

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
ΕΜΒΟΛΙΜΗ ΠΕΡΙΟΔΟΣ ΙΑΝΟΥΑΡΙΟΥ 2021
ΕΙΣΗΓΗΤΗΣ: Ι. ΞΕΖΩΝΑΚΗΣ

Θέμα 1 (Βαθμοί 3.4):

α) (Βαθμοί 1.8). Στη `main()` ενός προγράμματος να δηλώσετε ένα πίνακα ακεραίων $N \times N$, τον `mat`. Να καλείτε μια συνάρτηση, την `result()`, η οποία περιγράφεται παρακάτω και να γράφετε στην οθόνη την τιμή επιστροφής της.

Η συνάρτηση `result()` θα διαπιστώνει εάν ο πίνακας `mat` είναι αραιός και συμμετρικός. Εάν ο πίνακας είναι αραιός και συμμετρικός η συνάρτηση θα επιστρέφει 1, διαφορετικά θα επιστρέφει 0.

α) (Βαθμοί 1.6). Σε ένα πρόγραμμα έχετε δηλώσει ένα πίνακα ακεραίων 3×5 , τον `pass`, ο οποίος σχηματικά παρουσιάζεται παρακάτω:

1	3	5	2	-6
9	17	12	13	18
11	-4	15	12	7

Αν η αποθήκευση του πίνακα έχει αρχίσει στην διεύθυνση μνήμης 5000, ποια είναι η τιμή καθενός από τα παρακάτω (κάθε ακέραιος χρειάζεται χώρο 4 byte):

α. `*pass`

β. `*pass + 5`

γ. `*pass + 2`

δ. `pass[2] + 3`

Θέμα 2 (Βαθμοί 2.8):

Υπενθύμιση: σε μια κυκλική λίστα οι κόμβοι δημιουργούν κύκλο, δεν υπάρχει πρώτος και τελευταίος.

Διαθέτετε μια κυκλική απλά συνδεδεμένη λίστα, κάθε κόμβος της οποίας περιγράφεται από μια δομή του είδους `node` και περιέχει ως «χρήσιμα δεδομένα» ένα ακέραιο. Να δώσετε την περιγραφή των κόμβων της δομής.

Γνωρίζετε ένα δείκτη, τον `ptr`, ο οποίος δείχνει σε κάποιο κόμβο της λίστας. Να γράψετε μια συνάρτηση, την `cyclic`, η οποία θα κάνει τα εξής:

- Θα βρίσκει τον κόμβο, ο οποίος περιέχει τον μέγιστο ακέραιο της λίστας.
- Θα τοποθετεί σε αυτό τον κόμβο ένα δείκτη, έστω τον `temp`.
- Θα «κόβει» την λίστα, κάνοντας κεφαλή της λίστας τον κόμβο στον οποίο δείχνει ο `temp`. Η λίστα δηλαδή δεν θα είναι πια κυκλική, αλλά θα έχει πρώτο κόμβο αυτόν στον οποίο δείχνει ο `temp` και τελευταίο τον αμέσως προηγούμενό του.
- Θα επιστρέφει στη `main()` τον δείκτη στην κεφαλή της λίστας (τον `temp`).

Θέμα 3 (Βαθμοί 3.8):

α) **(Βαθμοί 2.1).** Οι κόμβοι ενός δυαδικού δέντρου περιγράφονται από δομές του είδους *tree*. Τα «χρήσιμα δεδομένα» κάθε κόμβου είναι ένας πίνακας χαρακτήρων *N* θέσεων, ο *pin*, στον οποίο έχει τοποθετηθεί μια συμβολοσειρά.

Η *main()* γνωρίζει ένα δείκτη, τον *ptr*, σε κάποιο κόμβο του δέντρου.

Να γράψετε μια συνάρτηση, την *mostleft()*, η οποία θα κάνει τα εξής:

- Εάν ο κόμβος στον οποίο δείχνει ο *ptr* έχει δύο παιδιά, η συνάρτηση να εναλλάσσει μεταξύ τους τα περιεχόμενα των δύο αυτών παιδιών.
- Η συνάρτηση να επιστρέφει στη *main()* ένα δείκτη στον πρώτο χαρακτήρα της συμβολοσειράς *pin* του πιο αριστερού παιδιού του κόμβου στον οποίο δείχνει ο *ptr* (εννοούμε το πιο αριστερό παιδί που υπάρχει για τον κόμβο *ptr*).

β) **(Βαθμοί 0.7).** Να σχεδιάσετε ένα δυαδικό δέντρο αναζήτησης, στο οποίο οι τιμές των κόμβων του σας δίνονται με την εξής σειρά:

15 25 24 22 23 21 10 8 39 11 6

Είναι το δέντρο που προέκυψε AVL; Ναι ή όχι και γιατί;

γ) **(Βαθμοί 0.5).** Να διατρέξετε το δέντρο που δημιουργήσατε με μεταδιατεταγμένο τρόπο.

δ) **(Βαθμοί 0.5).** Να σχεδιάσετε ένα δυαδικό σωρό μεγίστων, ο οποίος να περιέχει τις τιμές της παραπάνω ερώτησης (β) (δεν ζητείται να δημιουργήσετε τον σωρό βήμα προς βήμα, απλώς να τον σχεδιάσετε «με το μάτι»).