



## WORDLE

Το παιχνίδι WORDLE έχει «κυριεύσει» το διαδίκτυο το τελευταίο διάστημα. Ο χρήστης προσπαθεί μέσα σε έξι προσπάθειες να μαντέψει μια κρυφή λέξη που αποτελείται από 5 γράμματα. Δεν αποκλείεται κάποιο γράμμα να χρησιμοποιείται παραπάνω από μία φορές μέσα στην κρυφή λέξη. Κάθε φορά που ο χρήστης ολοκληρώνει την εισαγωγή των 5 γραμμάτων της εκάστοτε προσπάθειας του και αν η λέξη που επέλεξε υπάρχει στη βάση δεδομένων του παιχνιδιού, ενημερώνεται για τα εξής:

- ❖ Αν κάποιο ή κάποια από τα γράμματα της λέξης που μάντεψε βρίσκονται στις αντίστοιχες σωστές θέσεις της κρυφής λέξης (Πράσινο χρώμα στην παραπάνω εικόνα).
- ❖ Αν κάποιο ή κάποια από τα γράμματα της λέξης που μάντεψε υπάρχουν μεν στην κρυφή λέξη αλλά ΔΕ βρίσκονται στις αντίστοιχες σωστές θέσεις της κρυφής λέξης (Κίτρινο χρώμα στην παραπάνω εικόνα).
- ❖ Αν το γράμμα της λέξης του δεν υπάρχει πουθενά στην κρυφή λέξη. (Γκρι χρώμα στην παραπάνω εικόνα).

**Να γραφεί πρόγραμμα** σε ΓΛΩΣΣΑ στο οποίο θεωρήστε πως:

- ❖ υπάρχουν αποθηκευμένοι **ήδη** στη βάση δεδομένων του παιχνιδιού:
  - πίνακας SOURCE[2022,5] που περιέχει με αλφαβητική σειρά όλες τις **αποδεκτές** λέξεις που μπορούν να συμμετάσχουν στο παιχνίδι.
  - παράλληλος του SOURCE πίνακας USED[2022] που περιέχει τη λογική τιμή ΨΕΥΔΗΣ σε όλες τις θέσεις του, κάτι που αντιστοιχεί στο γεγονός πως αρχικά δεν έχει χρησιμοποιηθεί καμία λέξη της βάσης δεδομένων του παιχνιδιού ως κρυφή.
- ❖ υπάρχει **ήδη** υλοποιημένη η διαδικασία RANDOM, η οποία χρησιμοποιεί τους πίνακες SOURCE και USED για να δημιουργήσει τον πίνακα WORDLE[5] που περιέχει την **κρυφή** λέξη κάθε γύρου. Θεωρήστε πως, μέσω του πίνακα USED, η διαδικασία επιλέγει την κρυφή λέξη μόνο αν αυτή δεν έχει ξαναχρησιμοποιηθεί.
- ❖ σε κάθε θέση των πινάκων SOURCE και WORDLE περιέχεται **ΜΟΝΟ** ένας χαρακτήρας - γράμμα.

Υλοποιήστε το παιχνίδι ως εξής:

Στην αρχή κάθε γύρου (έξι **το πολύ** προσπάθειες) του παιχνιδιού, να εκτελείται κατάλληλα η διαδικασία RANDOM για τη δημιουργία της κρυφής λέξης.

Κάθε λέξη που δίνει ο χρήστης πρέπει να ελέγχεται αν μπορεί να συμμετάσχει στο παιχνίδι. Σε αντίθεση περίπτωση, αφού ενημερωθεί από το παιχνίδι με κατάλληλο μήνυμα, ο χρήστης να επαναλαμβάνει την εισαγωγή μέχρι η προσπάθεια του να είναι αποδεκτή. Οι αποδεκτές λέξεις που δίνει ως λύσεις ο χρήστης να αποθηκεύονται σταδιακά στον πίνακα TRY[6,5].

Να υλοποιήσετε, στο τέλος του προγράμματος, τη λογική Συνάρτηση FOUND την οποία και να χρησιμοποιήσετε για τον παραπάνω έλεγχο (**ΠΡΟΑΙΡΕΤΙΚΑ**: εκμεταλλευτείτε κατά την υλοποίηση της συνάρτησης το γεγονός πως οι αποδεκτές λέξεις είναι τοποθετημένες στη βάση δεδομένων **αλφαβητικά**).

