**Task 3 Analysis and Design – Tom H**

***Introduction:***

The problem:

File compression is an effective method of taking texts and shortening them down to reduce transfer times. Effective compression methods need to be fast but lossless so I need to make a program that can read a text file and compress then decompress the contents. I need to find a way of taking a text file and compressing it so it is easier to transfer; then I have to decompress the file and return it to its original state.

How I intend to solve the problem:

The program I will make will allow the user to either make a new text file or open a premade one for compression. The program will take this text file and compress it as a separate file. The same program should be able to open an existing text file that has been compressed and decompress it. It should return the original file including all the relevant punctuation and capitalisation.

Success Criteria:

My success criterion for my program is:

* My program should be able to save a text file to the user’s desired folder.
* My program should be able to take this text file and compress it down into as simple a form as possible.
* My program should be able to open a pre-existing text file that has already been compressed and decompress it.
* My program must be aware of any punctuation and capitalisation and make sure that it’s compressed and decompressed with the text file.

Testing:

In order to test my program I must create my program in each stage of the success criteria and validate that it works before continuing on to the next point. After every change, I will debug the program to check for errors and make note of each major change in a development document. Once my program works successfully, I will need to go through a long testing document that checks any validation that I might be required.

***Design:***

Inputs and Outputs:

*Inputs:*

* The text entered by the user
* The chosen file the user wants to decompress

Outputs:

* A compressed sentence

1. A list of unique words
2. A list of unique positions

* A decompressed sentence

User Interface:

\*See sheet of paper!\*

Visual Basic: This is a high level programming language that allows you to make a program with efficient code and an intuitive design system.

Flow Chart:

\*See sheet of paper!\*

Pseudocode:

Validation:

To make my program as efficient as possible I need to use validation to overcome and potential problems that may cause errors in the program:

* Has the user entered a file name?
* Has the user cancelled out of the save file dialog box?
* Has the user entered a sentence?
* Is there any punctuation or capitalisation in the text?

Variables: