

Esercizi 07-12-22

Riccardo Gennaro

November 2022

NOTA

Tutti gli esercizi vanno svolti rispettando le buone pratiche di allocazione dinamica. Potete implementare su più file per esercitarvi.

Esercizio 1

Un messaggio viene rappresentato mediante una struct

```
struct messaggio {  
    char testo[10000];  
    int priorit ;  
}
```

dove priorit  ha un range crescente da 1 a 10. Scrivere una struttura dati "coda a priorit " in cui e' possibile inserire messaggi, da cui vengono estratti in modo FIFO per classi di priorit : (prima quelli a priorit  10, poi quelli a priorit  9,...)

Esercizio 2

Scrivere un programma, nel file `esercizio1.cc`, che, presi come argomenti del main i nomi di due file, copi il primo file nel secondo correggendone la sintassi e generando in tal modo un testo "corretto" secondo le seguenti regole:

- la prima parola del testo deve iniziare con una lettera maiuscola;
- tutte le parole che seguono i seguenti caratteri: ".", "?", "!", devono iniziare con una lettera maiuscola.

Se ad esempio l'eseguibile   `a.out`, il comando `./a.out testo testocorretto` creer  un nuovo file di nome `testocorretto` e vi copier  il contenuto del file dato testo, modificando le parole quando queste non verificano le regole descritte sopra. Nelle figure 1 e 2 un esempio di file `testo` e `testocorretto`.

filastrocca delle parole:	filastrocca delle parole:
Fatevi avanti! chi ne vuole?	Fatevi avanti! chi ne vuole?
di parole ho la testa piena,	di parole ho la testa piena,
con dentro la \luna" e la \balena"	con dentro la \luna" e la \balena".
ci sono parole per gli amici:	ci sono parole per gli amici:
Buon giorno, Buon anno, Siate felici!	Buon giorno, Buon anno, Siate felici!
parole belle e parole buone;	parole belle e parole buone;
parole per ogni sorta di persone.	parole per ogni sorta di persone.
di G. Rodari.	di G. Rodari.

justification=centering figuretesto	justification=centering
	Table 1: testocorretto

NOTA 1: Per semplicità si assuma che il testo contenuto nel primo file inizi con un carattere alfabetico, non contenga “...” e che “.”, “?” e “!” siano sempre preceduti da una parola e seguiti da uno spazio.

NOTA 2: Per semplicità si assuma che ogni parola contenuta nel testo del primo file abbia al massimo lunghezza 30 caratteri.

NOTA 3: E' ammesso l'uso della funzione `strlen` della libreria `cstring`, non è ammesso l'uso di altre funzioni di libreria, in particolare della funzione `toupper`.

NOTA 4: il programma deve potenzialmente funzionare con ogni possibile codifica dei caratteri secondo le regole di tali codifiche viste a lezione (quindi non solo ASCII). Per realizzare la conversione da caratteri minuscoli in maiuscoli, è vietato l'uso di tabelle o di 26 if o switch-case, uno per ogni carattere.