

ΠΡΟΣΟΧΗ: Η ΓΡΑΠΤΗ ΑΥΤΗ ΕΞΕΤΑΣΗ ΣΥΜΒΑΛΛΕΙ ΣΤΟ 85% ΤΗΣ ΤΕΛΙΚΗΣ ΒΑΘΜΟΛΟΓΙΑΣ

Οι απαντήσεις να αποθηκευτούν στο ECLASS: exams-dai.uom.gr στην αντίστοιχη εργασία.

Ως απάντηση να σταλεί το αρχείο του πηγαίου κώδικα (grades_analysis.c)

Να γράψετε πρόγραμμα (όνομα αρχείου **grades_analysis.c**) το οποίο κάνει μια επεξεργασία των βαθμών ενός/μιας φοιτητή/φοιτήτριας, όπως περιγράφεται παρακάτω. Τα αρχικά δεδομένα προς επεξεργασία βρίσκονται ήδη σε μορφή πίνακα στο δοθέν αρχείο **grades_analysis.c** τον εμπλουτισμό του οποίου καλείστε να συνεχίσετε. Συγκεκριμένα, στο δοθέν αρχείο συμπεριλαμβάνεται ένας διδιάστατος πίνακας **grades_table**, οι γραμμές του οποίου αντιστοιχούν σε 4 έτη φοίτησης και οι στήλες του στους 8 βαθμούς που ελήφθησαν σε εκείνο το έτος. Για παράδειγμα, αν το στοιχείο `grade_table[0][1]` είναι 6, σημαίνει ότι στο πρώτο έτος (γραμμή 0) και στο μάθημα με αύξοντα αριθμό 1 ο φοιτητής έλαβε βαθμό 6. Τα έτη φοίτησης είναι στο διάστημα 0 έως και 3, ενώ οι αύξοντες αριθμοί μαθημάτων είναι στο διάστημα 0 έως και 7.

(1) Να περιλάβετε στο πρόγραμμά σας:

Τη δήλωση ενός κατάλληλου τύπου δομής (struct) με όνομα **failedT**, που θα μπορεί να αναπαραστήσει τις ακόλουθες πληροφορίες για το/τη φοιτητή/φοιτήτρια: (α) έτος φοίτησης (ακέραιος αριθμός), (β) αύξων αριθμός μαθήματος έτους (ακέραιος αριθμός). Με βάση τη συγκεκριμένη δομή, να συμπεριληφθεί μονοδιάστατος πίνακας με όνομα **failed_table** 32 θέσεων τύπου **failedT**.

Βαθμοί: 10 από 100.

Το πρόγραμμα θα περιλαμβάνει όλες τις συναρτήσεις/διαδικασίες που περιγράφονται παρακάτω. Μπορείτε αν θέλετε να ορίσετε και επιπλέον συναρτήσεις.

(2) Τη δήλωση (πρωτότυπο) και την υλοποίηση της συνάρτησης `void analyze_table`, η οποία δέχεται τον πίνακα **grades_table** και τον πίνακα **failed_table** και τοποθετεί σε στοιχεία του πίνακα **failed_table** τους αριθμοδείκτες (έτος – αύξων αριθμός) των μαθημάτων που έχουν βαθμό ΚΑΤΩ από 5 (δεν περάστηκαν). Επίσης, η συνάρτηση θα “επιστρέφει” το ΠΛΗΘΟΣ των βαθμών που είναι ΚΑΤΩ από 5 ΚΑΙ το ΠΛΗΘΟΣ των βαθμών που είναι ΑΚΡΙΒΩΣ 10 του πίνακα **grades_table**. Για παράδειγμα, αν το πρώτο στοιχείο του πίνακα **grades_table** με βαθμό κάτω του 5 είναι το **grades_table[0][3]** (βαθμός 2), τότε στο πρώτο στοιχείο του πίνακα **failed_table** θα τοποθετηθεί στο αντίστοιχο πεδίο έτους το 0 και στο πεδίο του αύξοντα αριθμού το 3.

Βαθμοί: 20 από 100.

(3) Τη δήλωση του πρωτοτύπου και την υλοποίηση μιας συνάρτησης `void print_data`, η οποία θα τυπώνει στην οθόνη τα περιεχόμενα του πίνακα **failed_table**. Επιπροσθέτως, θα τυπώνει και το πλήθος των βαθμών που είναι ακριβώς 10 καθώς και το ποσοστό τοις εκατό (float) επί του συνόλου των βαθμών (32) στο οποίο αντιστοιχεί το πλήθος των βαθμών που είναι 10. Για παράδειγμα, αν από τους 32 βαθμούς οι 4 είναι ακριβώς 10, το ποσοστό είναι: **12.50%** (δείτε παράδειγμα εκτέλεσης παρακάτω).