Το παιχνίδι να ενημερώνει σταδιακά τον πίνακα RESULTS[6,5] ο οποίος θα περιέχει, σύμφωνα με τους κανόνες του παιχνιδιού, τα εξής:

- ❖ το σύμβολο '@' στη θέση που ο χρήστης μάντεψε σωστά ότι υπάρχει το ίδιο γράμμα στην αντίστοιχη θέση της κρυφής λέξης, (όπως θα μας ενημέρωνε το **πράσινο** χρώμα στο παραπάνω γραφικό περιβάλλον)
- ❖ το σύμβολο '#' στη θέση που ο χρήστης μάντεψε κάποιο γράμμα που υπάρχει μεν στην κρυφή λέξη **αλλά** σε διαφορετική θέση (όπως θα μας ενημέρωνε το **κίτρινο** χρώμα στο παραπάνω γραφικό περιβάλλον)
- ❖ το σύμβολο '\_' σε κάθε άλλη θέση. (όπως αντίστοιχα θα μας ενημέρωνε το **γκρι** χρώμα παραπάνω).

Για το σκοπό αυτό, μετά από κάθε *αποδοκτή* προσπάθεια του χρήστη, να χρησιμοποιείτε κατάλληλα τη Διαδικασία CHECK (την οποία και να υλοποιήσετε στο τέλος του προγράμματος) που θα εξετάζει τη συγκεκριμένη προσπάθεια του χρήστη σε σχέση με την κρυφή λέξη και θα ενημερώνει κατάλληλα τον πίνακα RESULTS.

Το παιχνίδι να εμφανίζει **για κάθε αποτυχημένη** προσπάθεια που έχει κάνει **μέχρι εκείνη τη στιγμή** ο χρήστης:

- ❖ τη λέξη που αποτελούσε την προσπάθεια του **ΚΑΙ**
- ❖ τις πληροφορίες μέσω του πίνακα RESULTS που ίσως τον βοηθήσουν να μαντέψει πιο σωστά στην επόμενη προσπάθεια του.

Στο τέλος του κάθε γύρου:

- ❖ Αν βρήκε την κρυφή λέξη:
  - Να εμφανίζει σε ποια προσπάθεια από τις έξι που είχε στη διάθεση του τα κατάφερε τελικά.
  - Να ενημερώνει τη δομή που εξασφαλίζει ότι η κρυφή λέξη δε θα επαναληφθεί.
- ❖ Αν δεν κατάφερε να βρει τελικά την κρυφή λέξη, να αποκαλύπτει ποια ήταν αυτή εμφανίζοντας κατάλληλο μήνυμα. Σε αυτή την περίπτωση δεν αποκλείεται η κρυφή λέξη να ξαναχρησιμοποιηθεί μέσω της RANDOM.

Στη συνέχεια να τον ρωτά αν θέλει να ξαναπαιξει και να αποδέχεται μόνο το «**N**» ή το «**O**» ως θετική ή αρνητική απάντηση αντίστοιχα.

Το παιχνίδι τερματίζεται όταν:

- ❖ ο χρήστης απαντήσει αρνητικά, **ή**
- ❖ όταν ο χρήστης μαντέψει σωστά **όλες** τις κρυφές λέξεις που έχει το παιχνίδι στη βάση δεδομένων του, **ή**
- ❖ όταν ο χρήστης δεν καταφέρει να μαντέψει την κρυφή λέξη **13 συνεχόμενες** φορές.

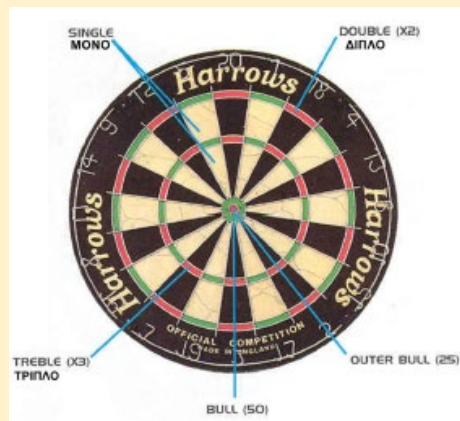
Στην πρώτη περίπτωση, το παιχνίδι να εμφανίζει το ποσοστό των γύρων που ο χρήστης δεν κατάφερε τελικά να βρει την κρυφή λέξη.

