

## ΕΡΓΑΣΤΗΡΙΟ 9

### ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥΙ, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2009-2010

#### Στόχοι

- Αναδρομή

#### Πριν ξεκινήσετε

Δημιουργήστε ένα φάκελο του οποίου το όνομα είναι

επώνυμο\_όνομα\_AEM\_email

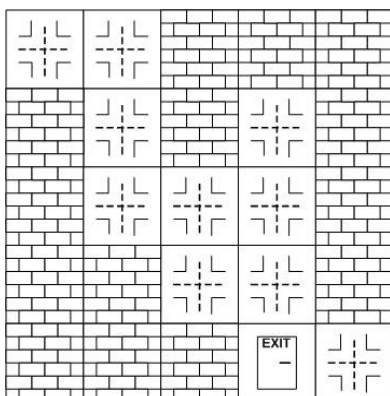
Μέσα σε αυτό το φάκελλο θα πρέπει να αποθηκεύσετε την παραδοτέα άσκηση.

#### Λυμένη άσκηση

Γράψτε ένα πρόγραμμα το οποίο διαβάζει την περιγραφή ενός λαβύρινθου και χρησιμοποιεί μια αναδρομική συνάρτηση για να βρει αν είναι δυνατή η έξοδος από το λαβύρινθο.

#### Ο λαβύρινθος

Ο λαβύρινθος θα αναπαρασταθεί ως ένας δισδιάστατος πίνακας 10x10, όπου κάθε κελί μπορεί να περιέχει τοίχο (W, wall) ή μονοπάτι (P, path), ή κάποια έξοδο (E, Exit). Για παράδειγμα, ο 5x5 λαβύρινθος που φαίνεται κάτω αριστερά περιγράφεται όπως δείχνει το σχήμα στα δεξιά.



P	P	W	W	W
W	P	W	P	W
W	P	P	P	W
W	W	P	P	W
W	W	W	E	P

Η είσοδος του λαβύρινθου είναι το πάνω αριστερά κελί (με συντεταγμένες (0,0)). Μπορείτε να υποθέσετε ότι το κελί αυτό περιέχει πάντα μονοπάτι.

Αν βρισκόμαστε σε ένα κελί τότε μπορούμε να κινηθούμε μόνο οριζοντίως ή καθετίως και μόνο σε κελί που περιέχει μονοπάτι ή έξοδο. Απαγορεύεται να κινηθούμε έξω από το λαβύρινθο.

Ο λαβύρινθος μπορεί να έχει περισσότερες από μία εξόδους ή και καμία έξοδο.

## Είσοδος/Εξοδος

Η είσοδος του προγράμματος γίνεται από το πληκτρολόγιο. Ο λαβύρινθος εισάγεται γραμμή γραμμή, με τη μορφή που φαίνεται στο παραπάνω παράδειγμα. Υπάρχει ένα κενό ανάμεσα σε κάθε δύο χαρακτήρες στην ίδια γραμμή.

Η έξοδος του προγράμματος είναι ένα μήνυμα YES ή NO που ενημερώνει το χρήστη αν είναι δυνατή η έξοδος από το λαβύρινθο ή όχι.

## Ο Αλγόριθμος

### Αναζήτηση εξόδου:

Ο αλγόριθμος αναζήτησης παίρνει ως είσοδο το λαβύρινθο και τις συντεταγμένες του κελιού στο οποίο θέλουμε να κινηθούμε. Επιστρέφει TRUE ή FALSE ανάλογα με το αν η κίνηση είναι δυνατή.

- Αν το κελί είναι εκτός λαβυρίνθου, επιστρέφουμε FALSE
- Αν το κελί περιέχει τοίχο, επιστρέφουμε FALSE
- Αν το κελί περιέχει έξοδο, επιστρέφουμε TRUE
- Αν το κελί περιέχει μονοπάτι, τότε πρέπει να αναζητήσουμε κάποια έξοδο ξεκινώντας από αυτό το κελί. Στην ουσία, προσπαθούμε να λύσουμε μια μικρότερη έκδοση του ιδίου προβλήματος (δηλαδή, να λύσουμε ένα μικρότερο τμήμα του λαβύρινθου, αυτό που ξεκινά από το νέο κελί). Η διαδικασία επίλυσης του προβλήματος είναι λοιπόν αναδρομική:
  - Αναζητάμε έξοδο ξεκινώντας από το κελί που είναι βόρεια του τρέχοντος. Αν υπάρχει, επιστρέφουμε TRUE
  - Αναζητάμε έξοδο ξεκινώντας από το κελί που είναι ανατολικά του τρέχοντος. Αν υπάρχει, επιστρέφουμε TRUE
  - Αναζητάμε έξοδο ξεκινώντας από το κελί που είναι νότια του τρέχοντος. Αν υπάρχει, επιστρέφουμε TRUE
  - Αναζητάμε έξοδο ξεκινώντας από το κελί που είναι δυτικά του τρέχοντος. Αν υπάρχει, επιστρέφουμε TRUE

