

## I raccia A

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 punto\_di\_sella che restituisce (in maniera opportuna usando i puntatori) il valore, la riga e la colonna del primo punto di sella presente nella matrice.

Si definisce punto di sella un elemento  $m[i,j]$  tale che esso risulti essere contemporaneamente il massimo fra tutti gli elementi della riga  $i$ -esima e il minimo fra tutti gli elementi della colonna  $j$ -esima oppure risulti essere contemporaneamente il minimo fra tutti gli elementi della riga  $i$ -esima e il massimo fra tutti gli elementi della colonna  $j$ -esima. Il metodo ritorna false se non esistono punti di sella. Si consiglia di trovare tutti i massimi di una colonna e metterli in un vettore e fare lo stesso per i minimi delle righe.

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 (**verifica\_duplicati.sh**) che riceva come parametri due file di testo (il primo conterrà una serie di coppie cartellaA cartellaB, una per riga del file, mentre il secondo dovrà essere creato dal programma).

Esempio d'uso sarà : **verifica\_duplicati.sh file\_cartelle.txt file\_output.txt**.

Per ogni coppia di cartelle il programma dovrà verificare che per ogni file contenuto nella cartellaA, non sia duplicato (cioè contenuto nella cartellaB). Se il file risulta duplicato, il programma dovrà scrivere nel file *file\_output.txt* il nome del file duplicato seguito dalla *cartellaB* in cui il file si trova.

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

### Esercizio 2 (Programmazione in Python, 12 punti)

Scrivere un programma python che analizzi una serie di numeri che rappresentano i voti di un certo partito per ogni giorno dell'anno (360 per semplicità).

Il programma deve :

- 1) leggere dal file voti.txt tutti i voti e caricarli su una struttura dati adeguata
- 2) Trovare il numero di voti massimo, minimo e la moda (cioè il numero di voti più presente nei dati).
- 3) Supponendo che i mesi sono tutti di 30 giorni, rifare le stesse statistiche per il mese e la settimana (cioè trovare il mese in cui si hanno avuto più voti, di meno e la moda).

Suggerimento: Per calcolare la moda, conviene utilizzare un dizionario.

Sarà apprezzato l'inserimento dei metodi all'interno di una classe AnalizzaVoti.