

# ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι :

## ΕΙΣΑΓΩΓΗ ΣΤΗ C++

### Θέματα Εξετάσεων Ιουνίου 2015

1. Γράψτε μία συνάρτηση με όνομα `digit` που να δέχεται δύο ακέραια ορίσματα,  $N$  και  $d$ . Η συνάρτηση θα επιστρέφει το ψηφίο στη θέση  $d$  του αριθμού  $N$ . Προσέξτε ότι το  $N$  μπορεί να είναι αρνητικός. Θεωρούμε ότι στην πρώτη θέση είναι το ψηφίο των μονάδων. Για παράδειγμα, το `digit(N=57960,d=2)` πρέπει να επιστρέφει τον αριθμό 6. Αν το  $d$  είναι μεγαλύτερο από το πλήθος των ψηφίων του  $N$ , η συνάρτηση θα επιστρέφει 0.

Αποθηκεύστε στην περιοχή σας (με δεξί κλικ, Save Page As ...), το αρχείο στη διεύθυνση <http://tinyurl.com/ints201411>. Περιέχει 3590 ακέραιους, σε ξεχωριστή γραμμή ο καθένας.

Χρησιμοποιήστε τη συνάρτηση που γράψατε για να βρείτε το τρίτο ψηφίο των αριθμών του αρχείου που κατεβάσατε. Τα ψηφία που θα βρείτε, να τα γράψετε στο αρχείο "digit.txt", ένα σε κάθε σειρά.

2. Μία διαμόρφωση ενός αρχείου κειμένου (ASCII) που μπορεί να χρησιμοποιηθεί για την αποθήκευση εικόνας με αποχρώσεις του γκρι είναι η ακόλουθη:

- η πρώτη γραμμή του αρχείου πρέπει να γράφει: P2.
- η δεύτερη πρέπει να γράφει τις διαστάσεις: πλάτος ύψος (δηλαδή τους δύο αριθμούς με κενό μεταξύ τους).
- η τρίτη πρέπει να έχει ένα θετικό ακέραιο αριθμό  $K$  που αντιπροσωπεύει τη μέγιστη τιμή του γκρι. Πρέπει να είναι μικρότερη από 65536. Τυπική τιμή για αυτή είναι το 255.
- ακολουθούν τα pixels της εικόνας *κατά γραμμές*. Κάθε pixel αντιπροσωπεύεται από ένα ακέραιο αριθμό από το 0 έως και το  $K$ . Μεταξύ των τιμών πρέπει να υπάρχει ένα τουλάχιστον κενό ή αλλαγή γραμμής. Οι γραμμές του αρχείου πρέπει να έχουν λιγότερους από 70 χαρακτήρες. Διευκολύνει επομένως αν κάθε pixel είναι γραμμένο σε ξεχωριστή γραμμή.

Η διαμόρφωση αυτή αποτελεί ένα αρχείο τύπου plain pgm (portable graymap) που μπορούμε να το δούμε με προγράμματα απεικόνισης.

Γράψτε ένα πρόγραμμα που θα διαβάζει το αρχείο `input.ppgm` και θα δημιουργεί μία νέα εικόνα στο `output.ppgm` ως εξής: Το pixel  $(i, j)$  στη νέα εικόνα θα είναι ο μέσος όρος του pixel  $(i, j)$  της αρχικής και των γειτονικών του (μέχρι γείτονες τάξης  $p$ ). Επομένως, αν το  $(i, j)$  είναι μακριά από τα άκρα, τα pixels που χρησιμοποιούμε στον υπολογισμό είναι τα  $(i \pm p, j \pm p)$ . Αν το  $(i, j)$  είναι στα άκρα, οι γείτονες είναι λιγότεροι.

Το πρόγραμμά σας θα ζητά από τον χρήστη τον ακέραιο θετικό αριθμό  $p$ . Για να το δοκιμάσετε, χρησιμοποιήστε το αρχείο στη διεύθυνση <http://fla.st/1KxdatB>, αφού το αποθηκεύσετε με το όνομα `input.ppgm`.

3. (α) Να γράψετε συνάρτηση που να αναλύει ένα μη αρνητικό ακέραιο (α' όρισμα) στα ψηφία του και να τα αποθηκεύει σε `std::list` (β' όρισμα) με ακέραια στοιχεία. Ο container θα θεωρείται αρχικά κενός.
- (β) Να γράψετε συνάρτηση που να υπολογίζει το άθροισμα δύο μη αρνητικών ακεραίων με οποιοδήποτε πλήθος ψηφίων. Κάθε ακέραιος θα δίνεται ως `std::list` ακέραιων ψηφίων που θα ορίζεται από τους `iterators` αρχής και τέλους. Το άθροισμά τους θα αποθηκεύεται σε `std::list` με επαρκές πλήθος στοιχείων. Η συνάρτησή σας θα δέχεται πέντε `iterators` (αρχής και τέλους της πρώτης λίστας, αρχής και τέλους της δεύτερης λίστας και αρχής της λίστας για το άθροισμα).

Γράψτε πρόγραμμα που να χρησιμοποιεί τις προηγούμενες συναρτήσεις για να αναλύσει και να προσθέσει τους αριθμούς 1958723584 και 60945983. Να τυπώνει το άθροισμά τους στην οθόνη.