**Οδηγίες τελικής εργασίας**

**Γενική ιδέα**

Δημιουργήστε το παιχνίδι "μάντεψε τον αριθμό". To παιχνίδι επιλέγει ένα τυχαίο αριθμό μεταξύ 1 και 100 και στη συνέχεια να ζητήσει από τον παίκτη να μαντέψει τον αριθμό με 10 προσπάθειες. Μετά από κάθε επανάληψη, ο παίκτης θα πρέπει να ενημερωθεί αν είναι σωστός ή λάθος - και, εάν κάνει λάθος, να ενημερωθεί αν η πρόβλεψη ήταν πολύ χαμηλή ή πολύ υψηλή. Αν ο χρήστης δώσει οποιοδήποτε αριθμό εκτός του ορίου 1 έως 100 (αρνητικό ή θετικό) θα θεωρηθεί ως προσπάθεια. Δεν ζητείται αμυντικός προγραμματισμός στην είσοδο δεδομένων. Ελέγχουμε μόνο την περίπτωση που δεν μπορούμε να πάρουμε ούτε ένα μέρος της εισόδου σαν ακέραιο σύμφωνα με τις υπάρχουσες μεθόδους της javascript μετατροπής συμβολοσειράς σε ακέραιο. Θα πρέπει επίσης να ενημερωθεί ο παίκτης σχετικά με τους αριθμούς που προηγουμένως έχει μαντέψει. Το παιχνίδι θα τελειώσει μόλις ο παίκτης μαντέψει σωστά, ή όταν τελειώσουν οι επιτρεπόμενες επαναλήψεις. Όταν τελειώσει το παιχνίδι, ο παίκτης πρέπει να έχει τη δυνατότητα να ξεκινήσει πάλι.

**Οδηγίες**

1. (5 μονάδες) Δίνεται αρχείο [final-test-template.html](https://mathesis.cup.gr/asset-v1:ComputerScience+CS3.1+20B+type@asset+block/final-test-template.html). Να γίνουν οι εξής αλλαγές:
   * ο τίτλος να είναι "μάντεψε έναν αριθμό"
   * το μέγεθος του text (κουτί εισαγωγής αριθμού) να είναι 5 χαρακτήρες
   * να συνδεθεί η σελίδα με τα εξωτερικά αρχεία [final-test-style.css](https://mathesis.cup.gr/assets/courseware/v1/605deec775c50673a3d830052329cd2e/asset-v1:ComputerScience+CS3.1+20B+type@asset+block/final-test-style.css) και [final-test-script.js](https://mathesis.cup.gr/assets/courseware/v1/907ecc8a411d1c91108156e1cda042b2/asset-v1:ComputerScience+CS3.1+20B+type@asset+block/final-test-script.js) τα οποία δίδονται και στα οποία θα πρέπει να γίνουν αλλαγές όπως ορίζεται στη συνέχεια.

Στο αρχείο [final-test-script.js](https://mathesis.cup.gr/assets/courseware/v1/907ecc8a411d1c91108156e1cda042b2/asset-v1:ComputerScience+CS3.1+20B+type@asset+block/final-test-script.js) να συμπληρωθεί ο κώδικας javascript που επιτρέπει την συμπεριφορά που προδιαγράφτηκε στις γενικές οδηγίες ως ακολούθως:

