ICA – Uke 6

# Oppgave 1.2.1

a)

Table 1 – A, Table 2 – B, Table 3 - C

|  |  |  |
| --- | --- | --- |
| **Mottatt** | **String Table** | **Dekodet** |
| 2 | Table 2 | B |
| 3 | Table 4 - bc | C |
| 3 | Table 5 – cc | C |
| 1 | Table 6 – ca | A |
| 3 | Table 7 – ac | C |
| 4 | Table 8 – cb | Bc |
| 5 | Table 9 – bcc | Cc |
| 10 | Table 10 – ccc | Ccc |
| 11 | Table 11 – cccc | Cccc |
| 6 | Table 12 - ccccc | ca |

**Kode er:**

B C C A C B C C C C C C C C C C C A

Noen ASCII tabeller gir uttrykk for at hvert symbol består av 7 bits, mens i teksten fra uke 5 står det at det er 8 bits. Vi tar derfor utgangspunkt i at det er 8 bits.

Dette gir oss da – 144 bits når hvert symbol 8 bits.

b) ***Se Python fil Oppgave 1.2.1 B***

c)

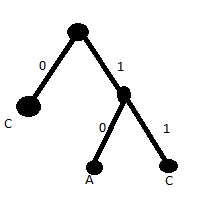
d) Huffman kode:

B C C A C B C C C C C C C C C C C A

A forekommer 2 ganger, B forekommer 2 ganger, C forekommer 14 ganger

Det gir en sannsynlighet på:

C = 0.78 ~ 0.8, A = 0.1, B = 0.1

 Koden blir da: 1000100110000000000010 = 22 bits

Original melding var på 44 bits, og vi har med Huffman koding fått en komprimeringsgrad på ~85%

e)

f) Ved å komprimere Shakespear filen, klarte vi å få den til å gå fra 2,080kb til 2,448kb

Dette gir en prosentvis nedgang på ca 15%

2448 – 2080 = 368

(368/2448) x 100 = 15.03%