

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**ΤΜΗΜΑΤΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΕΠΩΝΥΜΟ: \_\_\_\_\_ ΟΝΟΜΑ: \_\_\_\_\_  
ΑΜ: \_\_\_\_\_ ΑΚΑΔ. ΕΤΟΣ: 2014-2015 ΕΞΕΤ. ΠΕΡΙΟΔΟΣ: ΙΟΥΝΙΟΥ  
ΜΑΘΗΜΑ: ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ  
ΔΙΔΑΣΚΩΝ: Γ. ΔΗΜΗΤΡΙΟΥ

26 Ιουνίου 2015

Μέρος Α (25%). Βρείτε τι θα εκτυπώσει καθένα από τα παρακάτω αποσπάσματα κώδικα C. Κάθε σωστή απάντηση προσθέτει 5% στο βαθμό σας, ενώ κάθε λάθος απάντηση αφαιρεί 3% από το βαθμό σας.

|  |   |
|--|---|
| <pre>int y[] = {0, 3, 2, -1}; int i; for (i=10;i&gt;0;i=i-3)     printf("%d ",y[i%4]);</pre> | A. 2 -1 0 3<br>B. 3 2 0 -1<br>C. 0 2 -1 3<br>D. άλλο ή θα έχουμε σφάλμα |
|--|---|

|   |   |
|---|---|
| <pre>char *x[] = {"nikos","sakis","elena"}; char y[10]; int i; for (i=0;i&lt;6;i++) y[i]=x[i/2][i]; printf("%s",y);</pre> | A. sakis<br>B. elsani<br>C. nikia<br>D. άλλο ή θα έχουμε σφάλμα |
|---|---|

|   |  |
|---|--|
| <pre>int y[] = {1, 4, 0, -2}; int *x = y+1; x--; printf("%d",*x*2);</pre> | A. 0<br>B. 2<br>C. ό,τι περιέχει η διεύθυνση 1<br>D. άλλο ή θα έχουμε σφάλμα |
|---|--|

|   |  |
|---|--|
| <pre>int i, y[] = {3, -2, 0, -1}; int *x = (int*)malloc(8*sizeof(int)); for (i=0;i&lt;8;i++) *(x+7-i) = *(y+i/2); for (i=0;i&gt;-8;i-=2) printf("%d ",x[-i]);</pre> | A. 3 -2 0 -1<br>B. -1 3 0 -2 -2 0 3 -1<br>C. -1 0 -2 3<br>D. άλλο ή θα έχουμε σφάλμα |
|---|--|

|  |  |
|--|--|
| <pre>struct {int a,b[2];} x[] = {0, 3, -2, 5, 1, 0, -3, -1, 1, 4, 0, 2}; int i, *y = (int*)malloc(4*sizeof(int)); for (i=0;i&lt;4;i++) *(y+i) = x[i].a*x[3-i].b[i/2]; for (i=3;i&gt;=0;i--) printf("%d %d ",x[i].b[i/2],y[3-i]);</pre> | A. -2 8 0 0 1 -5 2 0<br>B. 0 3 -5 1 0 1 -8 2<br>C. 2 0 1 -5 1 0 3 -8<br>D. άλλο ή θα έχουμε σφάλμα |
|--|--|

Μέρος Β (75%). Γράψτε μέσα από την πλατφόρμα *Code::Blocks* ένα πρόγραμμα C το οποίο να διαβάζει από ένα αρχείο έναν πίνακα με αστεράκια και κενά, σε σχήμα λαβυρίνθου, να προσπαθεί να επιλύσει το λαβύρινθο ακολουθώντας επαναληπτικά μια τυχαία πορεία μέχρι να βρει την έξοδο, και να εκτυπώνει στην οθόνη το μήκος της επιτυχημένης διαδρομής.

Ειδικότερα, εκκινήστε και δημιουργήστε στην Επιφάνεια Εργασίας ένα φάκελο-project με όνομα το όνομά σας και τον ΑΜ σας, πχ «*ΓιώργοςΓεωργίου113*». Θα πρέπει να δουλέψετε σε αυτό το φάκελο.

Το πρόγραμμα που θα γράψετε θα πρέπει:

1. Να ζητάει από το χρήστη το μέγεθος  $N$  του πίνακα ( $N \times N$ ). Η είσοδος του  $N$  θα γίνεται από το πληκτρολόγιο.
2. Να ζητάει από το χρήστη το όνομα του αρχείου στο οποίο βρίσκεται ο λαβύρινθος, και αν αυτό είναι έγκυρο, να ανοίγει το αρχείο και να διαβάζει το λαβύρινθο σε κάποιον κατάλληλο πίνακα.
3. Να ξεκινάει από το σημείο  $(0,0)$  και ακολουθώντας χαρακτήρες κενού (' ') να προχωράει μέχρι να φτάσει στο σημείο  $(N-1,N-1)$ . Όπου έχει επιλογή κίνησης, να ακολουθεί τυχαία κατεύθυνση. Αν συναντήσει σημείο που ξανασυνάντησε ή αδιέξοδο, να αρχίζει από την αρχή. Κάθε φορά να σημειώνει και να μετράει τους χαρακτήρες που περνάει.
4. Να εκτυπώνει στην οθόνη το μήκος της τελικής διαδρομής.

Για την υλοποίηση των παραπάνω ζητούμενων, το πρόγραμμά σας μπορεί να δηλώνει και να χρησιμοποιεί οσεσδήποτε μεταβλητές, απλές ή πίνακες. Οι απλές μεταβλητές θα πρέπει να είναι τοπικές, αν δεν έχει νόημα να είναι καθολικές.

Από εκεί και πέρα, το πρόγραμμά σας θα πρέπει να χρησιμοποιεί με τον καλύτερο δυνατό τρόπο τις δομές και εντολές που έχετε μάθει. Έτσι, θα μετρήσει η ποιότητα του κώδικα, όπως για παράδειγμα η καλή επικοινωνία προγράμματος-χρήστη, η καλή χρήση βρόχων, η σωστή χρήση δεικτών (pointers), η σωστή χρήση συναρτήσεων (δικών σας ή βιβλιοθήκης), η σωστή χρήση αρχείων, η καλή στοίχιση καθώς και ο σχολιασμός.

Για έλεγχο σωστής λειτουργίας του κώδικά σας, διατίθεται το αρχείο "labyrinth.txt" που περιέχει ένα λαβύρινθο μεγέθους  $20 \times 20$ , στη θέση που θα σας υποδειχτεί.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