1. (5 μονάδες) Να δημιουργηθούν οι συνδέσεις μεταξύ συμβάντων που προκύπτουν από ενέργειες του χρήστη και χειριστών συμβάντων checkKey, checkGuess, restart
2. (5 μονάδες) Να ορίσετε συνάρτηση newRandom που βρίσκει τυχαίο αριθμό μεταξύ 1 και 100 και τον εκχωρεί στη μεταβλητή theGuess
3. (5 μονάδες) Να ορίσετε συνάρτηση checkKey που όταν ο χρήστης πατήσει <<enter>> να καλεί τη συνάρτηση που αποτελεί τον κεντρικό ελεγκτή του παιχνιδιού.
4. (5 μονάδες) Να ορίσετε συνάρτηση checkGuess η οποία καλείται είτε όταν ο χρήστης πατήσει <<enter>> στο πεδίο "new-guess" είτε όταν πατήσει το πλήκτρο "check", η οποία είναι ο κεντρικός ελεγκτής, καλεί τη συνάρτηση processGuess (η οποία αποφαίνεται για την ορθότητα του αριθμού) και κάνει τις κατάλληλες ενέργειες για να μην μπορεί να εισάγει ο χρήστης νέο αριθμό ή να ανασταλεί η λειτουργία του <<enter>>, εμφάνιση του πλήκτρου 'restart' και την εξαφάνιση του πλήκτρου 'check' σε περίπτωση ολοκλήρωσης του παιχνιδιού.
5. (5 μονάδες) Να ορίσετε συνάρτηση processGuess(newValue) η οποία καλείται από τη συνάρτηση checkGuess, περιέχει τη λογική του παιχνιδιού, ελέγχει αν η τιμή του χρήστη είναι σωστή, ή αν το παιχνίδι έχει τελειώσει χωρίς ο χρήστης να έχει βρει τον αριθμό, και επιστρέφει αντίστοιχα την τιμή "win", ή "lost", δημιουργεί και εμφανίζει τα κατάλληλα μηνύματα, αλλάζοντας το χρώμα του στοιχείου μηνυμάτων. Όλα τα μηνύματα του προγράμματος εμφανίζονται από την processGuess(). Σε περίπτωση που το παιχνίδι δεν έχει ακόμα τελειώσει, η συνάρτηση μπορεί είτε να μην επιστρέφει κάποια ιδιαίτερη τιμή, είτε να επιστρέφει κάποια τιμή της επιλογής σας.
6. (5 μονάδες) Να ορίσετε συνάρτηση restart η οποία καλείται όταν ο χρήστης πατήσει το πλήκτρο 'restart' και επανεκκινεί τη διαδικασία
7. (10 μονάδες) Στο αρχείο final-test-style.css να κάνετε τις κατάλληλες αλλαγές που αφορούν στα στοιχεία που εμφανίζονται στον χρήστη, προσπαθώντας να επιτύχετε το αποτέλεσμα που φαίνεται στην παρουσίαση. Ιδιαίτερη αναφορά γίνεται ώστε το χρώμα να οριστεί με βάση τα χρώματα που ορίζονται από τις μεταβλητές CSS.

Μπορείτε να κατεβάσετε και τα 3 αρχεία σε ένα zip εδώ: [final-test.zip](https://mathesis.cup.gr/assets/courseware/v1/e01dccd5ce6dd18941a1e005413f6c80/asset-v1:ComputerScience+CS3.1+20B+type@asset+block/final-test.zip)

*Υποσημειώσεις*:

* Η δυνατότητα υποβολής έχει ενεργοποιηθεί και έχετε προθεσμία για την υποβολή της μέχρι και την Κυριακή 21 Ιουνίου στις 12 το βράδυ. Στις 22 Ιουνίου θα ξεκινήσει η διαδικασία αλληλοαξιολόγησης η οποία και θα ολοκληρωθεί στις 28 Ιουνίου.
* από το 3'50'' του βίντεο της άσκησης, το μήνυμα "Λάθος, είσαι πιο χαμηλά" θα πρέπει να εμφανίζεται σε κόκκινο φόντο όπως συμβαίνει νωρίτερα
* Στο 1'16'' και 2'10'' του βίντεο της άσκησης εμφανίζεται ένα γαλάζιο πλαίσιο γύρω από το στοιχείο που έχει το focus. Επίσης το μέγεθος του πεδίου εισαγωγής αριθμού μπορεί να εμφανίζεται μικρότερο στο βίντεο. Το γαλάζιο πλαίσιο όπως και το μέγεθος του πεδίου, οφείλονται στον συγκεκριμένο φυλλομετρητή σε συνδυασμό με το λειτουργικό σύστημα macOS και δεν ανήκει στα ζητούμενα της άσκησης.
* **5/6/2020 UPDATE:** Έχει πραγματοποιηθεί μια μικρή αλλαγή στο HTML αρχείο. Συγκεκριμένα ένα tag που εκ παραδρομής είχε ονομαστεί "heading" άλλαξε σε "header". Αυτή η αλλαγή δεν επηρεάζει την λειτουργία της εφαρμογής.
* **8/6/2020 UPDATE:** Παρατηρήθηκε μια διαφορετική συμπεριφορά μεταξύ των φυλλομετρητών όσον αφορά το refresh (ανανέωση). Συγκεκριμένα στον firefox αν το πεδίο κειμένου είναι συμπληρωμένο, και επιλέξουμε το refresh δεν σβήνει τα περιεχόμενα του.