

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 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 **trova_proprietà** che restituisce (in maniera opportuna usando i puntatori) il numero di righe che soddisfino la seguente proprietà (e inoltre restituisce true se la proprietà è verificata almeno una volta, false se non è mai verificata):

Una riga verifica la proprietà se esiste un elemento X non appartenente a quella riga per cui la somma degli elementi della riga è uguale a $X * 5$.

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 (**elimina_foto.sh**) che riceva 3 parametri, i primi due sono i nomi di estensioni di foto (esempio **jpg, gif**), il terzo è il nome di una cartella (esempio **cart_foto**)

Esempio d'uso sarà quindi: **elimina_foto.sh jpg gif cart_foto**.

Il programma dovrà eliminare tutte le foto (contenute nella cartella **cart_foto**) con estensione **jpg** per le quali non esiste il corrispondente file **gif** (esempio: se nella cartella sono presenti i file: **a.jpg b.jpg c.jpg d.jpg e.jpg a.gif c.gif e.gif**, dovranno essere eliminati i file **b.jpg d.jpg**).

Alla fine il programma dovrà stampare il numero di file eliminati e il numero di jpeg rimanenti.

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

Esercizio 3 (Programmazione in Python, 12 punti)

Scrivere un programma python che analizzi il contenuto testuale di un file **Esame.txt**.

Supponendo che esista già (chi vuole la può realizzare facilmente) la funzione **isNumber(s)** che restituisce **True** se la stringa **s** è un numero, **False** altrimenti.

Il programma deve :

- 1) leggere tutte le parole contenute nel file e caricarle su una struttura dati adeguata, separando numeri e parole (per semplicità escludiamo i segni di punteggiatura).
- 2) Trovare tutte le parole ripetute più di 4 volte e stamparle.
- 3) Fare la media, il massimo e il minimo di tutti i numeri trovati che siano compresi fra 0 e 30.
- 4) Stampare il numero di occorrenze dei numeri ripetuti più di 1 volta, dal maggiore al minore, sempre che verifichino la proprietà di cui sopra. (Esempio: 30 5 volte, 29 6 volte, 28 4 volte).