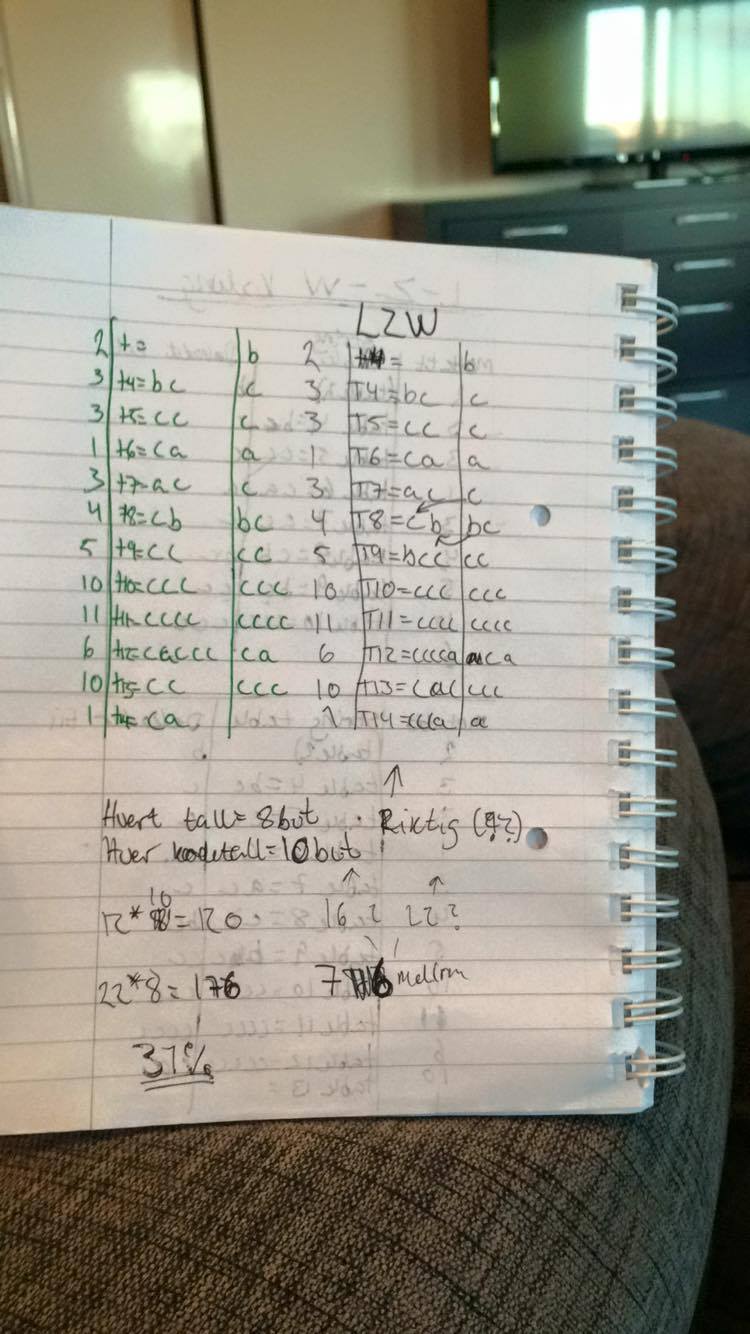
ICA 03

**b) ImplementerLZWalgoritmeiPythonbasertpåpseudokodefrateksten(teori**

**for uke05).**

Vi valgte å importere en LZW algoritme fra nettet for å spare tid.

