

1. Να δημιουργήσετε σε πρόγραμμα της C και να αξιοποιήσετε μια **στατική στοίβα**. Χρησιμοποιώντας τη στοίβα αυτή να υπολογίζετε την **οκταδική παράσταση** του AEM σας. Το πρόγραμμά σας θα ορίζει τη στοίβα με διάσταση 5.

Με τη χρήση της ίδιας στοίβας να εξετάζετε αν η οκταδική παράσταση του AEM σας είναι **παλινδρομικός** αριθμός ή όχι; Παλινδρομικός χαρακτηρίζεται ο αριθμός που διαβάζεται το ίδιο από το τέλος προς την αρχή π.χ. 3443 ή 6666

2. Να δημιουργήσετε μονοδιάστατο πίνακα 50 θέσεων στον οποίο να καταχωρείτε σε κάθε του θέση τον AEM φοιτητή με τον τυχαίο βαθμό σε κάποιο μάθημα. Οι AEM των φοιτητών θα είναι σε αύξουσα σειρά και θα είναι οι πρώτοι αριθμοί από το 2000 έως το 2400. Ο δε βαθμός κάθε φοιτητή, με ένα δεκαδικό ψηφίο, θα προκύπτει τυχαία με τη χρήση της συνάρτησης rand() και κυμαίνεται στο διάστημα [0.0, 10.0] . Για την αρχικοποίηση της σειράς των τυχαίων να χρησιμοποιήσετε την συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας.

Στη συνέχεια, το πρόγραμμά σας να ζητά να του πληκτρολογηθεί ένα AEM προκειμένου να εμφανίσει την βαθμολογία του φοιτητή. Η αναζήτησή του να γίνεται με τον αλγόριθμο της δυαδικής αναζήτησης. Ως εφαρμογή να εντοπίσετε και να εμφανίσετε στην οθόνη τον βαθμό του φοιτητή με AEM 2371.

Να εμφανίζονται επίσης στην οθόνη αναλυτικά τα βήματα της αναζήτησης. Δηλαδή ποια θέση εξέτασε πρώτα, ποια στη συνέχεια και τελικά να εμφανιστεί όλη η διαδοχική σειρά των θέσεων που εξέτασε. Αν δεν εντοπίσει την καταχώριση να εμφανίσει μήνυμα ότι δεν υπάρχει τέτοια καταχώρηση.

Να βρείτε και να εμφανίσετε στην οθόνη πόσοι από τους καταχωρημένους φοιτητές έχουν μεγαλύτερο βαθμό από τον βαθμό δεδομένου φοιτητή. Εφαρμόστε το για τον φοιτητή με AEM 2371. Να εμφανισθεί επίσης ο αριθμός των συνολικών βημάτων εξέτασης που έκανε το πρόγραμμά σας για να απαντήσει στο ερώτημα αυτό, δηλαδή εξέτασε-έλεγε για παράδειγμα 23 περιπτώσεις.

Υπόδειξη : Να δημιουργήσετε κατάλληλη συνάρτηση που να βρίσκει τους πρώτους αριθμούς και να τους καταχωρείτε ως AEM. Πρώτος λέγεται ο αριθμός που διαιρείται ακριβώς μόνο με τον εαυτό του και τη μονάδα. Στο πιο πάνω διάστημα οι πρώτοι αριθμοί είναι 2003, 2011, 2017, 2027, 2029, 2039, 2053, 2063, 2069, 2081, 2083, 2087, 2089, 2099, 2111, 2113, 2129, 2131, 2137, 2141, 2143, 2153, 2161, 2179, 2203, 2207, 2213, 2221, 2237, 2239, 2243, 2251, 2267, 2269, 2273, 2281, 2287, 2293, 2297, 2309, 2311, 2333, 2339, 2341, 2347, 2351, 2357, 2371, 2377, 2381.

