

LAB1 ΛΥΣΗ

Ευρετήριο απαντήσεων ανά τμήμα:

[Τμήμα 1](#)

[Τμήμα 2](#)

[Τμήμα 3](#)

[Τμήμα 4](#)

[Τμήμα 5](#)

Ευρετήριο απαντήσεων ανά μήνυμα λάθους/προειδοποίησης

[error: invalid preprocessing directive #includes](#)

[error: syntax error before ...](#)

[error: ... undeclared \(first use in this function\)](#)

[error: studio.h: No such file or directory](#)

[error: missing terminating " character](#)

[warning: division by zero](#)

[warning: implicit declaration of function ...](#)

[error: empty character constant](#)

[warning: "/" within comment](#)

[error: invalid lvalue in assignment](#)

[error: redeclaration of 'l' with no linkage \ previous declaration of 'l' was here](#)

[error: #include expects "FILENAME" or <FILENAME>](#)

[error: syntax error before 'double'](#)

[error: missing terminating " character](#)

[error: invalid suffix ... on integer constant](#)

Τμήμα 1

error: invalid preprocessing directive #includes

Το μήνυμα εμφανίστηκε στη γραμμή `#includes<stdio.h>` και μας λέει ότι η directive που έχουμε χρησιμοποιήσει δεν είναι έγκυρη. Αυτό συνήθως σημαίνει πως έχουμε κάποιο τυπογραφικό λάθος. Πράγματι, έχουμε γράψει `#includes` αντί για `#include`

error: syntax error before ...

Αυτό το μήνυμα είναι αρκετά γενικό και δε δίνει αρκετές πληροφορίες για το τι έχουμε κάνει λάθος, οπότε η μόνη μας επιλογή είναι να διαβάσουμε προσεκτικά τον κώδικα που προηγείται του σημείου όπου ανιχνεύτηκε το λάθος.

Στο `lab1.c` του τμήματος 1 η γραμμή του λάθους και η αμέσως προηγούμενη είναι:

```
int main (int argc, char *argv[]) (  
    int tempF, tempC;
```

Προσεκτική εξέταση του κώδικα μας δείχνει το λάθος στην πρώτη γραμμή: Αντί για το αριστερό άγκιστρο που προσδιορίζει το σώμα της `main`, υπάρχει παρένθεση.

Στο `lab1.c` του τμήματος 2 η γραμμή του λάθους είναι:

```
int main (int argc, char *argv[] {
```

Προσεκτική εξέταση του κώδικα μας δείχνει το σε αυτή τη γραμμή: Λείπει μια δεξιά παρένθεση πριν το άγκιστρο.

error: ... undeclared (first use in this function)

Αυτό το μήνυμα εμφανίζεται όταν ο compiler ανιχνεύει μια μεταβλητή η οποία δε φαίνεται να έχει δηλωθεί. Υπάρχουν αρκετά ενδεχόμενα για το ποιο είναι το ακριβές λάθος:

- Έχουμε ξεχάσει να δηλώσουμε τη συγκεκριμένη μεταβλητή, ή
- Την έχουμε δηλώσει, αλλά έχουμε κάποιο τυπογραφικό λάθος στο σημείο που τη χρησιμοποιούμε
- Η μεταβλητή είναι εκτός εύρους (περισσότερα σε επόμενο μάθημα)

Το ποια περίπτωση είναι εξαρτάται από το τι θέλουμε να κάνει το πρόγραμμα στο συγκεκριμένο σημείο.

Στο `lab1.c` των τμημάτων 1 και 2 και 4 το λάθος εμφανίστηκε στη γραμμή: `return 0;`

Γνωρίζουμε ότι η εντολή `return` στη `main` πρέπει να επιστρέφει μια ακέραια τιμή. Αυτή συνήθως είναι το μηδέν, αλλά μπορεί να είναι και οποιαδήποτε έκφραση αποτιμάται σε ακέραιο. Αφού ο compiler παραπονιέται για μεταβλητή, συμπεραίνουμε πως το `0` είναι όνομα μεταβλητής (και για την ακρίβεια το κεφαλαίο γράμμα `o`), κι όχι ο αριθμός μηδέν. Ένα επιπλέον στοιχείο προς αυτό το συμπέρασμα είναι το γεγονός πως το `0` δεν είναι χρωματισμένο από τον επεξεργαστή κειμένου μας με το χρώμα που χαρακτηρίζει αριθμητικά literals.

