

ΗΥ-150 Προγραμματισμός

Εαρινό Εξάμηνο 2021

Πανεπιστήμιο Κρήτης

Σειρά Ασκήσεων 2.

Ημερομηνία έναρξης:

19-03-21

Ημερομηνία παράδοσης:

02-04-21, ώρα 23:59

1. **(20%)** Γράψτε μια αναδρομική συνάρτηση `recursiveLinearSearch` όπου αναδρομικά και γραμμικά ελέγχει κάθε στοιχείο ενός δυναμικού πίνακα `vector` στην αναζήτηση ενός στοιχείου-κλειδιού που έχει ορίσει ο χρήστης. Η συνάρτηση πρέπει να λαμβάνει μεταξύ άλλων, το στοιχείο-κλειδί καθώς και αρχική θέση στο `vector` σαν ορίσματα. Τα ορίσματα αυτά, τα οποία θα δώσετε ως δεδομένα εισόδου στην αρχή του προγράμματος, θα πρέπει να βεβαιωθείτε ότι είναι αποδεκτά δεδομένα εισόδου. Αν το στοιχείο-κλειδί βρεθεί στο `vector`, να επιστρέφει την θέση του στο `vector` αλλιώς την τιμή `-1`. Κάθε κάλεσμα της αναδρομικής συνάρτησης πρέπει να ελέγχει ένα στοιχείο στο `vector`. Αρχικοποιήστε κατάλληλα με κώδικα ένα `vector` με όλους του ζυγούς αριθμούς μέχρι και το 150.
2. **(30%)** Υλοποιήστε έναν καινούργιο τύπο ορθογώνιου `Rectangle` με χαρακτηριστικά `length` και `width`, με αρχικές τιμές 1.0. Προσθέστε κατάλληλες συναρτήσεις-μέλη που να υπολογίζουν την περίμετρο (`perimeter`) και εμβαδόν (`area`) του ορθογώνιου. Επίσης υλοποιήστε κατάλληλες μεθόδους για `set()/get()` των χαρακτηριστικών `length`, `width`. Αυτές οι συναρτήσεις πρέπει να επιβεβαιώνουν αν οι τιμές των παραμέτρων αυτών είναι στα όρια 0.0 έως 30.0 με την χρήση των `standard Exceptions` και ειδικά της `invalid_argument`. Δοκιμάστε να αρχικοποιήσετε ένα `Rectangle` με τις αρχικές, μετά με αποδεκτές και τέλος με ανεπαρκείς τιμές ώστε να επιβεβαιωθούν τα `Exceptions` με ανάλογα `output statements`. Τα `declarations` και `definitions` του `Rectangle` πρέπει να βρίσκονται σε χωριστά αρχεία.
3. **(30%)** Μια βιβλιοθήκη χρεώνει 2 EUR ελάχιστη αμοιβή τον δανισμό ενός βιβλίου για 3 μέρες. Επίσης χρεώνει επιπλέον 1.5 EUR/ημέρα για κάθε μέρα πάνω από τις 3 ημέρες. Η μέγιστη χρέωση για 2 εβδομάδες (14 ημέρες) είναι 17.50 EUR. Υποθέστε ότι δεν δανίζει βιβλία για πάνω από 2 εβδομάδες. Υλοποιήστε ένα πρόγραμμα που υπολογίζει και εκτυπώνει σε ένα αρχείο τις χρεώσεις βιβλίων για πέντε(5) πελάτες τις τελευταίες 15 ημέρες. Ζητήστε από τον χρήστη να εισάγει τις μέρες που έχει δανειστεί ο κάθε πελάτης. Το πρόγραμμά σας εκτυπώνει τα αποτελέσματα σε ένα αρχείο σε μορφή πίνακα και υπολογίζει και εκτυπώνει το συνολικό ποσό χρεώσεων του δεκαπενθήμερου που πέρασε. Το πρόγραμμά σας πρέπει να υλοποιήσει μια συνάρτηση `calculateCharges` που να υπολογίζει την χρέωση για κάθε πελάτη.

*Ενδεικτικό περιεχόμενο αρχείου

Table_of_Clients:		
	Charge	Days
Client_1	2.00	03
Client_2	6.50	06
Client_3	12.50	10
Client_4	17.50	14
Client_5	9.50	08
Sum:	48.00	41

4. **(20%)** Δημιουργήστε μια στατική βιβλιοθήκη (στον ίδιο φάκελο του ερωτήματος αυτού), η οποία θα περιλαμβάνει μια κλάση που θα έχει κατάλληλες συναρτήσεις-μέλη που θα κάνουν κοινές μαθηματικές πράξεις όπως πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση πάνω σε 2 floats.
- Χτίστε και συνδέστε (**build & link**) την στατική αυτή βιβλιοθήκη μέσω του cmake.
 - Υλοποιήστε παραδείγματα που την καλούν (ένα για κάθε συνάρτηση-μέλος της κλάσης).