

Ηλεκτρονική εξέταση στον Προγραμματισμό Υπολογιστών με C++

Εξεταστική περίοδος Φεβρουαρίου 2020

Εκφώνηση

Στο διαγώνισμα, καλείστε να υλοποιήσετε μια κλάση η οποία μοντελοποιεί και υλοποιεί μια εικόνα και κάποιες βασικές πράξεις πάνω σε αυτή. Η κλάση αυτή πρέπει:

- Να έχει το όνομα **Image**
- Να ανήκει σε ένα namespace με όνομα `graphics`
- Να περιέχεται σε ένα αρχείο που θα δημιουργήσετε με όνομα **Image.h** (δήλωση και υλοποίηση).
- Να κληρονομεί από την κλάση `ImageStorageTraits` που βρίσκεται στο αρχείο `ImageStorageTraits.h` που σας δίνεται και να αξιοποιεί τα μέλη και πεδία που περιέχονται σε αυτή.

Η κλάση `ImageStorageTraits` υποδεικνύει κάποια βασικά πεδία που πρέπει να χρησιμοποιήσετε στην παράγωγη κλάση, καθώς και μια βοηθητική μέθοδο για να έχετε πρόσβαση σε μεμονωμένες τιμές που είναι αποθηκευμένες στην εικόνα. Στο σχετικό αρχείο (`ImageStorageTraits.h`) περιλαμβάνονται οδηγίες στα σχόλια. Διαβάστε τες προσεκτικά.

Την κλάση τη χρησιμοποιείτε για τη σωστή μεταγλώττιση και λειτουργία των τριών συναρτήσεων `Test1,2,3` που καλεί η `main` στο αρχείο `exam2021.cpp`. Πρέπει να διαβάσετε τον κώδικα των `Test` και να συμπεράνετε ποιες μέθοδοι και τελεστές λείπουν από την `Image`, ώστε να τις δηλώσετε και υλοποιήσετε σωστά μέσα στην κλάση `Image` στο αρχείο `Image.h`. Στον κώδικα των `Test`, επισημαίνονται για βοήθεια οι σχετικές γραμμές με το σχόλιο:

```
// To be implemented
```

(θα σας τις επισημάνει και ο compiler βέβαια έτσι κι αλλιώς)

Για να εκτελεστεί το τελικό πρόγραμμα, πρέπει να του περάσετε σαν παράμετρο (command line argument) τον αριθμό μητρώου σας (π.χ. 3100001, ΧΩΡΙΣ το 'ρ'). Επίσης, η εφαρμογή σας περιμένει να βρει στον ίδιο φάκελο με το εκτελέσιμο που θα παραχθεί, μια εικόνα `input.ppm` που σας δίνεται στο zip αρχείο που κατεβάζετε (βρίσκεται μέσα στο φάκελο `data`).

Στον κώδικα που σας δίνεται, υπάρχουν ακόμα τα ακόλουθα αρχεία:

- `ppm.h`: δηλώσεις 2 συναρτήσεων για να φορτώνεται και αποθηκεύεται μια εικόνα τύπου PPM. Θα χρειαστεί να τις καλέσετε εσωτερικά από το δικό σας κώδικα για την υλοποίηση των αντίστοιχων μεθόδων που απαιτούνται στα `Tests2` και `3`.
- `ppm.cpp`: Η υλοποίηση των παραπάνω 2 συναρτήσεων.
- `codec.h`: Υλοποιεί έναν απλό encoder/decoder για μια στοιχειώδη κρυπτογράφηση των εικόνων. Ενσωματώνεται από την `exam2021.cpp` και χρειάζεται στο `Test3`.

Προσοχή: **ΔΕΝ ΠΡΕΠΕΙ ΝΑ ΜΕΤΑΒΑΛΕΤΕ ΚΑΝΕΝΑ ΑΡΧΕΙΟ ΑΠΟ ΑΥΤΑ ΠΟΥ ΣΑΣ ΔΙΝΟΝΤΑΙ.** Γράφετε το κώδικά σας στο αρχείο Image.h και κάνετε compile το πρόγραμμα έτσι ώστε ο κώδικας που υπάρχει μέσα στις 3 συναρτήσεις Test1,2,3 που καλούνται από τη main να δουλεύει. Κατά την ανάπτυξη του κώδικά σας, μπορείτε να βάλετε σε σχόλια όποια από αυτές θέλετε, μέχρι να είναι σε θέση να εκτελεστούν. Προφανώς, τοπικά μπορείτε να πειραματιστείτε με τον κώδικά σας όπως επιθυμείτε, αλλά για τη βαθμολόγηση δε θα ληφθεί υπόψη καμία παρέμβαση στα περιεχόμενα των αρχείων που σας δίνονται έτοιμα. Αν κάποια από τις Test1..3 συναρτήσεις δε δουλεύει όταν ολοκληρώσετε τον κώδικά σας, αφήστε τη σε σχόλια, καθώς θα χρειαστεί να καταγράψετε την έξοδο της εκτέλεσης του προγράμματος.

