

ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι: ΕΙΣΑΓΩΓΗ ΣΤΗ C++

Θέματα Εξετάσεων Σεπτεμβρίου 2017

1. Γράψτε συνάρτηση που να υλοποιεί τον αλγόριθμο `std::search`: συμπληρώστε τον κώδικα 35/100

```
template<typename ForwardIt1, typename ForwardIt2>
ForwardIt1
search(ForwardIt1 first1, ForwardIt1 last1,
       ForwardIt2 first2, ForwardIt2 last2)
{ }
```

Η συνάρτησή σας θα εντοπίζει την πρώτη εμφάνιση της ακολουθίας μεταξύ των iterators `first2,last2` στο διάστημα `[first1,last1)`. Η σύγκριση των στοιχείων θα γίνεται με τον τελεστή `==`. Η συνάρτηση θα επιστρέφει iterator στην αρχή της πρώτης εμφάνισης ή `last1` αν δεν βρεθεί η ζητούμενη ακολουθία. Οι iterators `first1,last1` και `first2,last2` είναι μονής κατεύθυνσης.

Χρησιμοποιήστε τη συνάρτηση που γράψατε ώστε να εντοπίσετε την ακολουθία `{13, 12, 11}` στη σειρά `{15, 14, 13, 12, 11, 15, 14, 13, 12, 11}`.

2. Το λειτουργικό σύστημα UNIX υπολογίζει το χρόνο με βάση τον αριθμό των δευτερολέπτων που πέρασαν από την 1/1/1970, στις 00:00:00. Ποια ημέρα και ώρα συμπληρώθηκαν 10^9 δευτερόλεπτα; Πότε θα συμπληρωθούν $2^{31} - 1$ δευτερόλεπτα (και θα πάψουν να μετρούν σωστά το χρόνο τα συστήματα UNIX 32bit); 30/100

Υπόδειξη I: Ένα έτος που διαιρείται ακριβώς με το 4 αλλά όχι με το 100 είναι δίσεκτο. Είναι επίσης δίσεκτο αν διαιρείται με το 400.

Υπόδειξη II: Αγνοήστε τα εμβόλιμα δευτερόλεπτα (leap seconds).

3. Η διαμόρφωση `plain ppm` ενός αρχείου, που μπορεί να χρησιμοποιηθεί για την αποθήκευση έγχρωμης εικόνας είναι η ακόλουθη: 35/100

- 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 , μέχρι το 255, 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 παραγάγουμε όλα τα χρώματα.

Έστω η μιγαδική συνάρτηση μιγαδικής μεταβλητής $p(z)$. Μια οποιαδήποτε αρχική τιμή z_0 (για την οποία ισχύει $p'(z_0) \neq 0$) θα συγκλίνει σε μία από τις ρίζες της $p(z)$, στα σημεία δηλαδή που μηδενίζεται η $p(z)$, αν την μεταβάλλουμε ως εξής:

$$z_{i+1} = z_i - \frac{p(z_i)}{p'(z_i)}, \quad i = 0, 1, 2, \dots$$

Δηλαδή, αν ξεκινήσουμε από μία τιμή z_0 , η εφαρμογή του τύπου θα μας δώσει τη z_1 . Με νέα εφαρμογή του τύπου θα υπολογίσουμε τη z_2 , κλπ., έως ότου πλησιάσουμε όσο κοντά θέλουμε σε μία από τις ρίζες της $p(z)$, όταν δηλαδή $|p(z_i)| \leq \epsilon$ με ϵ ένα πολύ μικρό θετικό αριθμό. Αυτή η επαναληπτική διαδικασία αποτελεί τον *αλγόριθμο Newton-Raphson* για εύρεση ρίζας, εφαρμοσμένο σε μιγαδικές συναρτήσεις.

Η μιγαδική συνάρτηση μιγαδικής μεταβλητής $p(z) = z^3 - 1$ έχει ρίζες τα $a = 1$, $b = e^{i2\pi/3}$, $c = e^{-i2\pi/3}$. Οποιαδήποτε αρχική τιμή στο μιγαδικό επίπεδο, εκτός από την $z = 0 + i0$, θα συγκλίνει σε μία από τις ρίζες. Μπορούμε να δημιουργήσουμε μία έγχρωμη εικόνα αν σε κάθε σημείο στο μιγαδικό επίπεδο αντιστοιχήσουμε ένα χρώμα ανάλογα με το σε ποια ρίζα καταλήγει. Έτσι, π.χ., όσα σημεία καταλήγουν στην a τα χρωματίζουμε κόκκινα (RGB = (255, 0, 0)). Όσα καταλήγουν στην b τα χρωματίζουμε πράσινα (RGB = (0, 255, 0)) και όσα καταλήγουν στη c τα χρωματίζουμε μπλε (RGB = (0, 0, 255)). Το σημείο $0 + i0$ το χρωματίζουμε λευκό (RGB = (255, 255, 255)).

- (α) Επιλέξτε στον άξονα των πραγματικών $N = 512$ ισαπέχουσες τιμές στο διάστημα $[-1, 1]$ (τα άκρα περιλαμβάνονται): $x_i, i = 0, \dots, N - 1$.
- (β) Επιλέξτε στον άξονα των φανταστικών $M = 512$ ισαπέχουσες τιμές στο διάστημα $[-1, 1]$ (τα άκρα περιλαμβάνονται): $y_j, j = 0, \dots, M - 1$.
- (γ) Σχηματίστε τον μιγαδικό αριθμό $z = x_i + iy_j$ και βρείτε το «χρώμα» του με τη διαδικασία που περιγράφηκε παραπάνω.
- (δ) Αποθηκεύστε τα pixels (i, j) με το αντίστοιχο χρώμα τους σε αρχείο με όνομα *newton.pppm*. Χρησιμοποιήστε τη διαμόρφωση plain ppm. Η εικόνα στο *newton.pppm* είναι ένα Newton fractal.

Να στείλετε τους κώδικες που θα γράψετε, ως συνημμένους σε email στο ety215@materials.uoc.gr.

Διάρκεια: 3 ώρες

Καλή επιτυχία!