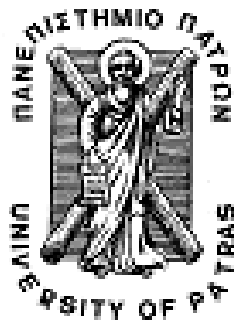


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
3^ο ΣΕΤ ΑΣΚΗΣΕΩΝ

Οι ασκήσεις αυτού του φυλλαδίου καλύπτουν τα παρακάτω θέματα και δίνονται ενδεικτικά οι αντίστοιχες ενότητες στο βιβλίο The GNU C Programming Tutorial που μπορείτε συμβουλευτείτε (<http://crasseux.com/books/ctutorial/>) .

- Συναρτήσεις (κεφάλαιο Functions)
- Πίνακες (κεφάλαιο Arrays)
- Δείκτες

Το τρίτο σύνολο ασκήσεων θα εξετασθεί μαζί με το τέταρτο την εβδομάδα 07-13/01/13 (ο κάθε φοιτητής θα εξεταστεί την ώρα του τμήματος του, για κάθε υποερώτημα ο φοιτητής θα πρέπει να έχει το αντίστοιχο αρχείο κώδικα C αποθηκευμένο στο λογαριασμό του στο diogenis.ceid.upatras.gr το οποίο θα πρέπει να έχει μεταγλωττιστεί χωρίς σφάλματα, ενώ για το σύνολο των ερωτημάτων θα πρέπει να έχει φέρει και μια εκτυπωμένη αναφορά με τις απαντήσεις του).

Άσκηση 1η

Γράψτε πρόγραμμα το οποίο να δηλώνει ένα μονοδιάστατο πίνακα μεγέθους N όπου θα τον γεμίζει με στοιχεία και στη συνέχεια θα καλεί μία συνάρτηση που θα αντιστρέφει τα στοιχεία του πίνακα και θα εκτυπώνει τον αρχικό και τον ανεστραμμένο πίνακα. Το μέγεθος του πίνακα N θα ορίζεται μέσω της οδηγίας #define του προεπεξεργαστή.

Παράδειγμα:

```
Original array: 0 1 2 3 4 5 ... 19
Inverted array: 19 18 17 16 ... 0
```

Άσκηση 2η

Γράψτε πρόγραμμα που θα κατασκευάζει δυναμικά έναν πίνακα θετικών ακεραίων. Το πρόγραμμα θα περιμένει έναν ακέραιο αριθμό, μέχρις ότου δοθεί αρνητικός αριθμός.

Κατόπιν θα εκτυπώνει τα στοιχεία που εισήχθηκαν, τη διεύθυνση στη μνήμη όπου βρίσκεται ο πίνακας, τη διεύθυνση στη μνήμη όπου βρίσκεται το πρώτο στοιχείο του πίνακα, τη διεύθυνση στη μνήμη όπου βρίσκεται το τελευταίο στοιχείο του πίνακα και το μέγεθος του πίνακα σε bytes.

Στη συνέχεια το πρόγραμμα θα ταξινομεί τα στοιχεία με χρήση της qsort() (<http://www.cplusplus.com/reference/cstdlib/qsort/>), και τέλος θα επαναεκτυπώνει τις πληροφορίες για το μέγεθος του πίνακα και τις διευθύνσεις πίνακα, πρώτου και τελευταίου στοιχείου.

Παράδειγμα εισόδου για πίνακα 5 στοιχείων :

```

Enter a positive integer value (give negative to end): 5
Enter a positive integer value (give negative to end): 4
Enter a positive integer value (give negative to end): 10
Enter a positive integer value (give negative to end): 2
Enter a positive integer value (give negative to end): 1
Enter a positive integer value (give negative to end): -1
Numbers entered:
numbers[0] = 5
numbers[1] = 4
numbers[2] = 10
numbers[3] = 2
numbers[4] = 1
Length Of Array= 5
Array address [0x103900840]      First element address
[0x103900840]      Last element address [0x103900850]
Array holds 20 bytes
Sorted Numbers
numbers[0] = 1
numbers[1] = 2
numbers[2] = 4
numbers[3] = 5
numbers[4] = 10
Length Of Array= 5
Array address [0x103900840]      First element address
[0x103900840]      Last element address [0x103900850]
Array holds 20 bytes

