



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
<http://courses.ece.tuc.gr>

ΗΜΜΥ ΠΛΗ 102

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

1^η - ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ: 24.03.2015

ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ: ΘΑ ΟΡΙΣΘΕΙ.

ΥΠΕΥΘΥΝΟΙ ΕΡΓΑΣΤΗΡΙΟΥ: Ανέστης Γιώργος, Μαραγκουδάκης Γιάννης, Αράπη Ξένια

(ganest, imarag, xenia)@ced.tuc.gr

Δυναμική διαχείριση μνήμης & Αναδρομή

Η παρακάτω άσκηση αντιστοιχεί στο 25% του συνολικού βαθμού του εργαστηρίου για το μάθημα ΗΜΜΥ ΠΛΗ 102 και εκπονείται σε ομάδες του ενός ατόμου.

Η αποστολή της άσκησης πρέπει να γίνει σύμφωνα με τις οδηγίες που δίνονται παρακάτω μέχρι τις 13:00 της **24ης Μαρτίου 2015**.

Η υποβολή του προγράμματος αυτόματα δηλώνει ότι είστε οι μοναδικοί συγγραφείς της λύσης της άσκησης. Εάν το πρόγραμμα ή μέρος αυτού έχει αντιγραφεί, θα πρέπει να αναφέρεται η πηγή του και ο συγγραφέας του.

Όλα τα παρακάτω θέματα είναι στη γλώσσα προγραμματισμού C. Σε όλες τις αναδρομικές συναρτήσεις (δεύτερο κομμάτι κάθε θέματος) απαγορεύεται η χρήση οποιουδήποτε βρόχου επανάληψης (for, while...) και η δήλωση καθολικών (global) και στατικών (static) μεταβλητών. Επιτρέπεται όμως η χρήση τοπικών (μη στατικών) μεταβλητών εντός της συνάρτησης.

Σε κάθε θέμα το πρόγραμμα να λειτουργεί επαναληπτικά, δηλαδή ο χρήστης να μπορεί να δώσει πολλές φορές διαφορετική είσοδο. Μπορείτε να χρησιμοποιήσετε βρόχο επανάληψης ΜΟΝΟ για το μενού χρήστη (πρώτο κομμάτι κάθε θέματος, όπου περιγράφεται η είσοδος των στοιχείων), ώστε να μπορεί να καλείται αυτό επαναληπτικά.

Βαθμολογούνται μόνο τα 2 πρώτα θέματα – το 3^ο θέμα είναι για την καλύτερη προετοιμασία σας. Σημειώνεται ότι πρέπει να κάνετε προσπάθεια να ελαχιστοποιήσετε τον αριθμό παραμέτρων των αναδρομικών συναρτήσεων. Επίσης, το 2^ο ερώτημα πρέπει να λυθεί χωρίς να περάσετε ως παράμετρο τον αριθμό στοιχείων της λίστας (σε αντίθετη περίπτωση θα υπάρχει ποινή στο βαθμό).

Στα προγράμματά σας η δέσμευση μνήμης θα πρέπει να είναι δυναμική (κάνοντας χρήση της malloc). Επίσης, θα πρέπει να φροντίσετε για την κατάλληλη αποδέσμευση της μνήμης που έχετε δεσμεύσει.

1^ο Βαθμολογούμενο Ερώτημα

Γράψτε ένα πρόγραμμα σε C το οποίο:

- 1) Να ζητεί από το χρήστη να δώσει τα στοιχεία ενός πίνακα ακεραίων. Ο χρήστης να μπορεί να δώσει οποιοδήποτε αριθμό ακεραίων επιθυμεί (μπορεί να περιέχεται ακόμα και ένας μόνο ακεραίος στον πίνακα).
- 2) Στη συνέχεια, γράψτε μία **ΑΝΑΔΡΟΜΙΚΗ** συνάρτηση που να ελέγχει εάν υπάρχει σε έναν πίνακα ακεραίων κατοπτρικό ζεύγος με άθροισμα ίσο με το πρώτο στοιχείο του πίνακα. Ένα κατοπτρικό ζεύγος αποτελείται από δύο αριθμούς οι οποίοι βρίσκονται σε θέσεις συμμετρικές ως προς το "μέσον" του πίνακα (π.χ. κατοπτρικά ζεύγη αποτελούν το πρώτο με το τελευταίο στοιχείο, το δεύτερο με το προτελευταίο στοιχείο, κτλ). Αν ο αριθμός των στοιχείων του πίνακα είναι περιττός, θεωρήστε ότι κατοπτρικό στοιχείο του μεσαίου στοιχείου είναι ο εαυτός του. Για παράδειγμα, στον πίνακα με τους ακεραίους 25 4 6 15 19 46 36, το κατοπτρικό ζεύγος «6 19» έχει άθροισμα ίσο με το πρώτο

