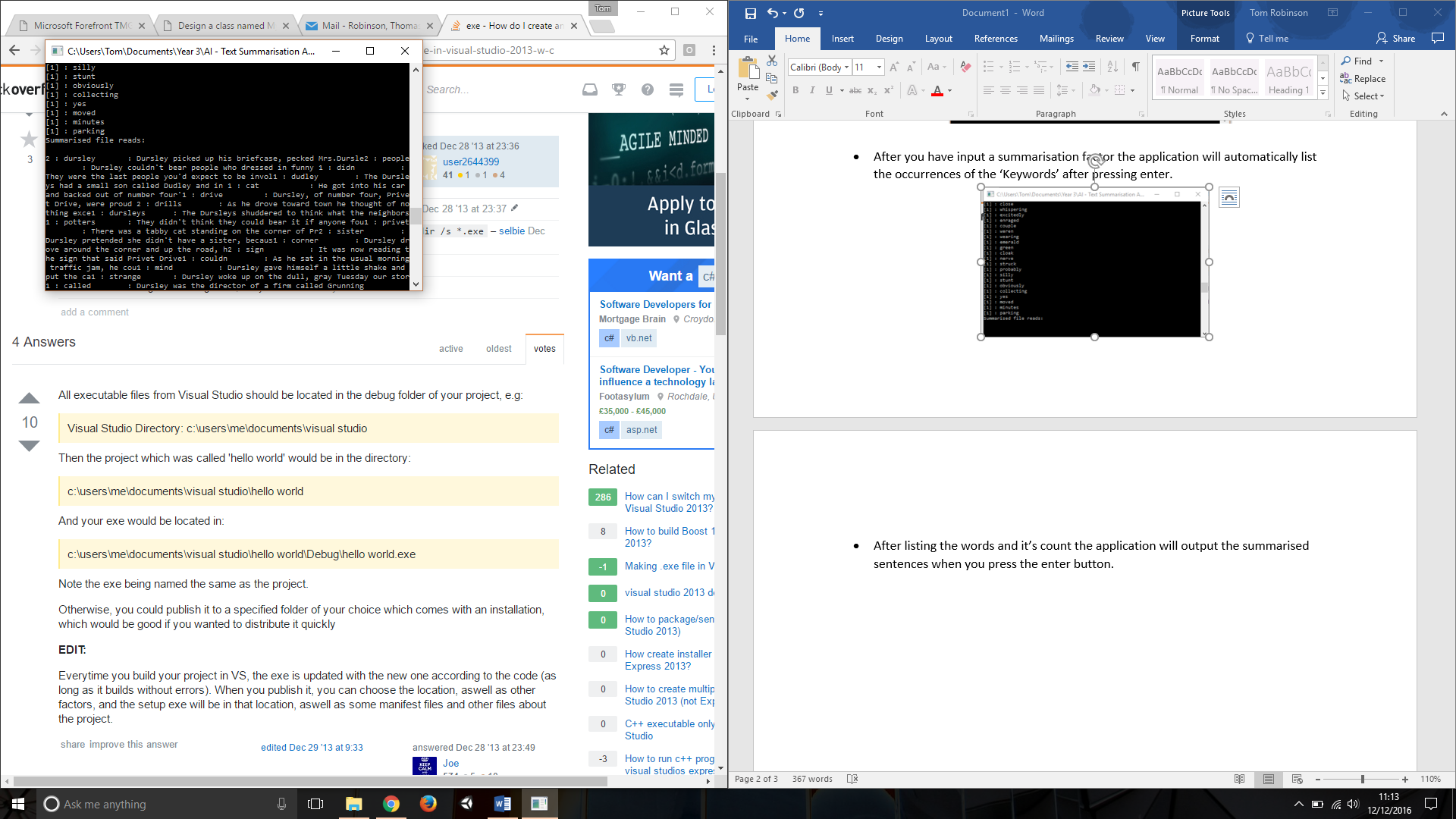


* Please then enter the name of the file which you added to the ‘Debug’ folder. (For demonstrative purposes, it will be called ‘Test.txt’)
* You will then be prompted to input a summarisation factor which you would like the text to be summarised by. Shown below:



* After you have input a summarisation factor the application will automatically list the occurrences of the ‘Keywords’ after pressing enter.



* After listing the words and it’s count the application will output the summarised sentences when you press the enter button.
* After the sentences have been displayed, press enter again and the application will show you the actual summarisation percentage, after pressing enter once more it will tell you what the file name is for the summarised text. This file will be within the same folder you saved your original text document.

