

Viene fornito un programma in grado di stimare le frequenze dei simboli di un file e calcolarne i codici canonici di Huffman. Come esempio dell'italiano si utilizzi `bibbia.txt`.

Esercizio 1

Bisogna realizzare una libreria in grado di comprimere e decomprimere stringhe di testo. L'interfaccia dovrà essere la seguente:

```
struct hufstr {  
    hufstr(const string& filename);  
    vector<uint8_t> compress(const string& s) const;  
    string decompress(const vector<uint8_t>& v) const;  
};
```

In pratica, bisogna modificare il programma fornito in modo che salvi in un formato a piacere la tabella dei codici di Huffman. Poi bisogna creare la classe `hufstr` che nel costruttore carichi la tabella e prepari quanto necessario alle funzioni successive.

La funzione `compress` prende una stringa e manda su un vettore di byte tutti i codici di Huffman corrispondenti ai caratteri di input, effettuando eventualmente un padding di 1.

La funzione `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 ad esempio la stringa "aeio" si ottiene un vettore di due byte (in esadecimale): 42 B1.

Esercizio 2

Trovare una soluzione per inserire la tabella direttamente nel sorgente C++ in modo statico, ottenendo una classe `hufstr` che non ha bisogno di file esterni.