

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΠΜΣ – ΡΟΗ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

Η γλώσσα προγραμματισμού C

ΕΡΓΑΣΤΗΡΙΟ 3.3: Πίνακες, βρόχοι, αλφαριθμητικά, αρχεία, δομές, συναρτήσεις+

Μάιος-Ιούνιος 2019

Το σημερινό εργαστήριο αποτελεί εμβάθυνση σε μετρίως προχωρημένα θέματα της γλώσσας C, από πίνακες, πολλαπλά φωλιασμένους βρόχους, αλφαριθμητικά, αρχεία, δομές `struct` και συναρτήσεις. Η παρακάτω άσκηση είναι το τελευταίο μέρος των πιο προχωρημένων εργαστηρίων του μαθήματος που μπορούν να παραδοθούν μέχρι το τέλος της εξεταστικής.

Άσκηση: «Παιχνίδι λέξεων τύπου scrabble»

Ξεκινήστε μια νέα εργασία `lab3_3` και δημιουργήστε ένα αρχείο προγράμματος με όνομα `ONOMA_lab3_3.1.c`, όπου αντί για `ONOMA` θα βάλετε το ονοματεπώνυμό σας. Συνδέστε το αρχείο αυτό με την εργασία.

Στην άσκηση αυτή καλείστε να γράψετε ένα πρόγραμμα παιχνιδιού, το οποίο μοιάζει τόσο με σταυρόλεξο όσο και με το γνωστό παιχνίδι “scrabble”. Ο χρήστης και ο υπολογιστής, ως αντίπαλοι, συμπληρώνουν ένα ταμπλό με λέξεις και συγκεντρώνουν βαθμούς ανάλογα με το μήκος της κάθε λέξης και της θέσης της στο ταμπλό. Το ταμπλό είναι τετράγωνος πίνακας χαρακτήρων μεγέθους $n \times n$, με κάθε κελί του να δέχεται ένα γράμμα της λέξης που τοποθετείται. Οι λέξεις επιλέγονται από κάποια λίστα λέξεων. Ειδικότερα:

- Οι λέξεις τοποθετούνται οριζόντια – από αριστερά προς τα δεξιά, ή κατακόρυφα – από επάνω προς τα κάτω.
- Κάθε λέξη που τοποθετείται διαγράφεται από τη λίστα.
- Η πρώτη λέξη επιλέγεται στην τύχη και τοποθετείται έτσι ώστε να περνάει πάνω από το κεντρικό κελί του ταμπλό.
- Κάθε επόμενη λέξη τοποθετείται σε επαφή με ήδη υπάρχουσα λέξη, έτσι ώστε να έχει τουλάχιστον ένα γράμμα κοινό με τη λέξη, το οποίο είτε προϋπάρχει είτε προστίθεται με τη νέα λέξη.
- Μια νέα λέξη μπορεί να τέμνει περισσότερες από μία λέξεις σε κοινά γράμματα.
- Κάθε επαφή γραμμάτων πρέπει να σχηματίζει λέξη της λίστας.
- Ο χρήστης και ο υπολογιστής παίζουν εναλλάξ.
- Το παιχνίδι τελειώνει μόλις δε μπορεί να τοποθετηθεί άλλη λέξη στο ταμπλό.

Η τοποθέτηση λέξεων αποφέρει βαθμούς στον κάθε παίχτη, ως εξής:

- Κάθε λέξη αποφέρει κατ’ αρχήν τόσους βαθμούς όσο είναι το μήκος της.
- Κάθε γράμμα που τοποθετείται σε ειδικά σημειωμένα κελιά μετράει διπλά.

Η τοποθέτηση της λέξης TAINIA κάθετα στη λέξη ΦΥΤΟ δημιουργεί και τη λέξη ΑΣ, οπότε λαμβάνονται 7 βαθμοί για την πρώτη και 4 βαθμοί για τη δεύτερη, εφόσον μετράει διπλά επειδή τέμνει δύο λέξεις, συνολικά 11 βαθμοί:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Φ | | | | | |
| | | | | | Υ | | | | | |
| | | | | | Τ | Α | Ι | Ν | Ι | Α |
| Τ | Ε | Λ | Ι | Κ | Ο | Σ | | | | |
| | | | | Α | | | | | | |
| | | | Τ | Ρ | Α | Π | Ε | Ζ | Ι | |
| | | | | Ε | | | | | | |
| | | | | Κ | | | | | | |
| | | | | Λ | | | | | | |
| | | | | Α | | | | | | |
| | | | | | | | | | | |

