

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εργαστήριο 1: Διαδικαστικός προγραμματισμός – δυναμική δέσμευση & αποδέσμευση μνήμης

ΕΡΓΑΣΤΗΡΙΟ 1:

Διαδικαστικός προγραμματισμός – Δυναμική δέσμευση & αποδέσμευση μνήμης

Σκοπός της ενότητας αυτής είναι να θυμηθούμε κάποιες έννοιες του διαδικαστικού προγραμματισμού και να εξασκηθούμε σε αυτές. Πιο συγκεκριμένα θα επικεντρωθούμε στην δυναμική δέσμευση και αποδέσμευση μνήμης στη C++ σε πολυδιάστατους πίνακες και structures.

Άσκηση 1.1

Δημιουργήστε μία συνάρτηση που θα δέχεται ως είσοδο έναν ακέραιο και θα επιστρέφει 0 αν είναι άρτιος και 1 αν είναι περιττός.

Άσκηση 1.2

Δημιουργήστε μία συνάρτηση που θα δέχεται ως είσοδο έναν ακέραιο και θα επιστρέφει 0 αν είναι πρώτος αριθμός και 1 αν όχι. Υπόδειξη: Πρώτοι είναι οι αριθμοί που διαιρούνται μόνο με τον εαυτό τους και την μονάδα (1).

Άσκηση 1.3

Κατασκευάστε το πρόγραμμα primes.cc το οποίο θα δέχεται ως είσοδο έναν ακέραιο και

- i. θα βρίσκει όλους τους πρώτους αριθμούς μέχρι τον συγκεκριμένο ακέραιο και θα τους γράφει σε έναν μονοδιάστατο πίνακα.
- ii. θα γράφει σε έναν δεύτερο μονοδιάστατο πίνακα τους περιττούς πρώτους αριθμούς που έχει βρεί και
- iii. θα γράφει σε έναν τρίτο μονοδιάστατο πίνακα τους άρτιους πρώτους αριθμούς.
- iv. θα εκτυπώνει τους πίνακες με τα στοιχεία του κάθε πίνακα χωρισμένα με στηλογνώμονα (tab).

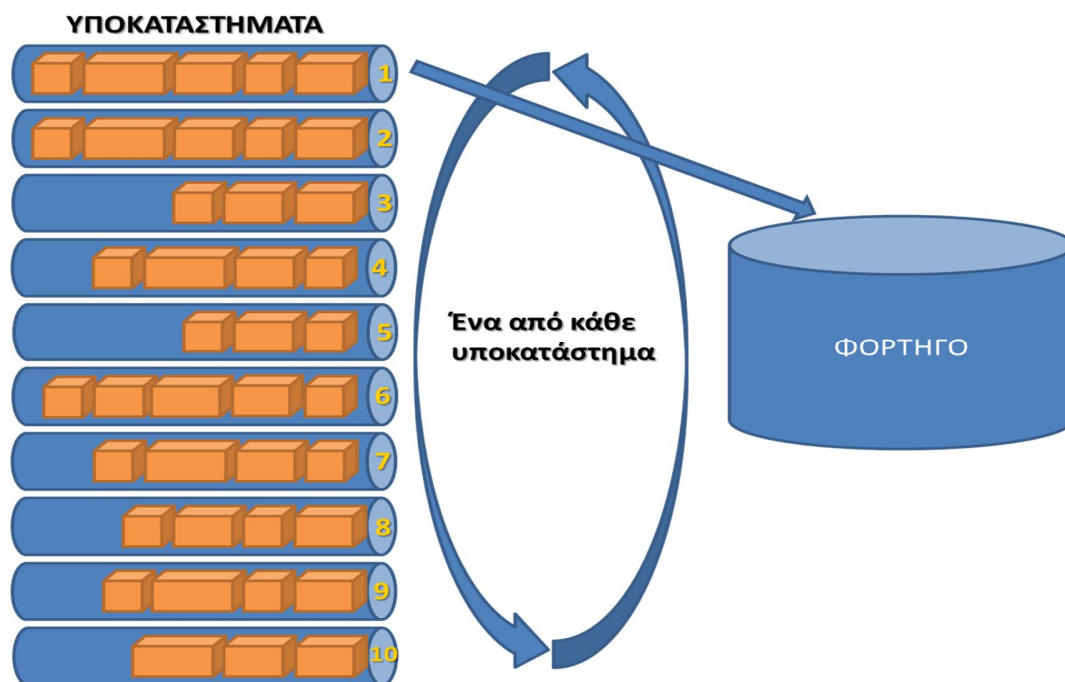
ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εργαστήριο 1: Διαδικαστικός προγραμματισμός – δυναμική δέσμευση & αποδέσμευση μνήμης