Στη δεύτερη περίπτωση, το παιχνίδι να εμφανίζει με κατάλληλο μήνυμα πόσες φορές βρήκε ο χρήστης την κρυφή λέξη με τις λιγότερες προσπάθειες σε κάποιο γύρο καθώς και πόσες ήταν αυτές (οι λιγότερες προσπάθειες).

π.χ. «Οι λιγότερες προσπάθειες που έκανες σε οποιονδήποτε γύρο ήταν 2 και έτσι θρήκες 131 κρυφές λέξεις».

Στην τελευταία περίπτωση, το παιχνίδι να εμφανίζει κατάλληλο μήνυμα που θα περιέχει την **πρώτη** και την **τελευταία** κατά σειρά από τις **13** αυτές κρυφές λέξεις που τον δυσκόλεψαν περισσότερο, μάντεψε δηλαδή τα λιγότερα γράμματα τους που τελικά ήταν στη σωστή θέση της κρυφής λέξης, καθώς και τον αριθμό αυτό. Υποθέστε πως υπάρχουν τουλάχιστον δυο λέξεις που τον δυσκόλεψαν περισσότερο.

π.χ. «Τα λιγότερα γράμματα στη σωστή θέση που θρήκες σε κάποιες από τις 13 συνεχόμενες αποτυχημένες προσπάθειες σου ήταν 2. Την πρώτη φορά που έγινε αυτό κρυφή λέξη ήταν η PROXY, και την τελευταία φορά κρυφή λέξη ήταν η WALTZ.»



## DARTS

Το παιχνίδι **Darts** (Βελάκια) παίζεται από 2 παίκτες. Ο στόχος είναι χωρισμένος με σύρμα σε 20 τμήματα (φέτες). Τα τμήματα είναι αριθμημένα με ένα δαχτυλίδι που είναι αποσπώμενο από το στόχο για να μπορούμε να τον γυρίζουμε και να χρησιμοποιούμε όλη την επιφάνειά του εξίσου. Εκτός από τα 20 τμήματα υπάρχει και η περιοχή του κέντρου του στόχου. Εκεί έχουμε την εξωτερική ζώνη (outer bull ή bull) που αξίζει 25 πόντους και το κέντρο (bull's eye ή double bull) που αξίζει 50 πόντους.

Πάνω στο στόχο βλέπουμε επίσης δύο δαχτυλίους που περνούν από όλα τα νούμερα. Ο δακτύλιος που βρίσκεται στην περιφέρεια ορίζει την περιοχή του διπλού (η αξία της φέτας επί δύο), ενώ ο δακτύλιος που βρίσκεται μεταξύ περιφέρειας και κέντρου ορίζει την περιοχή των τριπλών (η αξία της φέτας επί τρία).

