Προχωρημενος Προγραμματισμος I : Εισαγωγη στη C++

Θέματα Εξετάσεων Ιανουαρίου 2013

1. Ένας θετικός ακέραιος αριθμός λέγεται παλίνδρομος όταν μπορεί να "διαβαστεί" το ίδιο και από δεξιά προς τα αριστερά. Π.χ. οι αριθμοί 85158 και 20988902 είναι παλίνδρομοι. 2/10

   Γράψτε κώδικα που να τυπώνει στο αρχείο "palindrome.dat" όλους τους παλίνδρομους αριθμούς μέχρι το 1000000. Στην οθόνη να τυπώνει το πλήθος τους.

   Υπόδειξη: Θα χρειαστείτε τουλάχιστον δύο συναρτήσεις: μία που θα υπολογίζει το πλήθος των ψηφίων ενός ακέραιου αριθμού και μία άλλη που θα αναλύει τον αριθμό στα ψηφία του.

2. Η επίλυση της χρονικά ανεξάρτητης εξίσωσης Schrödinger για τον μονοδιάστατο αρμονικό ταλαντωτή (μάζα m σε δυναμικό $V = kx^2/2$) καταλήγει στις ιδιοσυναρτήσεις 4/10

$$\psi_n(y) = \sqrt{\frac{1}{2^n n! \sqrt{\pi}}} H_n(y) \mathrm{e}^{-y^2/2} \ ,$$

   όπου $y = x\sqrt{\sqrt{km}/\hbar}$.

   Να τυπώσετε στο αρχείο με όνομα "harmonic.dat" τις τιμές της πυκνότητας πιθανότητας ($\psi\psi^*$) για $n = 5$ σε 60 ισαπέχοντα σημεία x στο διάστημα $[-6 : 6]$, μαζί με τα αντίστοιχα σημεία x (δηλαδή το αρχείο θα περιέχει δύο στήλες, x και $\psi\psi^*$).

   Για να το κάνετε αυτό:

   (α΄) Γράψτε συνάρτηση που να υπολογίζει το παραγοντικό ενός μικρού ακεραίου.

   (β΄) Γράψτε συνάρτηση που να υπολογίζει την τιμή των πολυωνύμων Hermite, $H_n(x)$, για κάποια τιμή του x. Τα πολυώνυμα αυτά ικανοποιούν την αναδρομική σχέση:

$$H_n(x) - 2xH_{n-1}(x) + 2(n-1)H_{n-2}(x) = 0 \ , \qquad n \geq 2 \ ,$$

   με $H_0(x) = 1$, $H_1(x) = 2x$.

   (γ΄) Χρησιμοποιείστε τις συναρτήσεις που γράψατε παραπάνω για να γράψετε μια νέα συνάρτηση για την ψ. Αυτή θα δέχεται ως ορίσματα τα n, x. Θεωρήστε ότι $m = k = \hbar = 1$.

3. (α΄) Υλοποιήστε τον αλγόριθμο `std::equal()` της STL σε δικιά σας συνάρτηση με όνομα equal. Ο αλγόριθμος `std::equal()` δέχεται τρεις iterators, beg1, end1, beg2. Οι δύο πρώτοι "δείχνουν" στον ίδιο container ενώ ο τρίτος iterator "δείχνει" σε άλλο container. Ο αλγόριθμος 4/10

συγκρίνει τα στοιχεία του πρώτου container μεταξύ των beg1, end1 με τα αντίστοιχα του δεύτερου (ξεκινώντας από το beg2). Αν όλα τα αντίστοιχα στοιχεία είναι μεταξύ τους ίσα επιστρέφει **true** αλλιώς επιστρέφει **false**. Συμπληρώστε, επομένως, τον κώδικα

```
template<typename Iterator>
bool
equal(Iterator beg1, Iterator end1, Iterator beg2) {
.....
}
```

(β΄) Χρησιμοποιώντας το `std::list`, δημιουργήστε δύο λίστες ακέραιων αριθμών και αποθηκεύστε τις ακολουθίες $1, 3, 5, 88, 92, 4, 91$ στην πρώτη και $1, 3, 5, 88, 92, 4, 2, 91$ στη δεύτερη. Καλέστε τη δικιά σας συνάρτηση `equal()` για να ελέγξετε αν είναι ίσες οι δύο ακολουθίες (προφανώς πρέπει να σας επιστρέψει **false**).