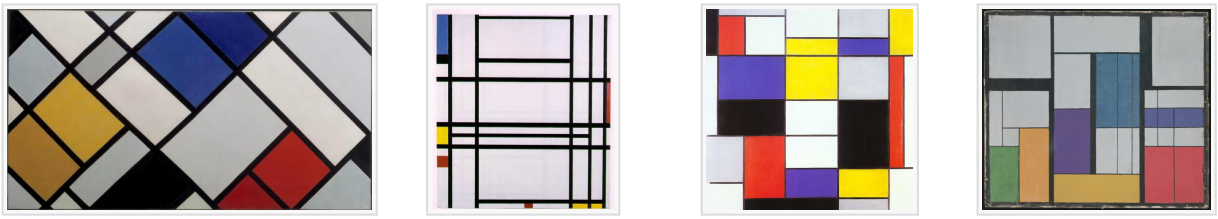
art class

Italian — 1.0

L'esame di storia dell'arte si sta avvicinando, ma ultimamente a scuola hai passato molto più tempo sull'informatica che sull'arte! Non ti resta che scrivere un programma che faccia l'esame al posto tuo.

L'esame consiste nell'analisi di una serie di dipinti. Ciascun dipinto è un esempio di uno fra quattro stili distinti, numerati da 1 a 4.

Lo stile 1 corrisponde all'arte moderna neoplastica. Per esempio:



Lo stile 2 corrisponde a paesaggi impressionistici. Per esempio:

Lo stile 3 corrisponde all'espressionismo astratto. Per esempio:

Lo stile 4 corrisponde alla pittura a campi di colore. Per esempio:

Data una rappresentazione digitale di un dipinto, il tuo compito è determinare a quale stile esso appartiene.

La giuria delle IOI ha raccolto varie immagini per ciascuno stile. Tra queste, nove immagini per stile sono state scelte a caso e incluse nel materiale presente sul computer, per permettere di esaminarle a mano e utilizzarle per il testing. Le immagini rimanenti sono utilizzate per la valutazione.

L'immagine è composta da $H \times W$ pixel, organizzata in righe numerate $0, \ldots, (H - 1)$ dall'alto verso il basso, e in colonne numerate $0, \ldots, (W - 1)$ da sinistra a destra.

L'immagine è rappresentata tramite tre array bidimensionali `R`, `G` e `B`, rispettivamente la quantità di rosso, verde e blu in ciascun pixel dell'immagine. I valori variano da `0` (assenza di rosso, verde o blu) a `255` (massimo rosso, verde o blu).

Implementazione

Devi sottoporre un file che implementa la funzione `style()`, come segue:

Funzione: `style()`

C/C++

```
int style(int H, int W,
        int R[500][500], int G[500][500], int B[500][500]);
```

Pascal

```
type artArrayType = array[0..499, 0..499] of longint;

function style(H, W : LongInt;
        var R, G, B : artArrayType) : LongInt;
```

Descrizione

Questa funzione deve determinare lo stile dell'immagine in input.

Parametri

- `H`: Il numero di righe dell'immagine.
- `W`: Il numero di colonne dell'immagine.
- `R`: Un array bidimensionale di dimensione `H×W`, contenente la quantità di rosso in ciascun pixel dell'immagine.
- `G`: Un array bidimensionale di dimensione `H×W`, contenente la quantità di verde in ciascun pixel dell'immagine.
- `B`: Un array bidimensionale di dimensione `H×W`, contenente la quantità di blu in ciascun pixel dell'immagine.
- *Restituisce*: Lo stile dell'immagine, che deve essere `1`, `2`, `3` o `4`, come descritto sopra.

Ogni elemento `R[i][j]`, `G[i][j]` e `B[i][j]` si riferisce al pixel nella riga `i` e colonna `j`, ed è un intero compreso tra `0` e `255` inclusi.

Limiti

- Tempo limite: 5 secondi.
- Limite di memoria: 64 MiB
- $100 \le H \le 500$
- $100 \le W \le 500$

Punteggio

Per questo problema non ci sono subtask, invece il punteggio sarà determinato in base a quante immagini il programma è in grado di classificare correttamente.

Se il tuo programma classifica correttamente il P percento delle immagini ($0 \le P \le 100$):

- Se $P < 25$ il programma totalizzerà 0 punti.
- Se $25 \le P < 50$ il programma totalizzerà un punteggio tra 0 e 10 punti, calcolato su una scala lineare. Precisamente, il punteggio sarà $10 \times (P - 25) / 25$, arrotondato per difetto.
- Se $50 \le P < 90$ il programma totalizzerà un punteggio tra 10 e 100 punti, calcolato su una scala lineare. Precisamente, il punteggio sarà $10 + (90 \times (P - 50) / 40)$, arrotondato per difetto.
- Se $90 \le P$ il programma totalizzerà 100 punti.

Testing

Il grader di esempio sul computer legge l'input dal file `artclass.jpg`. Questo file deve contenere un'immagine in formato JPEG.

Si ammette l'utilizzo di ogni applicazione di processing grafico per studiare le immagini, anche se questo non è necessario per risolvere il problema (vedi il menu *Applicazioni* > *Grafica*).

Note relative al linguaggio

C/C++	Devi inserire `#include "artclass.h"`.
Pascal	Devi definire `unit ArtClass`. Tutti gli array sono numerati a partire da `0` (non da `1`).

Vedi i template delle soluzioni sulla tua macchina per alcuni esempi.