### ΚΑΝΟΝΕΣ ΠΑΙΧΝΙΔΙΟΥ

Νικητής αναδεικνύεται όποιος φτάσει πρώτος στα **3** κερδισμένα **legs (Best of 5 legs)**. Οι παίκτες ξεκινούν κάθε leg με 501 πόντους και σκοπός τους είναι να μηδενίσουν το σκορ τους με όσο το δυνατόν λιγότερα βελάκια. Η σειρά εκκίνησης εναλλάσσεται σε κάθε leg. Το παιχνίδι ξεκινάει με απ' ευθείας αφαίρεση των πόντων από το πρώτο επιτυχημένο βελάκι **όμως τελειώνει πάντοτε με ένα βελάκι σε διπλό (double-out) ή σε bull's eye (διπλό του 25)**. Οι παίκτες ρίχνουν από 3 (το πολύ) συνεχόμενα βελάκια ο καθένας, εναλλάξ, μέχρι να μηδενίσει κάποιος το σκορ του. Σε περίπτωση που η βολή είναι εκτός στόχου, δε μειώνεται το σκορ. Αν κάποιος ρίξει ΚΑΙ τις τρεις βολές του εκτός στόχου ή το υπόλοιπό του γίνει **1** (αφού δε θα έχει τη δυνατότητα να τελειώσει με διπλό σε καμία περίπτωση στην επόμενη βολή του), νικητής αναδεικνύεται αυτομάτως ο άλλος. Αν ρίξει βολή που το αποτέλεσμά της είναι μεγαλύτερο από το σκορ που είχε αφήσει πριν τη βολή, τότε χάνει το δικαίωμα να ρίξει τις εναπομείνουσες 1 ή 2 βολές του και μένει στο σκορ που είχε πριν από αυτή τη βολή [π.χ. έχει 60 υπόλοιπο και με το πρώτο βελάκι ρίχνει διπλό 19. Για να τελειώσει το παιχνίδι θέλει 22 πόντους που επιτυγχάνονται με διπλό 11. Αν ρίξει διπλό 12 (24 πόντους) τότε ο παίκτης δε ρίχνει άλλο βελάκι και μένει στο 22. Αν όμως ρίξει στη δεύτερη βολή διπλό 6 (12 πόντους) τότε με νέο υπόλοιπο 10 θα προσπαθήσει να φέρει στην τρίτη βολή διπλό 5 για να μηδενίσει. Αν δεν τα καταφέρει ή θα μειωθεί περαιτέρω αν π.χ. ρίξει μονό 3 θα 'χει υπόλοιπο 7 ή αν ρίξει οτιδήποτε πάνω από 10 θα παραμείνει στο 10 το υπόλοιπο του.]

### Δημιουργία Προγράμματος

Να γράψετε πρόγραμμα σε «ΓΛΩΣΣΑ» που θα υλοποιεί ένα τουρνουά DARTS ως εξής:

Αρχικά να ζητά τα ονόματα των 64 παικτών του τουρνουά και να τα τοποθετεί στον πίνακα **ON[64]**. Θεωρήστε ότι τα ονόματα είναι όλα διαφορετικά μεταξύ τους.

Στην (1<sup>η</sup>) φάση των 64, θα τίθενται αντιμέτωποι οι παίκτες στις αντιδιαμετρικές θέσεις του πίνακα ως εξής: 1-64, 2-63, 3-62,..., 32-33. Οι νικητές κάθε παιχνιδιού θα τοποθετούνται σε βοηθητικό πίνακα **WIN** στις θέσεις 1-32 ως εξής: στη θέση 1 ο νικητής του 1-64, στη θέση 2 ο νικητής του 2-63 κ.ο.κ. Στην επόμενη 2<sup>η</sup> φάση των 32, τα ζευγάρια θα είναι 1-32, 2-31,..., 16-17 και οι νικητές θα τοποθετηθούν στις θέσεις 1-16 του πίνακα **WIN** αντίστοιχα.

Η ίδια διαδικασία θα ακολουθηθεί μέχρι τον Μεγάλο Τελικό ανάμεσα στους παίκτες που θα βρίσκονται εκείνη τη στιγμή στις θέσεις 1-2 του πίνακα **WIN**.

Στη φάση των 64, σε κάθε ζευγάρι, πρώτος ξεκινά ο παίκτης που βρίσκεται στη μικρότερη θέση του πίνακα. Σε όλες τις επόμενες φάσεις (έξι συνολικά : 64, 32, 16, 8, ημιτελικά, τελικός), σε κάθε ζευγάρι, πρώτος ξεκινά αυτός που έχει ρίξει τις λιγότερες συνολικά εύστοχες βολές μέχρι εκείνη τη στιγμή σε όλο το τουρνουά. Θεωρήστε ότι δεν υπάρχουν ισοβαθμίες όταν εξετάζεται η σειρά εκκίνησης.

Θεωρήστε πως έχουν ήδη υλοποιηθεί και μπορείτε να χρησιμοποιήσετε αν θέλετε, τα υποπρογράμματα:

- **SYNAPTHΣH FIND (NAME, key)** που επιστρέφει τη θέση που βρίσκεται αποθηκευμένη η τιμή *χαρακτήρα* **key** στον πίνακα *χαρακτήρων* **NAME [64]**.
- **SYNAPTHΣH SUM (Score, line, col)** που επιστρέφει το άθροισμα των στοιχείων ενός ακέραιου πίνακα **Score[64,6]** που βρίσκονται στη γραμμή **line** στις στήλες **1** μέχρι και **col**.
- **ΔΙΑΔΙΚΑΣΙΑ REPLACE (player1, player2)** που αντιμεταθέτει τα περιεχόμενα των μεταβλητών τύπου *χαρακτήρα* **player1** και **player2**.

