

## Istruzioni

- Tempo disponibile: 90 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Il programma sarà valutato per
  - Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
  - Corretta implementazione di strutture dati e algoritmi
  - Utilizzo efficiente delle risorse
  - Stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- I programmi non compilabili saranno valutati 0 punti.
- Fare l'upload di tutti i file che compongono il programma (elencati nella sezione "Ulteriori specifiche").

## Esercizio - Parte 1 (max 8 punti)

Un file di testo è composto da stringhe senza caratteri bianchi (che chiameremo parole) lunghe al massimo 30 caratteri, separate da caratteri bianchi come spazi e newline. Un esempio è l'allegato `testo.txt`.

Scrivere un programma in linguaggio C, da compilare in un eseguibile di nome `frequenze` che

- riceva come argomento della linea di comando il nome di un file del formato sopra indicato;
- stampi a video una riga per ogni parola che compare nel file di testo almeno 4 volte, dalla più lunga alla più breve, contenente la parola e il numero di volte che la parola compare nel testo.

Ad esempio, se `testo.txt` è il file allegato, l'invocazione

`./frequenze testo.txt`

deve produrre un output simile al seguente:

```
Catholic 6
was 6
her 4
a 6
```

Per tenere traccia delle frequenze delle parole è obbligatorio usare una lista collegata. Si suggerisce di memorizzare, in ogni elemento della lista,

- la parola;
- un contatore intero da incrementare ogni volta che la parola viene incontrata nel testo.

## Esercizio - Parte 2 (max 3 punti)

Fare in modo che il programma

- sia case insensitive (cioè il fatto che la stessa lettera compaia maiuscola in una parola e minuscola nell'altra non le renda diverse - ad esempio "zucchero" e "Zucchero" sarebbero la stessa parola)
- riceva altre parole come ulteriori argomenti della linea di comando e le escluda dalla lista.

## Ulteriori specifiche

- La lista collegata deve essere implementata come tipo di dato astratto (in modo cioè che il programma principale acceda alla lista solo attraverso le funzioni definite nell'interfaccia della lista).
- Verificare la correttezza della linea di comando e la corretta apertura dei file; in caso di errore, stampare un messaggio e terminare l'esecuzione.
- Il programma deve essere costituito dai seguenti file:
  - `main.c` contenente (tra eventuali altre) la funzione `main`;
  - `listaParole.c` con la definizione delle funzioni su liste (ed eventuali altre);
  - `listaParole.h` con le definizioni dei tipi di dato e le dichiarazioni delle funzioni definite in `listaParole.c` e utilizzate in `main.c`;
  - `Makefile` che permetta di costruire l'eseguibile con un singolo comando `make`.