

Esercizio 1 (Programmazione in C, 8 punti)

Scrivere un programma in C per la gestione di matrici di interi $n \times m$, che contenga una bozza del main (con almeno l'invocazione del metodo `verifica_mat`) e il seguente metodo:

Un metodo `verifica_mat` che riceve una matrice di interi, un vettore di interi di dimensione n e le opportune variabili e restituisce (in maniera opportuna usando i puntatori)

1) il numero di colonne della matrice, che verificano la seguente proprietà: ogni elemento della colonna, elevato al quadrato deve essere il triplo del corrispondente elemento del vettore.

2) il numero di righe della matrice in cui tutti gli elementi sono pari e maggiori di tutti gli elementi del vettore.

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 (`copiaFile.sh`) che riceva 2 parametri, il primo è il nome di una cartella (`Cartella1` per esempio) e il secondo è il nome di un file (per esempio `prova.txt`).

Esempio d'uso sarà quindi: `copiaFile.sh Cartella1 prova.txt`

Il file `prova.txt` è fatto da una coppia di elementi `nomeFile numero` (esempio: `pippo.txt 5`). Se il file è contenuto nella cartella corrente, il programma dovrà copiare nella cartella `Cartella1`, x file (nell'esempio 5), chiamandoli, nell'esempio, `pippo1.txt`, `pippo2.txt`, ecc. .

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

Esame di Ambienti di Programmazione per il SW di Base - Data: 21 Luglio 2021 - Durata:
Traccia A

<i>Cognome</i>		<i>Nome</i>		<i>Matricola</i>	
----------------	--	-------------	--	------------------	--

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

Realizzare un programma C che, utilizzando le System Call di Linux, effettui le seguenti operazioni.

Il processo principale dovrà creare un certo numero di pipe e 3 processi figli F1, F2 e F3. Il padre leggerà una serie di numeri double dal file binario "pippo.bin", due alla volta. Quindi, il maggiore dei due lo manderà a F1, il minore a F2. Se i due numeri sono uguali ne manderà uno dei due a caso a F3 (lascio allo studente la scelta di quale dei due mandare. Quindi il padre manderà un numero speciale (magari 0) per indicare ai figli che non ci sono più numeri.

F1 dovrà fare la media dei numeri ricevuti e stampare in output. F2 e F3 dovranno trovare il massimo tutti i numeri comunicare il risultato al padre.

Infine il padre scriverà alla fine del file di testo "Risultato.txt", il minimo dei due numeri ricevuti.

Si utilizzino le pipe che si ritengono necessarie per far comunicare i vari processi.

