

Cognome		Nome		Matricola	
---------	--	------	--	-----------	--

Esercizio 1 (Programmazione in C, 8 punti)

Scrivere un programma in C per la gestione di matrici di float, che contenga i seguenti metodi:

Una funzione *create*, che riceve due interi (dimensione di righe e colonne), crea una matrice delle dimensioni indicate, e restituisce il suo puntatore.

Un metodo *read* per la lettura di una matrice da tastiera.

Un metodo *print* per la stampa della matrice su output.

Un metodo **verifica_vet_matrice**, a cui viene passato fra gli altri parametri anche un vettore V, e che restituisce (in maniera opportuna usando i puntatori) 2 interi, **matGrande** e **vetPiccolo**:

il primo (**matGrande**) corrisponde al numero di righe pari che sono più grandi della corrispondente riga dispari (la 2 rispetto alla 1, la 4 rispetto alla 3. E così via). Per più grande si intende che almeno la metà più uno degli elementi siano maggiori degli elementi corrispondenti dell'altra riga.

il secondo (**vetPiccolo**) è il numero di colonne della matrice in cui almeno metà +1 elementi siano più grandi del corrispondente elemento del vettore V (si suppone che la dimensione di V è pari al numero di righe della matrice).

Il metodo *main* che dichiara e alloca la matrice e invoca opportunamente i metodi precedenti.

NOTA: QUALSIASI SOLUZIONE ALTERNATIVA CHE NON USI I PUNTATORI SARA' CONSIDERATA ERRATA.

Esercizio 2 (Programmazione Shell, 10 punti)

Scrivere un programma shell (**trova_film.sh**) che riceva 2 parametri, il primo è il nome di una cartella (**cartellaFilm** per esempio) e il secondo è il nome di un film.

Esempio d'uso sarà quindi: **trova_film.sh cartellaFilm elenco_nomi_film.txt**.

Il file **elenco_nomi_film.txt** conterrà una serie di triple, nome del film, anno, estensione (Esempio: **Revenant 2015 avi**)

Per ognuno di questi nomi di film, il programma dovrà verificare se il film è contenuto nella sottocartella anno della cartella **cartellaFilm**. Se non lo trova dovrà cercare nelle altre sottocartelle della cartella (**cartellaFilm** nell'esempio).

Alla fine si scriverà in un file di testo **output.txt**, il nome del film con l'estensione seguito dall'anno in caso si trova il film, altrimenti il messaggio **Errore: <nome_film_con_estensione> <cartella_in_cui_si_trova> invece di <anno_reale>**

Esempio: (**Revenant 2015 avi**). Bisognerà cercare il file **Revenant.avi** nella cartella **cartellaFilm/2015**. Se non lo si trova bisognerà cercarlo in tutte le sottocartelle. Immaginiamo di trovarlo nella sottocartella 2013, bisognerà scrivere nel file di testo:

Errore: Revenant.avi 2013 invece di 2015

Gestire anche il controllo degli errori (parametri insufficienti, file di output già esistente, cartella inesistente, ecc..).

Esercizio 3 (Programmazione System Call di Linux, 12 punti)

Realizzare un programma C che, utilizzando le System Call di Linux, svolga le seguenti funzioni.

Un padre creerà un processo Figlio F1 e F2, mentre il solo F1 creerà un altro processo N (nipote). Si utilizzino le pipe che si ritengono necessarie per far comunicare i vari processi. Il padre leggerà un vettore di n interi positivi da un file binario, quindi leggerà un intero X da tastiera. A questo punto passerà al figlio F1 tutti i numeri pari mentre a F2 i numeri dispari. Il processo F1 troverà il massimo dei numeri pari e passerà al nipote tutti i numeri, mentre F2 troverà il minimo e lo restituirà al padre. Il nipote troverà la media di tutti i numeri minori di X e scriverà tale media su un file di testo "Output.txt".