

# Huffman String Table File

[Submit Solution](#)

Viene fornito il codice (`table_generator.cpp`) di un programma in grado di stimare le frequenze dei simboli presenti in un file (usiamo `bibbia.txt` come riferimento per l'italiano) e produrre una tabella di codici canonici di Huffman. È necessario modificare il programma fornito in modo da salvare nel file `table.bin`, in un formato a piacere, la tabella dei codici di Huffman per poterla poi utilizzare nel seguito.

Nei file `hufstr.h` e `hufstr.cpp`, bisogna realizzare una libreria in grado di comprimere e decomprimere stringhe di testo. L'interfaccia dovrà essere la seguente:

```
1. class hufstr {
2. public:
3.     std::vector<uint8_t> compress(const std::string& s) const;
4.     std::string decompress(const std::vector<uint8_t>& v) const;
5. };
```

Nel costruttore di default della classe `hufstr` si deve caricare la tabella di Huffman presente nel file `table.bin` e preparare quanto necessario alle funzioni successive.

Il metodo `compress` prende una stringa e ritorna un vettore di byte contenente un bitstream ottenuto codificando con i codici di Huffman i caratteri della stringa di input, effettuando eventualmente un padding di 1 sull'ultimo elemento.

Il metodo `decompress` decodifica il vettore fornito, interrompendosi appena non riesce a leggere un codice completo. Siccome la tabella è fissata, non è necessario includerla nello stream.

Utilizzando le frequenze ottenute da `bibbia.txt`, ad esempio la stringa `"aeio"` produce il vettore di due byte (in esadecimale): `42 B1`.

Infatti `a` è `0100`, `e` è `001`, `i` è `0101`, `o` è `1000`. Quindi i bit sono `0100 001 0101 1000`, che divisi in byte (con padding di 1) producono `0100.0010 1011.0001`, appunto `42 B1`.

Nel sottomettere la soluzione bisogna fornire anche la propria tabella nel file `table.bin`.