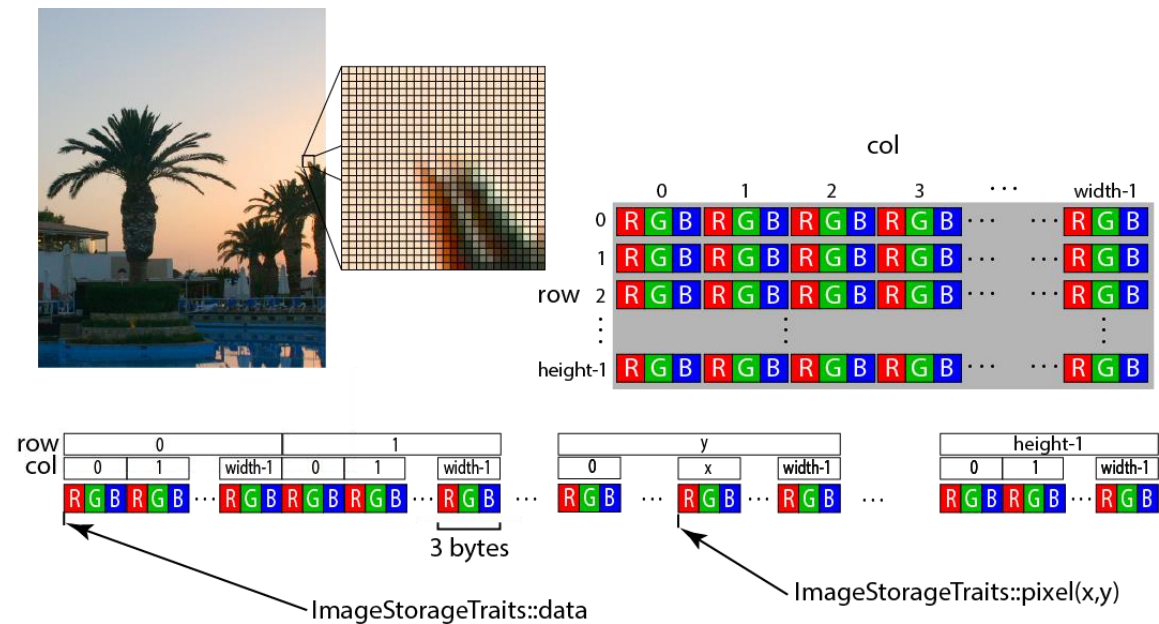
Για την ανάπτυξη του κώδικα μπορείτε προφανώς να χρησιμοποιήσετε οποιονδήποτε compiler C++ (έκδοση γλώσσας 11 και μετά), IDE και λειτουργικό σύστημα.

Η Εικόνα

Τυπικά, μια εικόνα αναπαρίσταται και αποθηκεύεται στη μνήμη του υπολογιστή σε μια συνεχόμενη και συμπαγή περιοχή μνήμης με μέγεθος

$width \times height \times channels \times BytesPerChannel$,

όπου $width$ το πλάτος της εικόνας, $height$ το ύψος της, $channels$ ο αριθμός των χρωματικών «καναλιών» (συνιστωσών, π.χ. RED, GREEN, BLUE) και $BytesPerChannel$ ο αριθμός των bytes που χρησιμοποιούνται για την αριθμητική αναπαράσταση της έντασης του κάθε καναλιού χρώματος.



Στη δική μας περίπτωση, καλούμαστε να διαχειριστούμε τριχρωματικές εικόνες (R,G,B) με 1 byte ανά κανάλι χρώματος (τιμές ανά χρώμα 0-255 τύπου unsigned char). Στην κλάση ImageStorageTraits που πρέπει να κληρονομεί η Image, η μνήμη αυτή πρέπει να δεσμευτεί στο πεδίο data. Η μέθοδος pixel μας επιστρέφει το offset μέσα στον buffer

της μνήμης όπου «ξεκινάει» το Pixel (x,y) (βλ. σχήμα). Κάθε διαδοχικό pixel απέχει στη μνήμη από το προηγούμενο 3 bytes (unsigned char).

Ζητούμενα

Ζητούμενο 1. Υλοποιήστε και ανεβάστε το αρχείο Image.h ΜΟΝΟ, ως ένα από τα 3 αρχεία που σας ζητούνται. ΜΗΝ ανεβάσετε κάποιο zip αρχείο με όλο τον κώδικά σας: Δε θα βαθμολογηθεί.

Ζητούμενο 2. Ανεβάστε το αρχείο output.ppm που παράγεται από το Test2.

Ζητούμενο 3. Ανεβάστε το αρχείο encrypted.ppm που παράγεται από το Test3.

Σχόλια

Η αναμενόμενη έξοδος για την εικόνα output.ppm είναι μια αλλαγμένη έκδοση της αρχικής. Η εικόνα encrypted.ppm θα φαίνεται παραμορφωμένη και με θόρυβο. Για να ελέγξετε τόσο την εικόνα Input.ppm που σας δίνεται και που φορτώνεται από το πρόγραμμά σας, όσο και τις εικόνες που παράγονται από αυτό (output.ppm, encrypted.ppm), μπορείτε να χρησιμοποιήσετε το site: <http://paulcuth.me.uk/netpbm-viewer/> ή κάποιον άλλο online image viewer που να υποστηρίζει το PPM format.

Τα Test1,2,3 που καλούνται από τη main σας κάνουν κάποιο βασικό validation των λειτουργιών που υλοποιείτε, οπότε πρέπει να ελέγχετε από την έξοδο στην κονσόλα της εφαρμογής σας αν τα τεστ που αναφέρει ότι εκτελεί ολοκληρώνονται επιτυχώς [PASS] ή όχι [FAIL].

Τα Tests είναι αυξανόμενης δυσκολίας και απαιτούν την ορθή υλοποίηση των προηγούμενων.

Προσοχή: Η έξοδος της εφαρμογής εξαρτάται από τον Αριθμό Μητρώου που εισάγετε ως παράμετρο στην κονσόλα και θα πιστοποιηθεί η ορθότητα των αποτελεσμάτων.