Διεθνές Πανεπιστήμιο της Ελλάδος
Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων
Μάθημα : Δομημένος Προγραμματισμός
Εξάμηνο : Α' – Χειμερινό 2020-2021
Διδάσκων : Γουλιάνας Κώστας

18 Ιανουαρίου 2021

ΘΕΜΑΤΑ Α' ΕΞΕΤΑΣΤΙΚΗΣ

ΘΕΜΑ 1 (3.7)

Υποτίθεται ότι έχετε γράψει τον κώδικα για τη συνάρτηση `showP(<παράμετροι>)`, η οποία θα **εμφανίζει** τα στοιχεία ενός πίνακα ακεραίων `p`.

Να γράψετε **ΣΕ ΜΙΑ ΞΕΧΩΡΙΣΤΗ ΣΕΛΙΔΑ** τη **δήλωση** για τη **συνάρτηση** `keepOnlyOne(<παράμετροι>)` και **ΝΑ ΣΧΕΔΙΑΣΕΤΕ** το αντίστοιχο **Διάγραμμα Ροής (ΟΧΙ ΨΕΥΔΟΚΩΔΙΚΑ – ΑΛΓΟΡΙΘΜΟ ΟΥΤΕ ΚΩΔΙΚΑ)**.

Η συνάρτηση `keepOnlyOne(<παράμετροι>)` θα **ελέγχει** όλα τα στοιχεία ενός πίνακα ακεραίων `n` θέσεων με ακέραιες τιμές που υποτίθεται ότι έδωσε ο χρήστης, τυχαίες, απ' το πληκτρολόγιο ή με δυναμική αρχικοποίηση και θα κρατάει **στις πρώτες θέσεις του πίνακα μόνο την πρώτη εμφάνιση** του κάθε αριθμού, ώστε οι τιμές του να είναι **μοναδικές**. Οι **περισσότερες από μία εμφανίσεις κάθε τιμής θα πηγαίνουν στις τελευταίες θέσεις** του πίνακα. Αν για παράδειγμα η πρώτη τιμή του πίνακα είναι 5, θα ελέγχει όλες τις υπόλοιπες τιμές του πίνακα και αν βρει και άλλο 5 σε κάποια θέση, θα μετακινεί όλα τα υπόλοιπα στοιχεία του πίνακα μια θέση μπροστά, ώστε να μπει το δεύτερο 5 στο τέλος. Αν βρει και άλλο 5 σε κάποια θέση, θα μετακινεί όλα τα υπόλοιπα στοιχεία του πίνακα μια θέση μπροστά, ώστε να μπει το τρίτο 5 στην προτελευταία θέση και συνεχίζει μ' αυτό τον τρόπο **μέχρι να φτάσει στον έλεγχο του τελευταίου διαφορετικού αριθμού του πίνακα**. Σε κάθε βήμα με την κλήση της μεθόδου `showP()` θα εμφανίζει και την νέα μορφή που θα παίρνει ο πίνακας. Μετά το πέρας της διαδικασίας, θα γεμίζει όλες τις θέσεις του πίνακα με τα στοιχεία που εμφανίστηκαν περισσότερες από μια φορά με την τιμή -999. **Η υλοποίηση θα γίνει με τις εντολές Για (for) και Όσο (while) και χωρίς τη χρήση της εντολής break**.

Στη συνάρτηση `main()` **:**

- Να **διαβάσετε** τον αριθμό θέσεων του πίνακα (θα πρέπει `n >= 10`) - ψευδοκώδικας
- Να **δηλώσετε** έναν πίνακα `p` ακεραίων `n` θέσεων και να τον γεμίσετε με δυναμική αρχικοποίηση- κώδικας.
- Να **καλέσετε** τη συνάρτηση `keepOnlyOne(<παράμετροι>)`, για να **κρατήσετε** στις πρώτες θέσεις του πίνακα την πρώτη εμφάνιση του κάθε διαφορετικού στοιχείου και στις τελευταίες τις παραπάνω από μια εμφανίσεις του κάθε στοιχείου του πίνακα `p` - κώδικας.