```

Άσκηση 3η

Σχεδιάστε αλγόριθμο ο οποίος διαβάζει ένα σύνολο από αλφαριθμητικά και: (i) καταγράφει αυτό με το μικρότερο μήκος, (ii) τα διατάσσει, με βάση την λεξικογραφική διάσταση, χρησιμοποιώντας τον αλγόριθμο διάταξης που αναφέρεται στην άσκηση 2.

Άσκηση 4η

Ζητείται η ανάπτυξη προγράμματος που στο κύριο σώμα ορίζεται και αρχικοποιείται ένας πίνακας 10 ακεραίων στοιχείων που αντιστοιχούν στους μισθούς 10 υπαλλήλων, ενώ δύο συναρτήσεις (στις οποίες θα περνιέται ο πίνακας σαν όρισμα) θα επιστρέφουν η μία το μεγαλύτερο μισθό και η δεύτερη το μέσο όρο της μισθοδοσίας.

Άσκηση 5η

Δημιουργείστε δισδιάστατο πίνακα με τιμές που δίνονται από το χρήστη τόσο για τις διαστάσεις όσο για το περιεχόμενο των κελιών και στη συνέχεια υπολογίστε το άθροισμα της διαγωνίου του πίνακα. Ο υπολογισμός του αθροίσματος πρέπει να γίνεται με κλήση συνάρτησης την οποία εσείς θα ορίσετε.

Άσκηση 6η

Να κατασκευάσετε πρόγραμμα το οποίο να κάνει τη μετατροπή δεκαδικών ψηφίων (0...9) στον κώδικα Μορς και αντίστροφα.

Συγκεκριμένα ο χρήστης θα επιλέγει τον τύπο της μετατροπής που θέλει να κάνει και κατόπιν θα εισάγει είτε μία σειρά δεκαδικών ψηφίων είτε μία σειρά τελείες / παύλες και το πρόγραμμα θα εκτυπώνει τον ισοδύναμο κώδικα Μορς ή τα ισοδύναμα ψηφία αντίστοιχα.

Παράδειγμα εκτέλεσης του προγράμματος (με **έντονα** η είσοδος του χρήστη):

```
1 - Ari8mos se kwdika Morse
2 - Kwdikas Morse se ari8mo
0 - Eksodos
1
Dwse ari8mo : 2009
0 isodynamos kwdikas Morse einai : ..--- -----
1 - Ari8mos se kwdika Morse
2 - Kwdikas Morse se ari8mo
0 - Eksodos
2
Dwse kwdika Morse: .-----.---.---..
0 isodynamos ari8mos einai : 1978
1 - Ari8mos se kwdika Morse
2 - Kwdikas Morse se ari8mo
0 - Eksodos
0
```

Υπόδειξη: Εσωτερικά στο πρόγραμμά σας μπορείτε να αποθηκεύετε σε ένα δυσδιάστατο πίνακα την αναπαράσταση των ψηφίων σε κώδικα Μορς (http://en.wikipedia.org/wiki/Morse_code).

Άσκηση 7η

Να υλοποιήσετε 2 συναρτήσεις οι οποίες θα δέχονται ως είσοδο τρεις μονοδιάστατους πίνακες ακεραίων A, B, C και θα υπολογίζουν η μία το άθροισμα (A+B) και η άλλη τη διαφορά (A-B) και θα αποθηκεύουν το αποτέλεσμα στον πίνακα C.

Οι συναρτήσεις θα καλούνται από κυρίως πρόγραμμα όπου θα δηλώνονται οι A, B, C μεγέθους n (όπου n σταθερά), θα δίνονται από τον χρήστη οι τιμές για τους A και B και μετά την κλήση κάθε συνάρτησης θα εκτυπώνεται ο C.

Δώστε δύο υλοποιήσεις μία με πίνακες και μία με δείκτες και δυναμική δέσμευση μνήμης.