Κάποια ομάδα "διόρθωσε" το λάθος με το να συμπεριλάβει το `0` σε `' '` (δηλαδή έγραψε `return '0';`). Αυτό δεν είναι σωστή διόρθωση, αν και ικανοποιεί τον compiler. Το `'0'` είναι μια τιμή

τύπου χαρακτήρας (char). Οι χαρακτήρες στη C αναπαρίστανται ως ακέραιοι (ASCII κωδικοί), οπότε το `return '0';`, όσον αφορά τον compiler, είναι ισοδύναμο με το `return 79;` διότι ο κωδικός ASCII του κεφαλαίου ο είναι 79.

Στο lab1.c του τμήματος 3 το λάθος εμφανίστηκε στη γραμμή `int tempF; tempC;` η οποία δηλώνει τις μεταβλητές του προγράμματος.

Παρατηρούμε πως για τη μεταβλητή `tempC` δεν έχει προσδιοριστεί κάποιος τύπος. Κατά πάσα πιθανότητα, ο προγραμματιστής έβαλε κατά λάθος ερωτηματικό αντί για κόμμα μετά το `tempF`. Η διόρθωση είναι `int tempF, tempC;` Εναλλακτικά, θα μπορούσαμε να προσδιορίσουμε τύπο `int` για το `tempC` αλλά τότε θα έπρεπε να βάλουμε τη δήλωση στην επόμενη γραμμή (μην ξεχνάτε: μια εντολή ανά γραμμή!) ως εξής:

```
int tempF;  
int tempC;
```

Επίσης στο lab1.c του τμήματος 3, το ίδιο λάθος εμφανίζεται στη γραμμή `tempC = (tempF-32)*5/9;` Εδώ όμως το πρόβλημα δεν είναι ότι ξεχάσαμε να δηλώσουμε μια μεταβλητή, αλλά ότι κάναμε τυπογραφικό λάθος κατά τη χρήση μιας ήδη δηλωμένης μεταβλητής (αντί για `tempF` γράψαμε `tenpF`).

Συμπέρασμα: Όταν βλέπετε αυτό το μήνυμα λάθους, δε σημαίνει απαραίτητα ότι πρέπει να δηλώσετε τη μεταβλητή για την οποία σας παραπονιέται. Αντίθετα, πρέπει να ΜΕΛΕΤΗΣΕΤΕ τον κώδικα για να εξακριβώσετε αν όντως ξεχάσατε τη δήλωση ή αν το μήνυμα οφείλεται σε κάτι διαφορετικό όπως πχ τυπογραφικό λάθος.

Στο lab1.c του τμήματος 5 το μήνυμα εμφανίζεται σε δύο σημεία. Στην πρώτη περίπτωση έχουμε ορθογραφικό λάθος στο όνομα μιας μεταβλητής (`radius` αντί για `radius`). Στη δεύτερη περίπτωση έχουμε λάθος στο όνομα μιας σταθεράς: αντί για κεφαλαίο `I` έχουμε χρησιμοποιήσει τον αριθμό 1. Το κεφαλαίο `I`, ο αριθμός ένα και το μικρό `el` είναι γενικά χαρακτήρες που μοιάζουν πολύ, οπότε πρέπει πάντα να αποφεύγουμε να τους χρησιμοποιούμε με τρόπο που μπορεί να προκαλέσει σύγχυση.

Τμήμα 2

error: studio.h: No such file or directory

Το μήνυμα προσδιορίζει ότι δεν υπάρχει το αρχείο που προσπαθούμε να κάνουμε include. Στην περίπτωση που το αρχείο που θέλουμε να κάνουμε include είναι της στανταρντ βιβλιοθήκης, τότε σημαίνει πως έχουμε κάνει κάποιο λάθος στο όνομα ή δεν έχουμε χρησιμοποιήσει < >. Θα επανέλθουμε σε αυτό το μήνυμα και πιθανές διορθώσεις όταν μάθετε πώς να κατασκευάζετε δικά σας header files.

Στη συγκεκριμένη περίπτωση πρόκειται για ορθογραφικό λάθος. Το αρχείο λέγεται stdio.h και όχι studio.h

error: syntax error before ...

[βλ. σχόλια στο τμήμα 1](#)

error: missing terminating " character

Αυτό το μήνυμα σημαίνει ότι έχουμε ξεχάσει να "κλείσουμε" μια συμβολοσειρά. Είναι εύκολο να το εντοπίσουμε και χωρίς χρήση compiler, γιατί ο επεξεργαστής κειμένου που χρησιμοποιούμε δίνει συγκεκριμένο χρώμα στις συμβολοσειρές. Αν δούμε ότι υπάρχει κομμάτι του κώδικα στο χρώμα συμβολοσειράς χωρίς να πρέπει, ξέρουμε αμέσως ότι κάπου πιο πριν έπρεπε να είχαμε κλείσει τη συμβολοσειρά χρησιμοποιώντας " (double quote).