στοιχείο (25) του πίνακα. Η συνάρτηση να επιστρέφει τη θέση του πρώτου στοιχείου του κατοπτρικού ζεύγους που ικανοποιεί τη συνθήκη. Σε περίπτωση αποτυχίας, η συνάρτηση να επιστρέφει -1.

- 3) Το πρόγραμμα να τυπώνει ΜΕΣΑ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ένα μήνυμα με τα 2 στοιχεία του κατοπτρικού ζεύγους που ικανοποιούν τη συνθήκη ή μήνυμα αποτυχίας αλλιώς. Επίσης, το πρόγραμμα να τυπώνει στη main την τιμή επιστροφής της συνάρτησης.

2° Βαθμολογούμενο Ερώτημα

Γράψτε ένα πρόγραμμα σε C, το οποίο:

- 1) Να ζητάει από το χρήστη να δώσει τα στοιχεία μίας λίστας (όχι ταξινομημένης) ακεραίων. Ο χρήστης να μπορεί να δώσει οποιοδήποτε αριθμό ακεραίων επιθυμεί (ακόμα και 0).
- 2) Στη συνέχεια, γράψτε μία **ΑΝΑΔΡΟΜΙΚΗ** συνάρτηση που να ελέγχει εάν σε μία τέτοια λίστα τα δεδομένα κάθε κόμβου (πέραν των 2 τελευταίων) είναι μικρότερο από το γινόμενο των δεδομένων των 2 επόμενων κόμβων στη λίστα. Για παράδειγμα, η λίστα με τους ακεραίους 5 4 2 3 25 15 6 ικανοποιεί αυτή τη συνθήκη. Η συνάρτηση να επιστρέφει NULL σε περίπτωση επιτυχίας. Σε περίπτωση λίστας με 2 ή λιγότερα στοιχεία μπορείτε να θεωρήσετε ότι ικανοποιείται το κριτήριο. Σε περίπτωση αποτυχίας να επιστρέφεται ένας δείκτης στον πρώτο κόμβο της τριάδας διαδοχικών αριθμών που δεν ικανοποιούν το κριτήριο.
- 3) Το πρόγραμμα να τυπώνει ΜΕΣΑ ΣΤΗ ΣΥΝΑΡΤΗΣΗ **ένα μήνυμα** για το αν η λίστα ικανοποιεί το ζητούμενο κριτήριο. Σε περίπτωση αποτυχίας να τυπώνεται κατάλληλο μήνυμα για την τριάδα διαδοχικών αριθμών που δεν ικανοποιούν το κριτήριο. Επίσης, το πρόγραμμα να τυπώνει στη main πληροφορία σχετικά με την τιμή επιστροφής της συνάρτησης.

Το ερώτημα αυτό πρέπει να λυθεί χωρίς να περάσετε ως παράμετρο τον αριθμό στοιχείων της λίστας (σε αντίθετη περίπτωση θα υπάρχει ποινή στο βαθμό).

3° Ερώτημα (Προετοιμασίας, Δε Βαθμολογείται)

Το παρακάτω θέμα παρέχεται για την καλύτερη προετοιμασία σας για την τελική εξέταση, ή την online εξέταση (για όσους την επιλέξουν). Το πρόγραμμα αυτό μπορεί να υποβληθεί αλλά δε βαθμολογείται.

