Προχωρημενος Προγραμματισμος I : Εισαγωγη στη C++

Θέματα Εξετάσεων Ιανουαρίου 2010

1. Ένας τρόπος να σχεδιάσουμε ένα διδιάστατο fractal είναι ο εξής: ξεκινάμε από ένα σημείο του επιπέδου, έστω το ($x = 0$, $y = 0$), και το μετακινούμε στη θέση (x', y') όπου 3/10

$$\begin{aligned} x' &= a \cdot x + b \cdot y + e \\ y' &= c \cdot x + d \cdot y + f \end{aligned}$$

και a, b, c, d, e, f σταθερές.

Το νέο σημείο το μεταφέρουμε με τον ίδιο μετασχηματισμό στο επόμενο σημείο του fractal (δηλαδή, θέτουμε $x' \to x$ και $y' \to y$ και παράγουμε το νέο (x', y')). Τη διαδικασία αυτή την επαναλαμβάνουμε επ' άπειρο. Η ακολουθία των σημείων (x, y) που παράγονται, αποτελεί το fractal.

Γράψτε ένα πρόγραμμα το οποίο:

(α') Θα διαβάζει από το αρχείο "in.dat" 4 γραμμές. Σε κάθε γραμμή θα υπάρχουν 7 πραγματικοί αριθμοί: οι 6 πρώτοι αντιστοιχούν στους συντελεστές a, b, c, d, e, f και ο τελευταίος στην πιθανότητα p να γίνει ο συγκεκριμένος μετασχηματισμός. Κάθε γραμμή αντιστοιχεί σε άλλο μετασχηματισμό. Το άθροισμα των πιθανοτήτων, $\sum p_i$, όλων των μετασχηματισμών είναι 1.

(β') Θα επιλέγει ένα τυχαίο πραγματικό αριθμό r στο διάστημα $[0, 1)$. Ανάλογα με την τιμή του θα εφαρμόζεται διαφορετικός μετασχηματισμός. Δηλαδή, αν ισχύει $0 \leq r < p_1$ θα εκτελείται ο πρώτος μετασχηματισμός, αν ισχύει $p_1 \leq r < p_1 + p_2$ θα εκτελείται ο δεύτερος κλπ.

(γ') Θα επαναλαμβάνει το προηγούμενο βήμα 1000 φορές σώζοντας κάθε φορά το σημείο που προκύπτει στο αρχείο "fractal.dat".

Δοκιμάστε το πρόγραμμά σας με τις εξής παραμέτρους στο "in.dat"

0	0	0	0.16	0	0	0.01
0.85	0.04	−0.04	0.85	0	1.6	0.85
0.2	−0.26	0.23	0.22	0	1.6	0.07
−0.15	0.28	0.26	0.24	0	0.44	0.07

και

0	0	0	0.25	0	−0.4	0.02
0.95	0.005	−0.005	0.93	−0.002	0.5	0.84
0.035	−0.2	0.16	0.04	−0.09	0.02	0.07
−0.04	0.2	0.16	0.04	0.083	0.12	0.07

Αν θέλετε, μπορείτε να σχεδιάσετε τα "fractal.dat" που προκύπτουν.

2. (α΄) Γράψτε μια συνάρτηση που να συγχωνεύει δύο ήδη ταξινομημένους πίνακες δημιουργώντας ένα τρίτο, επίσης ταξινομημένο.[1] Η συνάρτηση θα δέχεται για ορίσματα 4/10

   - δύο μονοδιάστατους πίνακες που θα θεωρούνται ταξινομημένοι από το μικρότερο στοιχείο στο μεγαλύτερο. Οι πίνακες μπορεί να έχουν διαφορετικό πλήθος στοιχείων. Ο τύπος των στοιχείων μπορεί να είναι οποιοσδήποτε αριθμητικός.
   - ένα τρίτο μονοδιάστατο πίνακα. Θα θεωρείται ότι έχει πλήθος στοιχείων τουλάχιστον ίσο με τον αριθμό των στοιχείων των δύο πινάκων εισόδου.

   Η συνάρτηση θα αντιγράφει τα στοιχεία των δύο πρώτων πινάκων στον τρίτο με τέτοιο τρόπο ώστε να είναι ταξινομημένα σε αυτόν από το μικρότερο στο μεγαλύτερο.

   (β΄) Χρησιμοποιήστε το για να συγχωνεύσετε τους πίνακες $[-4.21, -2.19, -0.1, 0.06, 2.3]$ και $[-3.2, -3.1, -1.7, 0.4, 0.88, 3.5]$ σε ένα τρίτο πίνακα, τον οποίο θα τυπώσετε στην οθόνη.

3. Ένας θετικός ακέραιος αριθμός μπορεί να ενταχθεί σε κατηγορίες ως εξής: υπολογίζουμε το άθροισμα των (θετικών) ακεραίων που τον διαιρούν ακριβώς, χωρίς να λαμβάνουμε υπόψη τον εαυτό του. Ο ακέραιος αριθμός λέγεται *perfect* (τέλειος) αν αυτό το άθροισμα είναι ίσο με τον ακέραιο αριθμό. Αν το άθροισμα είναι μεγαλύτερο από τον αριθμό, αυτός λέγεται *abundant*. Αν είναι μικρότερο, ο αριθμός λέγεται *deficient*. Έτσι π.χ., το 6 είναι τέλειος καθώς για τους διαιρέτες του, $1, 2, 3$ (αλλά όχι το 6), ισχύει $6 = 1 + 2 + 3$. 3/10

   (α΄) Να γράψετε συνάρτηση που να υπολογίζει αν το όρισμά της, ένας ακέραιος αριθμός, είναι perfect, abundant ή deficient. Ανάλογα με το τι θα βρει για τον αριθμό να επιστρέφει $0, 1$, ή -1 αντίστοιχα.

   (β΄) Χρησιμοποιήστε τη για να χαρακτηρίσετε τους αριθμούς από το 1 έως το 10000. Δημιουργήστε το αρχείο "perfect.dat", σε κάθε γραμμή του οποίου θα τυπώνετε έναν ακέραιο αριθμό (διαδοχικά όλους από το 1 ως το 10000) και δίπλα (αφήνοντας ένα κενό μεταξύ τους) μια από τις λέξεις perfect, abundant, deficient ανάλογα με την κατηγορία στην οποία ανήκει ο αριθμός.

[1] Υλοποιείστε, δηλαδή, τον αλγόριθμο std::merge.