Στην άσκηση του τμήματος 2, έπρεπε να προσθέσουμε " πριν τη δεξιά παρένθεση του printf.

warning: division by zero

Αυτό το μήνυμα μας προειδοποιεί για διαίρεση με το μηδέν. Το σχετικό κομμάτι κώδικα είναι `tempC = (tempF-32)*5/0;` Για να διορθώσουμε το λάθος, πρέπει να γνωρίζουμε τον τύπο μετατροπής θερμοκρασίας, ο οποίος περιγράφεται στα σχόλια του προγράμματος.

error: ... undeclared (first use in this function)

[βλ. σχόλια στο τμήμα 1](#)

Τμήμα 3

error: ... undeclared (first use in this function)

[βλ. σχόλια στο τμήμα 1](#)

warning: implicit declaration of function...

Αυτό το μήνυμα σημαίνει ότι χρησιμοποιούμε μια συνάρτηση η οποία δεν έχει δηλωθεί. Αν πρόκειται για συνάρτηση βιβλιοθήκης (όπως στο συγκεκριμένο μήνυμα όπου αναφέρεται στην printf) τότε εξετάζουμε τα `#include` και βεβαιωνόμαστε ότι έχουμε συμπεριλάβει το αρχείο με τις κατάλληλες δηλώσεις συναρτήσεων. Στη συγκεκριμένη άσκηση, πρέπει να προσθέσουμε `#include<stdio.h>` στην αρχή του προγράμματος.

error: empty character constant

Ένα literal που είναι χαρακτήρας περικλείεται από single quotes (μονά εισαγωγικά). Αν ο compiler δει δύο διαδοχικά single quotes χωρίς χαρακτήρα ανάμεσά τους, τότε εμφανίζει το παρακάτω λάθος. Επίσης, εμφανίζει προειδοποίηση αν δει περισσότερους από έναν χαρακτήρες μέσα σε single quotes (warning: multi-character character constant). Στο συγκεκριμένο πρόγραμμα, πρέπει να αντικαταστήσουμε τα δύο single quotes από ένα double quote (").

Επίσης, σημειώστε πως το γεγονός ότι το string του printf δεν είναι χρωματισμένο κόκκινο όπως θα περιμέναμε, μας δίνει μια νύξη για το ότι κάτι δεν πάει καλά με αυτό.

error: ... undeclared (first use in this function)

[βλ. σχόλια στο τμήμα 1](#)

Τμήμα 4

warning: *"/*" within comment*

Δεδομένου ότι οι χαρακτήρες `/*` ξεκινούν ένα σχόλιο, το παραπάνω μήνυμα υπονοεί ότι προσπαθούμε να "φωλιάσουμε" ένα σχόλιο μέσα σε ένα άλλο. Μπορεί να έχουμε κάνει κυριολεκτικά αυτό το λάθος ή απλά να έχουμε ξεχάσει να κλείσουμε ένα προηγούμενο σχόλιο. Σε κάθε περίπτωση, πρέπει να κοιτάξουμε στις προηγούμενες γραμμές κώδικα για να δούμε τι ακριβώς έχει συμβεί και να κάνουμε την κατάλληλη διόρθωση.

Το γεγονός ότι ο επεξεργαστής κειμένου δίνει ξεχωριστό χρώμα στα σχόλια (το Kate τα εμφανίζει γκρι) μπορεί να μας βοηθήσει πολύ στον εντοπισμό του λάθους.

error: *invalid lvalue in assignment*

Ο όρος `lvalue` αναφέρεται σε οντότητες που μπορούν να εμφανιστούν αριστερά ενός τελεστή ανάθεσης (`lvalue = right value`). Με βάση αυτά που έχετε μάθει μέχρι τα πρώτα 2 εργαστήρια, `lvalues` μπορεί να είναι μόνο μεταβλητές. Οπότε το παραπάνω μήνυμα σημαίνει ότι εμφανίστηκε μη-μεταβλητή στο αριστερό μέρος ενός `=`. Πράγματι ο σχετικός κώδικας είναι `1 = 1`; Αριστερά από το `=` βρίσκεται ο αριθμός ένα, ενώ δεξιά η μεταβλητή `l`. Έπρεπε να είναι ανάποδα.

