

# ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ I: ΕΙΣΑΓΩΓΗ ΣΤΗ C++

## Θέματα Εξετάσεων Ιανουαρίου 2019

- Το χρώμα ενός pixel μπορεί να αναπαρασταθεί με τρεις ακέραιους αριθμούς, τις συνιστώσες των χρωμάτων «κόκκινο» (R), «πράσινο» (G), «μπλε» (B). Κάθε συνιστώσα είναι ακέραιος από 0 έως και  $K$  και συνεπώς μπορούμε να έχουμε  $(K+1) \times (K+1) \times (K+1)$  συνδυασμούς. Ο συνδυασμός  $I$  μπορεί να γραφεί συναρτήσει των  $R, G, B$ , ως εξής:

$$I = R \times (K+1) \times (K+1) + G \times (K+1) + B.$$

Προφανώς  $0 \leq I < (K+1)^3$ .

Γράψτε συναρτήσεις που να μετατρέπουν από  $R, G, B$  σε  $I$  και αντίστροφα.

- Έστω  $c$  ένας μιγαδικός αριθμός. Ξεκινώντας από τον αριθμό  $z_0 = c$  μπορούμε να παραγάγουμε την ακολουθία

$$\begin{aligned} z_1 &= z_0^2 + c \\ z_2 &= z_1^2 + c \\ z_3 &= z_2^2 + c \\ &\vdots \quad \quad \quad \vdots \end{aligned}$$

Οι αριθμοί  $z_i$  έχουν μέτρο που, όταν  $i \rightarrow \infty$ , είτε τείνει στο άπειρο είτε είναι φραγμένο (μικρότερο δηλαδή από κάποια τιμή). Η συμπεριφορά αυτή εξαρτάται προφανώς από τον αριθμό  $c$ . Τα σημεία του μιγαδικού επιπέδου για τα οποία το  $|z_\infty|$  δεν είναι άπειρο αποτελούν το σύνολο Mandelbrot.

Μπορούμε να αποδώσουμε διάφορα χρώματα στα σημεία του μιγαδικού επιπέδου ως εξής:

Επιλέγουμε ένα ακέραιο  $K$  και θέτουμε  $N = (K+1)^3 - 1$ . Για κάθε σημείο  $c$  υπολογίζουμε την ακολουθία  $z_i$  με  $i = 0, 1, \dots, N-1$ . Ο δείκτης  $I$  του πρώτου όρου με μέτρο πάνω από 2 (ή το  $N$  αν δεν βρεθεί τέτοιος όρος) είναι ακέραιος στο  $[0, N]$ . Ο ακέραιος  $N - I$  (που και αυτός είναι στο  $[0, N]$ ) μπορεί να αναλυθεί σε χρωματικές συνιστώσες  $R, G, B$  στο διάστημα  $[0, K]$ .

Γράψτε συνάρτηση που να δέχεται ένα μιγαδικό αριθμό  $c$  και να υπολογίζει τα  $R, G, B$  σύμφωνα με τα παραπάνω.

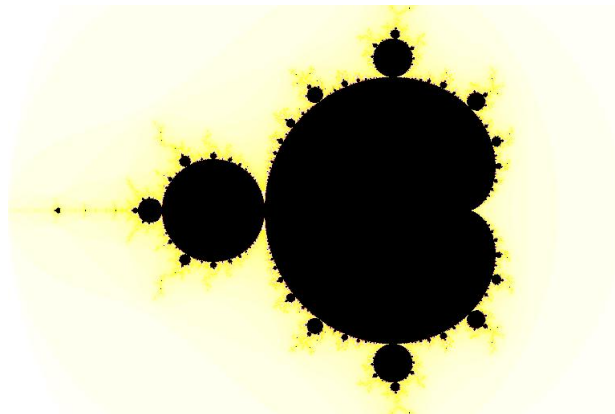
- Μια διαμόρφωση ενός αρχείου που μπορεί να χρησιμοποιηθεί για την αποθήκευση έγχρωμης εικόνας είναι η ακόλουθη:

- 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  $K$ , μέχρι το 255, που αντιπροσωπεύει τη μέγιστη τιμή του κάθε χρώματος.
- Να ακολουθούν τα pixels της εικόνας *κατά γραμμές*. Στο αρχείο θα γράφονται οι τιμές των χρωμάτων «κόκκινο» (R), «πράσινο» (G), «μπλε» (B) για το κάθε pixel, με ένα κενό μεταξύ τους. Ένα pixel που έχει χρώμα κόκκινο θα αναπαρίσταται 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.

Οι γραμμές του αρχείου πρέπει να έχουν έως 70 χαρακτήρες.