Άσκηση 8η

Ο μέγιστος κοινός διαιρέτης δύο αριθμών x, y μπορεί να υπολογιστεί σύμφωνα με τον αλγόριθμο του Ευκλείδη ως εξής: έστω $x > y$ (αν $x < y$ εναλλάσσουμε τους δύο αριθμούς), αν το y είναι 0 τότε το x είναι ο μέγιστος κοινός διαιρέτης, διαφορετικά βάλε στη θέση του x το y , αντικατέστησε το y με το υπόλοιπο της διαίρεσής του x με το y , και επανέλαβε τον έλεγχο.

Υλοποιείτε συνάρτηση που υπολογίζει το μέγιστο κοινό διαιρέτη δύο αριθμών με βάση τον συγκεκριμένο αλγόριθμο, και δώστε μία επαναληπτική και μία αναδρομική υλοποίηση.

Άσκηση 9η

Να υλοποιηθεί πρόγραμμα σε C το οποίο παράγει N τυχαίους αριθμούς με τιμές από το 0 μέχρι το 9 και στην συνέχεια υπολογίζει το ιστόγραμμα των παραπάνω αριθμών, δηλαδή πόσες φορές εμφανίζεται ο κάθε ένας από αυτούς. Για τον σκοπό αυτό να υλοποιηθούν οι ακόλουθες συναρτήσεις:

1. `void random_vector(int array[], int N)`: παράγει N τυχαίους αριθμούς από 0 ως 9 και τους αποθηκεύει στον πίνακα `array`.
2. `void calculate_hist(int array[], int N, int hist[10])`: αποθηκεύει στον πίνακα `hist`, το πλήθος των αριθμών που βρίσκονται στο διάνυσμα `array`, δηλαδή το `hist[j]` αποθηκεύει το πλήθος εμφανίσεων του αριθμού j στο πίνακα `array`.
3. `void print_hist(int hist[10])`: τυπώνει στην οθόνη το ιστόγραμμα με γραφικό τρόπο (ακολουθεί παράδειγμα)

Παράδειγμα εκτέλεσης:

```
Daste to N: 100
0: ***** (10)
1: ***** (20)
2: ***** (10)
3: ***** (7)
4: ***** (7)
5: ***** (8)
6: ***** (8)
7: ***** (10)
8: ***** (12)
9: ***** (8)
```

(στις παρενθέσεις εμφανίζεται το πλήθος εμφάνισης του αντίστοιχου αριθμού στον πίνακα `array`)

Άσκηση 10η

Ο αλγόριθμος συμπίεσης RLE (Run Length Encoding) δέχεται μία συμβολοσειρά εισόδου και τη συμπιέζει αντικαθιστώντας στη συμβολοσειρά συνεχόμενες

επαναλήψεις του ίδιου συμβόλου με πλήθος εμφανίσεων πάνω από 1, με το πλήθος εμφανίσεων και το σύμβολο, για παράδειγμα, η συμβολοσειρά `aaaxxxxttfejjjj` συμπιέζεται από την `3a3x3tef4j`.

Υλοποιείστε συνάρτηση που δέχεται σαν όρισμα συμβολοσειρά και την αντικαθιστά με τη συμπιεσμένη αναπαράσταση της. Προσέξτε ότι όταν ένα σύμβολο εμφανίζεται μόνο του, παραμένει ως έχει στη συμπιεσμένη αναπαράσταση.

Άσκηση 11η

Έστω η αναδρομική σχέση $a_n = (3a_{n-1}^2 - 1) / a_{n-1}$, με $a_0 = 1$, που ορίζει μία ακολουθία αριθμών.

(α) Υλοποιείστε αναδρομική συνάρτηση που παίρνει σαν όρισμα μια ακέραια τιμή n και επιστρέφει την τιμή a_n .

(β) Έστω $S(n) = a_0 + a_1 + \dots + a_n$. Υλοποιείστε συνάρτηση που να παίρνει σαν όρισμα μια ακέραια τιμή n και επιστρέφει την τιμή $S(n)$. Δώστε μια αναδρομική και μία επαναληπτική υλοποίηση.