Μετά από αρκετές κινήσεις, το ταμπλό συμπληρώνεται και καταλήγει στη μορφή:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Π | Ο | Δ | Ο | Σ | Φ | Α | Ι | Ρ | Ο | |
| Α | | | Ν | | Υ | | | | Χ | |
| Σ | | | | | Τ | Α | Ι | Ν | Ι | Α |
| Τ | Ε | Λ | Ι | Κ | Ο | Σ | | | Α | |
| Ε | | | | Α | | | | | | |
| Ρ | | | Τ | Ρ | Α | Π | Ε | Ζ | Ι | |
| Ι | Ο | Ν | | Ε | | Α | | Ω | | Π |
| Ω | | Ε | | Κ | | Ι | | Η | Ρ | Α |
| Σ | | Α | Χ | Λ | Α | Δ | Ι | | | Τ |
| Η | | Ρ | | Α | | Ι | | | | Ο |
| | | Η | | | | Α | Γ | Ι | Ο | Σ |

Παρατηρήστε ότι έχουν μείνει αρκετές λέξεις από τη λίστα, οι οποίες δε μπόρεσαν να τοποθετηθούν. Αν είχαν γίνει διαφορετικές επιλογές λέξεων, το ταμπλό θα κατέληγε σε διαφορετική τελική μορφή.

Το πρόγραμμά σας θα πρέπει να παίζει ως ένας από τους δύο αντιπάλους, αναζητώντας και τοποθετώντας λέξεις στο ταμπλό όσο γίνεται αποτελεσματικότερα, να δέχεται και να ελέγχει τις κινήσεις του χρήστη – δεύτερου παίχτη, και βέβαια να συμπληρώνει και να προβάλλει στην οθόνη την τρέχουσα μορφή του ταμπλό. Η λίστα λέξεων του παιχνιδιού θα πρέπει να διαβάζεται από κάποιο κατάλληλο αρχείο εισόδου, που δίνεται ως παράμετρος στο πρόγραμμά σας.

Μια ιδέα υλοποίησης της λίστας λέξεων είναι ως πίνακας συμβολοσειρών:

```
char * Lista_Lexewn[L_max];
```

όπου *L_max* κατάλληλη σταθερά που δηλώνεται με οδηγία *define* και περιέχει το μέγιστο επιτρεπτό πλήθος λέξεων, πχ 256. Ο πίνακας αυτός αρχικοποιείται σε NULL (ή

ισοδύναμα σε $(\text{char}^*)0$) για όλα τα στοιχεία του, και με κάθε λέξη που διαβάζεται από το αρχείο εισόδου σε κάποιον προσωρινό μονοδιάστατο πίνακα χαρακτήρων – έστω *temp*, δημιουργείται αντίγραφο αυτής στον πίνακα, με τη βοήθεια της συνάρτησης βιβλιοθήκης *strdup()*, η οποία με όρισμα μια συμβολοσειρά, δημιουργεί και επιστρέφει ως αποτέλεσμα ένα αντίγραφό της:

```
Lista_Lexewn[i] = strdup(temp);
```

για κατάλληλη τιμή μεταβλητής δείκτη *i*. Κάθε φορά που το παιχνίδι τοποθετεί μια από τις λέξεις του πίνακα στο ταμπλό, η λέξη διαγράφεται με τη βοήθεια των εντολών:

```
free(Lista_Lexewn[i]);  
Lista_Lexewn[i] = NULL;
```

και πάλι για κατάλληλη τιμή μεταβλητής δείκτη *i*.

Από την άλλη πλευρά, επειδή το ταμπλό περιέχει γράμματα, αλλά και ειδικούς χαρακτήρες που υποδηλώνουν διπλή αξία γράμματος, μια ιδέα υλοποίησης του ταμπλό είναι ως ένας δισδιάστατος πίνακας από δομές, όπου κάθε δομή έχει δύο πεδία, ένα για το γράμμα που έχει τοποθετηθεί στο αντίστοιχο κελί, και ένα για το αν το κελί είναι διπλής αξίας γράμματος ή όχι.

Σύμφωνα με τα παραπάνω, το πρόγραμμά σας θα πρέπει να υλοποιεί τα ακόλουθα βήματα:

1. Όρισε με οδηγία *define* ένα μέγιστο μέγεθος ταμπλό *n_max*. Δήλωσε τον τύπο δομής ενός κελιού – έστω *struct keli* – και το ταμπλό ως δισδιάστατο πίνακα τέτοιων δομών – έστω *A*.
2. Διάβασε από το πληκτρολόγιο την τιμή του *n* φροντίζοντας να είναι μέχρι *n_max*. Πρόσεξε ότι το *n* θα πρέπει να είναι περιττός αριθμός, ώστε να ορίζεται κεντρικό κελί στο ταμπλό.
3. Αρχικοποίησε τα δύο πεδία των δομών όλων των κελιών του ταμπλό, με ‘ ’ (κενό) και 0 αντίστοιχα.
4. Όρισε σκόρπια στο ταμπλό τα κελιά που υποδηλώνουν διπλή αξία γράμματος. Αυτά θα πρέπει να περιλαμβάνουν το κεντρικό κελί καθώς και τα κελιά στις τέσσερις γωνίες του ταμπλό, ενώ τα υπόλοιπα όρισέ τα με τυχαίο τρόπο. Για τα κελιά διπλής αξίας γράμματος βάλε 1 στο αντίστοιχο πεδίο της δομής κελιού. Το πλήθος των κελιών αυτών θα πρέπει να είναι ίσο με $n+2$.
5. Άνοιξε το αρχείο εισόδου, τερματίζοντας το πρόγραμμα με κατάλληλο μήνυμα αν αυτό δεν ανοίξει σωστά.
6. Διάβασε από το αρχείο μία-μία τις λέξεις του παιχνιδιού, τοποθετώντας τες στον πίνακα *Lista_Lexewn* με τον τρόπο που περιγράφεται πιο πάνω. Σταμάτησε την ανάγνωση αν φτάσεις στην τελευταία θέση του πίνακα πριν φτάσεις στο τέλος του αρχείου. Εκτύπωσε κατάλληλο μήνυμα με την ολοκλήρωση της ανάγνωσης και συνέχισε.
7. Εκτύπωσε την αρχική μορφή του ταμπλό, καθώς και τις διαθέσιμες λέξεις του παιχνιδιού.
8. Επανάλαβε:
 - 8.1. Ζήτη από το χρήστη την επιλογή μιας λέξης, της θέσης και της κατεύθυνσης τοποθέτησής της στο ταμπλό.
 - 8.2. Έλεγξε την ορθότητα επιλογής του χρήστη και επέστρεψε στο προηγούμενο βήμα σε περίπτωση λάθους.
 - 8.3. Τοποθέτησε τη λέξη στο ταμπλό και εκτύπωσε το.

- 8.4. Υπολόγισε τους βαθμούς που συγκεντρώνονται με την τοποθέτηση αυτή, πρόσθεσε τους βαθμούς στους βαθμούς του παίχτη, και εκτύπωσε στην οθόνη τους βαθμούς αυτούς.
- 8.5. Αφαίρεσε τη λέξη από τον πίνακα με τον τρόπο που περιγράφεται πιο πάνω.
- 8.6. Υλοποίησε την κίνηση του υπολογιστή:
 - 8.6.1. Εξέτασε κάθε μία από τις διαθέσιμες λέξεις, με φθίνουσα σειρά μήκους, ώστε να εξετάσεις πρώτα τις μεγαλύτερες λέξεις.
 - 8.6.2. Βρες πιθανές θέσεις τοποθέτησης της λέξης, αν υπάρχουν, σύμφωνα με τους πιο πάνω κανόνες.
 - 8.6.3. Επίλεξε την τοποθέτηση που θα αποφέρει τη μεγαλύτερη συγκομιδή βαθμών.
 - 8.6.4. Τοποθέτησε τη λέξη και αφαίρεσέ την από τον πίνακα των λέξεων.
 - 8.6.5. Εκτύπωσε το ταμπλό, τη βαθμολογία και τις διαθέσιμες λέξεις, μέχρι να μη γίνεται νέα τοποθέτηση.
9. Ανάδειξε το νικητή και εκτύπωσε μήνυμα τερματισμού.

Για τα παραπάνω βήματα θα πρέπει να υπάρχει η δυνατότητα δήλωσης αδυναμίας του χρήστη να βρει λέξη, με μετάβαση απ' ευθείας από το βήμα 8.1 στο βήμα 8.6, καθώς και του υπολογιστή, με μετάβαση από το βήμα 8.6.2 στο βήμα 8.6.5. Φυσικά το δεύτερο δεν θα πρέπει να συμβαίνει αν γίνεται ολοκληρωμένος έλεγχος για όλες τις δυνατές τοποθετήσεις λέξεων στο ταμπλό! Για να τερματιστεί ο βρόχος, θα πρέπει να υπάρξει ταυτόχρονη αδυναμία του χρήστη και του υπολογιστή για εύρεση λέξης.

Η εκτύπωση του ταμπλό γίνεται μόνο για τα πεδία χαρακτήρα των δομών των κελιών του ταμπλό. Βρείτε κάποιον τρόπο ώστε να δείχνετε τα κελιά διπλής αξίας γράμματος, για παράδειγμα με '\$' αν η θέση είναι κενή ή με μικρά έναντι κεφαλαίων αν η θέση είναι κατειλημμένη.

Προσπαθήστε να υλοποιήσετε τον κώδικά σας με τη βοήθεια συναρτήσεων, ώστε ο κύριος βρόχος του παιχνιδιού να είναι ευανάγνωστος. Για παράδειγμα, μπορείτε να δημιουργήσετε μια συνάρτηση, η οποία να υλοποιεί τα βήματα 8.1 και 8.2, διαβάζοντας και ελέγχοντας την κίνηση του παίχτη, καθώς και μια συνάρτηση, η οποία να υλοποιεί την κίνηση του υπολογιστή.

Μεταφράστε, επαληθεύστε, αποθηκεύστε και υποβάλετε το πρόγραμμά σας.