Άσκηση 2

Έστω ότι οι ταχυμεταφορές DiT Courier έχουν 10 υποκαταστήματα και ένα κέντρο διαχείρισης. Κάθε μέρα (κύκλος εκτέλεσης) συλλέγονται όλα τα δέματα-πακέτα στο κέντρο διαχείρισης και από εκεί φορτώνονται σε ένα φορτηγό, π.χ. για να μεταφερθούν στο επόμενο διακομιστικό κέντρο. Το φορτηγό έχει συγκεκριμένη χωρητικότητα (σε μονάδες) και ο τρόπος που φορτώνεται είναι ο παρακάτω:

Με μία κυκλική επανάληψη και για λόγους δικαιοσύνης το φορτηγό φορτώνεται με ένα δέμα από κάθε υποκατάστημα μέχρι να φτάσει στο όριο της χωρητικότητάς του. Μάλιστα στην τελευταία επανάληψη και επειδή τα δέματα μπορεί να έχουν διαφορετικά μεγέθη, οπότε άλλα να χρειάζονται μικρή χωρητικότητα ενώ άλλα μεγάλη, για κάθε ένα από τα δέκα υποκαταστήματα, αν το δέμα του δεν χωράει στο φορτηγό μπορεί να γίνει έλεγχος και για τα επόμενα υποκαταστήματα. Η παρακάτω εικόνα αναπαριστά ένα παράδειγμα:



Άσκηση 2.1

Με την βοήθεια της συνάρτησης “rand” κατασκευάστε μία συνάρτηση η οποία θα επιστρέφει το ακέραιο μέρος του λογαρίθμου του αποτελέσματος της συνάρτησης “rand” (αρχεία αναφοράς cstdlib, cmath).

Άσκηση 2.2

Κατασκευάστε μία struct, την “dema”, η οποία θα εμπεριέχει τα παρακάτω:

- i. id → ακέραιος
- ii. idKatasthnmatos → ακέραιος
- iii. xwros → ακέραιος

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εργαστήριο 1: Διαδικαστικός προγραμματισμός – δυναμική δέσμευση & αποδέσμευση μνήμης

Άσκηση 2.3

Κατασκευάστε συνάρτηση που θα παίρνει ως είσοδο έναν ακέραιο (αριθμός υποκαταστήματος 1 ως 10) και θα δημιουργεί τυχαίο αριθμό δεμάτων για το υποκατάστημα αυτό. Κάθε δέμα θα παίρνει κατά την δημιουργία του ως **“id”** έναν αύξοντα αριθμό (*ίδιο για όλα τα υποκαταστήματα), **“idKatastnmatos”** τον αριθμό του υποκαταστήματος (είσοδος) και ως **“xwros”** έναν τυχαίο. Η συνάρτηση αυτή να επιστρέφει μονοδιάστατο πίνακα με τα δέματα του υποκαταστήματος. Για τη δημιουργία τυχαίων αριθμών, χρησιμοποιείστε τον κώδικα που γράψατε για την Άσκηση 2.1.

Άσκηση 2.4

Κατασκευάστε συνάρτηση που θα δέχεται ως είσοδο δισδιάστατο πίνακα (υποκαταστήματα με δέματα) και θα επιστρέφει ένα μονοδιάστατο πίνακα με τα δέματα που μπορούν να χωρέσουν σε C μονάδες χωρητικότητας. Το C να είναι παράμετρος.

Άσκηση 2.5

Κατασκευάστε πρόγραμμα που για N επαναλήψεις θα δημιουργεί τυχαίο αριθμό δεμάτων για κάθε ένα από τα 10 υποκατάστημα. Σε κάθε επανάληψη θα γεμίζει το φορτηγό, χωρητικότητας C μονάδων, και θα εκτυπώνει για κάθε υποκατάστημα πόσα δέματα φορτώθηκαν στο φορτηγό και την συνολική τους χωρητικότητα. Στο τέλος της εκτέλεσης για κάθε υποκατάστημα να εκτυπωθούν πόσα δέματα και ποιας συνολικής χωρητικότητας παραμένουν χωρίς να εξυπηρετηθούν. Το N και το C να είναι παράμετροι του προγράμματος και να δίδονται από τη γραμμή εντολής. Η δημιουργία δεμάτων και το «γέμισμα» του φορτηγού να γίνονται με χρήση του κώδικα που γράψατε για τις Ασκήσεις 2.3 και 2.4, αντίστοιχα.