

Traccia A

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

Esercizio 1 (Programmazione in C, 8 punti)

Scrivere un programma in C per la gestione di matrici di interi, 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 `restituisce_max_vett` che restituisce (in maniera opportuna usando i puntatori) il massimo della matrice e un vettore `V` costruito come indicato di seguito.

Il vettore `V` è formato da tutti gli elementi della matrice che si trovino nelle righe pari e che siano maggiori di 2 volte il massimo della diagonale secondaria della matrice.

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

Esercizio 2 (Programmazione Shell, 10 punti)

Scrivere un programma shell (**`copia_modificati.sh`**) che riceva come parametri un file di testo e due cartelle. Esempio d'uso sarà: **`copia_modificati.sh file_da_copiare.txt cartellaA cartellaB`**. Il file conterrà per ogni riga il nome di un file (che è presente nella cartella **`cartellaA`**).

Per ogni file letto dal file **`file_da_copiare.txt`**, tale file dovrà essere copiato dalla cartella **`cartellaA`** alla cartella **`cartellaB`** (attenzione, i nomi delle cartelle sono passati come parametri, quindi non devono per forza essere **`cartellaA`** e **`cartellaB`**!!!!). Se il file è già presente nella cartella **`cartellaB`**, prima di copiarlo bisognerà chiedere all'utente "Vuoi sovrascriverlo?" e in caso di risposta affermativa bisognerà copiarlo e creare una versione di backup della copia pre-esistente (sostituendo l'estensione originale con `.bak`).

Alla fine si dovrà stampare in output il numero e il nome dei file copiati nella **`cartellaB`** e di quelli che non sono stati copiati. Gestire anche il controllo degli errori (parametri insufficienti, cartella inesistente, ecc..).

Esercizio 3 (Programmazione in Python, 12 punti)

Scrivere un programma python che faccia una serie di statistiche su una gara di atletica.

Il programma deve :

- 1) leggere dal file `risultati.txt` tutte le triple `nomeAtleta`, `altezza`, `tempo` (per semplicità supponiamo che `tempo` sia solo un intero che rappresenti il numero di secondi impiegato dall'atleta per arrivare al traguardo) e caricarle su una struttura dati adeguata.
- 2) leggere dal file `categorie.txt` tutte le coppie `nomeAtleta`, `categoria` e caricarle su una struttura dati adeguata (categoria potrà assumere i valori Junior, Senior e Amatore).
- 3) Restituire il numero e il nome degli atleti che abbiano la stessa altezza ordinati per il tempo impiegato.
- 4) Restituire le classifiche (cioè i nomi degli atleti ordinati per il tempo impiegato) sia assoluta (tutti gli atleti), che per categorie, cioè solo i Juniores, solo i Seniores e solo gli Amatori.

Si supponga che non esistano 2 atleti con lo stesso `nomeAtleta`.