Για την υλοποίηση του τουρνουά, να δημιουργήσετε και να χρησιμοποιήσετε κατάλληλως τις εξής **ΔΙΑΔΙΚΑΣΙΕΣ**:

- **SHOT (points, darts)** που δέχεται τους πόντους (**points**) με τους οποίους ξεκινά μια τριάδα προσπαθειών κάποιος παίχτης σε ένα leg. Για κάθε βολή διαβάζει την αξία της χωρίς έλεγχο εγκυρότητας (θεωρείστε πως για βολή εκτός στόχου εισάγεται ο αριθμός μηδέν) και, αν πρέπει, διαβάζει χωρίς έλεγχο εγκυρότητας τον πολλαπλασιαστή της βολής : 1, 2 ή 3. Η διαδικασία, σύμφωνα με τους κανόνες του παιχνιδιού, επιστρέφει στην ίδια παράμετρο το νέο υπόλοιπο του μετά τις βολές. Αν και οι τρεις δε βρουν στόχο, επιστρέφεται η τιμή **-1**. Στην παράμετρο (**darts**) επιστρέφει τον αριθμό των βολών (1, 2 ή 3) που τελικά έριξε ο παίχτης.
- **LEG (player1, player2, winner, darts1, darts2)** που θα υλοποιεί **ένα leg** κάποιου αγώνα. Δέχεται τα ονόματα των 2 παικτών: **player1** αυτός που ξεκινά να ρίχνει πρώτος και **player2**. Χρησιμοποιώντας κατάλληλα τη διαδικασία **SHOT**, σύμφωνα με τους κανόνες του παιχνιδιού, επιστρέφει το όνομα του νικητή (**winner**) του εν λόγω **leg** καθώς και το πλήθος των εύστοχων βολών των 2 παιχτών **darts1** και **darts2** αντίστοιχα.
- **MATCH (player1, player2, winner, darts)** που θα δέχεται τα ονόματα των 2 παικτών **player1, player2** και θα υλοποιεί **έναν αγώνα**. Χρησιμοποιώντας κατάλληλα τη διαδικασία **LEG**, θα επιστρέφει το όνομα του νικητή (**winner**) και το πλήθος (επίδοση) των εύστοχων βολών (**darts**) που απαιτήθηκαν από αυτόν για να κερδίσει τον αγώνα.
- **UPDATE (winner, darts, NAME, SCORE)** που θα δέχεται το όνομα του νικητή κάποιου αγώνα (**winner**) και το πλήθος εύστοχων των βολών (**darts**) που απαιτήθηκαν από αυτόν για να κερδίσει τον αγώνα, τον πίνακα *χαρακτήρων* **NAME [64]** και έναν ακέραιο πίνακα **Score [64,6]** τον οποίο και θα ενημερώνει κατάλληλα.

Εκτός από το όνομα του τελικού νικητή, το πρόγραμμα να εμφανίζει αν κέρδισε με επίδοση καλύτερη ή χειρότερη από τον μέχρι τότε μέσο όρο του.

Η άποψη πως όσο το τουρνουά προχωρούσε οι παίκτες παρουσιάζονταν όλο και πιο αγχωμένοι, ισχύει ή όχι; (η μέση επίδοση των νικητών της κάθε φάσης εκτός του τελικού, όλο και χειρότερη;)



## WINE

Ένας οινοπαραγωγός αγοράζει μούστο προς 1.20€/λίτρο με σκοπό να πουλήσει κρασί όταν ολοκληρωθεί η ζύμωση, διαδικασία που επιφέρει απώλειες 4% στον όγκο του μούστου. Ο οινοπαραγωγός πουλά το κρασί προς 2€/λίτρο. Να γραφεί ο αλγόριθμος ο οποίος :

Αρχικά να διαβάσει πόσα λίτρα μούστο, αγόρασε ο οινοπαραγωγός. Θεωρείστε τον αριθμό ακέραιο που ανήκει στο [20000, 50000].

