

## Προγραμματισμός Υπολογιστών με C++

1<sup>η</sup> Εργασία – Ακαδημαϊκό Έτος 2018-19

Δεξιότητες: Συγγραφή και μεταγλώττιση κώδικα c++, δημιουργία βιβλιοθηκών, χρήση αρχείων κεφαλίδας, υλοποίηση απλών συναρτήσεων και μεθόδων κλάσεων, χρήση τελεστών, δυναμική εκχώρηση και αποδέσμευση μνήμης, πράξεις με δείκτες, χειρισμός ορισμάτων γραμμής εντολών και εγγραφή/ανάγνωση δεδομένων σε/από αρχεία.

Ημερομηνία Παράδοσης Εργασίας: **2 Δεκεμβρίου 2018**

### 1. Εκφώνηση

Δίνεται ως είσοδος σε ένα πρόγραμμα C++ που τρέχει από τη γραμμή εντολών το όνομα ενός αρχείου εικόνας σε μορφότυπο (format) ppm. Να δημιουργηθεί η κατάλληλη λειτουργικότητα ώστε να μπορεί το πρόγραμμά σας να ανοίγει την αντίστοιχη εικόνα και να την αποθηκεύει σε στιγμιότυπο μιας κλάσης Image με τη χρήση μιας συνάρτησης ReadPPM (βλ. κώδικα εργασίας). Στη συνέχεια, το πρόγραμμά σας να αναφέρει στη γραμμή εντολών τις διαστάσεις της εικόνας, να υπολογίζει το αρνητικό της και να την αποθηκεύει σε ένα νέο αρχείο με τη βοήθεια της συνάρτησης WritePPM. Στην περίπτωση που ο χρήστης δε δώσει στη γραμμή εντολών το όνομα αρχείου προς ανάγνωση ως όρισμα στην εφαρμογή, να ζητείται κατά την εκτέλεση του προγράμματος. Το αρχείο που θα αποθηκεύεται θα πρέπει να έχει το επίθεμα `_neg` στο κύριο μέρος του ονόματός του (πριν την τελευταία τελεία μετά την οποία ακολουθεί η κατάληξη ppm). Το όνομα του εκτελέσιμου που θα δημιουργείται θα πρέπει να είναι `neg.exe`

**Παράδειγμα επιθυμητού τρόπου εκτέλεσης:**

|  |                         |
|--|-------------------------|
| Εκτέλεση του προγράμματος (τρόπος A: παρέχεται το όνομα αρχείου σαν όρισμα στην εφαρμογή)  | > <b>neg photo1.ppm</b> |
| Έξοδος:<br><br>Στην κονσόλα: <b>Image dimensions are: 1200 X 750</b><br>Δημιουργία στο ίδιο μονοπάτι του αρνητικού της εικόνας εισόδου με όνομα: <b>photo1_neg.ppm</b>   |                         |
| Εκτέλεση του προγράμματος (τρόπος B: Δεν παρέχεται αρχείο ως όρισμα)   | > <b>neg</b>            |
| Έξοδος:<br><b>File name of the image to load:</b><br>Είσοδος (στην ίδια γραμμή):<br><b>Image01.ppm</b><br>Έξοδος:<br><b>Image dimensions are: 1200 X 750</b><br>Δημιουργία στο ίδιο μονοπάτι του αρνητικού της εικόνας εισόδου με όνομα: <b>photo1_neg.ppm</b> |                         |

## 2. Μορφότυπος αρχείων εικόνας ppm

Στην εργασία σας θα χρειαστεί να φορτώσετε ένα αρχείο εικόνας με βάση έναν γνωστό, απλό μορφότυπο, τον PPM, υλοποιώντας τη συνάρτηση LoadPPM, της οποίας η ακριβής δήλωση σας δίνεται.

Για την εργασία θα υποστηρίξουμε την έκδοση P6 του ppm format. Το ppm format είναι ένα μικτό ASCII/binary format το οποίο είναι από τις απλούστερες αναπαραστάσεις αποθήκευσης εικόνων. Τις ακριβείς προδιαγραφές του μπορείτε να τις βρείτε στη διεύθυνση:

<http://paulbourke.net/dataformats/ppm/>

Παρατίθεται η επεξήγηση στα Ελληνικά εδώ:

Ένα αρχείο τύπου PPM αποτελείται από δύο μέρη, την κεφαλίδα και τα δεδομένα της εικόνας. Η κεφαλίδα απαρτίζεται από τουλάχιστο τρία τμήματα, χωρισμένα υπό φυσιολογικές συνθήκες με το χαρακτήρα της νέας γραμμής (CR/LF – “return”: ‘\n’), ή τουλάχιστο από τον κενό χαρακτήρα (space, tab). Το πρώτο τμήμα (ή η πρώτη γραμμή, αν είναι χωρισμένη η κεφαλίδα με “return”) είναι ο προσδιοριστής του τύπου του ppm format που ακολουθεί, ο οποίος μπορεί να είναι είτε “P3” είτε “P6” (προφανώς χωρίς τα εισαγωγικά!). Εμείς υποστηρίζουμε μόνο το P6 format.

