

# Esercitazione 3

Stringhe e libreria standard di IO, creazione di librerie statiche, macros, **extern** e **static**.

## Esercizio 1: controllo errori

Scrivere una funzione macro 'CHECK\_CALL\_PTR' che prende due argomenti, un puntatore ed una stringa, e stampa utilizzando **perror** l'**errno** e la stringa passata come argomento prima di uscire dal programma con **EXIT\_FAILURE**. Esempio di utilizzo:

```
FILE *fp;  
CHECK_CALL_PTR(fp=fopen("fileprova.txt", "r"), opening fileprova.txt);
```

NOTA: per il secondo argomento della macro non ci sono le virgolette!

Se il file 'fileprova.txt' non esiste il risultato sarà:

**opening fileprova.txt: No such file or directory**

Stampare il valore di ritorno di un programma che testa la macro 'CHECK\_CALL\_PTR' sia in caso di successo che di fallimento (ricordare che il valore di ritorno dell'ultimo comando bash eseguito si può avere accedendo alla variabile di ambiente bash **?**).

## Esercizio 2: matdump

Scrivere un programma che prende in ingresso un intero (piccolo, es.  $\leq 512$ ), alloca in memoria una matrice di  $512 \times 512$  elementi **float** in modo che siano contigui in memoria. Inizializzare tale matrice (M1) in modo arbitrario (ad esempio  $M1(i,j) = (i+j)/2.0$ ). Fare quindi il dump della matrice in formato binario su un file il cui nome è 'mat\_dump.dat'. Rileggere quindi la matrice dal file 'mat\_dump.dat' memorizzandola in una matrice differente da quella di partenza (chiamata M2). Al termine dell'operazione di lettura effettuare il confronto delle due matrici con la funzione 'confronta'. Tale funzione deve essere implementata in modo “generico” cioè prende come argomenti un puntatore a funzione con tipo opportuno, il puntatore alle due matrici da confrontare e la size. Testare il programma passando alla funzione 'confronta' la funzione di libreria **memcmp** (man 3 memcmp).