Βαθμοί: 15 από 100.

(4) Τη δήλωση του πρωτοτύπου και την υλοποίηση μιας συνάρτησης `int read_data`, η οποία θα δέχεται ως ορίσματα ένα αλφαριθμητικό και δύο ακέραιες τιμές, οι οποίες αντιστοιχούν σε άνω και κάτω όρια και θα τυπώνει στην οθόνη το αλφαριθμητικό, ζητώντας από το χρήστη να εισαγάγει μια ακέραια τιμή εντός των παραπάνω ορίων, συμπεριλαμβανομένων αυτών. Οι παραπάνω εντολές θα επαναλαμβάνονται μέχρι ο χρήστης να δώσει ακέραια τιμή εντός των ορίων, οπότε και η συνάρτηση θα επιστρέφει την (σωστή) τιμή που εισήγαγε ο χρήστης. Ένα παράδειγμα κλήσης της συνάρτησης για την απόδοση μιας τιμής από 0 έως και 10 στην ακέραια μεταβλητή **n** θα μπορούσε να είναι το ακόλουθο:

```
n = read_data("Grade (0-10): ", 0, 10);
```

με αποτέλεσμα να τυπωθεί στην οθόνη το ακόλουθο προτρεπτικό, αναμένοντας εισαγωγή αριθμού:

```
Grade (0-10):
```

Βαθμοί: 20 από 100.

(5) Τη δήλωση του **πρωτοτύπου** και την **υλοποίηση** μιας συνάρτησης void **update_data**, η οποία θα ζητά από τον χρήστη, με τα αντίστοιχα προτροπικά μηνύματα (δείτε παράδειγμα εκτέλεσης παρακάτω), (α) το έτος (έλεγχος αν $0 \leq \text{έτος} \leq 3$), (β) τον αύξοντα αριθμό μαθήματος του έτους (έλεγχος αν $0 \leq \text{αριθμός μαθήματος} \leq 7$), (γ) ένα νέο βαθμό (έλεγχος αν $0 \leq \text{βαθμός} \leq 10$). Με την επιτυχή ολοκλήρωση της εισαγωγής των τριών ακεραίων τιμών, η διαδικασία **update_data** θα ενημερώνει την αντίστοιχη θέση του πίνακα **grades_table** (δηλαδή η συγκεκριμένη θέση του πίνακα θα έχει νέα τιμή).
Βαθμοί: 15 από 100.

(6) Συνάρτηση **main()**, από την οποία θα καλούνται οι διαδικασίες **analyze_table** και **print_data** και θα ερωτάται ο χρήστης αν θέλει να ενημερώσει τον πίνακα των βαθμών (αποδεκτές απαντήσεις: **0** ή **1** και έλεγχος αποδεκτής απάντησης). Αν ο χρήστης εισαγάγει **0**, το πρόγραμμα τερματίζει. Αν ο χρήστης εισαγάγει **1**, καλείται η διαδικασία **update_data** και ακολούθως καλούνται και πάλι οι διαδικασίες **analyze_table** και **print_data** και ερωτάται ξανά ο χρήστης για πιθανή ενημέρωση του πίνακα βαθμών.
Βαθμοί: 20 από 100.

Παράδειγμα εκτέλεσης (με κόκκινο χρώμα επισημαίνονται πιθανές είσοδοι από το πληκτρολόγιο):

Failed grades:

Year: 0, Subject: 3
Year: 1, Subject: 1
Year: 1, Subject: 2
Year: 1, Subject: 3
Year: 1, Subject: 6
Year: 3, Subject: 0
Year: 3, Subject: 1
Year: 3, Subject: 2
Year: 3, Subject: 5
Year: 3, Subject: 6

Count of 10s: 2

Percentage of 10s: 6.25%

Do you want to update a grade? (1 for YES / 0 for NO): **4**

Do you want to update a grade? (1 for YES / 0 for NO): **1**

Year (0-3): **0**

Subject (0-7): **10**

Subject (0-7): **3**

Grade (0-10): **10**

Failed grades:

Year: 1, Subject: 1
Year: 1, Subject: 2
Year: 1, Subject: 3
Year: 1, Subject: 6
Year: 3, Subject: 0
Year: 3, Subject: 1
Year: 3, Subject: 2
Year: 3, Subject: 5
Year: 3, Subject: 6

Count of 10s: 3

Percentage of 10s: 9.38%

Do you want to update a grade? (1 for YES / 0 for NO): **0**