Να διαβάσει επαναληπτικά για κάθε πελάτη έναν ακέραιο αριθμό (θεωρείστε τους όλους διαφορετικούς) που να εξασφαλίζει ότι ανήκει στο [100, 1000], ο οποίος αντιστοιχεί στον αριθμό των λίτρων που θέλει να αγοράσει αυτός. Κάθε πελάτης που τελικά εξυπηρετείται (θεωρείστε πως το πλήθος τους είναι πάνω από ένα), ερωτάται να επιλέξει τους τρόπους συσκευασίας του κρασιού ανάμεσα σε μεταλλικούς ασκούς χωρητικότητας 20 και 5 λίτρων και σε γυάλινα μπουκάλια χωρητικότητας 1 λίτρου. Ο πελάτης τελικά απαντά δίνοντας 2 ακέραιους αριθμούς που αντιστοιχούν στο πόσα λίτρα από την παραγγελία θέλει να συσκευαστούν με ασκούς και πόσα με μπουκάλια. Αν δώσει δυο φορές το μηδέν, θα πάρει όλη την ποσότητα χύμα με δικό του τρόπο. Αν η αρχική του παραγγελία μπορεί να συσκευαστεί με τον τρόπο που ζητά (κι ως περισσεύουν κάποια λίτρα), να τους δέχεται αλλιώς να δίνει μήνυμα λάθους και να τους ξαναζητά μέχρι να δοθούν σωστά. Θεωρείστε πως εξαντλούνται πρώτα οι 20λιτροι ασκοί πριν χρησιμοποιηθούν οι 5λιτροι.

Ο οινοπαραγωγός αγοράζει σε τιμή χονδρικής κάθε συσκευασία ασκού των 20 λίτρων προς 1.6€ και των 5 λίτρων προς 1.1€, ενώ κάθε εμφιαλωμένο μπουκάλι του κοστίζει 0.40€. Το κόστος συσκευασίας το επωμίζεται τελικά ο κάθε πελάτης με επιβάρυνση 10% για κάθε ασκό και 5% για κάθε μπουκάλι.

Η επαναληπτική διαδικασία τελειώνει όταν δεν αρκεί το υπόλοιπο στις δεξαμενές του οινοπαραγωγού για να εξυπηρετήσει κάποιον πελάτη.

Μιας και πλησιάζει η γιορτή του πολιούχου του χωριού, όλοι οι πελάτες έχουν συμφωνήσει να δωρίσουν όποια ποσότητα κρασιού τυχόν περισσέψει από τη διαδικασία της συσκευασίας για την ενίσχυση του Πανηγυριού στην πλατεία του χωριού. Το ίδιο θα πράξει και ο οινοπαραγωγός.

Στο τέλος να εμφανίζονται τα ακόλουθα:

- 1) Η μέση ποσότητα κρασιού που προμηθεύτηκε κάθε πελάτης.
- 2) Το μέσο πλήθος ασκών των 20 λίτρων που προμηθεύτηκε κάθε πελάτης.
- 3) Ο συνολικός αριθμός μπουκαλιών που συσκευάστηκαν.
- 4) Πόσα ευρώ έβγαλε ο οινοπαραγωγός μόνο από την εμπορία των συσκευασιών.
- 5) Ποσα λίτρα κρασί προσφέρθηκαν στο πανηγύρι τελικά.
- 6) Ποιος κατά σειρά πελάτης
  - a. Έκανε τη μικρότερη παραγγελία σε λίτρα.
  - b. Άφησε το μεγαλύτερο κέρδος στον οινοπαραγωγό.
  - c. Είχε τη μέγιστη ποσοστιαία διαφορά σε λίτρα παραγγελίας από τον αμέσως προηγούμενο εξυπηρετηθέντα πελάτη, και πόσο είναι αυτό το ποσοστό;*Θεωρείστε πως και στις 3 παραπάνω περιπτώσεις ο πελάτης είναι μοναδικός.*
- 7) Ποια θα ήταν η απόλυτη και ποια η ποσοστιαία μεταβολή στο κέρδος του οινοπαραγωγού αν πούλαγε όλο το κρασί χύμα (ασυσκευάστο);



Αυτό το έργο διατίθεται με άδεια Creative Commons BY Greece 3.0

Αναφορά Δημιουργού

<http://creativecommons.org/licenses/by/3.0/gr/>