Ενδεικτική έξοδος του προγράμματος :

```
Give an integer n >= 10 : 3
Give an integer n >= 10 : 12

p =  5 2 5 5 4 2 2 3 2 3 3 1
p =  5 2 5 4 2 2 3 2 3 3 1 5
```

```
p =  5 2 4 2 2 3 2 3 3 1 5 5
p =  5 2 4 2 3 2 3 3 1 2 5 5
p =  5 2 4 3 2 3 3 1 2 2 5 5
p =  5 2 4 3 3 3 1 2 2 2 5 5
p =  5 2 4 3 3 1 3 2 2 2 5 5
p =  5 2 4 3 1 3 3 2 2 2 5 5
p =  5 2 4 3 1 -999 -999 -999 -999 -999 -999 -999
Press any key to continue . . .
```

ΘΕΜΑ 2 (1.4)

Να γράψετε την εντολή για τη δήλωση της συνάρτησης `<τύπος> findHMS(<παράμετροι>)` και τον **κώδικα** της συνάρτησης, η οποία δεν θα επιστρέφει κάποια τιμή, αλλά με τη χρήση **δεικτών** (**pointers**), όταν δίνεται η διάρκεια σε δευτερόλεπτα, θα **βρίσκει** τις **ώρες** τα **λεπτά** και τα **δευτερόλεπτα** που αντιστοιχούν σε αυτή τη διάρκεια. Αν π.χ. η διάρκεια είναι **8000 δευτερόλεπτα** η συνάρτηση θα υπολογίζει στους δείκτες τις τιμές **2, 13 και 20**, γιατί τα 8000 **δευτερόλεπτα** είναι 2 **ώρες** , 13 **λεπτά** και 20 **δευτερόλεπτα**. Να γράψετε και τις απαραίτητες εντολές στη `main()`, με τις οποίες θα διαβάζετε μια ακέραια τιμή στη μεταβλητή `diarkeia`, θα καλείτε τη συνάρτηση `findHMS(<παράμετροι>)` και μετά θα εμφανίζετε τις **ώρες** τα **λεπτά** και τα **δευτερόλεπτα** που αντιστοιχούν σε αυτή τη διάρκεια.
+0.5, αν ξαναγράψετε τη main() με τέτοιον τρόπο, ώστε για τους 3 δείκτες να χρησιμοποιηθεί η μνήμη σωρού (heap memory) και η εντολή malloc(), αντί της μνήμης στοίβας (stack memory).

ΘΕΜΑ 3 (4.9)

Να γράψετε μια δομή `struct playList`, όπου θα πρέπει να χρησιμοποιήσετε την `typedef` ώστε για λόγους συντομίας, αντί του `struct playList` θα χρησιμοποιείτε το `PL`. Η δομή `playList` θα περιέχει τα πεδία `titlos (string)`, `mnimi (σε Megabytes)`, `hours`, `minutes` και `seconds` (θετικοί ακέραιοι για τα 3 τελευταία πεδία) .

- Να γράψετε την συνάρτηση `fillstruct(<παράμετροι>)` με την οποία θα γεμίζετε μια απλή μεταβλητή (όχι στοιχείο πίνακα) τύπου `struct playList` με τιμές, τις οποίες θα περνάτε παραμετρικά - κώδικας.
- Να γράψετε την συνάρτηση `showstruct(<παράμετροι>)` με την οποία θα εμφανίζετε τις τιμές των πεδίων μιας απλής μεταβλητής (όχι στοιχείο πίνακα) τύπου `struct playList` - κώδικας.
- Να γραφεί **Αλγόριθμος** ή **Κώδικας** μιας συνάρτησης `fillUSB(<παράμετροι>)`, η οποία με τη χρήση της εντολής **while** ή **do…while** (**ΟΧΙ for και break**) θα **γεμίζει** ένα `usb` κάποιας χωρητικότητας σε `Gigabytes` με όσα στοιχεία ενός πίνακα τύπου `playList` μπορούν να χωρέσουν σε αυτό (ανάλογα με τη μνήμη που καταλαμβάνουν)

και θα επιστρέφει τη συνολική μνήμη σε Gigabytes (1 Gigabyte = 1024 Megabytes).

