

Codifica di huffman Ferrari, S4784573

Per prima cosa ho calcolato i singoli caratteri , le relative occorrenze e la probabilità con cui i caratteri capitano nel file:

```
Il carattere L volte: 11 con una probabilita' del: 0.011 %
Il carattere o volte: 3396 con una probabilita' del: 3.396 %
Il carattere r volte: 4428 con una probabilita' del: 4.428 %
Il carattere e volte: 9477 con una probabilita' del: 9.477 %
Il carattere m volte: 3590 con una probabilita' del: 3.59 %
Il carattere volte: 14636 con una probabilita' del: 14.636 %
Il carattere i volte: 7917 con una probabilita' del: 7.917 %
Il carattere p volte: 1949 con una probabilita' del: 1.949 %
Il carattere s volte: 6936 con una probabilita' del: 6.936 %
Il carattere u volte: 7651 con una probabilita' del: 7.651 %
Il carattere d volte: 2089 con una probabilita' del: 2.089 %
Il carattere l volte: 5042 con una probabilita' del: 5.042 %
Il carattere t volte: 6461 con una probabilita' del: 6.461 %
Il carattere a volte: 6293 con una probabilita' del: 6.293 %
Il carattere , volte: 1936 con una probabilita' del: 1.936 %
Il carattere c volte: 3229 con una probabilita' del: 3.229 %
Il carattere n volte: 4801 con una probabilita' del: 4.801 %
Il carattere g volte: 1164 con una probabilita' del: 1.164 %
Il carattere . volte: 2273 con una probabilita' del: 2.273 %
Il carattere A volte: 140 con una probabilita' del: 0.14 %
Il carattere C volte: 223 con una probabilita' del: 0.223 %
Il carattere q volte: 979 con una probabilita' del: 0.979 %
Il carattere b volte: 1090 con una probabilita' del: 1.09 %
Il carattere D volte: 181 con una probabilita' del: 0.181 %
Il carattere f volte: 529 con una probabilita' del: 0.529 %
Il carattere N volte: 277 con una probabilita' del: 0.277 %
Il carattere j volte: 86 con una probabilita' del: 0.086 %
Il carattere v volte: 1127 con una probabilita' del: 1.127 %
Il carattere I volte: 130 con una probabilita' del: 0.13 %
Il carattere h volte: 493 con una probabilita' del: 0.493 %
Il carattere V volte: 212 con una probabilita' del: 0.212 %
Il carattere P volte: 446 con una probabilita' del: 0.446 %
Il carattere Q volte: 64 con una probabilita' del: 0.064 %
Il carattere E volte: 96 con una probabilita' del: 0.096 %
Il carattere M volte: 140 con una probabilita' del: 0.14 %
Il carattere S volte: 220 con una probabilita' del: 0.22 %
Il carattere F volte: 137 con una probabilita' del: 0.137 %
Il carattere y volte: 54 con una probabilita' del: 0.054 %
Il carattere ; volte: 33 con una probabilita' del: 0.033 %
Il carattere U volte: 63 con una probabilita' del: 0.063 %
Il carattere volte: 1 con una probabilita' del: 0.001 %
```

Attraverso la funzione Entropy ho calcolato l'entropia di shannon:

$$\sum_i p_i \log_2 \frac{1}{p_i}$$

l'entropia vale : 4.250931125309729

E con la funzione `lunghezzaAttesa()` trovo la lunghezza attesa della codifica, calcolata nel seguente modo:

$$\sum_{x \in \mathcal{X}} p(x) L_C(x).$$

Ottenendo che:

La lunghezza attesa della codifica è': 4.28

Dopo di che costruisco un albero contenente un nodo per ogni carattere, e fino a quando non rimane un solo nodo nella lista prendo i due con probabilità minore, li assegna come codifica un 1 o uno 0, li rende figli di un nodo padre contenente come probabilità la somma dei figli e infine rimuove quest'ultimi dalla lista e aggiunge il corrispondente nodo padre.

Ad ogni iterazione devo cambiare la priorità la lista in modo da avere nelle prime due posizioni sempre i nodi con probabilità minore.

Infine, la funzione `compressioneFile` si occupa di creare il file compresso sostituendo ad ogni carattere la rispettiva codifica.

La compressione ottenuta è di : **La compressione è': 428282**

Dal momento che l'entropia si avvicina moltissimo alla lunghezza attesa possiamo dire che è a ridondanza nulla e che non ci sono margini di miglioramento per la compressione ottenibile.