<https://pypi.python.org/pypi/lzw/>

<https://github.com/joeatwork/python-lzw>

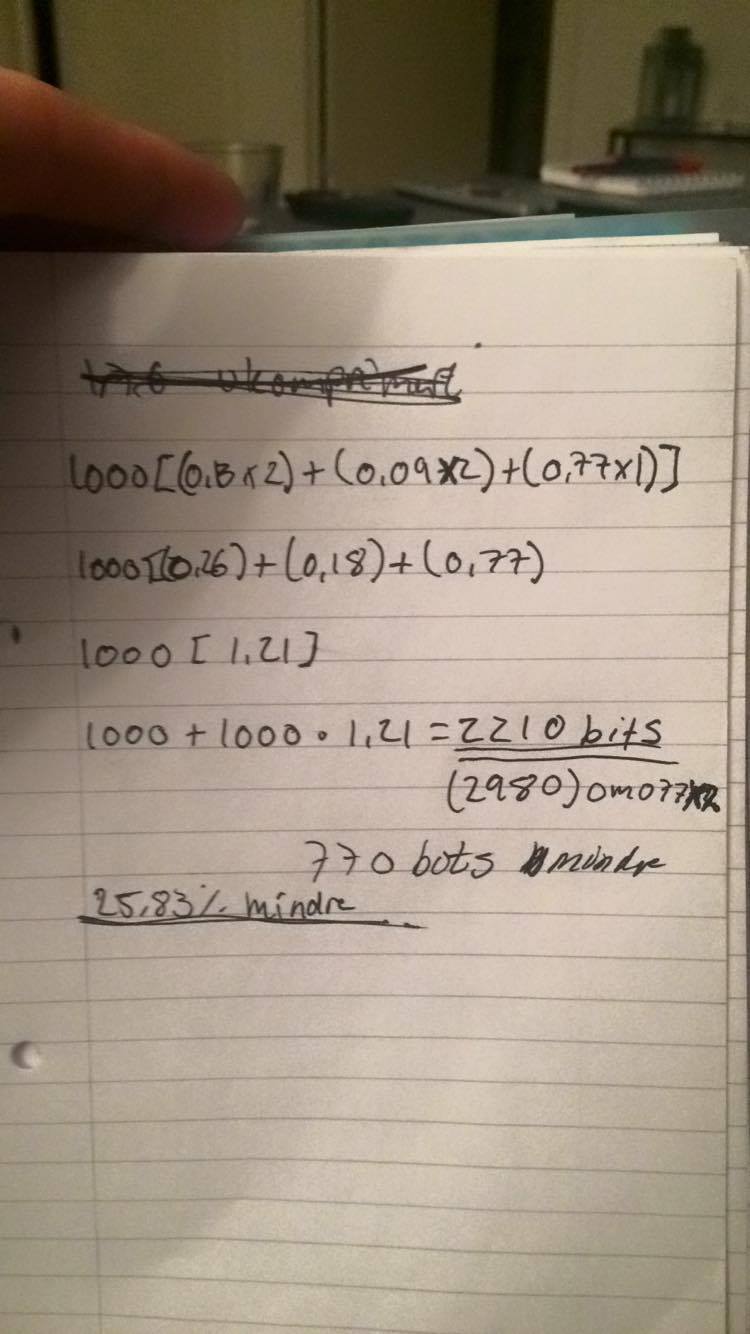
**c) Beregne komprimeringsgraden (i prosent) for dette tilfelle:**

Våre beregninger tilsier at komprimeringsgraden for LZW i dette tilfellet er 31%

**d) Basert på frekvensen av forekomsten for de tre symbolene(a,b,c),lag en Huffman kode, kode meldingen og beregn prosentvis komprimeringsgrad. Hvordan er den i forhold til LZW caset? Forklar.**

(Se bilder nedenfor!)

Huffman koden har en komprimeringsgrad på 25.83%. Dette er betydelig mindre enn komprimeringsgrad ved bruk av LZW-metoden. I meldingens nåværende form vil antall bytes spart være få, men ved større filer eller meldinger vil forskjellene bli større.

**f) Finne kompr****imeringsgraden for begge tekstfilene. Reflekter!**

Vi kodet og komprimerte to tekstfiler: Hamlet på 184KB og Complete Shakespeare på 5.6MB(5590KB). Etter koding var størrelsen på Hamlet 98KB og Complete Shakespeare 2.9MB(2863KB)

Det vil si at komprimeringsgraden for Hamlet var på ca 53%, mens for Complete Shakespeare var den ca 51%. Prosentvis er ikke forskjellen stor men jo større filen er jo større lagringsplass spares ved komprimering. (50% av 1000 er mer en 50% av 100.)