- Να γραφεί **Αλγόριθμος** ή **Κώδικας** μιας συνάρτησης totalHMS(<παράμετροι>), η οποία θα **αθροίζει** τις ώρες, τα λεπτά και τα δευτερόλεπτα όλων των στοιχείων ενός πίνακα τύπου struct playList και θα εμφανίζει στο τέλος το σύνολο του καθενός. Προσοχή, αν για n = 2 η διάρκεια των 2 λιστών αναπαραγωγής είναι 1 ώρα, 23 λεπτά, 30 δευτερόλεπτα και 1 ώρα, 37 λεπτά , 50 δευτερόλεπτα αντίστοιχα (σύνολο 2 ώρες, 60 λεπτά, 80 δευτερόλεπτα) θα εμφανίζει 3 ώρες , 1 λεπτό, 20 δευτερόλεπτα.

Στην συνάρτηση main() :

- Να **διαβάσετε μια ακέραια τιμή > 5 στην μεταβλητή n** - ψευδοκώδικας**.**
- **Να δηλώσετε** έναν πίνακα pl τύπου struct playList n θέσεων (-κώδικας).
- Να **γεμίσετε** τα πεδία των n στοιχείων του πίνακα pl διαβάζοντας σε τοπικές μεταβλητές της main() όσες τιμές χρειαστεί απ' το πληκτρολόγιο (- ψευδοκώδικας), καλώντας την συνάρτηση findHMS() (-κώδικας) να απομονώσετε από την διάρκεια που θα διαβάσετε απ' το πληκτρολόγιο τις ώρες, τα λεπτά και τα δευτερόλεπτα) και μετά θα αναθέτετε όλες τις παραπάνω τιμές στα αντίστοιχα πεδία του κάθε στοιχείου του πίνακα pl καλώντας την συνάρτηση fillstruct(<παράμετροι>) –κώδικας.
- Να **εμφανίσετε** τις τιμές **όλων** των πεδίων για τα n στοιχεία του πίνακα- κώδικας **.**
- Να διαβάσετε μια τιμή για τη χωρητικότητα ενός usb σε Gigabytes - ψευδοκώδικας **.**
- Να **καλέσετε** την συνάρτηση fillUSB(<παράμετροι>), για να **γεμίσετε** ένα usb αυτής της χωρητικότητας με όσα στοιχεία του πίνακα pl μπορούν να χωρέσουν σε αυτό και θα επιστρέφετε τη συνολική μνήμη σε Megabytes - κώδικας **.**
- Να **εμφανίσετε** τη συνολική μνήμη που καταλαμβάνουν τα στοιχεία σας στο usb - ψευδοκώδικας.
- Να **καλέσετε** την συνάρτηση totalHMS(<παράμετροι>), για να **αθροίσετε** τις ώρες, τα λεπτά και τα δευτερόλεπτα όλων των στοιχείων του πίνακα pl και θα **εμφανίζετε** το σύνολο του καθενός - κώδικας **.**

+1.0, αν ξαναγράψετε τη main() και τις συναρτήσεις με τέτοιον τρόπο, ώστε για όλες τις μεταβλητές να χρησιμοποιηθεί η μνήμη σωρού (heap memory) και η εντολή malloc(), αντί της μνήμης στοίβας (stack memory).

- Στη δήλωση (**υπογραφή**) και στην **κλήση** κάθε συνάρτησης θα πρέπει να γράψετε **κώδικα** C, όπως και στη **δήλωση** των **τοπικών μεταβλητών και πινάκων**, ενώ στις υπόλοιπες εντολές μπορείτε να γράψετε Αλγόριθμο, αλλά με τα σωστά ονόματα των **μεταβλητών, πεδίων και συναρτήσεων**.

Υποβάλλετε το αρχείο των απαντήσεων μαζί με το αρχείο με την ταυτότητά σας στην πλατφόρμα http://submit.iee.ihu.gr/, **Δομημένος Προγραμματισμός (Θεωρία)**. Μπαίνετε με τους κωδικούς Username : **10134** Password : **47639**

Διάρκεια Εξέτασης : 100 λεπτά
Καλή Επιτυχία