Οι επόμενες δύο πληροφορίες που περιέχονται στην κεφαλίδα είναι το πλάτος και το ύψος της εικόνας σε pixels, δοσμένα ως ASCII κείμενο.

Το τέταρτο τμήμα της κεφαλίδας αναγράφει την μέγιστη επιτρεπόμενη τιμή που μπορεί να αποθηκευτεί σε κάθε pixel. Αν η τιμή είναι μέχρι 255, τότε η εικόνα αποθηκεύει ένα byte για κάθε κανάλι χρώματος (κόκκινο, πράσινο, μπλε) σε κάθε pixel. Αν ο αριθμός που αναγράφεται είναι μεγαλύτερος του 255, τότε χρειάζονται 2 bytes. Εμείς υποστηρίζουμε μόνο 24bit εικόνες, δηλαδή τιμές μέχρι 255.

Επιπροσθέτως, μεταξύ των παραπάνω γραμμών ή στο τέλος μιας γραμμής (δηλαδή μετά την πληροφορία που περιέχει), μπορεί να τοποθετηθεί οπουδήποτε μέσα στην κεφαλίδα ένα σχόλιο. Το σχόλιο ξεκινάει πάντα με το χαρακτήρα ‘#’ και εκτείνεται ως το τέλος της γραμμής (μέχρι το χαρακτήρα “return” – ‘\n’).

Αν κατά την ανάγνωση της κεφαλίδας κάτι πάει στραβά, εμφανίζουμε ένα σχετικό σφάλμα στην κονσόλα στο οποίο και περιγράφουμε ακριβώς το πρόβλημα. Πιθανά σφάλματα είναι (με τη σειρά που μπορεί να προκύψουν): δεν υπάρχει το αρχείο που προσπαθούμε να ανοίξουμε ή δεν ανοίγει, προσπάθεια να διαβάσουμε ppm που δεν είναι τύπου P6, λείπει η οριζόντια ή/και η κάθετη διάσταση της εικόνας, απέτυχε να διαβαστεί η μέγιστη τιμή αποθήκευσης ή είναι μεγαλύτερη του 255.

Σημείωση: Για την απλούστερη υλοποίηση της συνάρτησης ανάγνωσης της εικόνας, να θεωρήσετε ότι δεν υπάρχουν σχόλια (δηλαδή τμήματα κειμένου που ξεκινάνε από το χαρακτήρα ‘#’) και επομένως, η κεφαλίδα περιέχει μόνο τα απαραίτητα πεδία της εικόνας.

Παραδείγματα κεφαλίδων:

|                 |                          |                       |
|-----------------|--------------------------|-----------------------|
| P6 1024 788 255 | P6<br>1024<br>788<br>255 | P6<br>1024 788<br>255 |
|-----------------|--------------------------|-----------------------|

Μετά το τελευταίο πεδίο της κεφαλίδας (τη μέγιστη τιμή δηλαδή) πάντα υπάρχει ένας ακριβώς διαχωριστικός χαρακτήρας μεταξύ της κεφαλίδας και του σώματος των δεδομένων, συνήθως ένα κενό (space) ή '\n'.

Όταν ο τύπος του format είναι P6, τότε τα δεδομένα της εικόνας είναι αποθηκευμένα σε bytes (δυαδικό format), ένα byte για κάθε χρωματική συνιστώσα (R,G,B). Τα pixels σώζονται κατά γραμμές που περιέχουν τριπλέτες RGB από αριστερά προς τα δεξιά με πρώτη γραμμή (γραμμή 0) την πάνω και τελευταία την κάτω.

Προσοχή: Θα πρέπει να υλοποιήσετε το μηχανισμό ανάγνωσης και εγγραφής του μορφώτυπου ppm **μόνοι σας** και όχι να ενσωματώσετε κάποια υπάρχουσα βιβλιοθήκη που τον υλοποιεί (προφανώς επίσης θα πρέπει να είστε σε θέση να εξηγήσετε τι κάνει ο κώδικας).