Εδώ να σημειώσουμε το πόσο σημαντικό είναι να δίνουμε περιγραφικά ονόματα στις μεταβλητές μας όχι μόνο για να είναι κατανοητός ο κώδικας αλλά και για να αποφεύγουμε λάθη όπως το παραπάνω: ο αριθμός ένα και το αγγλικό γράμμα `el` μοιάζουν πάρα πολύ και είναι εύκολο να τα μπερδέψουμε. Μεταβλητές όπως `l`, `ll` κτλ δεν πρέπει να χρησιμοποιούνται ποτέ!

error: *redeclaration of 'l' with no linkage \ previous declaration of 'l' was here*

Αυτό το μήνυμα εμφανίζεται όταν μια μεταβλητή έχει δηλωθεί περισσότερες από μία φορές στο ίδιο εύρος (`scope`). Είναι πολύ σημαντικό να διαβάσετε προσεκτικά το πρόγραμμα και να έχετε καταλάβει τι λειτουργία του, για να μπορέσετε να ανιχνεύσετε πού έγινε το λάθος. Πιθανώς δηλώσατε μια μεταβλητή χωρίς να προσέξετε ότι υπάρχει ήδη άλλη με το ίδιο όνομα. Πιθανώς κάνατε κάποιο τυπογραφικό λάθος στο όνομα της δεύτερης μεταβλητής, και τυχαίνει να είναι ίδιο με το όνομα της πρώτης. Στο συγκεκριμένο παράδειγμα, έχουμε ξαναγράψει κατά λάθος τις δηλώσεις κάποιων μεταβλητών χωρίς να χρειάζεται, οπότε αρκεί να σβήσουμε ολόκληρη τη γραμμή 19.

error: *... undeclared (first use in this function)*

[βλ. σχόλια στο τμήμα 1](#)

Τμήμα 5

warning: *"/" within comment*

[βλ. σχόλια στο τμήμα 4](#)

error: *#include expects "FILENAME" or <FILENAME>*

Λάθος στη σύνταξη του `#include`. Το όνομα του αρχείου πρέπει να περικλείεται σε `< >` αν πρόκειται για header file της C ή σε `" "` αν πρόκειται για header file που κατασκευάσατε εσείς (θα μάθετε αργότερα πώς).

error: *syntax error before 'double'*

Αυτό το μήνυμα δε μας δίνει πολλές πληροφορίες και μπορεί να σημαίνει οτιδήποτε. Εξετάζουμε προσεκτικά τον κώδικα για να δούμε πού μπορεί να είναι το λάθος. Στη συγκεκριμένη περίπτωση έχουμε κόμμα αντί για ερωτηματικό στο τέλος της προηγούμενης εντολής.

error: *missing terminating " character*

Οι συμβολοσειρές (strings) σε ένα πρόγραμμα C πρέπει να περικλείονται σε double quotes (διπλά εισαγωγικά). Το παραπάνω μήνυμα σημαίνει πως στο πρόγραμμα εμφανίζεται μια συμβολοσειρά στην οποία δεν "κλείνουν" τα double quotes. Ένας εύκολος τρόπος να εντοπίσουμε το λάθος χωρίς καν να δούμε το μήνυμα είναι από το χρώμα που δίνει ο επεξεργαστής κειμένου στον κώδικα. Στο Kate οι συμβολοσειρές εμφανίζονται με κόκκινο. Αν έχουμε ξεχάσει το `"`, τότε θα είναι κόκκινος περισσότερος κώδικας από ότι πρέπει.

error: *... undeclared (first use in this function)*

[βλ. σχόλια στο τμήμα 1](#)

error: *invalid suffix ... on integer constant*

Το μήνυμα μας λέει ότι έχουμε μια ακέραια σταθερά στο πρόγραμμά μας (το 2 στη συγκεκριμένη περίπτωση) η οποία ακολουθείται από κάτι που δεν επιτρέπεται να εμφανίζεται μετά από αριθμό. Παρατηρούμε πως στο σημείο με το λάθος γράφουμε `2PI`. Για να διορθώσουμε το πρόβλημα πρέπει να έχουμε καταλάβει τι ακριβώς πρέπει να κάνει το πρόγραμμα σε αυτό το σημείο.

Από τα σχόλια του προγράμματος βλέπουμε πως πρέπει να υπολογίζουμε την περίμετρο ενός κύκλου, άρα συμπεραίνουμε πως το σωστό είναι `2*PI`

Εδώ να σημειώσουμε πως σε κάποιες περιπτώσεις επιτρέπεται να εμφανιστεί συγκεκριμένο γράμμα μετά από αριθμό. Για παράδειγμα το `2U` σε ένα πρόγραμμα σημαίνει πως θέλουμε ο αριθμός 2 να χρησιμοποιηθεί στο συγκεκριμένο σημείο ως unsigned integer.