

ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι: ΕΙΣΑΓΩΓΗ ΣΤΗ C++

Θέματα Εξετάσεων Ιουνίου 2016

1. Μία διαμόρφωση ενός αρχείου που μπορεί να χρησιμοποιηθεί για την αποθήκευση έγχρωμης εικόνας είναι η ακόλουθη:

- Η πρώτη γραμμή του αρχείου πρέπει να γράφει: P3.
- Η δεύτερη πρέπει να γράφει τις διαστάσεις της εικόνας: 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 κάθε χρώματος. Τυπική τιμή είναι το 255.
- Να ακολουθούν σε ξεχωριστές σειρές τα pixels της εικόνας κατά γραμμές. Σε κάθε σειρά του αρχείου θα αναγράφεται η τιμή των χρωμάτων «κόκκινο» (R), «πράσινο» (G), «μπλε» (B) για το κάθε pixel, με ένα κενό μεταξύ τους. Ένα pixel που έχει χρώμα κόκκινο θα αναπαριστάται από την τριάδα $K\ 0\ 0$ (αν το K είναι 255 θα γράφουμε 255 0 0). Το pixel με «πράσινο» χρώμα θα αντιστοιχεί στη γραμμή $0\ K\ 0$. Το μαύρο χρώμα είναι το $0\ 0\ 0$ ενώ το λευκό $K\ K\ K$. Το κίτρινο είναι $K\ K\ 0$. Σε κάθε συνιστώσα RGB μπορούμε γενικά να έχουμε οποιαδήποτε τιμή μεταξύ 0 και K ώστε να παράγουμε όλα τα χρώματα.

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

Δημιουργήστε ένα αρχείο με όνομα *belgium.pppm* με τη σημαία του Βελγίου¹, σε 512×768 pixels, χρησιμοποιώντας την παραπάνω διαμόρφωση.

2. Η συνάρτηση Bessel πρώτου είδους, ακέραιας τάξης n , $J_n(x)$, μπορεί να οριστεί ως εξής

$$J_n(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!(m+n)!} \left(\frac{x}{2}\right)^{2m+n}.$$

Να τυπώσετε στο αρχείο *bessel.dat* τις τιμές των συναρτήσεων $J_0(x)$, $J_1(x)$, $J_2(x)$ σε 150 ισαπέχοντα σημεία x_i στο διάστημα

¹τρεις κατακόρυφες λωρίδες ίσου πλάτους: μαύρο, κίτρινο, κόκκινο.

[0, 20]. Το αρχείο θα έχει σε κάθε γραμμή τις τιμές

$$x_i \quad J_0(x_i) \quad J_1(x_i) \quad J_2(x_i)$$

Υπόδειξη I: Στο άθροισμα δεν μπορούμε, φυσικά, να πάρουμε άπειρους όρους. Να σταματήσετε τον υπολογισμό του στον πρώτο όρο που κατ' απόλυτη τιμή είναι μικρότερος από 10^{-12} .

Υπόδειξη II: Παρατηρήστε ότι ο κάθε όρος στο άθροισμα προκύπτει από τον προηγούμενό του με πολλαπλασιασμό κατάλληλης ποσότητας. Μπορεί να σας βοηθήσει.

3. Θέλουμε να γράψουμε συνάρτηση που να δέχεται

- δύο iterators beg1, end1 που ορίζουν ένα διάστημα σε ένα container (αρχή του διαστήματος το beg1, μία θέση μετά το τέλος το end1)
- ένα άλλο iterator beg2 σε ίδιου τύπου container.

Η συγκεκριμένη συνάρτηση θα αντιγράφει όλα τα στοιχεία του διαστήματος [beg1,end1) σε διαδοχικές θέσεις από το beg2 και μετά. Αν εντοπίσει ομάδες ίσων και διαδοχικών στοιχείων θα παραλείπει κατά την αντιγραφή όλα τα στοιχεία της ομάδας εκτός από το πρώτο. Επιστρέφει iterator στο νέο container, στην πρώτη θέση μετά το τελευταίο στοιχείο που δεν έχει γραφτεί.

Επομένως, συμπληρώστε τον κώδικα

```
template<typename Iterator>
Iterator
uniq(Iterator beg1, Iterator end1, Iterator beg2) { }
```

Κατόπιν, δημιουργήστε ένα `std::vector<int>` και αποθηκεύστε σε αυτό τους αριθμούς 1, 6, 6, 5, 2, 2, 2, 8, 9, 9, 8, 9, 7, 0, 1, 1. Καλέστε τη συνάρτηση που γράψατε για να αντιγράψετε τα μοναδικά στοιχεία του σε άλλο `std::vector<int>`. Ο νέος container πρέπει να έχει τις τιμές 1, 6, 5, 2, 8, 9, 8, 9, 7, 0, 1.