### 3. Υλοποίηση

Για την υλοποίηση της εργασίας, σας δίνεται μια βάση κώδικα με δηλώσεις της κλάσης Image και της κλάσης Color που χρησιμοποιεί η Image, καθώς και οι δηλώσεις των συναρτήσεων ReadPPM και WritePPM. **Πρέπει να υλοποιήσετε:**

- Μια εξωτερική βιβλιοθήκη με όνομα **ppm.lib** (ή libppm.a για χρήστες του g++) που να αντιστοιχεί στην υλοποίηση των συναρτήσεων **ReadPPM** και **WritePPM** που ορίζονται στο **ppm.h**. Ο ρόλος της ReadPPM είναι να διαβάζει από ένα αρχείο τα δεδομένα της αντίστοιχης εικόνας και να παράγει έναν buffer δεδομένων εικόνας (πλάτος X ύψος τριάδες από float δεδομένα). Τα δεδομένα αυτά τα χρησιμοποιείτε εσείς στη συνέχεια για να αρχικοποιήσετε κατάλληλα ένα αντικείμενο τύπου Image. Η WritePPM παίρνει σαν όρισμα μια υπάρχουσα εικόνα σε μορφή πίνακα δεδομένων όμοιο με αυτόν που επιστρέφει η ReadPPM, και γράφει τα δεδομένα σε ένα αρχείο σύμφωνα με το μορφώτυπο PPM.
- Όλες τις μεθόδους της κλάσης Image, χωρίς να τροποποιήσετε με οποιονδήποτε τρόπο τη δήλωση της κλάσης αυτής ή να αλλοιώσετε το αρχείο κεφαλίδας Image.h. Αυτή η κλάση θα αποτελεί το βασικό υποδοχέα των δεδομένων μιας εικόνας και παρέχει μεθόδους για την πρόσβαση και αλλαγή δεδομένων μιας εικόνας. Οι υλοποιήσεις των μεθόδων load και save θα πρέπει να χρησιμοποιούν τις συναρτήσεις ReadPPM και WritePPM αντίστοιχα που βρίσκονται στη βιβλιοθήκη ppm που φτιάξατε.
- Την κύρια εφαρμογή που εκτελεί τα βήματα της λειτουργικότητας που είδατε στην ενότητα 1. Η κύρια εφαρμογή θα πρέπει να συνδέει τη βιβλιοθήκη ppm.lib (ή αντίστοιχα τη libppm.a)

που φτιάξατε προκειμένου να χρησιμοποιεί την υλοποίηση για τις συναρτήσεις ReadPPM και WritePPM.

Προσοχή: τα δεδομένα της έντασης κάθε καναλιού χρώματος είναι αποθηκευμένα μέσα στο αρχείο PPM σε ακέραια μορφή με ελάχιστη τιμή 0 και μέγιστη 255. Η Image εσωτερικά χρησιμοποιεί δεδομένα σε μορφή float. Κατά την ανάγνωση, να μετατρέπονται οι τιμές από το διάστημα ακεραίων [0,255] στο κανονικοποιημένο διάστημα αριθμών κινητής υποδιαστολής [0.0,1.0], στο οποίο και θα πρέπει να γίνονται οποιεσδήποτε πράξεις που αφορούν χρωματική πληροφορία. Για τις πράξεις μεταξύ χρωμάτων σας δίνεται έτοιμη η υλοποίηση τελεστών +/- στο αρχείο Color.h.

Το αρνητικό μιας εικόνας υπολογίζεται παίρνοντας κάθε Pixel  $p(x,y)$  χωριστά και κάνοντας την πράξη:

$$p'(x,y) = (1,1,1) - p(x,y)$$

#### 4. Γενικές οδηγίες

Για την υλοποίηση των προγραμμάτων, μπορείτε να χρησιμοποιήσετε όποιο περιβάλλον θέλετε αλλά ο κώδικας που θα παραδώσετε θα πρέπει να μπορεί να γίνει compile και link είτε με τον gcc σε περιβάλλον Windows, είτε με το Visual Studio 2015, πάλι προφανώς σε περιβάλλον Windows. Παραδίδετε έτοιμο Visual Studio project (ή αντίστοιχα makefile ή κάποιο script εκτέλεσης εντολών μεταγλώττισης για τον g++) και όχι μόνο τα .cpp και .h αρχεία και έχετε επιβεβαιώσει ότι γίνονται σωστά build.

Μπορείτε να ελέγξετε το αποτέλεσμα με το λογισμικό προεπισκόπησης που σας δίνεται στο eclass ή με οποιοδήποτε άλλο πρόγραμμα που αναγνωρίζει το μορφότυπο PPM.

**Προσοχή:** Σε οποιαδήποτε περίπτωση διαπιστωθεί **αντιγραφή** κώδικα ή αδυναμία του εξεταζόμενου φοιτητή να **εξηγήσει την υλοποίησή του**, η εργασία **μηδενίζεται** αυτομάτως. Επίσης, το εκτελέσιμο της εργασίας **πρέπει να δουλεύει** για να βαθμολογηθεί η τελευταία.