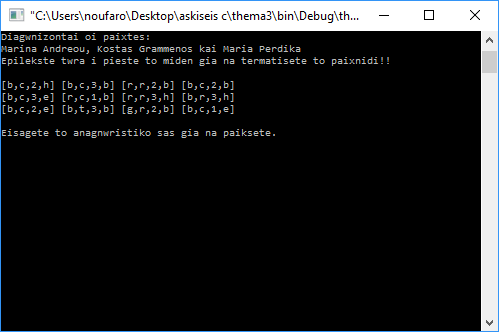
Κατ’ αρχήν δημιουργούμε δύο δομές, μια για τους 3 παίκτες (όνομα, αναγνωριστικό, βαθμολογία, όπου η βαθμολογία είναι τύπου short int) και μια για τα χαρακτηριστικά της κάθε κάρτας, όπου προστέθηκε και μια μεταβλητή checked τύπου int, για να μας βοηθήσει να κάνουμε τους ελέγχους στη συνέχεια.

Δημιουργήσαμε 3 πίνακες. 2 τύπου struct card, όπου στον πρώτο (9\*9) αρχικοποιήσαμε όλους τους πιθανούς συνδυασμούς καρτών (81) και έναν (3\*4) όπου θα καταχωρείται η νέα 12άδα που θα τυπώνεται σε κάθε γύρο του παιχνιδιού. Για να εξασφαλιστεί ότι σε κάθε 12άδα δε θα υπάρχουν δύο ίδιες κάρτες, δημιουργήσαμε έναν πίνακα τύπου int 81 θέσεων με μηδενική τιμή, όπου για κάθε τυχαία κάρτα που δημιουργούμε (με τη rand()%81 και αφού έχουμε θέσει όρισμα στην srand την time(NULL)), ο μοναδικός πλέον αριθμός της κάρτας checked θα αυξάνει κατά 1 το αντίστοιχο πεδίο. Με ένα while loop εξασφαλίζουμε ότι όταν ο αριθμός του πεδίου είναι ήδη 1 και δημιουργηθεί νέα ίδια κάρτα, αυτή δε θα αποθηκεύεται αλλά θα δημιουργείται απ’ ευθείας καινούρια.

Δημιουργούμε πίνακα δομής των παικτών με αρίθμηση το αναγνωριστικό τους (πχ p[‘a’].score), έτσι ώστε όταν κάποιος παίκτης επιλέγει να διαλέξει 3 κάρτες, όλες οι άλλες ενέργειες (πχ αλλαγή βαθμολογίας) να υλοποιούνται για τον συγκεκριμένο παίκτη. Μόλις ο παίκτης εισάγει το αναγνωριστικό του, καλείται η συνάρτηση epiloges\_paikti όπου κάνει όλες τις υπόλοιπες ενέργειες. Στην ίδια συνάρτηση γίνεται έλεγχος αν κάθε ένα από τα χαρακτηριστικά των 3 καρτών επιλογής του παίκτη είναι ίδιο ή διαφορετικό και με βάση το αποτέλεσμα αποθηκεύεται η νέα βαθμολογία.

Υλοποίηση:

α)



β)