Η διαμόρφωση αυτή αποτελεί ένα αρχείο τύπου plain ppm (portable pixmap).

Στο μιγαδικό επίπεδο, επιλέξτε 768 ισαπέχοντα σημεία  $x_i$  στον άξονα των πραγματικών, στο διάστημα  $[-2, 1]$ , και 512 ισαπέχοντα σημεία  $y_j$  στον άξονα των φανταστικών, στο διάστημα  $[-1, 1]$ . Για κάθε ζεύγος σημείων  $(x_i, y_j)$  σχηματίστε το μιγαδικό  $c = x_i + iy_j$  και βρείτε το χρώμα του. Αποθηκεύστε την εικόνα σε αρχείο με διαμόρφωση ppm. Ο κώδικάς σας θα πρέπει να βγάλει την εικόνα του σχήματος 1.



Σχήμα 1: Σύνολο Mandelbrot με  $K = 31$

- (α) Γράψτε μια συνάρτηση που να δέχεται τρία ορίσματα, first, mid, last, που είναι forward iterators στην ίδια ακολουθία τιμών. Η συνάρτηση να περιστρέφει τα στοιχεία στο διάστημα [first,last) ώστε το στοιχείο στο

οποίο δείχνει ο iterator mid να έρθει πρώτο. Το στοιχείο του προηγούμενου iterator από τον mid γίνεται τελευταίο. Η συνάρτηση επιστρέφει iterator στη θέση first+(last-mid).

Επομένως, συμπληρώστε τη συνάρτηση

```
template<typename Iterator>
Iterator f(Iterator first, Iterator mid, Iterator last)
{}
```

Να μη χρησιμοποιήσετε έτοιμο αλγόριθμο της C++.

*Υπόδειξη:* Σε ένα forward iterator a επιτρέπονται οι πράξεις \*a, ++a, a!=b, a==b, όπου b άλλος forward iterator.

- (β') Αποθηκεύστε τη σειρά {1,2,3,4,5,6,7,8} σε std::forward\_list<int>. Χρησιμοποιήστε τη συνάρτηση που γράψατε για να περιστρέψετε την ακολουθία από το 2 μέχρι και το 6 έτσι ώστε το στοιχείο 5 να γίνει πρώτο. Μετά την περιστροφή, τυπώστε τη. Θα πρέπει να είναι n {1,5,6,2,3,4,7,8}.

3. Σε διάφορα προγράμματά μας χρειάζεται να χειριστούμε πίνακα δύο διαστάσεων, με στοιχεία οποιουδήποτε τύπου και με διαστάσεις που υπολογίζονται κατά την εκτέλεση των προγραμμάτων. Επιθυμούμε να είναι συμβατός ο πίνακας με συναρτήσεις της Fortran, επομένως θέλουμε να αποθηκεύει τα στοιχεία του διαδοχικά κατά στίλεις.

Ένα τέτοιο πρόγραμμα είναι το ακόλουθο:

```
#include <iostream>

#include "array2d.h"

int main()
{
    int n,m;
    std::cin >> n >> m;

    array2d<double> a(n,m); // array2d με διαστάσεις n x m
    a(1,1) = 2.5; // τιμή στο πρώτο στοιχείο του a

    for (std::size_t i{1}; i <= n; ++i) {
        for (std::size_t j{1}; j <= m; ++j) {
            a(i,j) = i+j+3.5; // τιμές για το στοιχείο aij
        }
    }

    array2d<double> b{a}; // b αντίγραφο του a
```

```

array2d<double> c(n,m);
c = a; // c αντίγραφο του a

std::cout << c(n/2,m/2) <<'\n'; // η τιμή στο (n/2,m/2)

auto k = c.size(); // το πλήθος των στοιχείων του c

auto d1 = a.dim(1); // Η πρώτη διάσταση του a (δηλαδή n).
auto d2 = a.dim(2); // Η δεύτερη διάσταση του a (δηλαδή m).

b.diagonal(3) = 2.4; // το στοιχείο (3,3)

c = a.transpose(); // ο ανάστροφος του a

c += a; // c=c+a
c -= a; // c=c-a
c *= a; // c=c*a
c = a+b; // το άθροισμα των a,b
c = a-b; // η διαφορά των a,b
c = a*b; // το γινόμενο των a,b
c = 2.5 * a;
}

```

Γράψτε το αρχείο *array2d.h* στο οποίο θα ορίζεται ο νέος τύπος (η κλάση *array2d*) ως *template*.

**Να στείλετε τους κώδικες που θα γράψετε συνημμένους σε email  
στο [ety215@materials.uoc.gr](mailto:ety215@materials.uoc.gr).**

**Διάρκεια: 3 ώρες**

**Καλή επιτυχία!**