3. Σε **Super Market** υπάρχουν τρία αριθμημένα ταμεία για την εξυπηρέτηση των πελατών. Οι

πελάτες εξυπηρετούνται με τη γνωστή σειρά σχηματίζοντας ουρές σε καθένα από αυτά. Ο μέγιστος αριθμός καταχωρήσεων στις ουρές είναι 10 πελάτες. Αν όλες οι ουρές υπερβούν τις 10 καταχωρήσεις το πρόγραμμα αγνοεί κάθε νέα προσέλευση πελάτη. Βέβαια κάθε πελάτης επιλέγει το ταμείο με την μικρότερη ουρά. Με τη σειρά της κάθε υπάλληλος του ταμείου χρειάζεται χρόνο για να εξυπηρετεί τους πελάτες ανάλογα με το πλήθος των αγορών τους.

Να θεωρηθεί ότι κάθε τρία προϊόντα εξυπηρετείται ένας πελάτης. Παράδειγμα ο πελάτης με 2 προϊόντα θεωρείται ως μια εξυπηρέτηση, ενώ ο πελάτης με 14 προϊόντα θα θεωρείται ως τέσσερεις εξυπηρετήσεις. Αυτό σημαίνει ότι ο χρόνος εξυπηρέτησης των πελατών που περιμένει σε κάθε ουρά, εξαρτάται από το πόσα προϊόντα είχαν στο καλάθι οι προηγούμενοι.

Να δημιουργήσετε ένα πρόγραμμα που να προσομοιάζει τη λειτουργία αυτή του Super Market. Θα χρησιμοποιήσετε τη δομή της ουράς. Σε κάθε μια ουρά θα καταχωρούνται ως μέλη κόμβοι στους οποίους καταχωρούνται το πλήθος των προϊόντων του πελάτη.

Για τις ανάγκες του προβλήματος θεωρίστε ότι όταν υπάρχει ο ίδιος αριθμός πελατών στις ουρές ο προσερχόμενος πελάτης διαλέγει τυχαία, μεταξύ των ισοδύναμων ουρών. Ο πελάτης που επέλεξε και καταχωρήθηκε σε κάποια ουρά δεν έχει τη δυνατότητα να αλλάξει ακόμα και αν έχει αδειάσει οποιαδήποτε άλλη ουρά.

Ο μηχανισμός με τον οποίο προσέρχεται ο επόμενος πελάτης ή εξυπηρετείται από την ταμία καθορίζεται από το αποτέλεσμα ζαριάς. Αν το αποτέλεσμα της ζαριάς είναι μικρότερο του 3 θεωρίστε ότι προσέρχεται στα ταμεία νέος πελάτης, ο οποίος και θα διαλέξει μια από τις τρεις ουρές. Αν η ζαριά είναι μεγαλύτερη του 2 θεωρίστε ότι όλοι οι ταμίες εξυπηρετούν έναν πελάτη τους, κατά μία μονάδα, δηλαδή κατά τρία προϊόντα. Δεν μεταφέρεται η εξυπηρέτηση σε επόμενο πελάτη. Δηλαδή αν ο τρέχων τους πελάτης έχει ένα προϊόν, δεν μεταφέρεται η εξυπηρέτηση του ταμία στον επόμενο πελάτη ώστε να τον εξυπηρετήσει κατά τα δύο προϊόντα που υπολείπονται.

Για την αρχικοποίηση του προγράμματος θα χρησιμοποιήσετε τη συνάρτηση `srand(AEM)` όπου `AEM` ο αριθμός μητρώου σας. Για παράδειγμα αν ο αριθμός μητρώου μου είναι το 9999 θα καλείται με `srand(9999)`. Με την τυχαία συνάρτηση `rand()` θα χειρίζεστε το πλήθος των προϊόντων που έχει κάθε πελάτης στο καλάθι αγορών του και αυτό θα κυμαίνεται από 2 μέχρι 17 προϊόντα. Επίσης με την `rand()` θα φέρνετε τις ζαριές που θα καθορίζουν την κατάσταση, αν δηλαδή προσέρχεται νέος πελάτης ή γίνεται ενέργεια εξυπηρέτησης από τις ταμίες.

Το σύστημα σε κάθε νέο κύκλο εμφανίζει στην οθόνη την κατάσταση, δηλαδή σε κάθε ταμείο πόσοι πελάτες περιμένουν στην ουρά του και πόσοι πελάτες έχουν ήδη εξυπηρετηθεί. Το πρόγραμμα ολοκληρώνεται όταν εξυπηρετηθούν 60 συνολικά πελάτες, και στα τρία ταμία.