Γράψτε ένα πρόγραμμα σε C, το οποίο:

- 1) Να ζητάει από το χρήστη να δώσει τα στοιχεία μίας λίστας (μη ταξινομημένης) ακεραίων. Ο χρήστης να μπορεί να δώσει οποιοδήποτε αριθμό ακεραίων επιθυμεί (ακόμα και 0).
- 2) Στη συνέχεια γράψτε μία **ΑΝΑΔΡΟΜΙΚΗ** συνάρτηση που να ελέγχει εάν σε μία λίστα ακεραίων υπάρχουν 3 διαδοχικοί κόμβοι με ίσα δεδομένα. Για παράδειγμα, η λίστα με τους ακεραίους 5 4 4 4 8 10 20 20 20 21 περιέχει 2 τέτοιες τριάδες ίσων δεδομένων. Η συνάρτηση να επιστρέφει NULL σε περίπτωση αποτυχίας εύρεσης κατάλληλης τριάδας κόμβων. Σε περίπτωση επιτυχίας να επιστρέφεται ένας δείκτης στον πρώτο κόμβο της πρώτης τριάδας διαδοχικών αριθμών που ικανοποιούν το κριτήριο.
- 3) Το πρόγραμμα να τυπώνει ΜΕΣΑ ΣΤΗ ΣΥΝΑΡΤΗΣΗ **ένα μήνυμα** για το αν η λίστα ικανοποιεί το ζητούμενο κριτήριο. Σε περίπτωση επιτυχίας να τυπώνεται κατάλληλο μήνυμα για την πρώτη τριάδα διαδοχικών αριθμών που ικανοποιούν το κριτήριο. Επίσης, το πρόγραμμα να τυπώνει στη main πληροφορία σχετικά με την τιμή επιστροφής της συνάρτησης.
- 4) Γράψτε και το αντίστοιχο πρόγραμμα αν θέλουμε να τυπώνεται πληροφορία για την τελευταία τριάδα διαδοχικών κόμβων με ίσα δεδομένα. Αντίστοιχα, να επιστρέφεται ένας δείκτης στον πρώτο κόμβο της τελευταίας τριάδας που ικανοποιεί τη συνθήκη (ή NULL σε περίπτωση αποτυχίας.)

1. Τι πρέπει να παραδώσετε

Τα τρία αρχεία με τον πηγαίο κώδικά σας με όνομα της μορφής «**x_set1_exer_Y.c**», όπου **Y** βάλτε 1,2,3 για κάθε θέμα αντίστοιχα. Υπενθυμίζεται ότι το θέμα 3 είναι προαιρετικό και δε βαθμολογείται. Προσοχή στη στοίχιση του κώδικα και τα σχόλια, τα οποία θα πρέπει να είναι επεξηγηματικά του κώδικα. Στις πρώτες γραμμές του πηγαίου κώδικα να αναφέρετε το όνομά σας και τον αριθμό του φοιτητικού μητρώου σας.

2. Οδηγίες για την Ηλεκτρονική Υποβολή της Άσκησης

(Βήμα 1): Δημιουργήστε ένα συμπιεσμένο αρχείο (.zip) με όνομα της μορφής x_set1.zip, όπου x θα το αντικαταστήσετε με τον κωδικό της ομάδας σας (π.χ. LAB111030_set1.zip). Στο αρχείο αυτό συμπεριλάβετε τα αρχεία που περιγράφονται στην παράγραφο 1 («Τι πρέπει να παραδώσετε»). Μη χρησιμοποιήσετε άλλο πρόγραμμα συμπίεσης (rar, 7zip, κλπ.) πέραν του WinZip και φροντίστε τα αρχεία να βρίσκονται στον ίδιο φάκελο (directory).

Π.χ., αν είστε η ομάδα LAB111030 θα δημιουργήσετε ένα zip αρχείο με όνομα LAB111030_set1.zip και περιεχόμενο τα παρακάτω αρχεία:

LAB111030_set1_exer_1.c
LAB111030_set1_exer_2.c
LAB111030_set1_exer_3.c
(προαιρετικό)

(Βήμα 2):

Αφού βεβαιωθείτε ότι έχετε εισάγει σωστά τον κωδικό της ομάδας σας, υποβάλετε το συμπιεσμένο αρχείο που δημιουργήσατε μέσω της σελίδας αποστολής εργασιών του Ιστοχώρου Μαθημάτων (<http://courses.ece.tuc.gr>).

ΣΗΜΕΙΩΣΗ-1: Μπορείτε να υποβάλλετε την εργασία σας όσες φορές επιθυμείτε. Κάθε φορά κρατείται από το σύστημα η τελευταία εργασία που υποβάλλατε. Αν εργαζόσαστε στην άσκηση την τελευταία μέρα, σιγουρευτείτε ότι εντός της προθεσμίας έχετε υποβάλει κάποια εργασία για να είστε ασφαλείς.

ΣΗΜΕΙΩΣΗ-2: Εργασίες που αποστέλλονται μέσω email δεν θα γίνονται αποδεκτές !