

ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι :

ΕΙΣΑΓΩΓΗ ΣΤΗ C++

Θέματα Εξετάσεων Σεπτεμβρίου 2007

1. Από τα Μαθηματικά γνωρίζουμε ότι ισχύει 2/10

$$\pi = 3 + 2 \sum_{k=1}^{\infty} \frac{k(5k+3)(2k-1)!k!}{2^{k-1}(3k+2)!}.$$

Χρησιμοποιήστε την παραπάνω σχέση για να υπολογίσετε το π με ακρίβεια 10^{-6} . αυτό σημαίνει ότι στον υπολογισμό του αθροίσματος θα σταματήσετε στον πρώτο όρο που είναι μικρότερος από 10^{-6} . Συγκρίνετε το αποτέλεσμα σας με την “σωστή” τιμή.

Υπόδειξη: Στον υπολογισμό σας μπορείτε να βασιστείτε στο ότι ο κάθε όρος στο άθροισμα προκύπτει από τον προηγούμενο με πολλαπλασιασμό κατάλληλης ποσότητας.

2. Υλοποιήστε με αναδρομική συνάρτηση τον αλγόριθμο ταξινόμησης quick-sort. Η ρουτίνα θα δέχεται ως όρισμα ένα μονοδιάστατο vector της STL, με στοιχεία οποιουδήποτε τύπου, το οποίο θα ταξινομεί. Για τη διευκόλυνσή σας, η ρουτίνα θα παίρνει ως ορίσματα τους iterators που προσδιορίζουν την αρχή και το “τέλος” του vector. 4/10

Η απλή εκδοχή του αλγορίθμου είναι η ακόλουθη:

- (α) Επιλέγουμε ένα οποιοδήποτε στοιχείο της αρχικής συλλογής (π.χ. το μεσαίο).
- (β) Δημιουργούμε ένα νέο, *κατάλληλο* container της STL, με μοναδικό στοιχείο ένα αντίγραφο του επιλεγμένου στο προηγούμενο βήμα.
- (γ) Αντιγράφουμε σε διαδοχικές θέσεις *στα αριστερά* του μοναδικού στοιχείου στο νέο container τα στοιχεία του αρχικού vector που είναι *μικρότερα* από το επιλεγμένο στοιχείο.
- (δ) Αντιγράφουμε σε διαδοχικές θέσεις *στα δεξιά* του αρχικού στοιχείου στο νέο container τα στοιχεία του αρχικού vector που είναι *μεγαλύτερα ή ίσα* από το επιλεγμένο στοιχείο.
- (ε) Αντιγράφουμε τα στοιχεία του νέου container στον παλιό (σβήνοντας τα δικά του).
- (ς) Επαναλαμβάνουμε τη διαδικασία για τους δύο υπο-vectors εκατέρωθεν του επιλεγμένου στοιχείου, *αν έχουν, βέβαια, διάσταση μεγαλύτερη από 1* (αλλιώς δεν χρειάζονται ταξινόμηση).

Χρησιμοποιήστε τη ρουτίνα σας για να ταξινομήσετε τα στοιχεία (-1.2, 9.1, 8.2, -6.4, 3.8, 0.34, 5.67, 8.99, -9.01, 0.001, 4.8, 3.8).

3. Ένας τρόπος για να αναπαραστήσουμε σε πρόγραμμα τα φύλλα μιας τράπουλας είναι ο εξής: αντιστοιχούμε τους αριθμούς 1 – 13, 14 – 26, 27 – 39, 4/10

40 – 52 στα τέσσερα “χρώματα” της τράπουλας. Οι αριθμοί 11, 12, 13 αντιστοιχούν στα J , Q , K (τις “φιγούρες”) του πρώτου χρώματος· αντίστοιχα ισχύουν για τα υπόλοιπα χρώματα.

Αποθηκεύστε σε έναν πίνακα¹ τους αριθμούς 1 – 52· αυτός θα αποτελεί την “τράπουλά” μας. Για να την ανακατώσουμε με τυχαίο τρόπο στον υπολογιστή μπορούμε να ακολουθήσουμε την εξής διαδικασία:

Δημιουργούμε ένα πίνακα 52 τυχαίων αριθμών (έστω πραγματικών στο διάστημα $[0, 1)$), προσέχοντας να μην επαναλαμβάνονται στοιχεία σε αυτόν. Κατόπιν τον ταξινομούμε, κάνοντας ταυτόχρονα στην τράπουλά μας οποιαδήποτε μετακίνηση στοιχείων υπαγορεύει η διαδικασία ταξινόμησης. Π.χ., αν κατά την ταξινόμηση χρειαστεί να εναλλάξουμε τα στοιχεία 15 και 28 του τυχαίου πίνακα, κάνουμε ταυτόχρονα την εναλλαγή των ίδιων στοιχείων της τράπουλάς μας. Στο τέλος της διαδικασίας, ο τυχαίος πίνακας θα είναι ταξινομημένος και η “τράπουλά” μας θα είναι ανακατωμένη με τυχαίο τρόπο.

Υλοποιείστε τον παραπάνω αλγόριθμο για το ανακάτεμα. Θα σας χρειαστούν:

- Μία συνάρτηση παραγωγής (ψευδο)τυχαίων αριθμών.
- Ένα υποπρόγραμμα αναζήτησης συγκεκριμένης τιμής σε πίνακα.
- Ένα υποπρόγραμμα ταξινόμησης (με κατάλληλη τροποποίηση).

Τυπώστε στο αρχείο “shuffle.dat” 50 τυχαία ανακατωμένες “τράπουλες”. Η κάθε μία “τράπουλα” θα τυπώνεται σε ξεχωριστή γραμμή. Τα “χαρτιά” της τράπουλας, δηλ. τα στοιχεία του πίνακα, θα τυπώνονται με ένα κενό ανάμεσά τους.

¹ή σε όποιο container σας διευκολύνει.