

Αντικειμενοστραφής Προγραμματισμός 2018-2019

Πρώτη Εργασία

Στόχος της πρώτης εργασίας είναι η κατανόηση μερικών βασικών προγραμματιστικών εννοιών. Θα υλοποιηθούν δύο δομές και στην συνέχεια μερικές συναρτήσεις για τροποποίηση του περιεχομένου τους. Σκοπός είναι να γίνουν κατανοητές οι έννοιες των ορισμάτων των συναρτήσεων και η τιμή επιστροφής.

Ερώτημα 1

Ορίστε ένα *struct* **Cook** το οποίο θα αναπαριστά μια εγγραφή ενός μάγειρα. Το *struct* θα πρέπει να περιέχει πέντε πεδία.

- ❖ Ένα πεδίο **name** το οποίο θα είναι ένας πίνακας χαρακτήρων *char[]* πενήντα θέσεων.
- ❖ Ένα πεδίο **spec_cuisine** το οποίο θα είναι ένας πίνακας χαρακτήρων *char[]* πενήντα θέσεων.
- ❖ Ένα πεδίο **age** το οποίο θα πρέπει να είναι ένας μη προσημασμένος ακέραιος *unsigned short int*.
- ❖ Ένα πεδίο **score** το οποίο θα πρέπει να είναι ένας πραγματικός αριθμός χαμηλής ακρίβειας.
- ❖ Ένα πεδίο **chef** το οποίο θα πρέπει να είναι μία λογική μεταβλητή.

Τα πεδία περιγράφουν τα χαρακτηριστικά του μάγειρα όπως η κουζίνα στην οποία είναι καλός, το όνομα και η ηλικία του. Το πεδίο **chef** θα έχει την τιμή 1 αν είναι Chef ενώ την τιμή 0 αν είναι ερασιτέχνης.

Στην συνέχεια ορίστε ακόμα ένα *struct* **Trial** το οποίο θα περιγράφει μια δοκιμασία μαγειρικής. Αυτό το *struct* θα έχει τέσσερα πεδία.

- ❖ Ένα πεδίο **trial_name** το οποίο θα είναι ένας πίνακας χαρακτήρων *char[]* πενήντα θέσεων.
- ❖ Ένα πεδίο **trial_cuisine** το οποίο θα είναι ένας πίνακας χαρακτήρων *char[]* πενήντα θέσεων.
- ❖ Ένα πεδίο **dif_level** το οποίο θα είναι ένας ακέραιος αριθμός.
- ❖ Ένα πεδίο **reward** το οποίο θα είναι ένας ακέραιος αριθμός.

Στόχος των παρακάτω ερωτημάτων είναι να υλοποιηθούν τρεις συναρτήσεις για να γίνει κατηγοριοποίηση των μαγείρων, πρόβλεψη του νικητή της δοκιμασίας και ανταμοιβή ενός μάγειρα αν νικήσει σε κάποια δοκιμασία.

Ερώτημα 2

Ορίστε μια συνάρτηση **show_level** τύπου *void*, που δέχεται ως όρισμα ένα *struct* τύπου **Cook**. Η συνάρτηση θα πρέπει να ελέγχει το πεδίο της ηλικίας και αν είναι Chef και να τυπώνει τα εξής:

- ❖ Αν ο μάγειρας είναι κάτω από 25 χρονών και είναι Chef τότε να τυπώνει το μήνυμα «Young Chef».
- ❖ Αν ο μάγειρας είναι πάνω από 25 χρονών και είναι Chef τότε να τυπώνει το μήνυμα «Old Chef».

Ερώτημα 3

Ορίστε μια συνάρτηση **predict_winner** τύπου *void*, που δέχεται ως όρισμα δύο διαφορετικά *struct* τύπου **Cook** καθώς επίσης και το *struct* τύπου **Trial**. Στόχος είναι ελέγχοντας το είδος της κουζίνας που θα διαγωνιστούν οι δύο μάγειρες καθώς και το *score* που έχουν να γίνει πρόβλεψη του νικητή. Συγκεκριμένα:

- ❖ Αν η κουζίνα ειδίκευσης **μόνο** του πρώτου μάγειρα είναι ίδια με αυτή της δοκιμασίας, τότε αυτός είναι πιθανότερο να κερδίσει. Η συνάρτηση θα πρέπει να τυπώνει το μήνυμα «*First Cook has an advantage*».
- ❖ Ομοίως, αν ο δεύτερος μάγειρας έχει την ίδια κουζίνα ειδίκευσης με αυτή της διαδικασίας τότε αυτός είναι πιθανότερο να κερδίσει. Η συνάρτηση θα πρέπει να τυπώνει το μήνυμα «*Second Cook has an advantage*».
- ❖ Αν οι δύο μάγειρες έχουν την ίδια κουζίνα ειδίκευσης τότε αυτός με το μεγαλύτερο *score* είναι πιθανότερο να κερδίσει. Σε αυτή την περίπτωση η συνάρτηση αρχικά θα τυπώνει το μήνυμα «*Same cuisine*» και στην συνέχεια έχουμε 3 περιπτώσεις:
 - Αν το *score* του πρώτου μάγειρα είναι μεγαλύτερο από του δεύτερου τότε θα πρέπει να τυπώνει το μήνυμα «*First Cook has greater Score so far, so he might win*».
 - Στην αντίθετη περίπτωση πρέπει να τυπώνει το μήνυμα «*Second Cook has greater Score so far, so he might win*».
 - Στην περίπτωση που έχουν ίδιο *score* τότε το μήνυμα είναι το εξής «*Equal Scores*».

Ερώτημα 4

Ορίστε μια συνάρτηση **modify_winner** τύπου *int*, που δέχεται ως όρισμα δύο *struct*, ένα τύπου **Cook** και ένα τύπου **Trial**. Αυτή θα τροποποιεί και θα επιστρέφει το *score* του μάγειρα ως εξής:

- ❖ Αν το επίπεδο της δοκιμασίας (**dif_level**) είναι μικρότερο ίσο με 3, τότε το *score* του νικητή μάγειρα πρέπει να αυξάνεται σύμφωνα με το γινόμενο (**dif_level**)*(**reward**) της δοκιμασίας.
- ❖ Σε κάθε άλλη περίπτωση προστίθεται σε αυτό μόνο το **reward** της δοκιμασίας.

Διευκρινίσεις/Παρατηρήσεις

- ❖ Όλα τα κενά αποτελούνται από ένα space.
- ❖ Το παραδοτέο να ονομαστεί XXXXX.cpp και στην πρώτη γραμμή του να περιέχει σχόλιο(//XXXXX) όπου XXXXX ο Α.Μ. σας.
- ❖ Το XXXX.cpp **να μην περιλαμβάνει συνάρτηση main()**. (Κατά τη διάρκεια της υλοποίησης μπορείτε και πρέπει να γράψετε την δική σας main() σε άλλο αρχείο, πχ main.cpp, ώστε να τεστάρετε τον κώδικα σας. Το main.cpp θα ενσωματώνει το XXXXX.cpp με: #include "XXXXX.cpp")
- ❖ Τα μηνύματα που πρέπει να τυπώνονται καθώς και τα ονόματα των συναρτήσεων και πεδίων των δομών, πρέπει να είναι ακριβώς όπως αναφέρονται στην εκάστοτε εκφώνηση.
- ❖ Οι παραπάνω διευκρινίσεις είναι πολύ σημαντικές γιατί οι εργασίες θα βαθμολογηθούν αυτόματα.
- ❖ Μη παραδώσετε εργασία που δεν κάνει compile σε τουλάχιστον έναν από τους compilers: Dev C++ (σε Windows συστήματα), g++(σε linux/MacOS συστήματα). Αν έχετε την δυνατότητα, προτιμήστε τον g ++. Εργασίες που δεν μεταγλωττίζονται βαθμολογούνται αυτόματα με μηδέν.
- ❖ **Η εργασία είναι ατομική.** Σε περίπτωση αντιγραφής, όλες οι ίδιες/παρόμοιες εργασίες θα μηδενιστούν και οι παραβάτες θα θεωρηθούν μετεξεταστέοι στο Εργαστήριο, δηλαδή **δεν θα μπορέσουν να πάρουν μέρος στην γραπτή εξέταση για αυτή την ακαδημαϊκή χρονιά.**

**WHEN
IN
DOUBT
GOOGLE
IT!**

Καλή επιτυχία ☺