International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

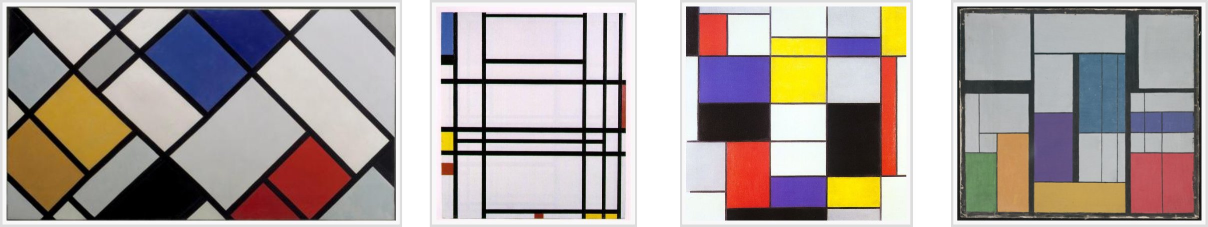
art class

English — 1.1

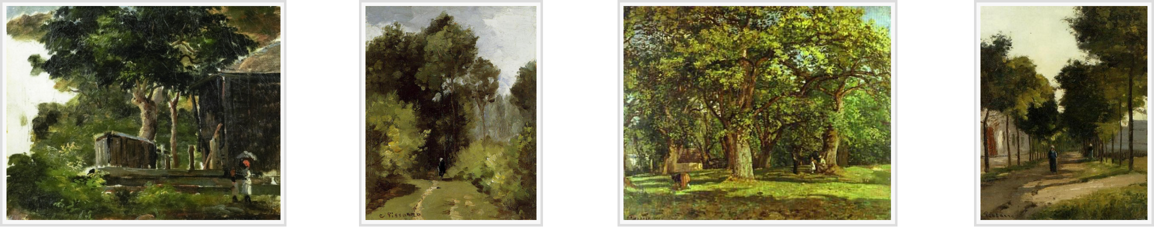
You have an Art History exam approaching, but you have been paying more attention to informatics at school than to your art classes! You will need to write a program to take the exam for you.

The exam will consist of several paintings. Each painting is an example of one of four distinctive styles, numbered 1, 2, 3 and 4.

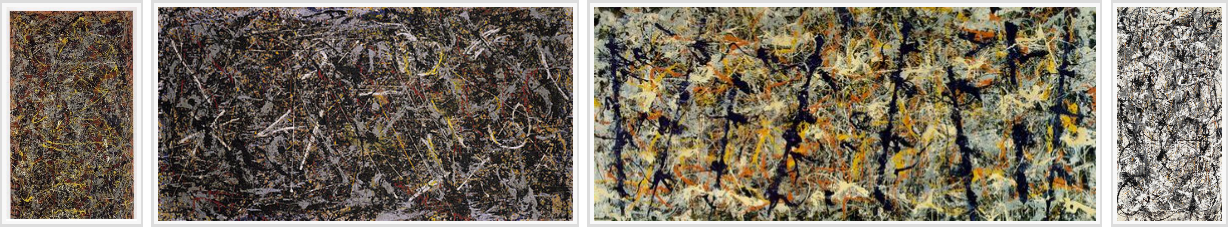
Style 1 contains neoplastic modern art. For example:



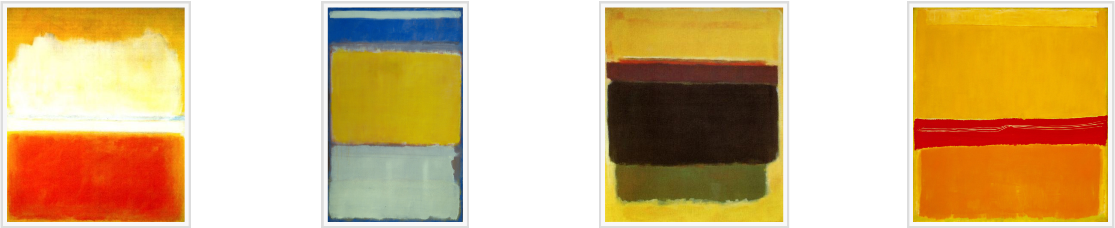
Style 2 contains impressionist landscapes. For example:



Style 3 contains expressionist action paintings. For example:



Style 4 contains colour field paintings. For example:



Your task is, given a digital image of a painting, to determine which style the painting belongs to.

The IOI judges have collected many images in each style. Nine images from each style have been chosen at random and included in the task materials on your computer, so that you can examine them by hand and use them for testing. The remaining images will be given to your program during grading.

The image will be given as an `H×W` grid of pixels. The rows of the image are numbered `0, …, (H - 1)` from top to bottom, and the columns are numbered `0, …, W - 1` from left to right.

The pixels are described using two-dimensional arrays `R`, `G` and `B`, which give the amount of red, green and blue respectively in each pixel of the image. These amounts range from `0` (no red, green or blue) to `255` (the maximum amount of red, green or blue).

Implementation

You should submit a file that implements the function `style()`, as follows:

Your Function: `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;
```

Description

This function should determine the style of the image.

Parameters

- `H`: The number of rows of pixels in the image.
- `W`: The number of columns of pixels in the image.
- `R`: A two-dimensional array of size `H×W`, giving the amount of red in each pixel of the image.
- `G`: A two-dimensional array of size `H×W`, giving the amount of green in each pixel of the image.
- `B`: A two-dimensional array of size `H×W`, giving the amount of blue in each pixel of the image.

- *Returns*: The style of the image, which must be `1`, `2`, `3` or `4`, as described above.

Each array element `R[i][j]`, `G[i][j]` and `B[i][j]` refers to the pixel in row `i` and column `j`, and will be an integer between `0` and `255` inclusive.

Constraints

- Time limit: 5 seconds
- Memory limit: 64 MiB
- $100 \le H \le 500$
- $100 \le W \le 500$

Scoring

There are no subtasks. Instead, your score for this task will be based on how many images your program correctly classifies.

Suppose you correctly classify P percent of the images (so $0 \le P \le 100$):

- If $P < 25$ then you will score 0 points.
- If $25 \le P < 50$ then you will score between 0 and 10 points, on a linear scale. Specifically, your score will be $10 \times (P - 25) / 25$, rounded down to the nearest integer.
- If $50 \le P < 90$ then you will score between 10 and 100 points, on a linear scale. Specifically, your score will be $10 + (90 \times (P - 50) / 40)$, rounded down to the nearest integer.
- If $90 \le P$ then you will score 100 points.

Experimentation

The sample grader on your computer will read input from the file `artclass.jpg`. This file must contain an image in JPEG format.

You are allowed to use any available graphics processing applications to study the images, but this is not necessary to solve the problem. (See the menu "Applications > Graphics".)

Language Notes

C/C++	You must `#include "artclass.h"`.
Pascal	You must define the `unit ArtClass`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.