

Esercizio 1

Bisogna realizzare una libreria in grado di comprimere e decomprimere **file qualsiasi** con un algoritmo di tipo RLE. L'interfaccia dovrà essere realizzata con un file .h e un file .cpp che implementino le seguenti funzioni:

```
bool rlecomp(std::istream& is, std::ostream& os);  
bool rledecomp(std::istream& is, std::ostream& os);
```

L'algoritmo funziona così: se c'è una sequenza di 2 o più caratteri ripetuti, manda in output un bit a 1, poi un numero binario a 6 bit da 1 a 63 che indica di ripetere 2-64 volte il byte successivo, altrimenti manda in output un bit a 0, poi un numero binario a 6 bit da 0 a 63 che indica di copiare i 1-64 byte successivi. La sequenza viene terminata con un bit a 1 seguito da 6 bit a 0. L'ultimo byte viene riempito con un padding di zeri.

Se ci sono sequenze, ripetute o meno, più lunghe di 64 byte, vengono codificati i primi 64 e poi si prosegue da lì.

Ad esempio, la sequenza
abcddddadaefgh
viene codificata (logicamente) come
02abc 14d 12a 03efgh 10

In binario
0 000010 01100001 01100010 01100011
1 000100 01100100
1 000010 01100001
0 000011 01100101 01100110 01100111 01101000
1 000000

Riorganizzando i bit
0000.0100 1100.0010 1100.0100 1100.0111 0001.0001 1001.0010 0001.0011 0000.1000 0011.0110
0101.0110 0110.0110 0111.0110 1000.1000 0000.0000

In esadecimale
04 C2 C4 C7 11 92 13 08 36 56 66 76 88 00