### Αποφυγή κύκλων:

Υπάρχει ένα ακόμη θέμα που πρέπει να λάβουμε υπόψη μας. Αν δεν προσέξουμε, υπάρχει περίπτωση να προχωράμε σε κύκλους. Πρέπει να βρούμε κάποιο τρόπο να ξέρουμε αν έχουμε ήδη εξαντλήσει όλα τα πιθανά μονοπάτια που ξεκινούν από ένα κελί. Για να επιτύχουμε αυτό, πρέπει για κάθε κελί να γνωρίζουμε αν έχει ήδη "εξερευνηθεί" ή όχι. Ένας τρόπος υλοποίησης είναι να δημιουργήσουμε ένα δεύτερο πίνακα 10x10, κάθε κελί του οποίου έχει την τιμή 0 ή 1, ανάλογα με το αν το αντίστοιχο κελί στο λαβύρινθο έχει ήδη ανακαλυφθεί. Κάθε φορά που βρισκόμαστε σε ένα κελί και είμαστε έτοιμοι να εξερευνήσουμε τα μονοπάτια που ξεκινούν από αυτό, το θέτουμε ως "καλυμμένο". Η συνάρτηση αναζήτησης πρέπει να παίρνει ως είσοδο και αυτόν το δεύτερο πίνακα.

Ένας πολύ καλύτερος τρόπος να υλοποιηθεί η άσκηση είναι με τη χρήση μιας δομής (struct) που αναπαριστά ένα κελί. Τα πεδία της δομής είναι δύο: το περιεχόμενο του κελιού και το αν το κελί έχει καλυφθεί ή όχι.

## Άλυτη άσκηση

Γράψτε ένα πρόγραμμα το οποίο αντικαθιστά κάθε εμφάνιση μιας συμβολοσειράς σε ένα κείμενο με μια άλλη συμβολοσειρά. Η συνάρτηση που εκτελεί την αντικατάσταση πρέπει να είναι αναδρομική. Η συμβολοσειρά που αντικαθίσταται και η συμβολοσειρά που την αντικαθιστά έχουν το ίδιο μήκος.

### Η συνάρτηση `replace` :

- Παίρνει ως παραμέτρους τρεις συμβολοσειρές:
  - Η πρώτη παράμετρος είναι το κείμενο
  - Η δεύτερη παράμετρος είναι η συμβολοσειρά που αντικαθίσταται
  - Η τρίτη παράμετρος είναι η συμβολοσειρά που αντικαθιστά τη δεύτερη
- Αναδρομικά αντικαθιστά κάθε εμφάνιση της δεύτερης συμβολοσειράς στο κείμενο με την τρίτη συμβολοσειρά.

### Η `main`:

- Διαβάζει τρεις συμβολοσειρές από το πληκτρολόγιο (με την ίδια σειρά που θα δοθούν ως παράμετροι στη `replace`).
- Καλεί τη συνάρτηση `replace` με παραμέτρους τις τρεις συμβολοσειρές
- Εκτυπώνει το αλλαγμένο κείμενο (την πρώτη συμβολοσειρά)

### Παράδειγμα:

Αν `text="alphabetalphaalphagamma"`, `togoaway="alpha"` και `replacement="hello"`, τότε μετά την κλήση στη `replace(text, togoaway, replacement)` το `text` έχει γίνει `helloworldhellohellogamma`

### Συνθήκες που πρέπει να ικανοποιούνται:

- Όλες οι συμβολοσειρές πρέπει να αποθηκεύονται σε δυναμικά δεσμευμένη μνήμη
- Όλη η δυναμικά δεσμευμένη μνήμη πρέπει να απελευθερώνεται
- Μπορείτε να υποθέσετε ότι η είσοδος θα είναι πάντα έγκυρη.
- Η έξοδος του προγράμματος πρέπει να ταιριάζει ΑΚΡΙΒΩΣ με αυτή στα δοσμένα αρχεία εξόδου (`expout?.txt`). Χρησιμοποιήστε την εντολή `diff` για να συγκρίνετε την έξοδο του προγράμματός σας με τη δική μας.
- Πρέπει να υπάρχουν σχόλια συναρτήσεων όπως περιγράφεται στο φυλλάδιο με τίτλο [Πρότυπα σχολιασμού προγραμμάτων](#)

Αποθηκεύστε το πρόγραμμά σας σε ένα αρχείο με όνομα `lab9.c`.

**Αρχεία προς παράδοση: lab9.c**

### **Πώς να παραδώσετε τη δουλειά σας**

Μεταφέρετε τα παραδοτέα αρχεία στο φάκελο που φτιάξατε με όνομα `επώνυμο_όνομα_AEM_email`. Δημιουργήστε ένα `tgz` αρχείο το οποίο περιέχει μόνο αυτό το φάκελο και έχει όνομα `επώνυμο_όνομα_AEM_email.tgz`

Στείλτε email στη διεύθυνση **ce120lab@gmail.com** και ένα αντίγραφο (CC) στον εαυτό σας με θέμα (subject) "CE120 lab9 tmimaX " (όπου X είναι ο αριθμός του τμήματος σας, δηλαδή 1-5) και επικολλημένο αρχείο το `επώνυμο_όνομα_AEM_email.tgz`

Υποβολές οι οποίες δεν ακολουθούν επακριβώς τις οδηγίες δημιουργίας των αρχείων και υποβολής δε θα λαμβάνονται υπόψη.